# Synthèse de vues et reconstruction de vues à partir de vidéos compressées multi-vues et multi-sources

Andrei Purica

## To cite this version:

HAL Id: tel-03533526
https://pastel.hal.science/tel-03533526

Submitted on 18 Jan 2022

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2017-ENST-0029

EDITE - ED 130

## Doctorat ParisTech

# THÈSE

**pour obtenir le grade de docteur délivré par**

## TELECOM ParisTech

**Spécialité « Signal et Images »**

*présentée et soutenue publiquement par*

**Andrei Iacob PURICA**

26 Juin 2017

# Synthèse et reconstruction de vues à partir de vidéos compressées multi-vues et multi-sources

Directeurs de thèse :
**Frédéric DUFAUX**
**Béatrice PESQUET-POPESCU**

**Jury**

| | |
|---|---|
| **Christine GUILLEMOT**, Dir. de Recherches, IRISA-INRIA | Rapporteur |
| **Peter SCHELKENS**, Professeur, Vrije Universiteit Brussel | Rapporteur |
| **Marc ANTONINI**, Dir. de Recherches, Lab. I3S - Univ. de Nice-Sophia Antipolis | Examinateur |
| **Mihai CIUC**, Professeur, University Politehnica of Bucharest | Examinateur |
| **Frédéric DUFAUX**, Dir. de Recherches, L2S - CentraleSupelec - Univ. Paris-Sud | Directeur de thèse |
| **Béatrice PESQUET-POPESCU**, Professeur, Télécom ParisTech | Directrice de thèse |
| **Bogdan IONESCU**, Professeur, University Politehnica of Bucharest | Co-Encadrant |

THÈSE

**TELECOM ParisTech**
école de l'Institut Mines-Télécom - membre de ParisTech
46 rue Barrault 75013 Paris - (+33) 1 45 81 77 77 - www.telecom-paristech.fr

# Synthése et reconstruction de vues à partir de vidéos compressées multi-vues et multi-sources

—

# View synthesis and view reconstruction from multi-view and multi-source compressed video

Andrei Iacob PURICA

# Long résumé

## 1 Introduction

Au cours de la dernière décennie, le monde a connu un “boom” de la connectivité avec les téléphones intelligents, passant progressivement d'un appareil de luxe et de niche à un outil presque indispensable dans la société d'aujourd'hui. Parallèlement, les services multimédia se sont déplacés vers une approche Cloud alors que les technologies de codage, de transmission et de stockage vidéo évoluaient à un niveau où la vidéo de haute qualité est facilement accessible sur Internet. Selon une enquête de Cisco, les vidéos représentaient 64% du trafic Internet en 2014, avec une prévision de 80% d'ici 2020. Par ailleurs, cette forte augmentation de la demande de contenus vidéo alimente une évolution rapide des technologies d'affichage. Les résolutions Ultra Haute Définition (UHD) sont désormais largement disponibles sur les téléviseurs et même sur les appareils mobiles. D'autres technologies qui fournissent une immersion supplémentaire, telles que la 3D stéréo ou la Haute Dynamique (HDR), sont déjà déployées sur une large gamme d'appareils, tandis que la réalité virtuelle (VR) et les vidéos 360° sont accessibles même sur les téléphones intelligents, grâce à l'utilisation d'écrans montés sur la tête.

En plus de l'amélioration et de l'évolution des technologies existantes, de nouvelles façons de fournir une expérience plus immersive sont continuellement étudiées. Les systèmes de téléconférence par immersion, la télédiffusion de Free Viewpoint TeleVision (FTV) et d'autres applications vidéo immersives sont maintenant possibles. La compression vidéo et la normalisation des formats vidéo jouent un rôle essentiel dans la mise en oeuvre de ces applications dans l'environnement interconnecté d'aujourd'hui. Suite à la finalisation récente de la norme de codage vidéo *High Efficiency Video Coding* (HEVC), une série d'extensions ont été développées pour répondre à diverses demandes. Les extensions de codage vidéo MultiView Video (MVV) et MultiView plus Depth (MVD) de HEVC sont déjà disponibles (MV-HEVC et 3D-HEVC) tandis que des expériences d'exploration pour les formats Divergent

MultiView qui permettent la vidéo 3D 360° ont récemment commencé. Une classe importante d'algorithmes qui exploitent les corrélations inter-vues, pour combler l'écart entre la vidéo 2D et 3D en générant de nouveaux points de vue virtuels, sont connus sous le nom de méthodes de synthèse de vue. En plus de permettre la conversion FTV ou 2D en 3D, ils sont également employés dans la compression ou le rendu vidéo 360°.

Pour résumer l'ensemble de la situation, nous sommes maintenant à un point de transition vers la vidéo 3D immersive au cours de laquelle de nouvelles technologies sont explorées et normalisées. En outre, l'adoption en cours de la dernière norme de codage vidéo HEVC, combinée à l'augmentation constante des résolutions d'affichage et à la transition dans les nuages des services vidéo, crée un intérêt significatif pour les algorithmes de super-résolution (SR) et d'amélioration de la qualité vidéo à partir de multiples sources compressées. Dans ce contexte, l'objectif principal de cette thèse est de développer de nouveaux outils visant à améliorer les méthodes de synthèse de vues utilisées dans les systèmes de compression et à combiner plusieurs sources vidéo compressées.

## 2 Codage vidéo

De la façon la plus simple, les vidéos peuvent être considérées comme un support électronique ou numérique qui stocke et facilite la représentation visuelle des médias en mouvement. Que le contenu reflète une scène du monde réel, une scène virtuelle ou un concept abstrait, le trait principal de toutes les vidéos qui les différencie des images est leur capacité à stocker l'information sur le mouvement. Pour cette raison, de grandes quantités d'informations doivent être stockées et transmises afin de partager une vidéo. En général, les vidéos sont formées d'une séquence d'images fixes qui sont affichées à une fréquence suffisamment élevée pour créer l'illusion de mouvement.

Alors que chaque encodeur vidéo a introduit de nouveaux algorithmes et outils, il existe une architecture générique basée sur quelques concepts communs à tous. Cette architecture est connue sous le nom de paradigme de codage vidéo hybride et les principaux concepts sont: quantification, transformations, codage prédictif et codage entropique.

Le paradigme de codage vidéo hybride est utilisé par toutes les normes de codage vidéo actuelles. L'architecture de base d'un encodeur vidéo hybride peut être considérée comme un squelette pour tous les codeurs vidéo modernes. Il utilise deux techniques différentes pour réduire la redondance spatiale et temporelle d'une séquence vidéo. La redondance spatiale est réduite grâce au codage par transformée combiné à la quantification qui réduit la taille des données en éliminant les hautes fréquences dans une image. Bien qu'il s'agisse d'une forme d'encodage avec perte car elle contient une étape de quantification, l'impact global sur la qualité perçue est acceptable en raison de la façon dont le Système Visuel Humain (HVS) perçoit l'information. La redondance temporelle est supprimée grâce au codage prédictif. L'idée générale est de prédire les données qui sont actuellement codées à partir des valeurs précédentes décodées et de n'encoder que la différence.

Figure 1 représente l'architecture générique d'un codeur vidéo hybride. Une fois qu'une nouvelle trame $I_k$ est entrée, le codeur peut fonctionner en deux modes selon le type d'encodage: intra-frame ou inter-frame. En mode intra, seul le codage par transformée est effectué. Tout d'abord, l'image est généralement transformée avec la transformée en cosinus discrÃ¨te (DCT). Les coefficients résultants sont quantifiés, puis une étape de codage sans perte est effectuée. Cette dèrniere consiste à appliquer le codage entropique sur les coefficients quantifiés. En fait, ce modèle simple représenté en rouge est également une méthode courante de compression d'images utilisée dans les normes développées par le Joint Photographic Experts Group (JPEG).

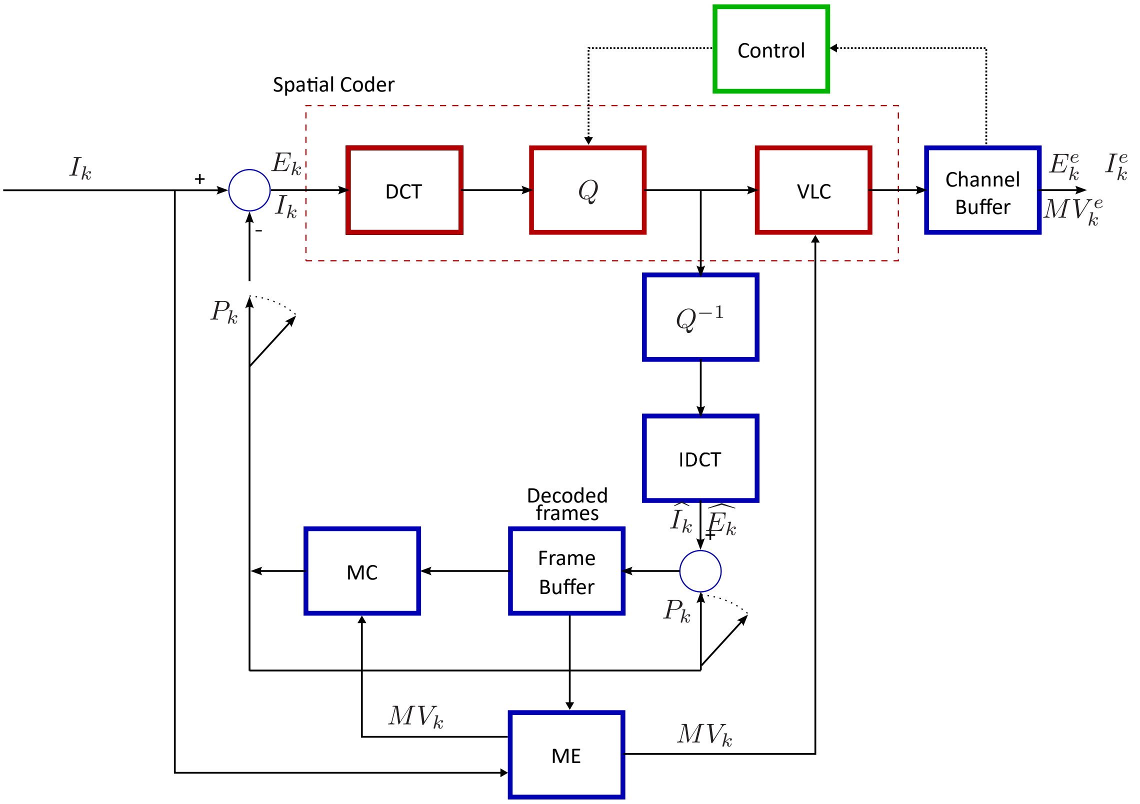


Figure 1 – Schéma générique du codeur hybride.

Le codage inter-image est un peu plus complexe. Tout d'abord un référentiel est décodé, il peut être soit intra ou inter. L'image intra frame est décodée en appliquant $Q^{-1}$ et la transformation inverse (IDCT). L'image résultante $\widehat{I_k}$ est alors stockée dans la mémoire tampon de l'encodeur. Lorsqu'une nouvelle trame est entrée, l'estimation de mouvement (ME) est effectuée entre la trame courante et la trame précédente stockée dans la mèmoire tampon, ce qui permet de calculer le champ de vecteur de mouvement $MV_k$ qui est également inclus dans le flux binaire. En utilisant la compensation de mouvement (MC), une prédiction $P_k$ de la trame est créée. L'erreur de prédiction de $P_k$, dénotée par $E_k$, est déterminée comme étant $I_k - P_k$, passée par le bloc de codage spatial et ajoutée au flux binaire comme étant $E_k^e$. L'erreur de prédiction est également décodée et additionnée avec $P_k$ afin de créer la référence ME pour une trame future.

High Efficiency Video Coding (HEVC) est la dernière norme de codage vidéo par l'équipe collaborative conjointe sur le codage vidéo (JCT-VC), rassemblant des experts de l'Union internationale des télécommunications (UIT) et de l'Organisation internationale de normalisation (ISO).

HEVC représente les données vidéo de manière hiérarchique. Au plus haut niveau, la séquence de données est composée de paramètres généraux (framerate, résolution spatiale, etc.). Un groupe d'images (GOP) définit une période de codage comme un certain nombre de trames (unité de séquence unique dans l'axe temporel).

Comme son prédécesseur, Advanced Video Coding (AVC), le modèle de codage vidéo hybride est réutilisé par la norme HEVC. évidemment, certains outils supplémentaires sont implémentés dans HEVC mais ne sont pas représentés dans Fig. 1, par exemple les filtres de déblocage ou Sample Adaptive Offset (SAO). Toutefois, il convient de noter que si HEVC a presque doublé l'efficacité de codage par rapport à h. 264/AVC, elle provient principalement des optimisations réalisées dans les éléments constitutifs essentiels (prédiction, transformation, codage entropique, etc.), alors que les outils supplémentaires (par exemple le SAO) ne peuvent apporter que des gains marginaux.

Étant donné une trame d'entrée de résolution arbitraire, un schéma de partitionnement de bloc est utilisé pour effectuer la compression au niveau d'un bloc de pixels. HEVC améliore considérablement la grille fixe de macrobloc 16x16 utilisée dans h.264/AVC en la remplaçant par une structure quadrangulaire plus flexible, ce qui permet une meilleure adaptation du partitionnement au contenu de l'image. L'arborescence quadrangulaire utilise une structure hiérarchique: la trame est d'abord divisée en *Coding Tree Units* (CTU) de taille fixe (de 64x64 à 16x16). Les CTU sont divisées (potentiellement récursivement) en *Unités de codage* (CU), formant la structure quadrangulaire. Ensuite, *Unités de Prédiction* (PU) et *Unités de Transformation* (TU) sont enracinées au niveau CU pour rassembler toutes les informations d'unité sur la prédiction (mode, vecteurs de mouvement, index de référence de trame, etc.) et la transformation utilisée respectivement. Il est important de noter que la taille d'une prédiction/transformation n'est pas liée à l'CU: les PU et les TU peuvent être subdivisées récursivement, et les PU et les TU sont indépendantes, de sorte que la prédiction et la transformation peuvent être effectuées à différentes tailles à l'intérieur d'une unité.

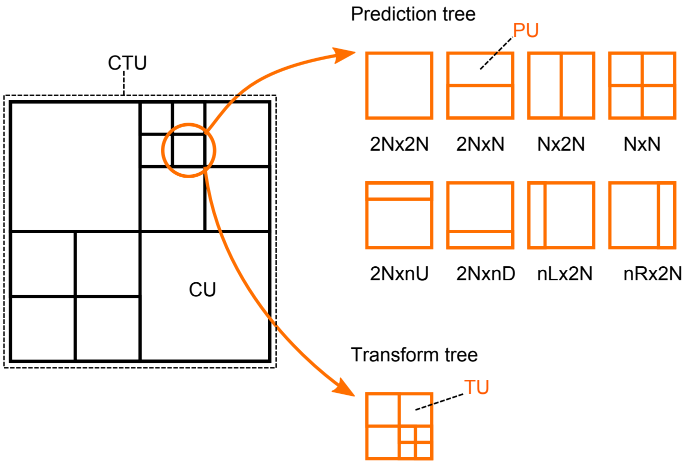


Figure 2 – Structure de partitionnement de trame dans HEVC.

Le HEVC met également en oeuvre une prédiction pour aborder les redondances spatiales dans un cadre, connu sous le nom de prédiction Intra. Cet outil est disponible pour tous les types de trames (I/P/B) et utilise des unités précédemment décodées comme référence pour prédire les valeurs des pixels de l'unité à encoder. Compte tenu de l'ordre de traitement de balayage matriciel du quadrilatère, les unités supérieure, supérieure gauche et gauche sont considérées comme le voisinage. Les modes Intra sont ordonnés selon l'angle de direction. Les directions verticale et horizontale sont associées à de faibles indices Intra (respectivement 1 et 2), tandis que les angles plus fins ont des indices Intra plus élevés, comme le montre la Figure 3.

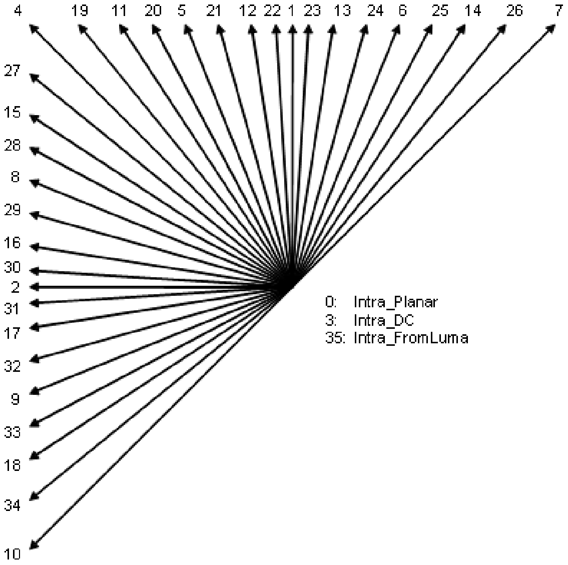


Figure 3 – HEVC modes intra.

Afin d'offrir à l'utilisateur une expérience de visualisation 3D, il faut au moins deux vues d'une scène (une pour chaque oeil). Ainsi, le format vidéo 3D le plus simple est la vidéo stéréoscopique classique (CSV), deux vues de la même scène sont acquises par deux caméras à une certaine distance (base de référence), comme le montre la Figure 4.

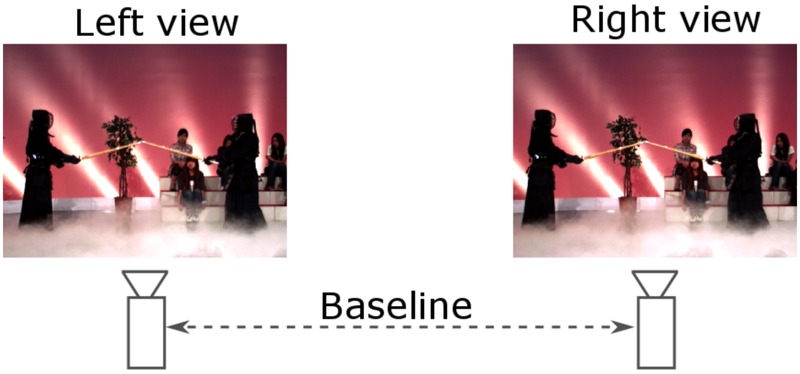


Figure 4 – Vidéo stéréo conventionnelle.

Les nouvelles technologies prennent également en compte la parallaxe de mouvement et visent à favoriser l'affichage de points de vue multiples de la scène. Pour permettre ce type de services, le format MultiView Video (MVV) a été introduit (voir Figure 5). Les données sont composées de $N$ vues acquises par des $N$ caméras dans une configuration spécifique en fonction de l'application. Certaines des configurations les plus courantes sont les matrices de caméras linéaires ou en arcs.

Selon le nombre de vues, les formats MVV peuvent nécessiter la transmission de grandes quantités de données. De plus, un utilisateur est limité à un ensemble fixe de positions. Ces problèmes sont traités par les formats MultiView-plus-Depth (MVD) qui associent une carte de profondeur à chaque vue et permettent la synthèse d'un nombre quelconque de vues virtuelles entre elles, comme le montre la Figure 5. Les cartes de profondeur n'utilisent qu'un seul plan d'image et fournissent une valeur pour chaque pixel qui mesure la distance entre la caméra et la projection réelle de ce pixel.

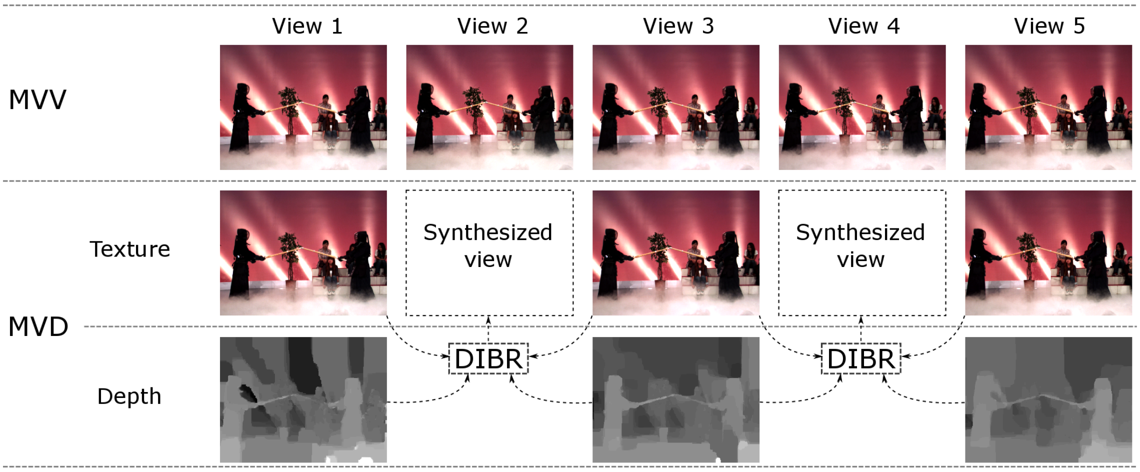


Figure 5 – Vidéo MultiView et MultiView-plus-depth.

## 3 Synthèse de vues exploitant la prédiction temporelle.

La synthèse de vues est le processus d'extrapolation ou d'interpolation d'une vue à partir d'autres vues disponibles. Les techniques de synthèse de vues peuvent être classées principalement en trois catégories. Les méthodes de la première catégorie, comme le *Depth-Image-Based-Rendering* (DIBR), requièrent des informations géométriques explicites telles que des cartes de profondeur ou de disparité pour déformer les pixels des vues disponibles à la bonne position dans la vue synthétisée. Les méthodes de la deuxième catégorie n'exigent qu'une géométrie implicite, comme certaines correspondances de pixels dans la vue disponible et synthétisée. Enfin, les méthodes de la troisième catégorie ne nécessitent aucune géométrie. Ils filtrent et interpolent de façon appropriée un ensemble d'échantillons préacquis. Un problème courant dans la synthèse de vues est celui des zones qui sont occultées dans les vues disponibles mais qui doivent être visibles dans les vues virtuelles. Ces zones apparaissent comme des trous dans les vues virtuelles, également appelées *disocclusions*. Ce problème est actuellement résolu par l'utilisation d'algorithmes de *inpainting*.

La plupart des algorithmes de synthèse de vues déforment la texture d'une trame donnée en utilisant les cartes de profondeur associées pour calculer des vecteurs de disparité (DVs). Cependant, les corrélations temporelles dans une séquence vidéo, sous forme de champs des vecteurs du mouvement (MVF), pourraient être utilisées pour l'améliorer davantage. Le défi consiste à obtenir une MVF qui peut être utilisée dans la vue synthétisée. Le calcul direct du MVF entre les trames synthétisées peut fournir une mauvaise estimation car les trames de référence et prédites sont affectées par les distorsions de synthèse. Une solution possible pour traiter les séquences MVD consiste à utiliser des corrélations inter-vues pour lier les MVF de vues différentes.

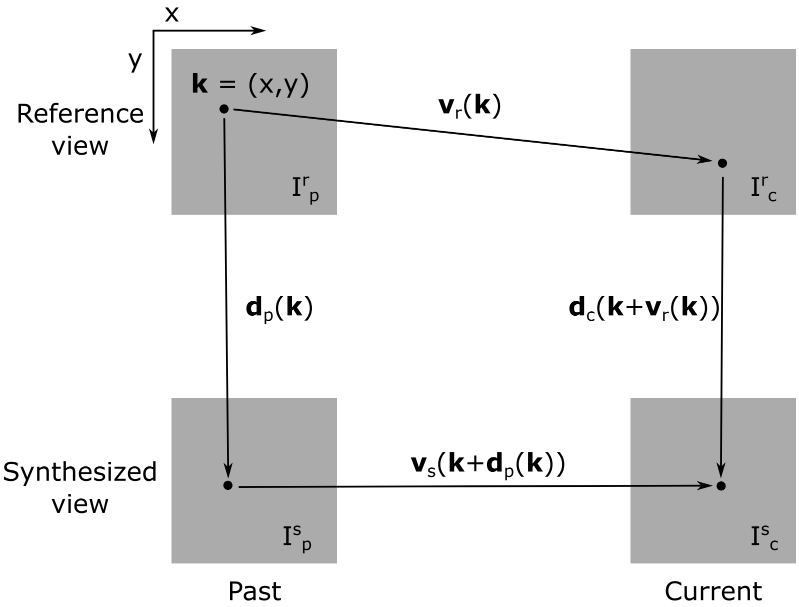


Figure 6 – Contrainte épipolaire, la relation entre les champs vectoriels de disparité (DVF) $\mathbf{d}_p$ et $\mathbf{d}_c$ à deux instants $c$ et $p$ respectivement, et les MVFs dans la vue synthétisée et de référence $\mathbf{v}_s$ et $\mathbf{v}_r$ respectivement pour une position $\mathbf{k}$ dans le cadre de référence $I_p^r$.

Figure 6 montre la relation entre les positions d'une projection de point du monde réel dans différentes vues et à différents instants. Considérons $I_p^r$, $I_c^r$, $I_p^s$, $I_c^s$ qui sont, respectivement, les cadres de vue de référence $(r)$ et les cadres de vue synthétisés $(s)$ à un ancien $p$ et actuel $c$ moment dans le temps. Si un point k n'est pas occulté, nous pouvons définir une contrainte dite épipolaire entre les quatre images:

$$\mathbf{v}_r(\mathbf{k}) + \mathbf{d}_c(\mathbf{k} + \mathbf{v}_r(\mathbf{k})) = \mathbf{d}_p(\mathbf{k}) + \mathbf{v}_s(\mathbf{k} + \mathbf{d}_p(\mathbf{k})). \tag{1}$$

## 3.1 Remplissage temporal de trou.

En raison du mouvement des objets de premier plan et de la caméra, les occultations de la vue synthétisée, varient dans le temps et produisent différents trous à différents moments. Ainsi, une partie de l'information manquante peut être disponible à différents moments. En exploitant la corrélation temporelle dans la séquence vidéo, il est possible de récupérer ces informations et de réduire la taille et le nombre de trous dans la synthèse.

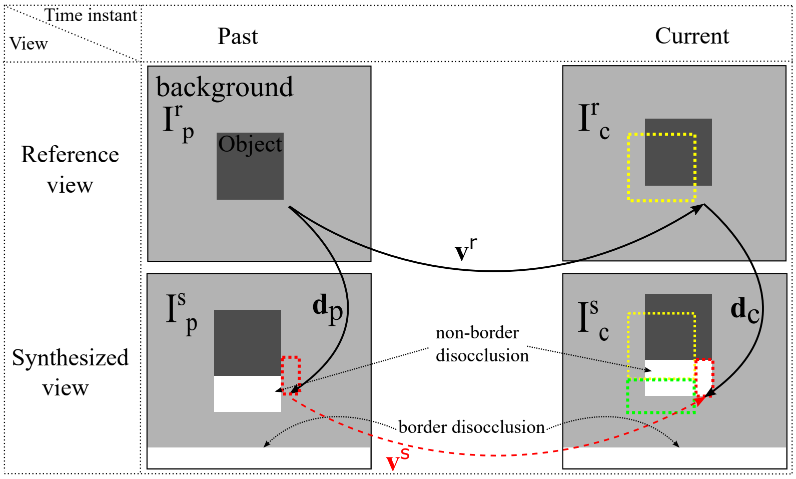


Figure 7 – Récupération temporelle de la zone non-occultées. Les carrés en pointillés jaunes marquent la position de l'objet dans le cadre précédent, le carré en pointillés vert montre les zone non-occultées dans le cadre précédent et les carrés en pointillés rouges indiquent la zone non-occultée qui était visible dans une trame précédente.

Dans la Fig. 7, un objet de premier plan est représenté dans deux vues à deux instants différents, les flèches noires représentent le MVF dans la vue de référence ($r$) et les DVFs pour un instant de temps passé ($p$) et actuel ($c$) ($\mathbf{v}^r$, $\mathbf{d}_p$, $\mathbf{d}_c$). Des lignes pointillées jaunes et vertes indiquent respectivement la position de l'objet et de la zone non-occultée dans le cadre précédent. On peut observer qu'une partie de la zone non-occultée du cadre courant était visible dans un cadre antérieur à cause du mouvement de l'objet (ceci est illustré sur la figure par une ligne pointillée rouge).

Dans la Fig. 8 nous montrons la relation entre MVF et les cartes de disparité pour trois vues d'une séquence MVD. Considérons deux vues de base, gauche ($L$) et droite ($R$), et une vue intermédiaire, qui est synthétisée du côté du décodeur en utilisant des algorithmes DIBR classiques. Les expressions $I^r_{pL}$, $I^r_{cR}$ et $I^s_f$ désignent respectivement les vues gauche, droite et synthétisée d'un instant de temps passé, présent ou futur ($p, c, f$). $\mathbf{v}$ et $\mathbf{d}$ sont les cartes de MVF et de disparité respectivement.

Cette approche a été testée en combinaison avec une technique de déformation de précision sub-pixel, utilisée à la fois pour déformer les vues gauche et droite et pour récupérer des informations temporelles. Des gains allant jusqu' à 1,34dB sur les zones non-occultées et 0,6dBs sur l'ensemble de l'image ont été rapportés, par rapport à la méthode de synthèse de vue recommandée utilisée dans le modèle d'essai 3D-HEVC de MPEG.

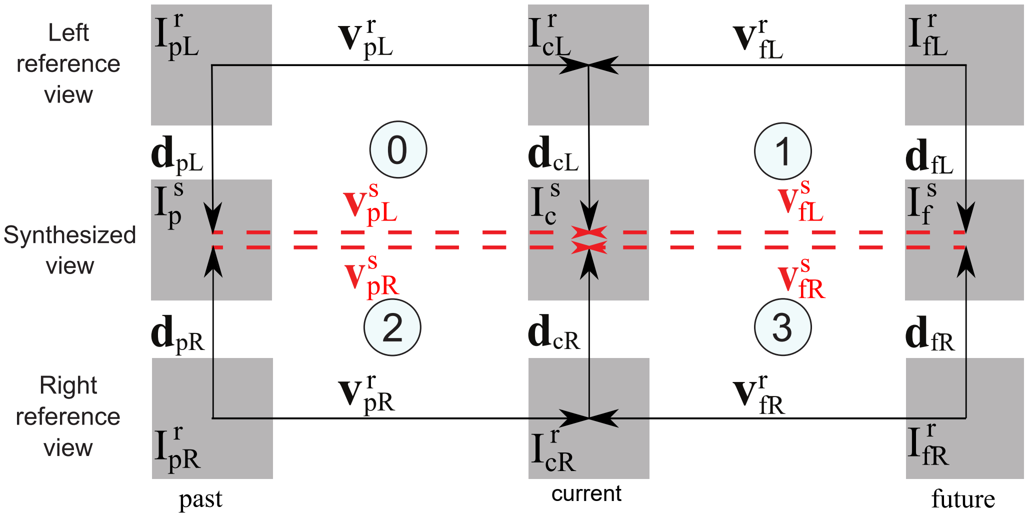


Figure 8 – Schéma de remplissage temporels de trous, pour deux vues de base et une synthèse intermédiaire, utilisant des trames synthétisées passées et futures pour récupérer des informations.

## 3.2 Synthèse de vue basée sur la prédiction temporelle

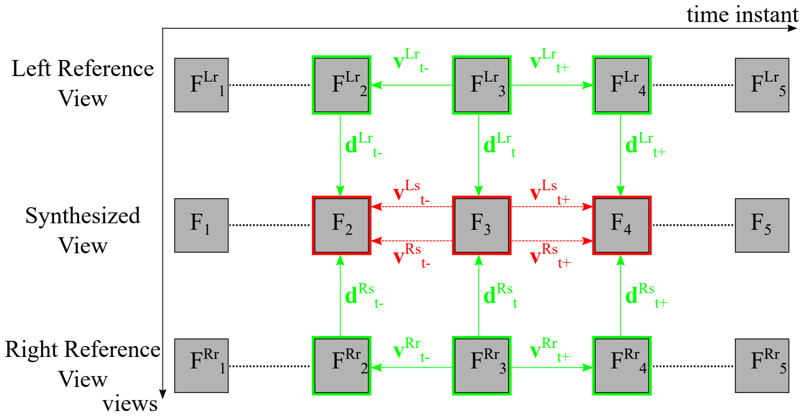


Figure 9 – Schéma de deux vues de référence utilisant des instants passés et futurs $(t_-, t_+)$. Vert: données d'entrée; rouge: l'étape MC.

Afin d'améliorer l'ensemble du cadre synthétisé par opposition aux seules zones non-occultées, nous avons introduit une nouvelle approche couplée à un mélange adaptatif de prévisions inter-vues et temporelles. Dans la Fig. 9, nous montrons le schéma général de la méthode proposée. Considérant une vue de référence gauche $(Lr)$ et une vue de référence droite $(Rr)$, avec leurs cartes de profondeur associées, nous cherchons à synthétiser une vue du milieu. En vert, nous représentons les entrées requises pour obtenir les MVF dans la vue synthétisée: les MVF dans les vues de référence pour un instant de temps passé $(t_-)$ et futur $(t_+)$ $(\mathbf{v}^{Lr}_{t-}, \mathbf{v}^{Lr}_{t+}, \mathbf{v}^{Rr}_{t-}, \mathbf{v}^{Rr}_{t+})$ et les six DVF ($\mathbf{d}_{t_-}^{Lr}$, $\mathbf{d}_{t}^{Lr}$, $\mathbf{d}_{t_+}^{Lr}$, $\mathbf{d}_{t_-}^{Rr}$, $\mathbf{d}_{t}^{Rr}$, $\mathbf{d}_{t_+}^{Rr}$).

En rouge dans la Fig. 9, nous montrons l'étape MC dans laquelle quatre prédictions de la trame courante sont obtenues en utilisant les quatre MVFs. Le schéma rouge et vert peut alors être itéré à travers tous les cadres de la vue synthétisée. Notez que la distance temporelle entre la prédiction et la référence dans le processus ME est constante et définie à 1 dans la Fig. 9. Comme chaque trame a des références temporelles différentes, l'algorithme nécessite une première synthèse DIBR.

## 3.3 Synthèse de vues exploitant la prédiction temporelle pour 3D-HEVC.

Comme les cadres de référence utilisés pour la prédiction temporelle sont également synthétisés, les gains sont quelque peu limités. Étant donné que certaines vues qui sont reconstituées par synthèse côté décodeur sont en fait disponibles côté codeur, nous pourrions maximiser l'efficacité de la prédiction temporelle en envoyant des informations supplémentaires sur la vue synthétisée. Principalement, nous envoyons une image codé intra par GOP pour la vue synthétisée.

Figure 10 illustre les étapes de l'algorithme de Synthese de Vues exploitant la Prédiction Temporelle (VSTP). Pour générer une prédiction temporelle, l'algorithme saisit deux trames de la vue de référence à deux instants, (i.e., un instant temporel actuel et futur ou passé, indiqué par $I_{c,L}^{r}$ et $I_{p,L}^{r}$ respectivement dans la figure) et calcule un MVF dense entre les deux ($\mathbf{v}_{r,p,L}$). Le MVF dense est ensuite déformée au niveau de la vue synthétisée à l'aide des cartes de disparité correspondantes ($\mathbf{d}_{c,L}$ et $\mathbf{d}_{p,L}$). Nous conservons également une carte des disparités correspondant au nouveau MVF ($\mathbf{d}'$). Ainsi, chaque pixel a un MV et un DV associés. L'étape suivante est le MC vers l'arrière, dans lequel on utilise une image clé ($I_p^s$) comme référence pour obtenir une première prédiction temporelle. En cas de superposition de valeurs, on utilise $\mathbf{d}'$ pour sélectionner le pixel de premier plan. Les valeurs $\widehat{I}_{p,R}^{s}$, $\widehat{I}_{f,L}^{s}$, $\widehat{I}_{f,R}^{s}$ sont obtenues en utilisant les mêmes étapes dans la vue de référence de droite au même instant de temps et à un instant futur dans les vues de référence de gauche et de droite respectivement, comme illustré dans la Figure 8. La synthèse finale est obtenue en effectuant une simple fusion entre les quatre prédictions temporelles ou une fusion inter-vue/temporelle. La prédiction de l'inter-vue est indiquée par $\widehat{I}^{i}$ dans la Figure 10.

Figure 11 montre la différence entre les deux schémas de prédiction temporelle. Le schéma "Direct" utilise la trame clé du GOP actuel et celle du GOP suivant comme

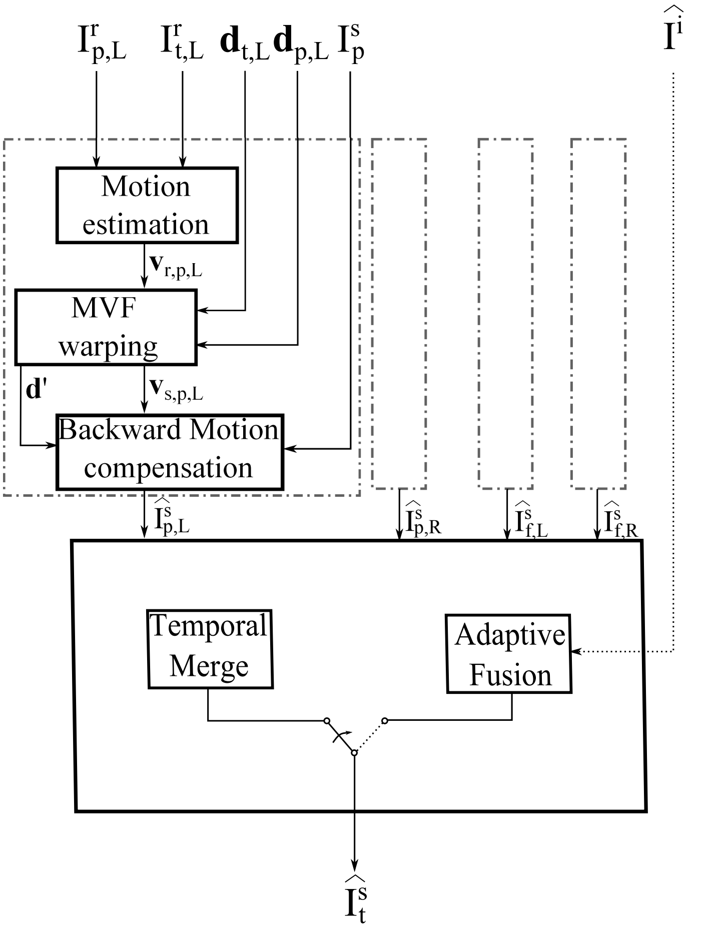
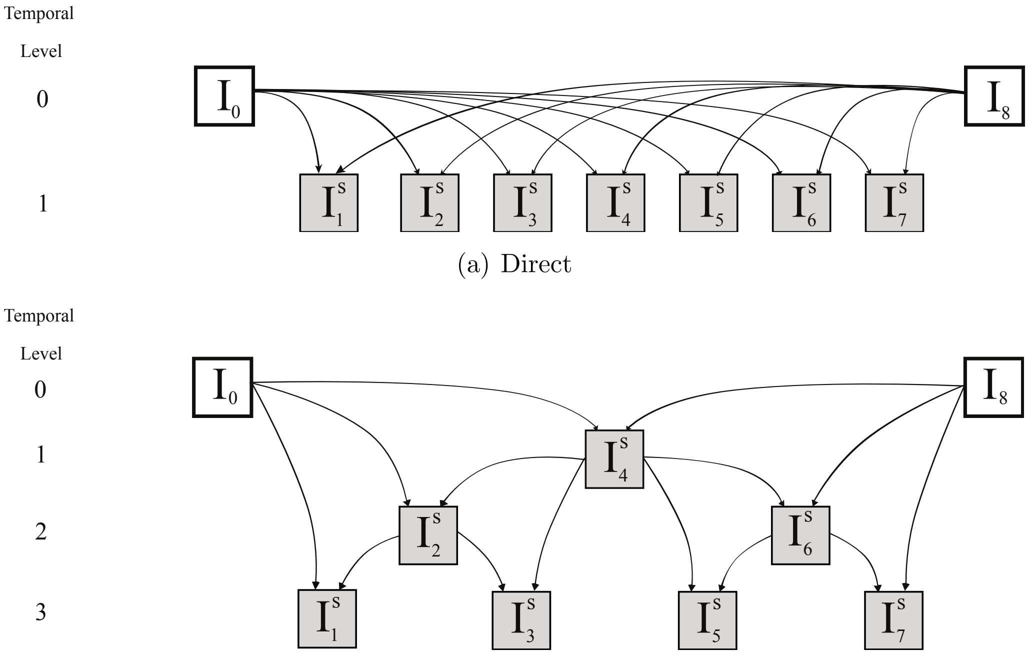


Figure 10 – Diagramme de flux pour la synthèse de vues exploitant la prédiction temporelle (VSTP).



(a) Direct

(b) Hierarchical

Figure 11 – Temporal prediction schemes inside a GOP of the synthesized view.

cadres de référence passés et futurs pour toutes les images restantes à synthétiser dans le GOP. Il en résulte une prédiction asymétrique, avec deux distances temporelles différentes entre chacune des deux images clés et l'image courante. On peut utiliser un schéma alternatif, appelé schéma hiérarchique, dans lequel les couches temporelles sont utilisées pour effectuer des prédictions symétriques (avec des distances temporelles égales). La distance temporelle maximale dans ce schéma est égale à la moitié de la taille du GOP.

Tableau 1 donne les valeurs BD-PSNR obtenues avec les deux schémas de prédiction avec fusion simple ("Direct" et "Hierarchical") et "Adaptive Fusion" appliquées dans le schéma "Hierarchical" ("HierarchicalAF") lorsque l'on considère seulement le PSNR de la vue intermédiaire 1/2 synthétisée avec VSTP. Dans le Tableau 2 nous montrons le BD-PSNR pour les 3 vues intermédiaires. Ici, le PSNR est calculé comme la moyenne entre les 3 (1/4, 3/4 synthétisé avec VSRS-1DFast et 1/2 avec VSTP). Une valeur positive dans ce tableau indique un gain. En moyenne, notre méthode apporte en moyenne une augmentation de 0,53dB, 0,59dB et 0,87dB de BD-PSNR avec les schémas "Direct" et "Hiérarchique" avec fusion des prédictions temporelles simples, et le schéma "Hiérarchique" avec la méthode "Fusion adaptative" respectivement, par rapport à la méthode de référence VSRS-1DFast. Dans la dernière colonne du tableau (HierAF+HierSynth) nous montrons le BD-PSNR obtenu si nous synthétisons les vues virtuelles 1/4 et 3/4 à partir de la vue de base gauche et de notre synthèse VSTP, et à partir de la synthèse VSTP et de la vue de base droite respectivement. La carte de profondeur pour la vue 1/2 est synthétisée à partir des vues de base droite et gauche. En utilisant cette synthèse hiérarchique, nous profitons de la qualité supérieure de notre méthode de rendu pour améliorer les vues 1/4 et 3/4 sans modifier le bitrate. Le delta-PSNR entre la référence et la nôtre pour des vues à 1/4 et 3/4 est de -0.09dB, -0.01dB, 1.58dB pour les séquences Balloons, Kendo et Newspaper en moyenne sur toutes les QPs. Comme on s' y attendait, ces résultats concordent avec le BD-PSNR indiqué dans le tableau 2 (HierAF+HierSynth par rapport à HierarchicalAF), puisque le taux n'est pas modifié.

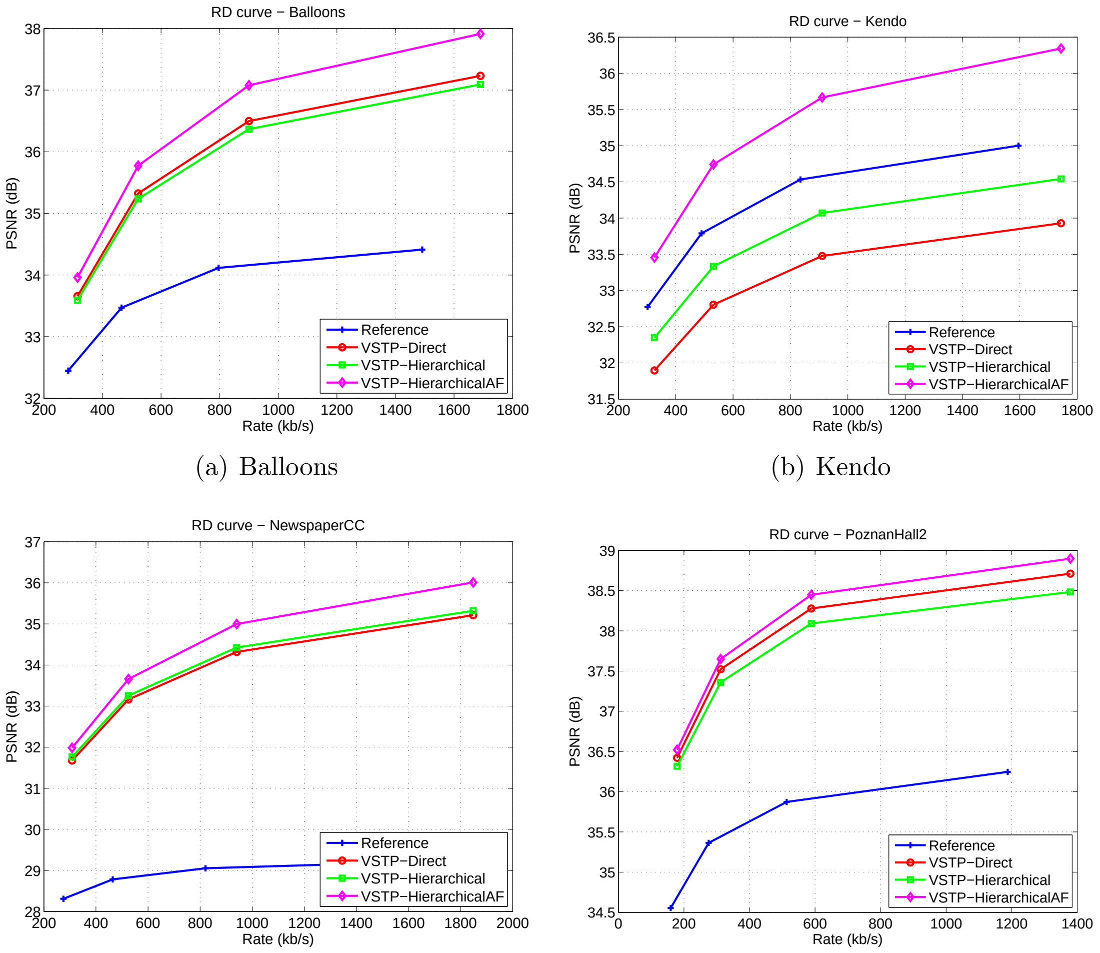
| **Sequence** | **BD-PSNR (in dB)** | | |
|---|---|---|---|
| | Direct | Hierarchical | HierarchicalAF |
| Balloons | 1.94 | 1.84 | 2.45 |
| Kendo | -1.12 | -0.56 | 0.93 |
| Newspaper | 4.70 | 4.80 | 5.28 |
| PoznanHall2 | 2.17 | 1.99 | 2.32 |
| **Average** | **1.92** | **2.01** | **2.74** |

Table 1 – Valeurs BD-PSNR pour un test, à 3 vues, obtenues avec des schémas de prédiction et de fusion adaptative dans la méthode proposée par rapport à la méthode de référence VSRS-1DFast.

| **Sequence** | **BD-PSNR (in dB)** | | | |
|---|---|---|---|---|
| | Direct | Hierarchical | HierarchicalAF | HierAF + HierSynth |
| Balloons | 0.52 | 0.49 | 0.69 | 0.64 |
| Kendo | -0.45 | -0.27 | 0.22 | 0.22 |
| Newspaper | 1.52 | 1.55 | 1.71 | 2.78 |
| **Average** | **0.53** | **0.59** | **0.87** | **1.21** |

Table 2 – Valeurs BD-PSNR pour un cas de test à 5 vues, obtenues avec les deux schémas de prédiction, la fusion adaptative et la synthése hiérarchique dans la méthode proposée par rapport à la méthode de référence VSRS-1DFast.

Les courbes de débit-distorsion (RD) du scénario de test à 3 vues pour la référence et la méthode proposée (pour les deux schémas et les méthodes de fusion) sont données dans la Figure 13, tandis que les 5 courbes RD du scénario de test de vue sont montrées dans la Figure 12. Ce scénario comprend également un schéma hiérarchique au sens inter-vues (HierAF+HierSynth).



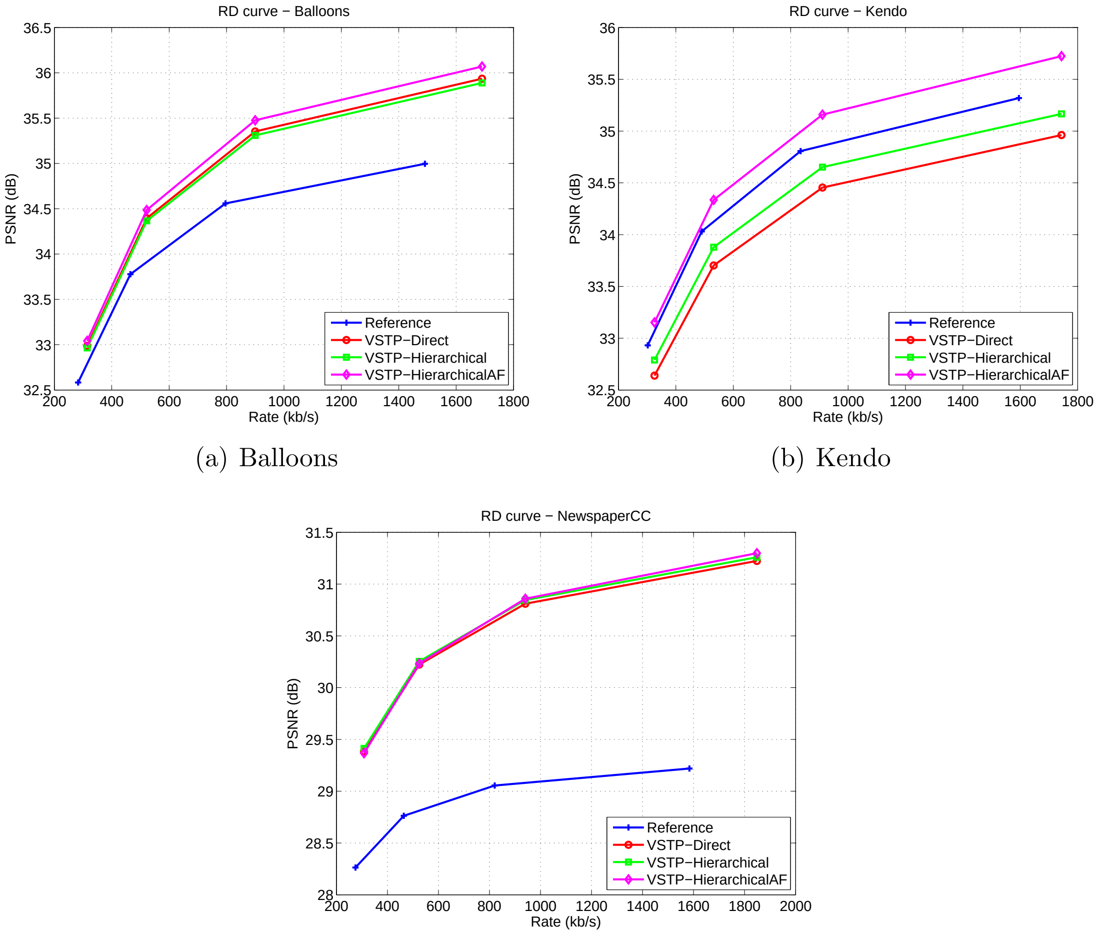
(a) Balloons

(b) Kendo

(c) Newspaper

(d) PoznanHall2

Figure 12 – Courbes RD de la référence et de la méthode proposée sur le scénario de test 3 vues pour les séquences Balloons, Kendo, NewspaperCC et PoznanHall2.



(a) Balloons

(b) Kendo

(c) Newspaper

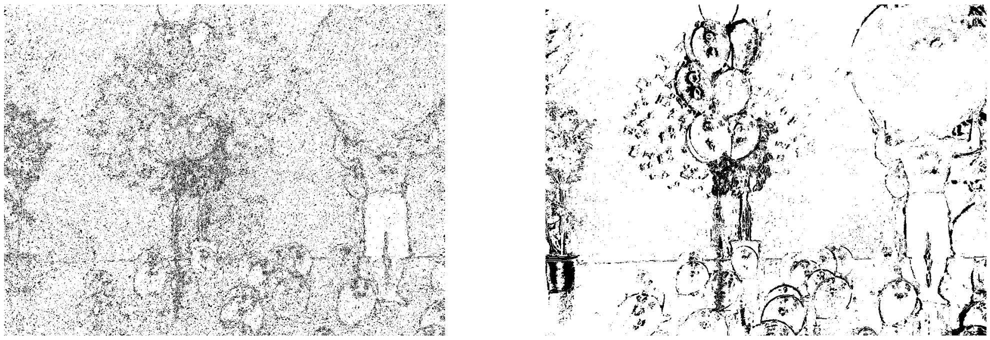
Figure 13 – Courbes RD de la référence et de la méthode proposée sur le scénario de test 5 vues pour les séquences Balloons, Kendo et NewspaperCC.

# 4 Évaluation de la qualité basée sur la région d'intérêt pour les techniques de synthèse de vue

Parce que les distorsions produits par la synthèse sont intrinsèquement différents de ceux du codage, l'évaluation de la qualité de la synthèse dans les systèmes utilisant le rendu DIBR n'est pas une question triviale. En particulier, l'objectif final de tels systèmes est de fournir une expérience 3D. Des mesures telles que le rapport signal de crête sur bruit (PSNR) fournissent une bonne évaluation objective, mais ne mettent pas l'accent sur les erreurs causées par les distorsions d'objet. Des méthodes d'évaluation qui prennent en compte la structure de l'image ont été créées, l'une des plus populaires étant la métrique basée sur la similarité structurelle (SSIM). Cependant, de petites distorsions sur une image peuvent masquer l'impact des distorsions localisées causées par la synthèse.

Notre objectif est d'évaluer comparativement plusieurs méthodes de synthèse de vues. Comme la plupart des méthodes d'évaluation, nous considérons que la référence est connue. Bien que cela ne soit pas vrai pour une vue virtuelle, il est généralement préférable, aux fins d'évaluation, de synthétiser une vue existante d'une séquence vidéo MVD. Ce chapitre propose une nouvelle technique d'évaluation de synthèse de vues, basée sur le SSIM, qui met l'accent sur la comparaison des artefacts de synthèse de vues autour de zones sensibles de l'image, sujettes aux erreurs. Deux méthodes différentes sont utilisées pour sélectionner les domaines d'intérêt dans l'évaluation de deux méthodes de synthèse. Tout d'abord, nous analysons la distribution des erreurs et séparons les erreurs de synthèse des erreurs de quantification. Une deuxième approche est axée sur l'évaluation directe des zones prédites différemment par les deux méthodes testées. Nous montrons cette technique pour apporter une meilleure différenciation des méthodes de synthèse par rapport à l'impact des artefacts de synthèse sur la qualité de l'image. De plus, des informations supplémentaires peuvent être extrapolées sur la localisation spatiale des distorsions par rapport à une évaluation SSIM ou PSNR.

Lors de l'essai de deux méthodes de synthèse de vues, une première façon de sélectionner les zones sujettes à des erreurs de synthèse serait de rechercher les pixels qui présentent une erreur absolue relativement élevée. Ceci peut fournir une bonne indication sur la qualité des méthodes de synthèse. Les erreurs produites par la quantification lors de l'encodage des vues de référence et les erreurs causées par la quantification de la carte de profondeur ou le processus d'interpolation sont généralement uniformément réparties et ne dépendent pas nécessairement de la structure de la scène ou de la méthode de synthèse des vues utilisée. Ceci peut également être observé dans la Fig. 14 où deux masques binaires sont affichés. Le noir indique les pixels dont l'erreur absolue est plus grande que le double de l'erreur absolue moyenne. La Fig. 14(a) affiche le masque d'une trame encodée avec 3D-HEVC à QP 25 et la Fig. 14(b) est obtenu à partir de la même trame synthétisée avec VSRS-1DFast à partir de vues de référence non codées. Il est facile de remarquer que dans le cas de l'encodage, des erreurs importantes sont réparties sur l'ensemble de l'image. Dans le cas de la synthèse, les erreurs sont concentrées et leur positionnement spatial dépend de la structure de la scène. La focalisation de l'évaluation de synthèse sur ces domaines peut fournir une meilleure indication de la qualité de la méthode pour la distorsion des objets, tout en ignorant d'autres sources d'erreurs moins percutantes, comme les erreurs de quantification produites par l'encodage. Le seuil utilisé pour générer le masque doit être choisi de manière à pouvoir séparer les grandes erreurs provenant de la synthèse. Pour ce faire, nous décrivons la histogramme des erreurs absolues pour une vue synthétisée. Dans Fig. 15, comme prévu, nous trouvons un grand pourcentage de pixels avec de petites erreurs. Ceci est normal pour les séquences codées car les erreurs sont normalement distribuées autour de zéro. Cependant, dans la Fig. 15 nous trouvons également une densité d'erreur accrue autour d'une valeur plus grande, marquée par une ligne rouge dans la figure. Ceci est causé par le processus de synthèse. Comme nous l'avons vu, la synthèse introduira des distorsions élevées par rapport aux erreurs de quantification, en particulier pour les QP faibles. Les erreurs de quantification sont limitées en valeur absolue par la moitié de l'intervalle de quantification, tandis que les erreurs de synthèse peuvent être plus élevées. Le seuil peut être déterminé en trouvant la valeur



(a) 3D-HEVC encoding

(b) Synthesis

Figure 14 – Les masques binaires sur l'image 1 de la séquence Balloons, en noir, indiquent des pixels avec des erreurs absolues élevées.

correspondant à une concentration des erreurs les plus élevées. Une autre option pour choisir les emplacements spatiaux pertinents qui doivent être évalués lors de la comparaison des méthodes de synthèse est d'examiner directement les différences entre les méthodes. Nous pouvons sélectionner ces zones en générant un nouveau masque de sélection contenant toutes les zones qui ont été rendues différemment par les deux méthodes.

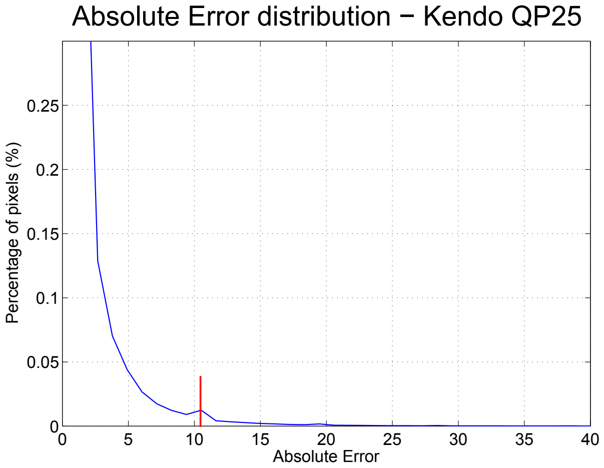


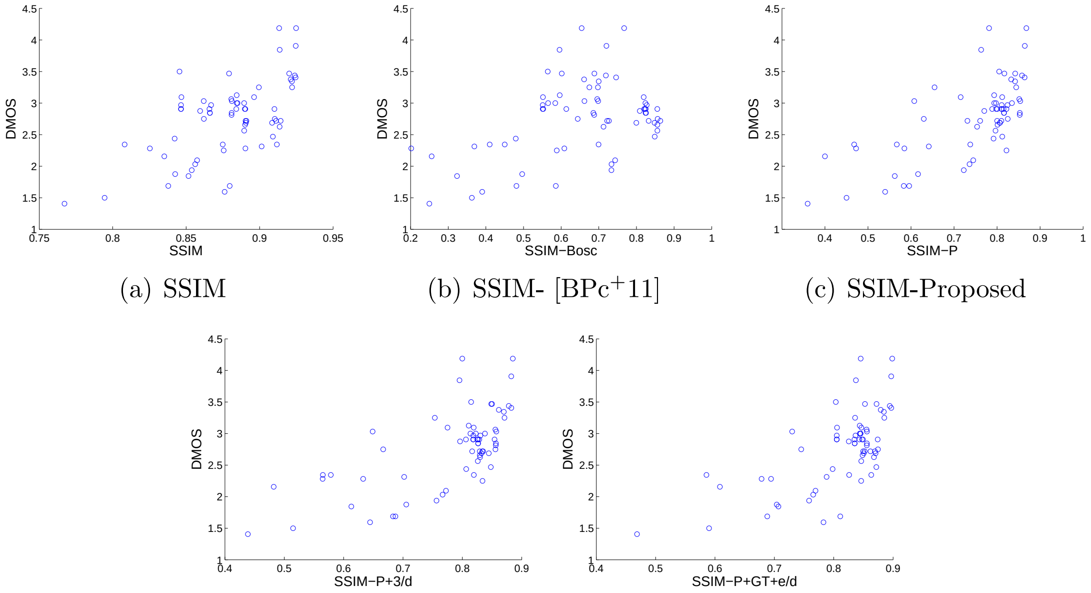
Figure 15 – Distribution d'erreur absolue pour une trame synthétisée dans une séquence de Kendo, à QP = 25.

Fig. 16(a) montre le comportement dans le temps de SSIM et notre technique d'évaluation proposée. Bien que le score SSIM soit relativement similaire d'une trame à l'autre, des variations peuvent être observées pour les scores SSIM avec des masque basé sur le histogramme (SSIM$_{hist}$) et les différences entre les méthodes (SSIM$_{epas}$). L'analyse de ces variations permet d'extrapoler des renseignements supplémentaires sur les forces ou les faiblesses d'une méthode. Regardons, par exemple, à trois instances de temps marquées dans la Fig. 16(a) avec des lignes verticales rouges ($t_1$, $t_2$ et $t_3$, frames 40,58 et 85 respectivement). On peut clairement remarquer une augmentation de SSIM$_{epas}$ à $t_2$ par rapport à $t_1$. C'est conforme aux SSIM, mais à peine perceptible. Regardons les masques SSIM$_{epas}$ pour les deux instances de temps dans les Fig. 16(b) et 16(c) pour identifier la cause. On peut voir la zone sujette à erreur marquée d'un carré rouge dans la Fig. 16(b). Dans la Fig. 16(c) cette zone est obstruée par une personne marchant devant elle et les erreurs sont cachées. Notez également que ΔSSIM$_{epas}$ est plus petit entre les frames 50 et 70. Cela indique que la méthode Wf permet d'obtenir une qualité supérieure à celle du VSRS-1DFast dans ce domaine. Évidemment, lorsque la zone est obstruée, le gain est réduit.

A $t_3$ on peut voir une baisse soudaine de SSIM$_{epas}$ qui n'est pas perceptible dans

SSIM. En regardant le masque de sélection, nous pouvons observer la personne qui s'approche d'un autre objet de premier plan qui est identifié comme une zone sujette aux erreurs par le masque de sélection. Ceci est marqué d'un carré rouge dans la Fig. 16(d). Ce type d'artefact apparaît en raison de la proximité des deux objets au premier plan. La zone située entre les deux n'est pas visible dans les vues de reference gauche ou droite (i.e. la zone non-occultée). Ces informations supplémentaires sur les méthodes testées, en fonction de la géometrie de la scène et des zones à risque d'erreur, ne peuvent pas être facilement extrapolées en utilisant uniquement SSIM ou PSNR.

La dernière partie de ce chapitre évalue la technique de génération de masque proposée en utilisant une base de données d'évaluation subjective de synthèse de vue. Nous générons les masques $SSIM_{epas}$ (P) pour évaluer plusieurs méthodes de synthèse. Pendant la génération du masque, nous utilisons également la vérité terrain ($GT$) et nous appliquons une opération d'érosion et de dilatation (e/d) pour éliminer les pixels isolés sélectionnés. Dans la Figure 17 on montre les diagrammes de dispersion pour SSIM et le ROI SSIM avec les masques binaires: [BPc$^+$11], P, P+e/d et P+GT+e/d. Chaque point représente le DMOS par rapport à la moyenne du score objectif sur toutes les trames d'une séquence/vue/méthode. Une amélioration peut être observée en utilisant l'approche proposée.

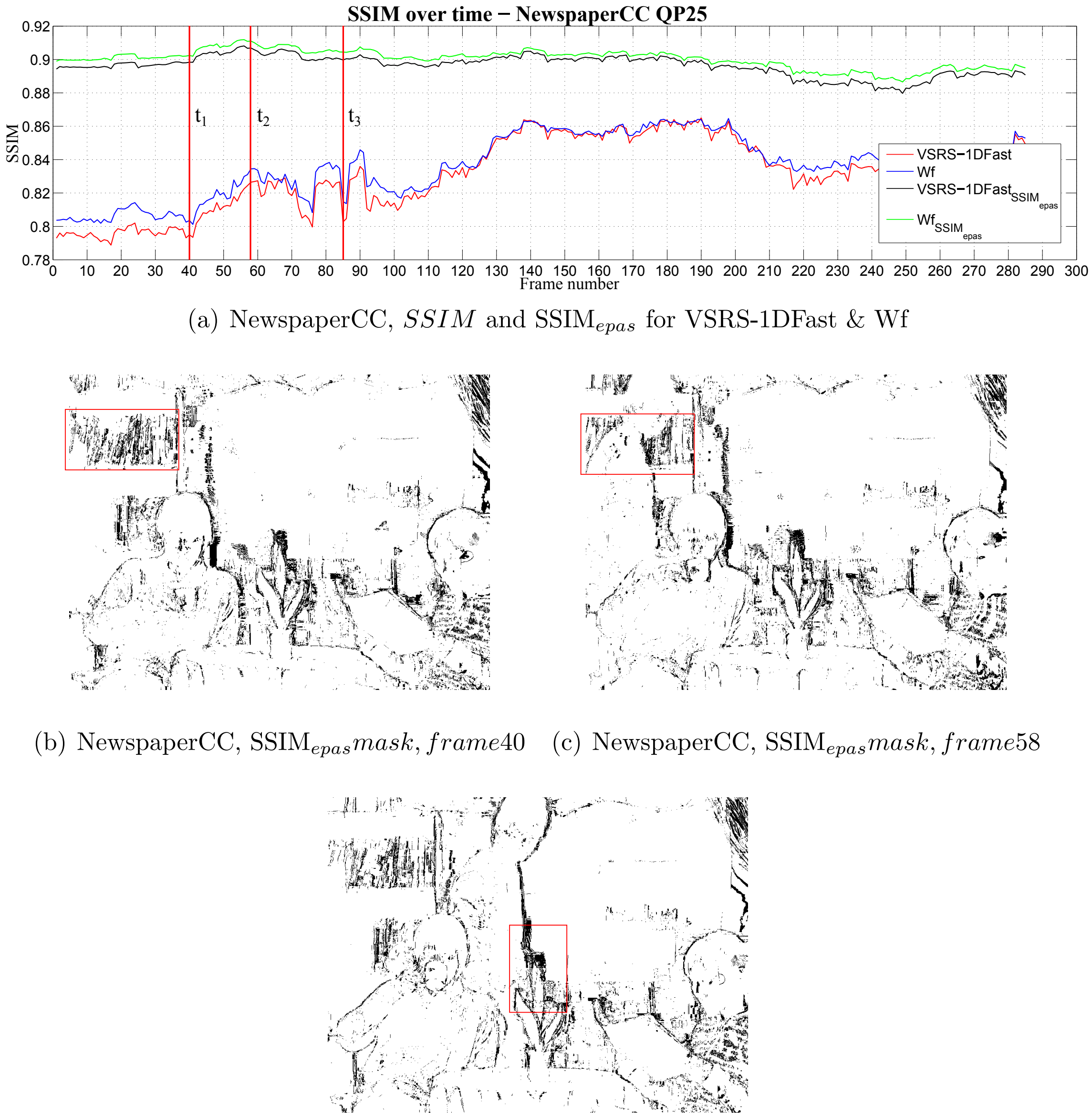


(a) SSIM (b) SSIM- [BPc$^+$11] (c) SSIM-Proposed

(d) SSIM-Proposed+e/d (e) SSIM-Proposed+GT+e/d

Figure 17 – Nuage de points des résultats objectifs pour le SSIM avec des ROI testés. Chaque point est le DMOS par rapport au score objectif moyen sur toutes les images pour une séquence, une synthèse et une méthode.

Bien que nous ayons pu démontrer que l'évaluation du region d'intêret (ROI) peut améliorer le rendement des mesures traditionnelles pour les images de synthèse, nos constatations indiquent que l'évaluation objective de la qualité de la synthèse visuelle demeure un sujet ouvert. Compte tenu des importantes incohérences entre les résultats objectifs et subjectifs, pour certaines méthodes de synthèse (e. g. les méthodes qui fournissent une image sans zone non-occultée mais qui ne sont pas géométriquement cohérentes), nous sommes amenés à conclure que la normalisation de l'évaluation subjective des séquences vidéo multivues et de la synthèse visuelle joue un rôle crucial. Toutefois, étant donné qu'une méthode de diffusion de contenu multi-vues au grand public n'est pas encore bien définie et que de multiples options sont encore à l'étude, les conditions d'évaluation subjectives et le rôle de la synthèse des vues pourraient changer radicalement à l'avenir.



(a) NewspaperCC, $SSIM$ and SSIM$_{epas}$ for VSRS-1DFast & Wf

(b) NewspaperCC, SSIM$_{epas}mask, frame40$ (c) NewspaperCC, SSIM$_{epas}mask, frame58$

(d) NewspaperCC, SSIM$_{epas}$ mask, frame 85

Figure 16 – Figure 16(a) - SSIM and SSIM$_{epas}$ au fil de temps. Figures 16(b), 16(c) and 16(d) montre les masques de sélection pour SSIM$_{epas}$.

# 5 Amélioration de la qualité et de la résolution vidéo à partir d'une vidéo compressée multi-sources

Ce chapitre aborde le problème de la reconstruction vidéo et de l'amélioration de la résolution. Le scénario est similaire à la situation rencontrée dans la synthèse en vue avec quatre prédictions temporelles et deux prédictions inter-vues d'une trame sans connaître la référence. Dans ce cas, il s'agit de multiples descriptions compressées d'une séquence vidéo. Chaque description peut être soumise à un certain niveau de compression avec des codeurs vidéo (VC) hybrides. De plus, les vidéos peuvent avoir des résolutions différentes. Pour résoudre ce problème, nous utilisons les caractéristiques clés qui régissent les codeurs vidéo hybrides et modélisons le problème comme une optimisation convexe, en construisant un cadre pratique capable de reconstruire et d'améliorer une séquence vidéo à partir de sources multiples.

Nous décrivons un modèle du problème de super-résolution dans la Fig. 18. A partir d'une séquence vidéo originale, nous appliquons différents modèles de dégradation qui consistent à sous-échantillonner ($L$) et comprimer la source avec un VC. Quatre opérations essentielles sont traditionnellement impliquées dans un codeur vidéo: la prédiction, la transformation, la quantification et le codage entropique. L'étape de prédiction permet une compression efficace des redondances présentes dans le signal source. Ensuite, une transformée linéaire vise à réduire davantage les corrélations dans le signal résiduel et compacte l'énergie dans un nombre limité de coefficients. Lors de la quantification, les coefficients de transformation sont mappés à un ensemble fini de mots-codes. Enfin, le codage par entropie exploite les redondances statistiques restantes dans les mots codés résultants et génère la représentation binaire du signal vidéo.

Nous proposons alors de minimiser le critère suivant, avec un paramètre $\beta \in [0;+\infty[$ permettant d'équilibrer les fonctions de coût:

$$\begin{aligned}\text{Find }\ \widehat{x} \in \underset{x\in\mathbb{R}^{K\times N}}{\operatorname{argmin}} \Bigg( & J_{\mathrm{DF}}(x) + \beta J_{\mathrm{SR}}(x) + \\ & \sum_{i=1}^{K}\sum_{m=1}^{M} \Big(\iota_{C_{m,i}}(T_{m,i}(L_{m,i}x_i - \widetilde{x_{m,i}}))\Big) + \\ & \sum_{i=1}^{K}\sum_{m=1}^{M} \Big(\sum_{s=1}^{S} \iota_{D_{s(m,i)}}(F_s x_i)\Big)\Bigg), \quad (2)\end{aligned}$$

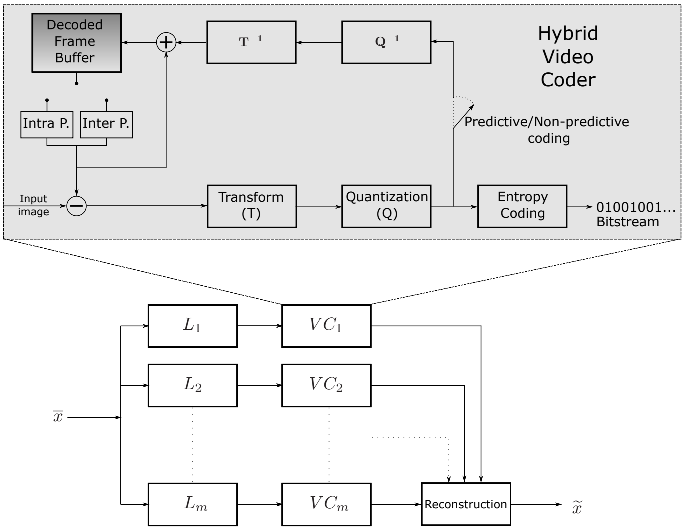


Figure 18 – Un modèle générique pour le sous-échantillonnage, la compression et la reconstruction de plusieurs sources vidéo.

où $m$ et $i$ représentent le flux vidéo et le numéro de trame, $\iota$ représentent la fonction caractéristique d'un ensemble convexe fermé, $T$ et $L$ représentent les opérateurs de transformation et de sous-échantillonnage et $\tilde{x}$ le résidu d'une trame.

La fidélité d'une observation est évaluée dans le domaine de la transformation. En l'absence d'indices supplémentaires, les niveaux de reconstruction représentent la meilleure référence de qualité (ce qui minimise l'erreur) pour la solution dans chaque domaine de transformation. Nous optons pour le choix raisonnable de minimiser la somme des distances entre les projections de la solution sous-échantillonnée sur les bases de transformation et les coefficients transformés quantifiés correspondants observés dans le flux binaire comprimé ($z$), selon une métrique appropriée $\phi_m$. Finalement, pour tenir compte de la fiabilité inégale des niveaux de reconstruction pour chaque version encodée, nous utilisons un paramètre supplémentaire $\alpha_m$:

$$J_{\mathrm{DF}}(\mathbf{x}) = \sum_{i=1}^{K} \sum_{m=1}^{M} \alpha_m \phi_m \left( T_{m,i}(L_{m,i}\widehat{x_i} - \widetilde{x_{m,i}}) - z_{m,i} \right). \qquad (3)$$

$J_{\mathrm{SR}}$ est utilisé pour équilibrer le critère de minimisation avec un *super-resolution prior*. A cet effet, considérons un ensemble d'opérateurs de suréchantillonnage $H_{m,i}$, qui peuvent être choisis pour adapter/compenser de manière optimale les opérateurs de sous-échantillonnage correspondants $L_{m,i}$. La super-résolution prior est défini ici comme la distance de la solution $\widehat{x}$ par rapport à ses versions successivement sous-échantillonnées et sur-échantillonnées, selon une métrique appropriée $\psi_m$, à savoir:

$$J_{\mathrm{SR}}(\mathbf{x}) = \sum_{i=1}^{K} \sum_{m=1}^{M} \psi_m \left( (\mathrm{Id}\ - H_{m,i} L_{m,i}) \widehat{x_i} \right). \tag{4}$$

Notez que $F_s$ introduit les contraintes d'amplitude et de fluidité telles que définies dans les Eq. (5) et Eq. (6), d'où $S = 2$ dans notre cas. La contrainte d'amplitude est directement appliquée à l'image:

$$F_1 = \mathrm{Id}\,,$$

$$D_1(m,i) = \{x \in \mathbb{R}^N : x^{(k)} \in [x_{\min}^{m,i}, x_{\max}^{m,i}]\ \forall k \in [1,N]\} \tag{5}$$

Pour la contrainte de fluidité isotropique basée sur la variation totale (TV), le gradient de l'image doit être calculé ($\nabla_h, \nabla_v$ étant les opérateurs de gradient horizontal et vertical respectivement):

$$F_2 = (\nabla_h, \nabla_v), D_2(m,i) = \{x \in \mathbb{R}^N : \sum_{k=1}^{N} \sqrt{\nabla_h^2 x^{(k)} + \nabla_v^2 x^{(k)}} \leq \eta_i\}. \tag{6}$$

Étant donné que notre problème est basé sur des opérateurs linéaires, notre choix de solveur repose sur l'algorithme primal dual, connu sous le nom d'algorithme Monotone Lipschitz ForwardBackward-Forward (M-LFBF). Cet algorithme, contrairement à d'autres méthodes similaires, assure une plus faible complexité de calcul pour les problèmes impliquant des opérateurs linéaires car il ne nécessite pas d'inversion de matrice. De plus, la structure itérative de bloc de l'algorithme permet des implémentations parallèles efficaces sur des architectures multicoeurs.

Dans un premier scénario, nous examinons la question des SR à partir de deux observations à faible résolution. La configuration expérimentale suit la Figure 19, et les deux observations sont générées par un sous-échantillonnage (par un facteur de 2 dans chaque dimension) d'une séquence d'entrée donnée en utilisant BicAA et BicNAA. Deux configurations de codage sont spécifiquement analysées, désignées par II et IP. Le mode II correspond à une configuration Intra: chaque trame de la séquence est traitée comme une trame Intra indépendante, sans outil d'estimation de mouvement et de compensation. Cependant, dans le cas du HEVC, les images I utilisent la prédiction spatiale Intra. A l'inverse, la configuration IP exploite les trames P pour améliorer l'efficacité de codage, et dans ce cas, une taille GOP de 8 a été choisie. Les évaluations sont effectuées sur 6 séquences CIF (352x288).

Le tableau 3 met en évidence l'efficacité du cadre proposé dans le scénario testé. Tout d'abord, nous montrons une amélioration significative par rapport à la référence

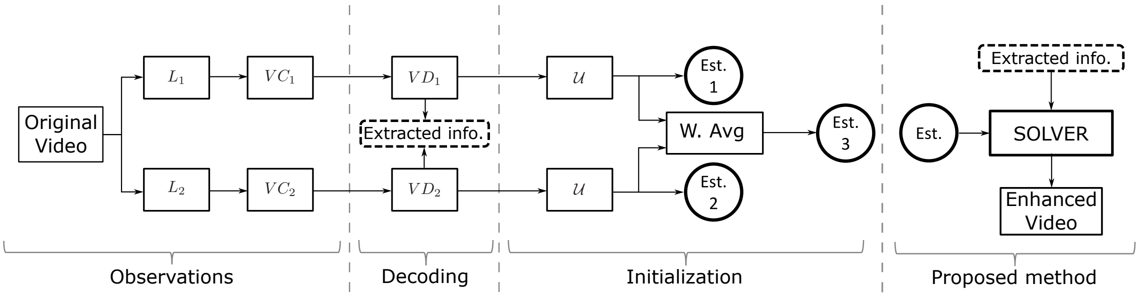


Figure 19 – Une vue schématique de la configuration expérimentale. Deux opérateurs de rééchantillonnage sont appliqués sur une séquence d'entrée. Chaque observation est compressée et décompressée, et des informations utiles sont extraites. Les observations décodées sont suréchantillonnées à leur résolution d'origine en utilisant l'inverse de l'opérateur de dégradation ou une méthode SOA SR. Ensuite, le cadre proposé est initialisé avec une estimation Ã haute résolution et les informations extraites pendant le décodage.

obtenue en faisant la moyenne des sur-échantillonnages bicubiques. Ce résultat tend à démontrer que les informations complémentaires recueillies à partir de chaque observation sont avantageusement utilisées par le cadre proposé. Les résultats du SSIM sont également présentés dans ce tableau. Toutefois, il convient de noter qu' à des QP élevées, les méthodes ont tendance à afficher des performances similaires, car le niveau de compression élevé combiné à l'opération de sous-échantillonnage conduit à une description très peu fiable pour déduire des informations supplémentaires.

| | | | QP1 | | | QP15 | | | QP25 | | | QP35 | | | Average | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Sequence | Mode | Ref | SOA | Prop. | Ref | SOA | Prop. | Ref | SOA | Prop. | Ref | SOA | Prop. | Ref | SOA | Prop. |
| PSNR (dB) | akiyo | II | 34.76 | 37.40 | **41.11** | 34.65 | 37.19 | **39.16** | 34.13 | 36.27 | **36.92** | 31.81 | 32.71 | **32.8** | 33.84 | 35.89 | **37.5** |
| | | IP | 34.76 | 37.4 | **40.97** | 34.62 | 37.1 | **38.84** | 34.16 | 36.24 | **36.87** | 32.17 | 33.13 | **33.14** | 33.93 | 35.97 | **37.46** |
| | foreman | II | 32.15 | 33.28 | **38.29** | 32.08 | 33.2 | **36.42** | 31.58 | 32.53 | **33.97** | 29.55 | 29.94 | **30.09** | 31.34 | 32.24 | **34.69** |
| | | IP | 32.15 | 33.28 | **38.06** | 32.01 | 33.12 | **36.03** | 31.25 | 32.07 | **32.99** | 28.78 | 29.06 | **29.06** | 31.05 | 31.88 | **34.03** |
| | bus | II | 26.84 | 27.93 | **31.91** | 26.82 | 27.91 | **30.48** | 26.62 | 27.70 | **28.83** | 25.23 | 25.99 | **26.16** | 26.38 | 27.38 | **29.35** |
| | | IP | 26.83 | 27.93 | **31.8** | 26.79 | 27.91 | **30.21** | 26.37 | 27.46 | **27.91** | 24.14 | 24.64 | **24.55** | 26.03 | 26.99 | **28.62** |
| | mobile | II | 22.69 | 23.7 | **27.6** | 22.68 | 23.69 | **26.54** | 22.61 | 23.61 | **25.05** | 21.97 | 22.83 | **23.16** | 22.49 | 23.46 | **25.59** |
| | | IP | 22.69 | 23.7 | **27.54** | 22.67 | 23.69 | **26.26** | 22.49 | 23.57 | **24.51** | 21.1 | 21.67 | **21.76** | 22.24 | 23.16 | **25.02** |
| | football | II | 28.01 | 29.85 | **33.27** | 27.99 | 29.82 | **31.83** | 27.72 | 29.46 | **30.37** | 26 | 27 | **27.18** | 27.43 | 29.03 | **30.66** |
| | | IP | 28.01 | 29.85 | **33.13** | 27.96 | 29.77 | **31.52** | 27.45 | 29.01 | **29.24** | 24.54 | 24.93 | **25.02** | 26.99 | 28.39 | **29.73** |
| | flower | II | 22.97 | 23.22 | **26.55** | 22.97 | 23.21 | **25.95** | 22.92 | 23.17 | **24.70** | 22.46 | 22.75 | **23.05** | 22.83 | 23.09 | **25.06** |
| | | IP | 22.97 | 23.22 | **26.51** | 22.96 | 23.21 | **25.77** | 22.84 | 23.15 | **24.29** | 21.92 | 22.09 | **22.18** | 22.67 | 22.92 | **24.69** |
| Average | | | **27.90** | **29.23** | **33.06** | **27.85** | **29.15** | **31.58** | **27.51** | **28.69** | **29.64** | **25.80** | **26.4** | **26.51** | **27.27** | **28.37** | **30.20** |
| SSIM | akiyo | II | .9642 | .978 | **.9816** | .9592 | .9729 | **.9743** | .9472 | .9595 | **.958** | .9066 | .9138 | **.9143** | .9443 | .9561 | **.957** |
| | | IP | .9642 | .978 | **.9813** | .959 | .9723 | **.9728** | .9484 | .9602 | **.9593** | .9136 | .9206 | **.921** | .9463 | .9578 | **.9586** |
| | foreman | II | .9402 | .9551 | **.9654** | .9356 | .9503 | **.9517** | .909 | .9205 | **.9213** | .851 | .8565 | **.8585** | .909 | .9206 | **.9242** |
| | | IP | .94 | .955 | **.9632** | .9319 | .9457 | **.9445** | .9002 | .9094 | **.9044** | .8327 | .8362 | **.8354** | .9012 | .9116 | **.9119** |
| | bus | II | .8524 | .8905 | **.9384** | .8498 | .8881 | **.9134** | .8272 | .8658 | **.8734** | .6996 | .728 | **.7277** | .8073 | .8431 | **.8632** |
| | | IP | .8523 | .8905 | **.9363** | .8472 | .8862 | **.9074** | .8093 | .8473 | **.8372** | .665 | .6886 | **.6859** | .7935 | .8281 | **.8417** |
| | mobile | II | .7873 | .8576 | **.9242** | .7857 | .8557 | **.8989** | .7756 | .8438 | **.8623** | .7136 | .7732 | **.7761** | .7656 | .8326 | **.8654** |
| | | IP | .7872 | .8575 | **.9227** | .7844 | .8539 | **.8906** | .7649 | .8318 | **.8341** | .6599 | .7036 | **.705** | .7491 | .8117 | **.8381** |
| | football | II | .8771 | .9145 | **.943** | .8748 | .9124 | **.924** | .849 | .8874 | **.8847** | .7029 | .7258 | **.7266** | .826 | .86 | **.8695** |
| | | IP | .877 | .9144 | **.9408** | .8724 | .9091 | **.9165** | .8358 | .8694 | **.8509** | .6507 | .6625 | **.662** | .809 | .8389 | **.8425** |
| | flower | II | .8292 | .8645 | **.9225** | .8276 | .8627 | **.9044** | .8192 | .8532 | **.8753** | .7782 | .8087 | **.8097** | .8135 | .8473 | **.878** |
| | | IP | .8291 | .8644 | **.9214** | .8265 | .8615 | **.8995** | .8129 | .8468 | **.8598** | .7562 | .7817 | **.7817** | .8062 | .8386 | **.8656** |
| Average | | | **.875** | **.91** | **.9451** | **.8712** | **.9059** | **.9248** | **.8499** | **.8829** | **.8851** | **.7608** | **.7833** | **.7836** | **.8392** | **.8705** | **.8846** |

Table 3 – Comparaison PSNR de la méthode de référence (bicubic), SOA [TSG15] et du cadre proposé, lorsque deux observations basse résolution sont disponibles. Ces résultats ont été obtenus en utilisant la compression HEVC.

Dans un deuxième scénario, nous considérons le cas où une observation est disponible à bassé résolution (LR) et l'autre à la résolution originale. Notre cadre est capable de combiner naturellement ces observations, puisque chaque observation est modélisée avec son propre modèle de dégradation. En général, les flux codés haute résolution (HR) présentent une qualité supérieure à celle des flux LR sur-échantillonnés, encodés avec des paramètres similaires. Ce comportement conduit à un ΔPSNR important entre les descriptions HR et LR. Intuitivement, si le ΔPSNR est très grand, il n' y a pas beaucoup d'informations qui peuvent être extraites d'une observation LR qui n'est déjà contenue dans la description HR. Par conséquent, nous commençons ce scénario avec un petit test effectué sur quelques images de la séquence Bus, avec VC générique en mode Intra. Notre objectif est d'analyser le comportement des algorithmes à partir du ΔPSNR des deux observations, noté $\Delta_{Obs}$ dans la table 4. $\uparrow_H$ Obs 1, $\uparrow_{SOA}$ Obs 1 et Prop représentent l'observation sur-échantillonnée avec les méthodes $H$ et SOA et le résultat obtenu par la méthode proposée. $\Delta$ est l'amélioration obtenue avec Prop sur Obs 2. La première colonne montre les QP utilisés dans les codages Obs 1 et 2, respectivement. Dans ce test, l'initialisation du solveur M-LFBF était Obs 2. Nous pouvons facilement remarquer que des gains plus élevés sont obtenus lorsque les descriptions sont plus similaires en termes de qualité. Une observation intéressante peut être faite pour les QP 1/20 et 15/20. Même si la qualité de l'Obs 1 augmente seulement de $0,02$ et que l'Obs 2 reste inchangé, nous pouvons constater une grande différence en $\Delta$ (de $0,69$ à $1,47$). Ce comportement peut s'expliquer par la dépendance des algorithmes vis-à-vis de la variété d'informations entre les descriptions plutôt que par leur qualité individuelle. Les tests effectués sur d'autres séquences révèlent un comportement similaire, cependant, par souci de brièveté, nous ne répétons pas ce test pour chaque codeur, configuration et séquence. En tant que tel, nous décidons d'effectuer un ensemble complet de tests en utilisant une combinaison QP qui fournit des observations de qualité similaire: QP 40 pour l'observation HR et QP 1 et 15 pour l'observation LR.

| QPs | $\uparrow_H Obs_1$ | $\uparrow_{SOA} Obs_1$ | $Obs_2$ | $\Delta_{Obs}$ | Prop. | $\Delta$ |
|---|---|---|---|---|---|---|
| 15 20 | 26.41 | 28.56 | 43.76 | **17.35** | 44.45 | **0.69** |
| 1 20 | 26.43 | 28.63 | 43.76 | **17.32** | 45.23 | **1.47** |
| 1 25 | 26.43 | 28.64 | 39.33 | **12.9** | 41.45 | **2.12** |
| 1 30 | 26.43 | 28.63 | 35.15 | **8.72** | 37.63 | **2.48** |

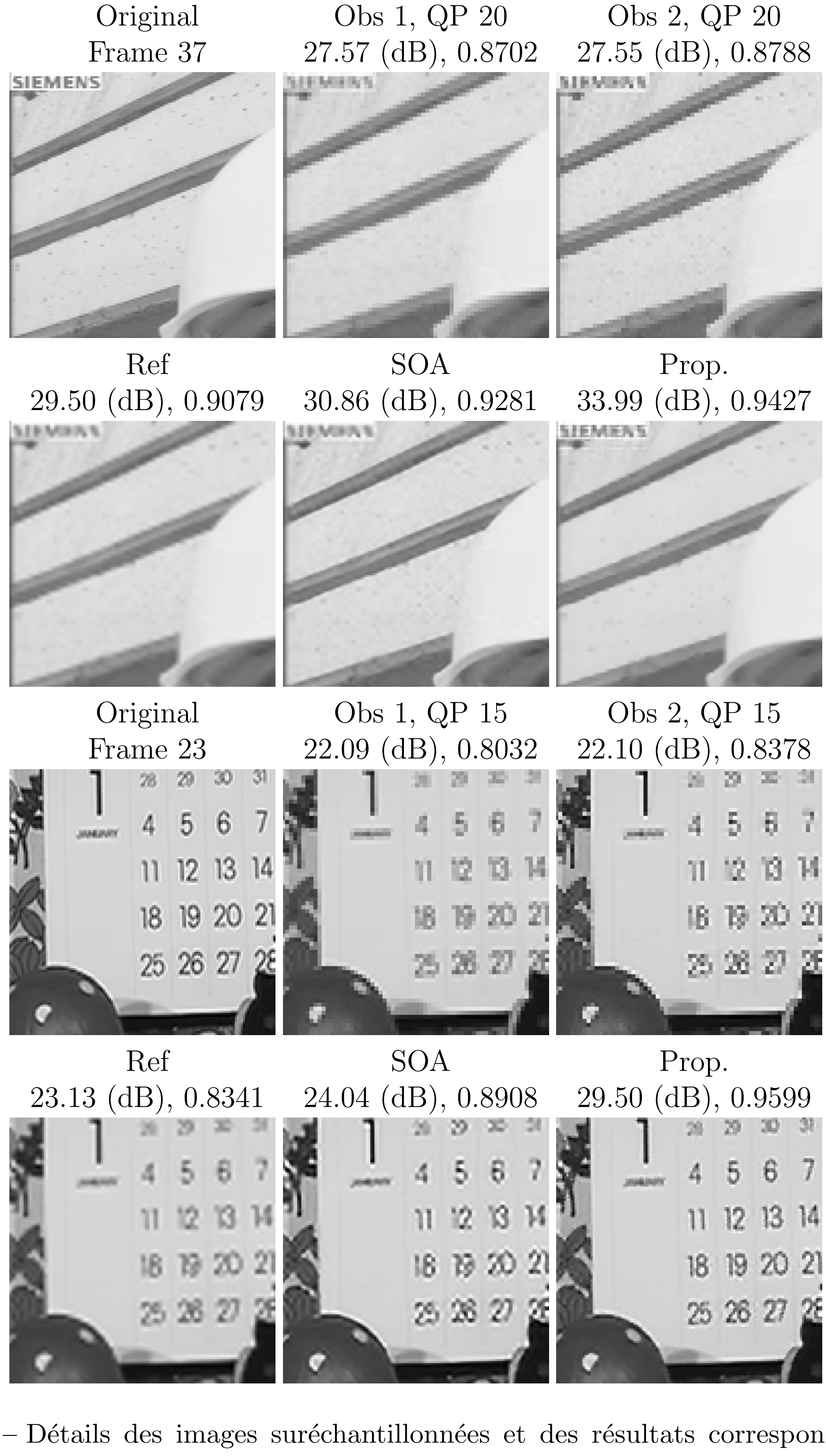
Table 4 – Comparaison PSNR (dB) de différentes combinaisons QP pour une description à basse résolution et à haute résolution sur une séquence de bus avec VC générique utilisant la configuration II.

L'observation LR est obtenue avec le sous-échantillonnage BicAA. Comme la qualité des observations est plus proche de celle de nos tests préliminaires, nous initialisons l'algorithme en utilisant la moyenne. Ref et SOA, dans ce cas, indiquent la moyenne entre l'Obs 2 et l'Obs 1 sur échantillonné avec $H$ et SOA, respectivement. Les résultats sont présentés dans le tableau 5. Notre algorithme surpasse les méthodes de référence et SOA sur toutes les séquences.

Dans la Figure 20 les détails de séquences Foreman et Mobile sont illustrés pour chaque observation et méthode testée. Les résultats PSNR et SSIM sont rapportés pour chaque image. Il est facile de constater que la méthode proposée donne les meilleurs résultats. Le texte qui est presque illisible dans les images super-résolues Ref et SOA est lisible en utilisant l'approche proposée.

| | | | QPs 1 & 40 | | | QPs 15 & 40 | | | Average | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Sequence | Mode | Ref | SOA | Prop. | Ref | SOA | Prop. | Ref | SOA | Prop. |
| PSNR (dB) | akiyo | II | 34.9 | 36.69 | **37.81** | 34.81 | 36.49 | **37.24** | 34.86 | 36.59 | **37.52** |
| | | IP | 35.29 | 37.09 | **38.3** | 35.18 | 36.84 | **37.42** | 35.24 | 36.97 | **37.86** |
| | foreman | II | 32.64 | 33.75 | **36.04** | 32.57 | 33.62 | **35.37** | 32.6 | 33.69 | **35.7** |
| | | IP | 32.28 | 33.25 | **35.19** | 32.16 | 33 | **34.21** | 32.22 | 33.12 | **34.7** |
| | bus | II | 28.1 | 29.38 | **30.31** | 28.08 | 29.33 | **30.19** | 28.09 | 29.36 | **30.25** |
| | | IP | 27.43 | 28.63 | **29.36** | 27.39 | 28.54 | **29.13** | 27.41 | 28.59 | **29.24** |
| | mobile | II | 25.23 | 26.54 | **27.3** | 25.22 | 26.52 | **27.34** | 25.23 | 26.53 | **27.32** |
| | | IP | 25.11 | 26.35 | **27.02** | 25.09 | 26.31 | **27.03** | 25.1 | 26.33 | **27.03** |
| | football | II | 28.93 | 30.59 | **31.69** | 28.9 | 30.53 | **31.53** | 28.91 | 30.56 | **31.61** |
| | | IP | 27.81 | 29.22 | **29.47** | 27.77 | 29.12 | **29.19** | 27.79 | 29.17 | **29.33** |
| | flower | II | 25.9 | 26.54 | **27.22** | 25.89 | 26.52 | **27.23** | 25.89 | 26.53 | **27.23** |
| | | IP | 25.34 | 25.91 | **26.53** | 25.32 | 25.88 | **26.55** | 25.33 | 25.9 | **26.54** |
| Average | | | **29.08** | **30.33** | **31.35** | **29.03** | **30.23** | **31.03** | **29.06** | **30.28** | **31.19** |
| SSIM | akiyo | II | 0.9517 | 0.9647 | **0.9726** | 0.9483 | 0.9597 | **0.9637** | 0.95 | 0.9622 | **0.9681** |
| | | IP | 0.9545 | 0.9664 | **0.9744** | 0.9511 | 0.9615 | **0.9645** | 0.9528 | 0.964 | **0.9695** |
| | foreman | II | 0.9127 | 0.9312 | **0.9509** | 0.9087 | 0.9246 | **0.9343** | 0.9107 | 0.9279 | **0.9426** |
| | | IP | 0.9072 | 0.9252 | **0.9378** | 0.9008 | 0.9144 | **0.9124** | 0.904 | 0.9198 | **0.9251** |
| | bus | II | 0.8278 | 0.8684 | **0.9029** | 0.8255 | 0.8647 | **0.8878** | 0.8267 | 0.8665 | **0.8953** |
| | | IP | 0.8204 | 0.8614 | **0.8914** | 0.8158 | 0.8539 | **0.8696** | 0.8181 | 0.8576 | **0.8805** |
| | mobile | II | 0.8494 | 0.8902 | **0.9065** | 0.8483 | 0.8883 | **0.9009** | 0.8488 | 0.8892 | **0.9037** |
| | | IP | 0.8565 | 0.8947 | **0.9046** | 0.8546 | 0.8912 | **0.8975** | 0.8556 | 0.893 | **0.901** |
| | football | II | 0.83 | 0.8743 | **0.9116** | 0.8279 | 0.8708 | **0.8953** | 0.829 | 0.8726 | **0.9035** |
| | | IP | 0.8081 | 0.8551 | **0.8788** | 0.8043 | 0.8486 | **0.8542** | 0.8062 | 0.8518 | **0.8665** |
| | flower | II | 0.8849 | 0.9056 | **0.9167** | 0.8836 | 0.9033 | **0.9104** | 0.8842 | 0.9045 | **0.9135** |
| | | IP | 0.8783 | 0.8992 | **0.9089** | 0.8763 | 0.8954 | **0.9012** | 0.8773 | 0.8973 | **0.905** |
| Average | | | **0.8735** | **0.903** | **0.9214** | **0.8704** | **0.898** | **0.9076** | **0.8719** | **0.9005** | **0.9145** |

Table 5 – Comparaison du PSNR de la méthode de référence, de la SOA et du cadre proposé, lorsqu'une observation basse résolution et une haute résolution sont disponibles. Ces résultats ont été obtenus en utilisant une compression HEVC.



Original Frame 37

Obs 1, QP 20 27.57 (dB), 0.8702

Obs 2, QP 20 27.55 (dB), 0.8788

Ref 29.50 (dB), 0.9079

SOA 30.86 (dB), 0.9281

Prop. 33.99 (dB), 0.9427

Original Frame 23

Obs 1, QP 15 22.09 (dB), 0.8032

Obs 2, QP 15 22.10 (dB), 0.8378

Ref 23.13 (dB), 0.8341

SOA 24.04 (dB), 0.8908

Prop. 29.50 (dB), 0.9599

Figure 20 – Détails des images suréchantillonnées et des résultats correspondants des méthodes super-résolution, testées sur les séquences Foreman et Mobile. Les valeurs PSNR et SSIM sont calculées sur le pièce d'image comparé.

# 6 Un algorithme de détection de route ferroviaire pour la surveillance des infrastructures avec des systèmes aéroportés durables

Ce dernier chapitre de la thèse aborde un sujet différent. Comme les drones ne se limitent plus aux applications militaires et sont même disponibles en tant que dispositifs de divertissement, la vidéo surveillance automatique des infrastructures est une réelle possibilité. C'est également l'objectif du projet SURICATE (SUrveillance de Reseaux et d'InfrastruCtures par des systemes AeroporTes Endurants), qui propose l'utilisation de véhicules aériens sans pilote (UAV) pour la surveillance d'infrastructures, telles que les voies ferrées ou les lignes électriques. Ce chapitre est centré sur ces idées et aborde un scénario spécifique: la surveillance des chemins de fer à l'aide de drones durables.

Afin de donner à un drone une idée de la position des voies ferrées, nous proposons un algorithme de détection des voies ferrées basé sur la transformée de Hough. Figure 21 décrit le schéma général de l'algorithme proposé. La méthode peut être divisée en 7 étapes, à partir de l'image d'entrée et en finalisant avec les coordonnées des lignes détectées.

Une fois qu'un ensemble de lignes est obtenu en utilisant la transformée de Hough, nous identifions les clusters de lignes en fonction de leur position (Rho), angle (Theta) et longueur. Un modèle de notation est proposé pour identifier le cluster correspondant au chemin de fer. De cette façon, nous pouvons obtenir une estimation de la position et de l'orientation du chemin de fer par rapport au drone.

La validation expérimentale est réalisée à partir de séquences réelles acquises par Airbus Defence&Space. Afin de mesurer la précision de l'algorithme, nous considérons qu'une ligne est détectée positivement si le cluster de lignes sélectionné est situé au-dessus du chemin de fer et a une bonne orientation. Dans le Tableau 6 nous reportons notre taux de détection. Comme on peut le constater, nous obtenons un très bon taux de détection pour les chemins de fer.

| **Sequence** | **Det. rate(%)** | **Sequence** | **Det. rate( %)** |
|---|---|---|---|
| Seq. 1 | 99.6 | Seq. 4 | 72,6 |
| Seq. 2 | 96.6 | Seq. 5 | 96.3 |
| Seq. 3 | 94.3 | Seq. 6 | 100 |

Table 6 – Taux de détection positifs sur les séquences testées.

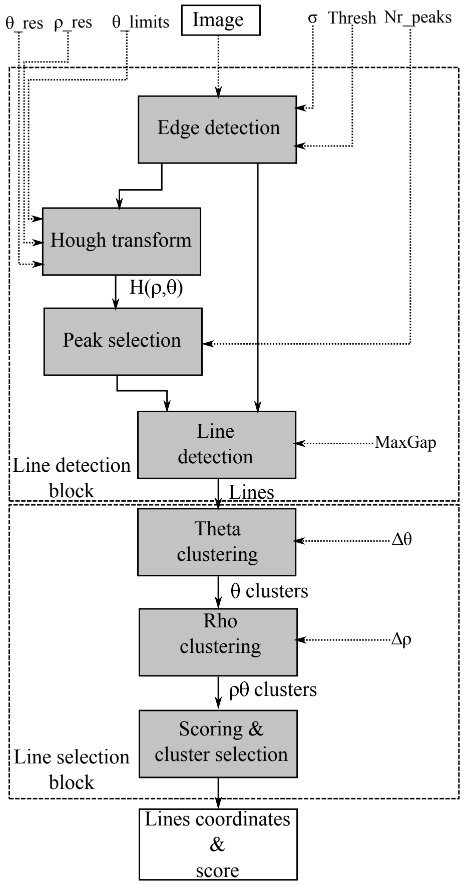
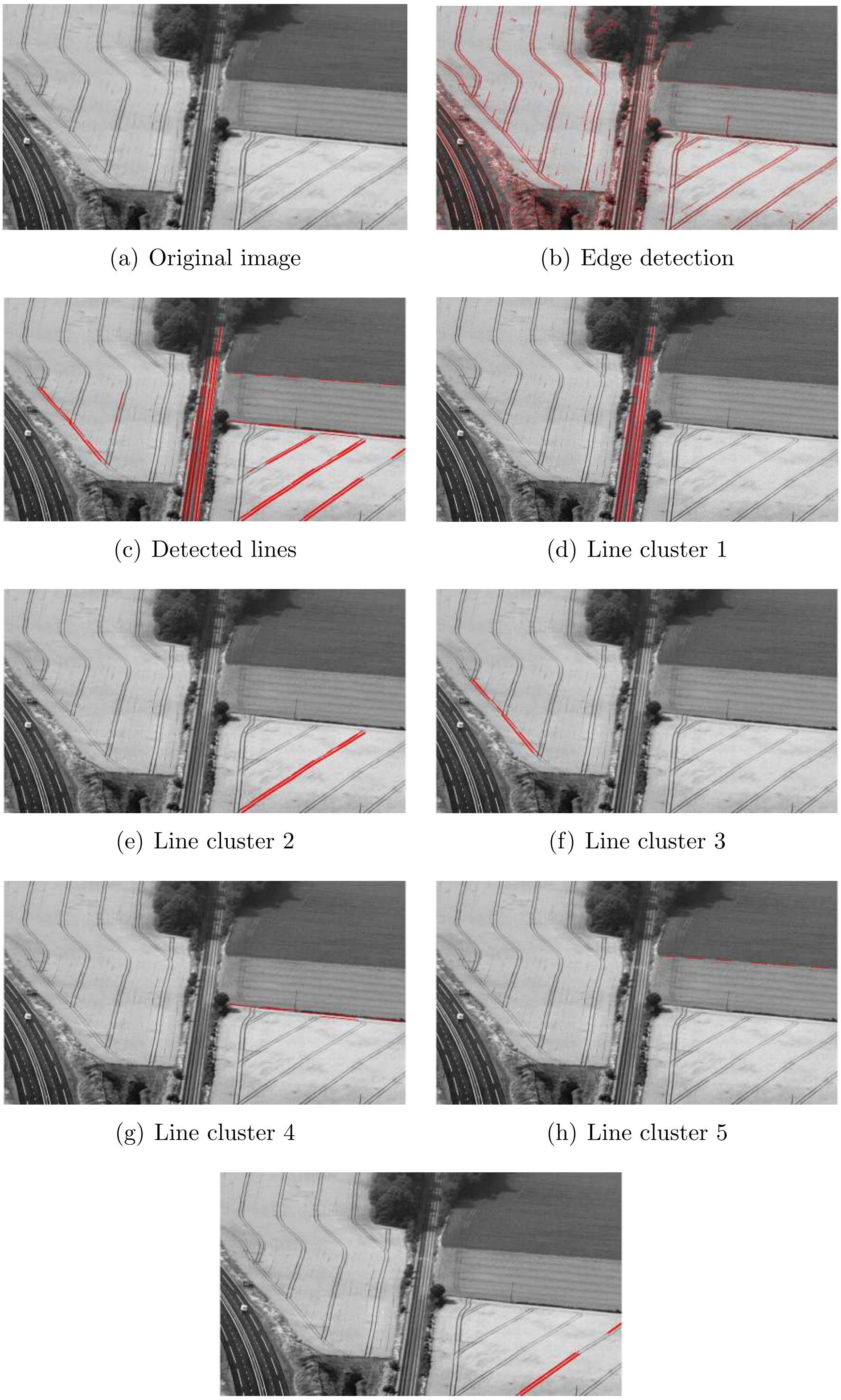


Figure 21 – Schéma général de l'algorithme.

Figure 22 montre un exemple du comportement de l'algorithme pour une trame. Les clusters de lignes détectées sont représentées, ainsi que l'étape de détection des bords. Dans la Figure 22(c) nous montrons toutes les lignes détectées. Le score déclaré ($\mathcal{S}$) pour les 6 clusters est de: 9,0932,5,9146,5,9701,4,6062,3,6431 et 3,0416. Comme on pouvait s' y attendre, le première cluster qui contient également le chemin de fer a obtenu le score le plus élevé et est choisie comme représentatif de la position et de l'orientation des chemins de fer.



(a) Original image

(b) Edge detection

(c) Detected lines

(d) Line cluster 1

(e) Line cluster 2

(f) Line cluster 3

(g) Line cluster 4

(h) Line cluster 5

(i) Line cluster 6

Figure 22 – Un exemple des lignes détectées et du processus de clustering pour l'image 40 de la Séquence 2.

# 7 Conclusion

L'objectif de cette thèse est de proposer de nouveaux algorithmes pour la synthèse visuelle dans les systèmes de compression vidéo MVD et d'aborder le problème de la reconstruction vidéo à partir de multiples sources vidéo compressées. Trois thèmes répondant à ces exigences ont été abordés. Tout d'abord, la synthèse de vues basée sur le DIBR peut être améliorée en tirant parti des corrélations temporelles dans une vue synthétisée. Comme les distorsions produites lors de la synthèse visuelle sont intrinsèquement différentes de celles introduites par la compression vidéo, le deuxième objectif de la thèse était de trouver de nouvelles façons d'évaluer la qualité et la performance des algorithmes de synthèse visuelle. Enfin, le troisième objectif de la thèse était de trouver des moyens de combiner des vidéos multi-sources avec des résolutions et des niveaux de compression éventuellement differents, afin de créer une représentation en haute résolution avec une qualité accrue.

Pour poursuivre ce travail, nous pouvons identifier plusieurs directions. Les algorithmes de synthèse de vues et de reconstruction vidéo peuvent être combinés et appliqués à divers scénarios de compression vidéo, une direction intéressante serait de les utiliser pour améliorer la création et la compression du format naissant pour la vidéo 3D 360°. La technique d'évaluation de la qualité peut être étendue en équilibrant les zones sujettes aux erreurs de synthèse de la vue avec le reste de l'image.

# Table of Contents

# List of Figures

# List of Tables

# List of Acronyms

| | |
|---|---|
| **3D-HEVC** | Three-dimensional High Efficiency Video Coding |
| **3DTV** | Three-dimensional Television |
| **AVC** | Advanced Video Coding |
| **BD** | Bjontegaard Delta |
| **CTC** | Common Test Conditions |
| **CTU** | Coding Tree Unit |
| **CU** | Coding Unit |
| **dB** | Decibel |
| **DIBR** | Depth Image Based Rendering |
| **DV** | Disparity Vector |
| **DVF** | Disparity Vector Field |
| **FTV** | Free viewpoint Television |
| **GOP** | Group Of Pictures |
| **HEVC** | High Efficiency Video Coding |
| **HR** | High Resolution |
| **HTM** | High efficiency video coding Test Model (reference software of 3D-HEVC) |
| **HVDE** | Horizontal, Vertical and Diagonal Extrapolation |
| **JCT-3V** | Joint Collaborative Team on 3D Video |

| | |
|---|---|
| **LR** | Low Resolution |
| **MC** | Motion Compensation |
| **MF-SR** | Multi Frame Super Resolution |
| **M-LFBF** | Monotone Lipschitz Forward-Backward-Forward |
| **MPEG** | Moving Pictures Expert Group |
| **MV** | Motion Vector |
| **MVD** | Multiview Video plus Depth |
| **MVF** | Motion Vector Field |
| **MVV** | MultiView Video |
| **PSNR** | Peak Signal to Noise Ratio |
| **PU** | Prediction Unit |
| **QP** | Quantization Parameter |
| **RD** | Rate-Distortion |
| **RDO** | Rate-Distortion Optimization |
| **ROI** | Region Of Interest |
| **SF-SR** | Single Frame Super Resolution |
| **SR** | Super Resolution |
| **SURICATE** | SUrveillance de Réseaux et d'InfrastruCtures (transport et énergie), de milieux, par des systèmes AéroporTés Endurants |
| **THF** | Temporal Hole Filling (proposed approach) |
| **VC** | Video Coder |
| **VSIM** | View Synthesis using Inverse Mapping |
| **VSP** | View Synthesis Prediction |
| **VSRS** | View Synthesis Reference Software |

| | |
|---|---|
| **VSTP** | View Synthesis exploiting Temporal Prediction (proposed approach) |
| **Wf** | sub-pixel precision Warping and filtering (proposed approach) |

# Introduction

## Context

Over the last decade the world experienced a "boom" in connectivity with smart phones gradually evolving from a luxury niche device to an almost required everyday tool in today's society. In parallel, multimedia services shifted to a cloud approach as video coding, transmission and storage technologies evolved to a level where high quality video is easily accessible over the Internet. According to a survey from Cisco, videos were responsible for 64% of all Internet traffic in 2014 [cis] with a predicted 80% by 2020. Furthermore, this significant increase in demand for video content is fueling a rapid evolution of display technologies. Ultra High Definition (UHD) resolutions are now widely available on TeleVision sets (TVs) and even mobile devices. Other technologies that provide further immersion such as stereo 3D or High Dynamic Range (HDR) are already deployed on a wide range of devices, while Virtual Reality (VR) and 360° videos are accessible even on smart phones or through the use of head mounted displays.

In addition to the improvement and evolution of exiting technologies, new ways of providing a more immersive experience are continuously investigated. Immersive teleconference systems, Free viewpoint TeleVision (FTV) and other immersive video applications are now possible[DPPC13] [TTFY11]. A critical role in enabling these applications in today's interconnected environment is played by video compression and format standardization. Following the recent finalization of the High Efficiency Video Coding (HEVC) [HEV13] standards, a series of extensions were developed to account for various demands. MultiView Video (MVV) and MultiView plus Depth (MVD) video coding extensions of HEVC are already available (MV-HEVC and 3D-HEVC) [TCM$^+$15] while exploration experiments for Divergent MultiView formats that enable 3D 360° video have recently began [BLTW16]. An important class of algorithms that exploit inter-view correlations, to bridge the gap between 2D and 3D video by generating new virtual viewpoints, are known as view synthesis methods.

In addition to enabling FTV or 2D to 3D conversion they are also employed in compression or 360° video rendering.

To summarize the overall situation, we are now at a transition point towards immersive 3D video during which new technologies are being explored and standardized. Furthermore, the underway adoption of the latest video coding standard HEVC, combined with the ever increasing display resolutions and cloud transition of video services creates significant interest for super-resolution (SR) and video quality enhancing algorithms from multiple compressed sources. In this context, the main goal of this thesis is to develop new tools aimed at improving view synthesis methods used in compression systems and combining multiple compressed video sources.

## Manuscript structure

This manuscript begins with an overview of video compression for both single and multi view video. The following chapters capture different subjects investigated during this theses. While (multiview) video coding is the preferred application throughout this work, additional details on each individual subject are available in each chapter. The structure of each chapter is as follows:

- Chapter 1 presents an overview of video coding. It begins with the basic concepts used in video compression and introduces the architecture shared by most video encoders. Then the key features of the latest video coding standard are discussed. The later part introduces the main concepts and formats used in 3D video compression. An overview of emerging 3D 360° video technology concludes the chapter.

- Chapter 2 presents a state-of-the-art of view synthesis followed by a detail description of two proposed approaches that exploit temporal correlations to improve view synthesis and their associated experiments. A 3D-HEVC integration and further extension of these ideas and a comprehensive discussion and experimental section conclude this part.

- Chapter 3 details our ROI approach to view synthesis quality evaluation. The first part introduces the subject and presents the main sources of errors in view synthesis. The next sections detail our proposed approaches and experiments.

- Chapter 4 begins with a state-of-the-art of video reconstruction and SR from multiple compressed sources and motivates our work. The following sections present a detailed description of our proposed video quality enhancement and SR framework. Various comparative experiments in multiple scenarios are thoroughly analyzed to validate the efficiency of the method.

- Chapter 5 presents our railroad detection algorithm for drones in the context of SURICATE project. Experimental results on real world, drone acquired, video sequences are presented and discussed.

The manuscript is concluded with a summary of the proposed methods and results as well as future work perspectives.

## Contributions

During this thesis, multiple subjects are investigated and various methods are proposed to improve different aspects of each subject. We can divide the contributions in four categories.

A first category is comprised of methods designed to improve the view synthesis by exploiting temporal correlations in the virtual views. These methods are designed for 3D-HEVC and provide video quality enhancement for MultiView-plus-Depth transmission systems that employ view synthesis. We thus proposed:

- A warping and filtering (Wf) technique that can be used for both Depth-Image-Based-Rendering (DIBR) and motion compensation. An intermediary filtering step is used to separate between foreground and background pixels. (Section 2.4.2)

- A Temporal Hole Filling (THF) method that derives motion information in the virtual view from a reference view based on a reverse epipolar constraint formulation for backward motion vectors. Backward motion compensation is used to retrieve additional information on disocclusions. (Section 2.4.1)

- A method that uses forward motion vectors in the reference view and merges full inter-view and temporal predictions of a frames from different time instants or views. (Section 2.5)

- A framework that adapts and integrates the above methods in 3D-HEVC and proposes a modification of the transmission scheme to further increase the capability of temporal prediction in view synthesis. An adaptive fusion is used to judiciously select between inter-view or temporal prediction at pixel level. (Section 2.6)

The second category contains methods to evaluate the quality of view synthesis algorithms.

- A first contribution proposes two fast Region-Of-Interest selection methods to identify areas in synthesized frames that are critical in the quality evaluation of a synthesis algorithm. The first method is designed to separate between compression and synthesis errors, while the second selects areas predicted differently by two synthesis methods. (Section 3.2)
- A method that further extends these ideas to simultaneously evaluate multiple synthesis algorithms. Several ROI generation possibilities are investigated and combined with different metrics. (Section 3.3)

The third subject investigated during this thesis resulted in the development of a robust Super-Resolution (SR) and video quality enhancement framework. Two contributions can be identified:

- A SR and reconstruction framework from multiple compressed video sources that accounts for the particularities of hybrid video coders. Due to it's robustness, the framework can be easily adapted for different problems in the future. (Sections 4.2 and 4.3)
- HEVC and a generic model for older coders are integrated in the framework and two practical applications are proposed: SR from two compressed videos and enhancing a High Resolution (HR) compressed video from a Low Resolution (LR) one. (Section 4.4 and 4.5)

Finally, the last contribution was developed as part of the *SUrveillance de Reseaux et d'InfrastruCtures par des systemes AeroporTes Endurants* (SURICATE) project. A slightly different subject is approached and our contribution is targeted towards a specific scenario: the surveillance of railroads using unmanned enduring airborne systems.

- A railroad detection algorithm designed to provide drones with a sense of the railroads position for tracking purposes. The method relies on Hough transform and an original clustering and scoring model was designed to detect railroads.

# Chapter 1

# Video coding

## Contents

## 1.1 Basic concepts

In the simplest way, videos can be viewed as an electronic or digital medium that stores and facilitates the visual representation of moving media. Whether the content reflects a real world scene, a virtual one or any abstract concept the main trait of all videos that differentiates them from images, is their ability to store motion information. For this reason large amounts of information have to be stored and transmitted in order to share a video. In general, videos are formed from a sequence of still images (frames) which are displayed at a high enough frequency to create the illusion of motion.

Using the above definition of a digital video sequence we can easily estimate the amount of information needed to store it. Let us consider a monochromatic video with a resolution of $1024 \times 768$, with the gray levels represented on 8 bits (256 levels) and a frame rate of 30 fps. Each frame requires approximately 786 kilobytes (KB). This value is tripled for color videos, for example in a YUV color space each pixel holds a value for luminance (Y) and two chrominance components (U & V). Thus, a color frame would require 2.2 Megabytes (MB). A second of video at 30 fps requires 66 MB while an hour is  237 Gigabytes (GB). In this scenario it is impossible to stream a video over the Internet and even storing is challenging. Fortunately, the information contained in a video is highly correlated and redundancies can be used to reduce the size of the data [Bov00].

The compression ratio can be defined as:

$$Compressionratio = \frac{UncompressedSize}{CompressedSize} \tag{1.1}$$

Even though lossless compression techniques allow the data to be perfectly reconstructed, they are not very efficient and achieve a compression rate of 3 - 4 [RS01]. Lossy compression [Sal07] methods are required to obtain high compression ratios. The challenge is to reduce the size of data without affecting the perceived quality of the video in a significant way.

### 1.1.1 Statistical redundancy

Statistical redundancies can be divided into two categories [SS08]: spatial redundancy and temporal redundancy. This implies there is a strong correlation between the pixels in a frame or those in a group of successive frames.

- Spatial redundancy refers to the statistical correlation between pixels belonging

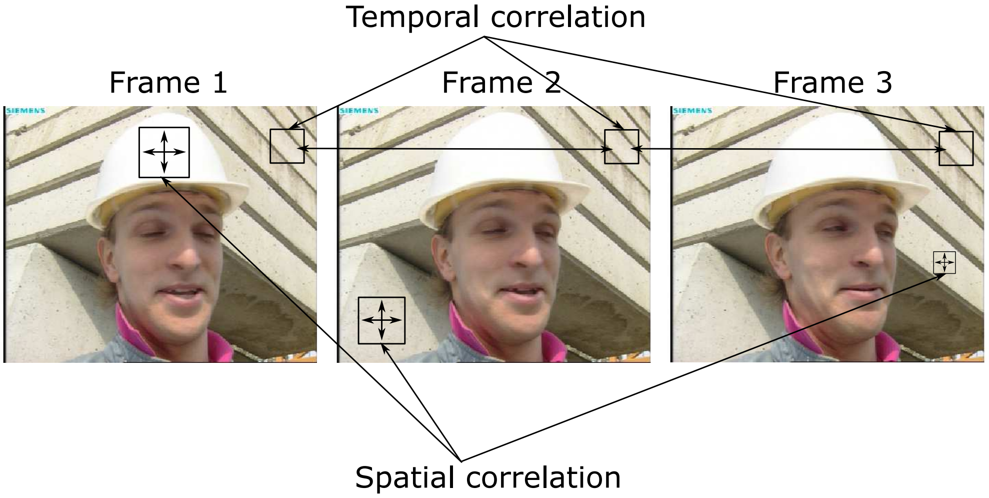


Figure 1.1 – Spatial and temporal correlation in a video sequence.

to the same frame. As each image in a video sequence commonly depicts a continuous representation of a scene, we often encounter areas that share similar color and luminance information. Thus, the luminance and color information of a pixel can be predicted from the neighboring pixels with a relatively small error. A large amount of data can be saved by removing the redundancy within a frame. In Figure 1.1 areas that have almost identical information can be seen.

- Temporal redundancy deals with the statistical correlation between consecutive frames of a video sequence. The illusion of motion is created by a fast displaying of images acquired at frequencies that are high enough to capture each movement in the scene. Therefore, we can expect pixels in consecutive frames to be highly correlated. This allows entire areas of an image to be predicted from neighboring frames. Figure 1.1 depicts three frames of a video sequence and shows an example of redundant temporal information.

### 1.1.2 Psychovisual redundancy

Psychovisual redundancy exploits the HVS's response to different stimuli. As such, certain types of information under various conditions can be perceived with a higher degree of sensitivity than others; the HVS does not perceive all information in an equal manner. This means that certain parts of the visual information can be ignored or represented using less data without affecting the perception [SS08]. While there are many aspects that define how the HVS perceives the world, we will refer here to those which are relevant for video compression. These are: luminance masking, texture masking, contrast sensitivity, temporal masking and color sensitivity.

- Luminance masking refers to the ability of the HVS to perceive brightness. The main concern here is the ability to detect one stimulus in the presence of another. This aspect is also known as luminance dependance or contrast masking [Wat87] [LF80]. A simple example is the ability to distinguish an object from the background. A similar level of luminance between the back ground and the object or a very low contrast will hide the object, making it harder to detect. More precisely, Weber's law [Fec60] states: "*Simple differential sensitivity is inversely proportional to the size of the components of the difference; relative differential sensitivity remains the same regardless of size*". If we consider the minimum luminance threshold $\Delta I$ which can be observed by the HVS, Weber's law can be expressed as:

  $$\frac{\Delta I}{I} \approxeq constant, \tag{1.2}$$

  where, the constant is approximately 0.02. Thus, the threshold for discrimination $\Delta I$ is directly proportional to the luminance $I$.

- Texture masking states that the discrimination threshold increases with picture details [CBL72]. The HVS perceives noise more easily in smooth image areas than textured areas that exibit high intensity variation.

- Contrast sensitivity refers to the HVS perception of a stimuli w.r.t. spatial frequency. This dependency can be modeled by a constrast sensitivity function (CSF), which, indicates how sensitive the HVS is to the frequency of the stimuli. Considering an image of black and white vertical stripes, above a certain frequency the image appears gray. The reason why patterns with high frequency can not be distinguished is the limited number of photo-receptors in our eyes.

- The temporal masking is caused by the HVS inability to instantly adjust to an abrupt scene change [Mit96]. More precisely, a sharp scene transition will leave the HVS less sensitive to details for a short time interval.

- Color sensitivity refers to the HVS perception of light. The HVS is more sensitive to certain wavelength. Thus, color spaces that use an equal representation of the primary colors perceived by the human eye (Red, Green, Blue) are not always efficient. A luminance chrominance color space is generally preferred for image and video compression.

### 1.1.3 Conclusions

Some of the most common fields that heavily rely on video data are entertainment, publicity, security or communications, but, besides these there are many others that benefit from the use of videos. Video technology can be found all around us, from our personal computer and television to drones or robots remote control applications, medical devices or security cameras. A key aspect in many applications is the ability to easily transmit video data and adapt it to each use case scenario. For example, some applications require a high amount of details to be recorded while in others the focus is on the speed of delivery. The devices used to display videos are also varied, from mobile phones to television sets or virtual reality (VR) headsets. Considering the large amount of information contained in digital video data fast and robust compression methods play a critical role in facilitating these applications. Video encoding algorithms are build around exploiting the statistical and psychovisual redundancies of video sequences.

## 1.2 Hybrid video coding

### 1.2.1 Quantization

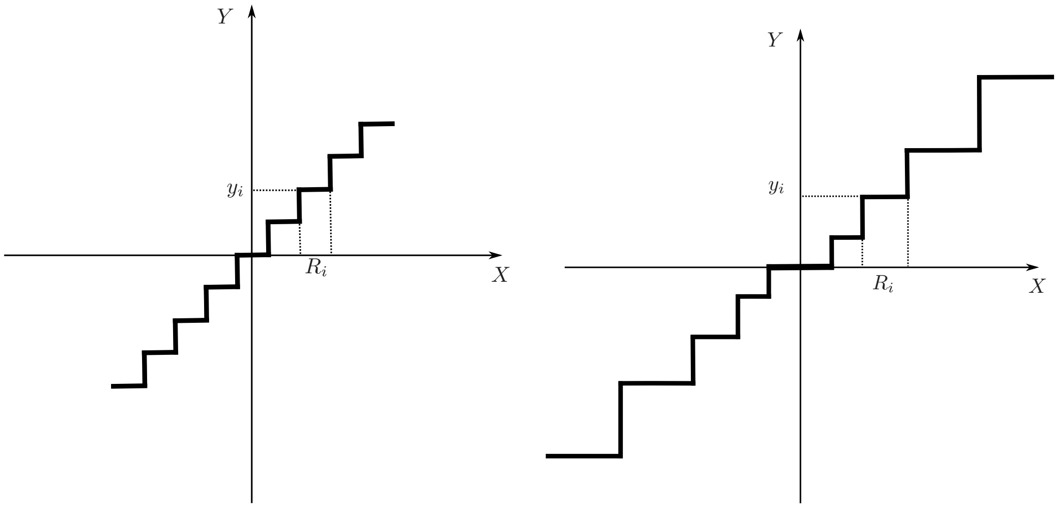
In the case of analogue signals, quantization constraints a continuous set of values to a discrete set. For digital signals quantization further reduces the discrete set. This operation can be interpreted as a mapping from a set $S$ to a discrete subset $C$ of cardinality $N$.

$$Q : x \in S \rightarrow C = y_1, y_2, ..., y_N \tag{1.3}$$

This is achieved by dividing the set $S$ into regions $R_i$ such that $\cup_{i=1}^{N} R_i = S$. A region $R_i$ can be defined as:

$$R_i = x \in S : Q(x) = y_i \tag{1.4}$$

Thus, quantization is comprised of two operations. An encoding step, during which the set $S$ is divided into regions and each value $x$ is associated to the index of a region $R_k$. The decoding step or inverse quantization consists in mapping the region to a reconstruction value $y_k$.



(a) Uniform quantization (b) Non uniform quantization

Figure 1.2 – Uniform and non uniform quantization.

If set $S$ is uniformly divided in regions $R_i$ and the reconstruction levels are the centers of the regions, then the quantizer is uniform (Figure 1.2(a)). In some cases however, it is better to allocate more resources (finner division of regions) in certain parts of the $S$. Considering a gray level image represented with 256 levels from black to white. A 128 levels representation can be achieved through quantization. If the image depicts a night scene it is easily understandable that more pixels will have values closer to black than white. Thus, using finer quantization intervals between 128 and 256 and expanding the rest may lead to a higher quality representation of the image. A similar rationale can be made in the case of transform coefficients, by focusing more on the part of the transform domain that holds information more relevant to the perceived quality.

Quantizers that use variable size regions are non-uniform (Figure 1.2(b)). The Lloyd-Max algorithm [Llo82] provides the best reconstruction levels and optimal regions with respect to the signal statistics.

### 1.2.2 Transform

Transform coding by itself is typically used to sparsify the data and is generally an almost lossless form of compression. However, the transform is generally coupled with a quantization process. As transforms are used to perform "energy compaction" (i.e. only a small number of coefficient have a significant impact on the HVS perception) a more targeted quantization can be achieved. In other words the transformation process is used to better select the information that will be discarded. In the case of images the discarded information is also chosen with respect to the HVS perception. As is the case of high frequency components for a DCT transformation.

A possible interpretation of the transform is the representation of an image as a weighted sum of basis images, where the weights are given by the transform coefficients. Each coefficient is a measure of the correlation between the basis and image. In general the transform will result in a compaction of energy in the transform domain resulting in a number of zero valued coefficients. Thus, a smaller number of values need to be encoded. Furthermore, the coefficients are less correlated than pixels and the information contained in each coefficient can often have a different impact on the HVS perceived quality.

Some of the most common transformations used in coding are: the Karhunen [Kar47], [Loe48], the Discrete Fourier Transform (DFT) [CT65], the Hadamard transform [Had93], the Walsh transform [Wal23] and the Discrete Cosine Transform (DCT) [ANR74]. In terms of coefficient correlation and energy compaction, the KLT provides the best results. However, this transform requires the calculation of the covariance matrix for each transformed image block, which is highly computationally expensive. A good compromise in terms of energy compaction and computational efficiency is the DCT, which is one of the most common transforms used in image or video compression.

### 1.2.3 Lossless coding and variable length coding

Lossless coding is a form of compression that allows the data to be retrieved without any loss of information. In the case of video coding this achieved through entropy coding.

Shannon's source coding theorem establishes that data cannot be compressed in a lossless manner with an average number of bits per symbol smaller than the Shannon entropy of the source [Sha48]. Considering a random variable $X$ with the possible values $x_1, ..., x_n$. The entropy of the random variable $H(x)$ is defined as:

$$H(X) = \sum_{i=1}^{n} P(x_i) \log_b P(x_i) \tag{1.5}$$

where $P(x_i)$ is the probability mass function, i.e. the probability of $\{x \in X : x = x_i\}$. If the base of the logarithm $b = 2$ then the result is expressed in bits.

Entropy coding aims to encode a sequence of symbols with the smallest possible bitrate, ideally the entropy. One of the most common approaches of entropy coding is variable length coding (VLC). The idea is to assign unique prefixes of variable length to each of the symbols in the transmitted data such that, the length of the prefix is inversely proportional to $P(x_i)$. Thus, the most common symbols are encoded with a small number of bits while rare symbols use longer prefixes. The most common entropy coding methods are: Huffman coding [Huf52] and arithmetic coding [Sha48].

### 1.2.4 Predictive coding

As discussed in Section 1.1 a high amount of correlation exists between the frames of video sequences. This enables the efficient application of predictive coding which relies on encoding the difference between the signal and its prediction, also known as residual.

More precisely, let's consider a gray level video sequence with frames $\{I_1, ..., I_k, ...I_n\}$, each frame is divided in M rows and N columns. When encoding frame $I_k$ the goal is to find a prediction $P_k$ from a another frame $I_r$ (prediction reference) that minimizes the $E_k$.

$$E_k = I_k - P_k \tag{1.6}$$

The amount of information in $E_k$ is inversely proportional to the precision of the prediction method used. Ideally $I_k = P_k$ and only the information needed to generate $P_k$ is sent to the decoder. In practice, the residual information is required in order to obtain an acceptable quality. Frames in a video sequence differ from each other mainly due to motion in the scene. As such, motion estimation and compensation is the preferred method of prediction in video coding.

#### 1.2.4.1 Motion estimation and compensation

Motion estimation (ME) aims to find a predictor of frame $I_k$ from the reference $I_r$. The prediction $P_k$ is obtained through motion compensation (MC). The predictor can be expressed as a field of vectors $\mathbf{v}(p) = (v_x, v_y)$ where $p = (x, y)$ is the position of a pixel associated with the vector. For computational and compression reasons a single vector can be estimated for blocks of pixels.

$$ME(I_r, I_k) = \mathbf{v} \tag{1.7}$$

$$MC(I_r, \mathbf{v}) = P_k \tag{1.8}$$

As can be seen from Figure 1.3 the residual $E_k$ and the amplitudes of motion vectors in $\mathbf{v}$ have much higher spatial correlation than the actual texture frame. Encoding and sending only $E_k$ and $\mathbf{v}$ is much more efficient than encoding a frame, with respect to compression ratio.

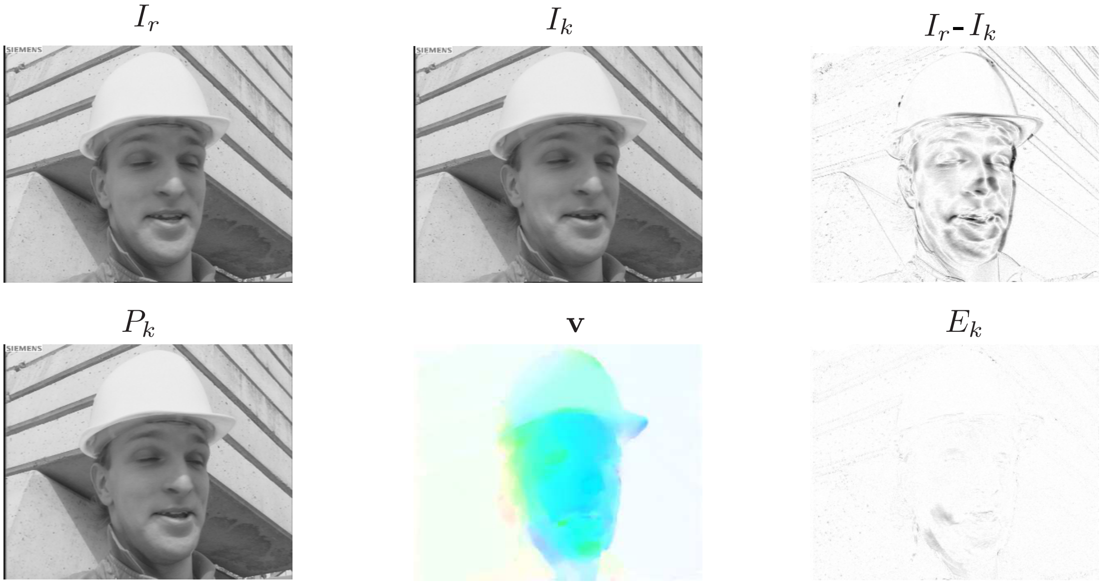


Figure 1.3 – An example of motion estimation and compensation.

#### 1.2.4.2 Coder architecture

While each video encoder introduced new algorithms and tools there is a generic architecture based on a few concepts which are common to all. These are: quantization, use of transforms, predictive coding and entropy coding. In what follows we will discuss this generic model of video compression know as hybrid video coding.

The hybrid video coding paradigm is used by all current video coding standards. The basic architecture of a hybrid video encoder can be thought of as a skeleton for all modern video coders. It uses two different techniques to reduce the spatial and temporal redundancy (see Section 1.1.1) from a video sequence. Spatial redundancy is reduced through transform coding combined with quantization which reduces the size of the data by eliminating high frequencies in an image. Even though this is a lossy form of encoding as it contains a quantization step, the overall impact on perceived quality is acceptable due to the way the HVS perceives information (see Section 1.1.2). Temporal redundancy is removed through predictive coding. The general idea is to predict the data which is currently encoded from decoded previous values and encode only the difference.

The Moving Pictures Experts Group (MPEG) standards divide video sequences in Groups Of Pictures (GOP). Figure 1.4 depicts a typical structure of a GOP in a hybrid video encoder. Depending on the way frames are encoded we can divide them in two types: Intra (I) and Inter frames. Inter frames can be further divided in P-type and B-type frames. The main characteristics of each type are:

- I-frames: are encoded independently from other frames. Temporal redundancy is not taken into account. Transform coding is employed to reduce spatial redundancy. In modern encoders predictive coding is also used between neighboring blocks of the image This is known as intra prediction.
- P-frames: temporal redundancy is reduced through predictive coding. The frame is predicted from another one. The reference of this temporal prediction is always a previous I or P type frame of the same GOP.
- B-frames: unlike P frames, the temporal prediction is bidirectional. Both a previous and a future frame can be used as references for the prediction.

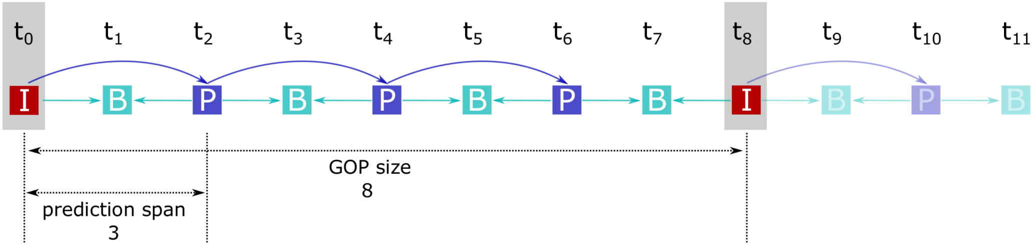


Figure 1.4 – Example of GOP structure.

Figure 1.5 depicts the generic architecture of a hybrid video coder. Once a new frame $I_k$ is inputted, the coder can work in two modes depending on the type of encoding: intra-frame or inter-frame. In intra mode only transform coding is performed. First the image is transformed usually with Discrete Cosine Transform (DCT). The resulting coefficients are quantized and then a lossless coding step is performed. This consists in applying variable length coding on the quantized coefficients. Actually, this simple model represented with red, is a also a common method of image compression used in the standards developed by the Joint Photographic Experts Group (JPEG).

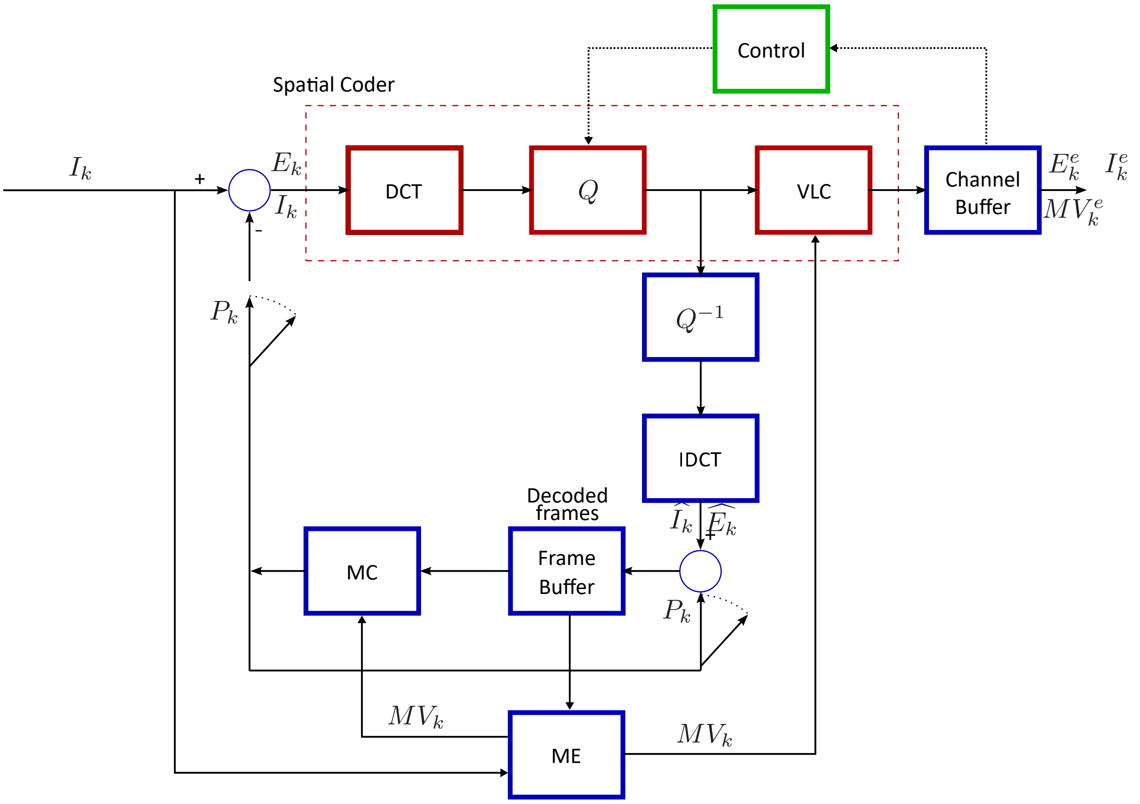


Figure 1.5 – Generic hybrid coder scheme.

The inter-frame coding is slightly more complex. First a reference frame is decoded, this can be either intra or inter. The intra frame is decoded by applying $Q^{-1}$ and the inverse transform (IDCT). The resulting image $\widehat{I_k}$ is then stored in the frame buffer. When a new frame is inputted, motion estimation is performed (ME) between the current frame and the previous frame stored in the buffer, thus computing the motion vector field $MV_k$ which is also included in the bitstream. Using motion compensation (MV) a prediction $P_k$ of the frame is created. The prediction error of $P_k$, denoted by $E_k$, is determined as $I_k - P_k$ passed through the spatial coder block and added to the bitstream as $E_k^e$. The prediction error is also decoded and summed with $P_k$ in order to create the ME reference for a future frame.

Figure 1.6 depicts the decoding process. Intra frames are decoded in three steps. Variable Length Decoding (VLD), $Q^{-1}$ and inverse transform (IDCT). The decoded frame is also stored in a buffer. Inter-frame use the same steps to decode $E_k^e$ and then sum it with $P_k$ obtained from MC a previous frame with the motion vector field $MV_k$.

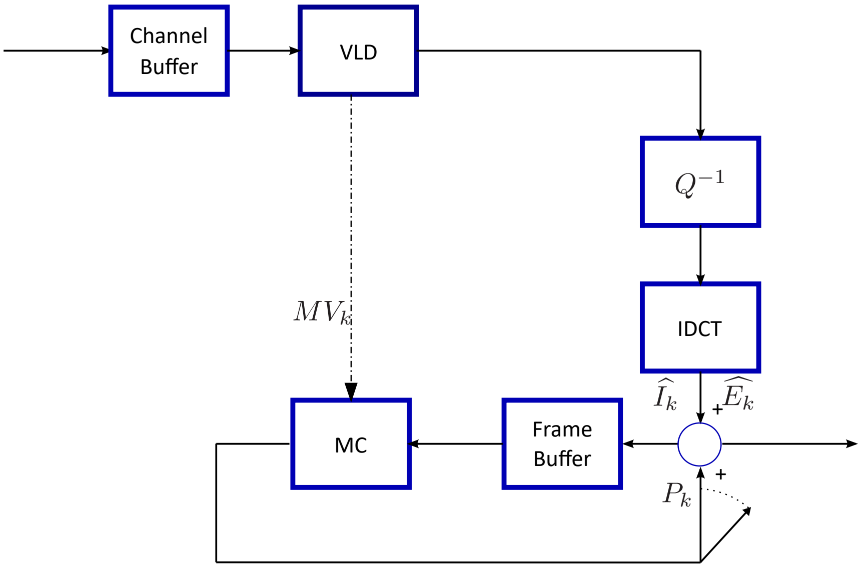


Figure 1.6 – Generic hybrid decoder scheme.

# 1.3 HEVC

## 1.3.1 Overview of HEVC

High Efficiency Video Coding (HEVC)[HEV13] is the latest video coding standard by the Joint Collaborative Team on Video Coding (JCT-VC), gathering experts from the International Telecommunication Union (ITU) and the International Organization for Standardization (ISO) [BHO+12].

HEVC represents video data in a hierarchical manner. At top level, the data sequence is comprised of general parameters (framerate, spatial resolution, etc.). A Group Of Pictures (GOP) defines a coding period as a number of frames (single sequence unit in the temporal axis). These frames can be further split in slices or tiles (e.g. to encode in parallel different portions of the frame).

As its predecessor Advanced Video Coding (AVC) [WSBL03], the HVC model discussed in Section 1.2 is reused by the HEVC standard. Obviously, some additional tools are implemented in HEVC but are not represented in Fig. 1.5, e.g de-blocking or Sample Adaptive Offset (SAO) filters. However, it is to be noted that if HEVC almost doubled the coding efficiency w.r.t. h.264/AVC [H2605], it mainly stems from the optimizations made in the essential building blocks (prediction, transform, entropy coding, etc.), whereas additional tools (e.g. SAO) can only provide marginal gains [GMM+13]. The following Section 1.3.2, further details the HEVC implementation of the different elements (e.g. block partition, transform, quantization).

## 1.3.2 HEVC structure details

### 1.3.2.1 Quadtree Partitioning

Given an input frame of arbitrary resolution, a block partitioning scheme is used to perform compression at a pixel block level. HEVC greatly improves the fixed 16x16 macroblock grid used in H.264/AVC by replacing it with a more flexible quadtree structure [HEV13], which allows a better adaption of the partitioning to the image content. The quadtree uses a hierarchical structure: the frame is first split into *Coding Tree Units* (CTUs) of fixed size (from 64x64 to 16x16). CTUs are split (potentially recursively) in *Coding Units* (CUs), forming the quadtree structure. Then, *Prediction Units* (PUs) and *Transform Units* (TUs) are rooted at the CU level to gather all the unit information on the prediction (mode, motion vectors, frame reference indexes, etc.) and the transform used respectively. It is important to note that the size of a prediction/transform is not related to the CU: both PUs and TUs can be recursively subdivided, and PUs and TUs are independent, so that prediction and transform can be made at different sizes inside a unit.

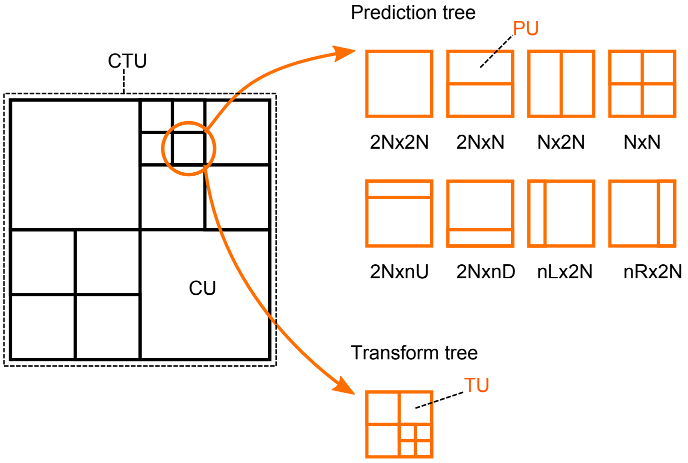


Figure 1.7 – Frame partitioning structure in HEVC.

### 1.3.2.2 Intra and Inter prediction

As its predecessor h.264/AVC, HEVC distinguishes three main different types of frames: I (Intra), P (Predicted), B (Bi-predicted). I frames are coded independently of all other images, whereas P/B frames can use motion estimation and compensation from a set of frames amongst previously encoded/decoded frames of the GOP. This latter process is known as Inter prediction, and allows to efficiently compress temporal redundancies in the source video stream. HEVC also implements a prediction for tackling spatial redundancies in a frame, known as Intra prediction. This tool is available for all types of frames (I/P/B) and uses previously decoded units as a reference to predict pixel values for the unit to be encoded. Given the raster scan processing order of the quadtree, top, top-left, and left units are the considered neighborhood. The Intra modes are ordered according to the direction angle. Vertical and horizontal directions are associated with low Intra indices (1 and 2 respectively), while finer angles have higher Intra indices, as shown in Figure 1.8.

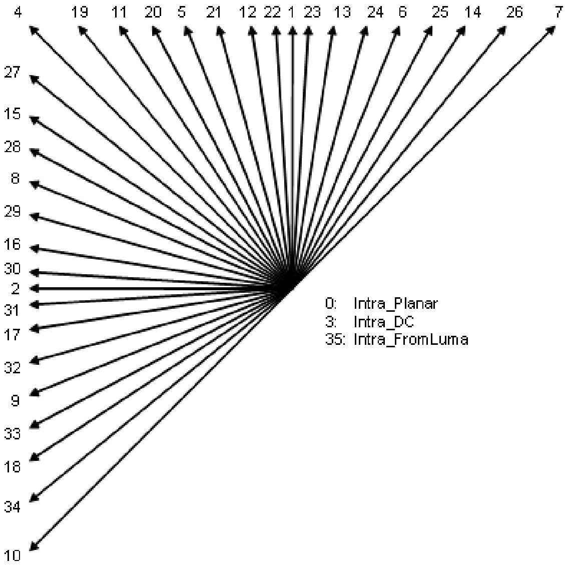


Figure 1.8 – HEVC intra modes.

The DC mode is a uniform prediction where each pixel is equal to the mean of the reference pixels. The planar mode performs a bilinear interpolation of the bottom row and the rightmost column of the current PU which are respectively substituted with the bottom-left and above-right causal reference samples.

As such, HEVC always computes a prediction for a CU (and more specifically, for each PU in a CU), either by Intra (I/P/B frames) or Inter prediction (P/B frames only). This observation implies that HEVC always encodes a CU residual.

#### 1.3.2.3 HEVC transforms

As detailed previously, HEVC computes the residual at a CU level by subtracting the prediction result of its PUs from the source signal. Then, the residual is transformed at the TU level. TUs are square pixel units that can be recursively subdivided, so different transform sizes have been specified in HEVC (4x4, 8x8, 16x16, 32x32).

Due to complexity considerations, HEVC relies on finite approximations of well-known transforms: the Discrete Cosine Transform (DCT) and its inverse (IDCT). Moreover, a Discrete Sine Transform (DST) is specifically used for 4x4 Intra units. The transform matrices are fully standardized and can be found in [BFB14].

#### 1.3.2.4 HEVC quantization

HEVC quantization is performed at a TU level on the transformed residual using a scalar quantizer. The applicable quantizer is indicated by a Quantization Parameter (QP) ranging from 0 to 51 which serves as an integer index to derive the applicable step size $\Delta_q$. HEVC follows a logarithmic structure : the step size doubles when the QP increases by 6. The first six step sizes (for QP ranging from 0 to 5) are presented below, alongside with the formula allowing to infer the step-size at higher QPs.

$$\Delta_{q,0..5} = \{2^{-4/6}, 2^{-3/6}, 2^{-2/6}, 2^{-1/6}, 1, 2^{1/6}\} \tag{1.9}$$

$$\Delta_q(QP) = \Delta_{q,QP\,mod\,6} \cdot 2^{\lfloor QP/6 \rfloor} \tag{1.10}$$

Furthermore, HEVC supports frequency dependent quantization, as human perception is more sensitive to information losses in the low frequency domain. Two default quantization matrices (for intra and inter frames) are thus specified, to leverage the quantization strength in each frequency band.

## 1.4 3D video

3D video refers to the ability of video to provide users with a perception of depth, thus providing a more realistic and immersive viewing experience. To provide a better understanding of how this is achieved we begin with a short look at the main factors that allow humans to perceive depth.

### 1.4.1 Depth perception

The human visual system relies on multiple depth cues that allow a person to perceive the geometry of a scene and build a 3D mental model [RHFL10]. Depending on the mechanisms used by the human visual system, depth cues can be classified as follows:

- **Oculomotor** depth cues which use the physical properties of the eyes to extract depth information.

  - **Accommodation**: the change of the eye's lens to adjust the focal length in order to bring objects into focus on the retina.
  - **Convergence**: the eyes rotation towards each other for close objects.
  - **Myosis**: the constriction of the pupil size in order to control the amount of light that is analyzed by the eye.

- **Visual** depth cues which use the visual information to extract information on scene depth.

  - **Monocular**:

    * **Static**: pictorial cues such as illumination or relative size differences.
    * **Dynamic**: motion parallax.

  - **Binocular**: the sensation of depth is created by the difference between the images captured on each retina.

Out of the above depth cues occulomotor ones are regarded as relatively weak, and are effective over short distances of up to 10 meters. Visual depth cues are more effective in providing a good perception of depth with binocular cues allowing a viewer to perceive depth from viewing distances ranging from a few centimeters to 100 meters.

In general 3D viewing technologies rely mostly on binocular cues for creating the illusion of depth. This is achieved by providing each eye with a different representation of a scene.

### 1.4.2 3D video formats

In order to provide a user with a 3D viewing experience at least two views of a scene are required (one for each eye). As such, the simplest 3D video format is the Conventional Stereo Video (CSV), two views of the same scene are captured by two cameras at a certain distance (baseline), as shown in Figure 1.9. An alternative to CSV is the Mixed Resolution Stereo [BSM$^+$09]. The binocular suppression theory states that if two views of different quality are multiplexed on a stereo display the resulting 3D image quality is closer to that of the higher quality view. Thus, one of the views can use a lower resolution without a significant loss in 3D quality.

Multi-resolution Frame Compatible (MFC) formats perform a spatial or temporal multiplexing in order to use a single support for both views. Spatial multiplexing

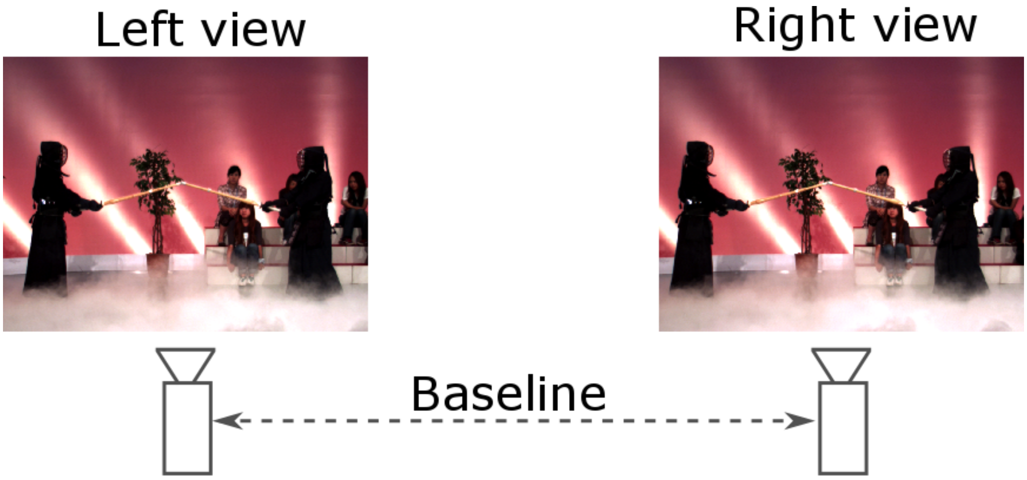


Figure 1.9 – Conventional stereo video.

reduces the horizontal or vertical resolution and fits both views on the same display while temporal multiplexing alternate views between frames.

New technologies also take into account motion parallax and aim at supporting the displaying of multiple view points of the scene [FCSK02] [CTMS03]. To enable this type of services the MultiView Video (MVV) format was introduced (see Figure 1.10). The data is composed of $N$ views captured by $N$ cameras in a specific configuration depending of the application. Some of the most common configurations are line or arc camera arrays.

Depending on the number of views, MVV formats may require large amounts of data to be transmitted. Furthermore a user is limited to a fixed set of positions. These issues are addressed by the MultiView-plus-Depth (MVD) formats that associates a depth map with each view and allow any number of virtual views to be synthesized in between them, as shown in Figure 1.10. Depth maps only use one image plane and provide a value for each pixel that measures the distance between the camera and the real world projection of that pixel.

### 1.4.3 MVD transmission system

While MFC video can be encoded efficiently using traditional encoders (HEVC), both MVV and MVD formats require additional tools to exploit inter-view correlations. Furthermore, MVD video also requires tools for the compression of depth maps. Two extensions of HEVC were developed to address these formats. A MultiView extension of HEVC that exploits inter-view correlation between views (MV-HEVC) and 3D-HEVC that incorporates tools for depth map compression [TCM+15] and

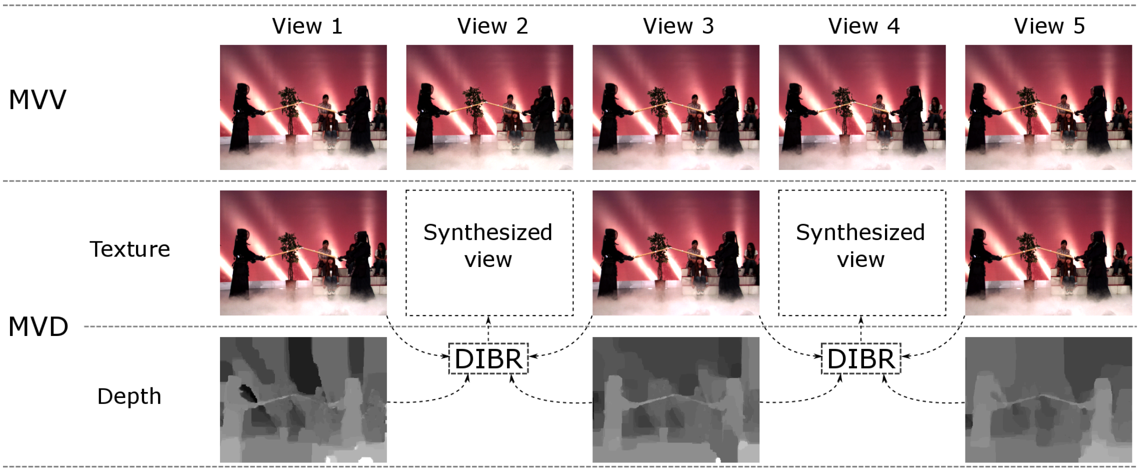


Figure 1.10 – MultiView and MultiView-plus-depth video.

proposes the use of Depth-Image-Based-Rendering (DIBR) based view synthesis.

Figure 1.11 depicts a general scheme of an MVD transmission system. $N$ views and their depth maps are inputted in a 3D video encoder which conjointly compresses the data into a single bitstream. In order to assure backward compatibility with legacy 2D formats this system should provide the possibility to independently decode a single view. The second view used for CSV formats and the remaining views can be encoded using the first one as a reference for inter-view prediction. The main difference w.r.t. an MVV transmission system is the usage of depth maps and DIBR based view synthesis. Thus, instead of sending a large number of views a limited number can be used to synthesize the others (see Figure 1.10) or render additional virtual ones.

### 1.4.4 Depth-Image-Based-Rendering

DIBR methods rely on pinhole camera geometry to re-project a point in the reference image into the real world and then onto another image plane. In order to describe this process let us consider two cameras with $C_1$ and $C_2$ centers and their associated image planes and coordinate systems as shown in Figure 1.12. The projections of a real world point $P$ onto each of the two image planes are denoted by $p_1$ and $p_2$.

Our goal is to express point $p_2$ as a function of $p_1$ and its depth. Points $p_1$ and $p_2$ with coordinates $(u_1, v_1, 1)$ respectively $(u_2, v_2, 1)$ can be expressed as a projection of

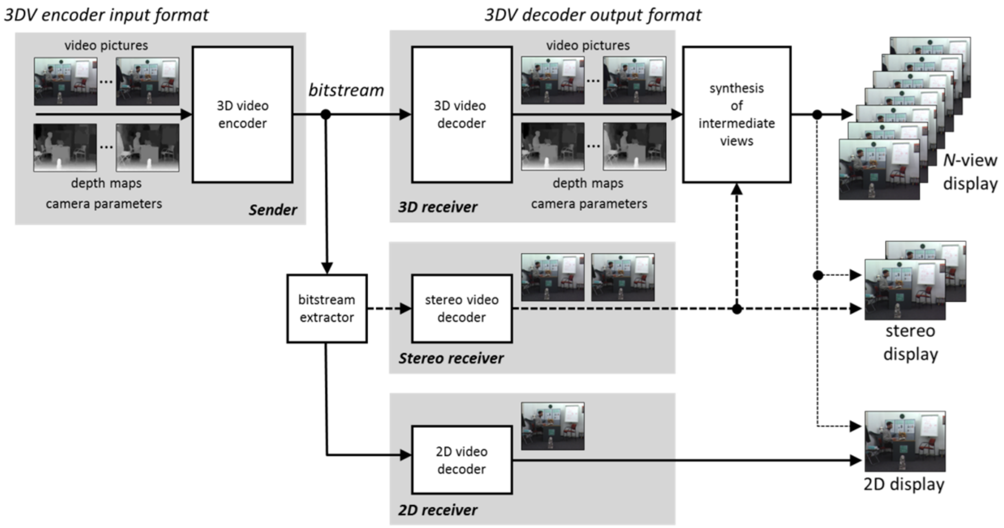


Figure 1.11 – MPEG's overview of an MVD transmission system [CTWY14].

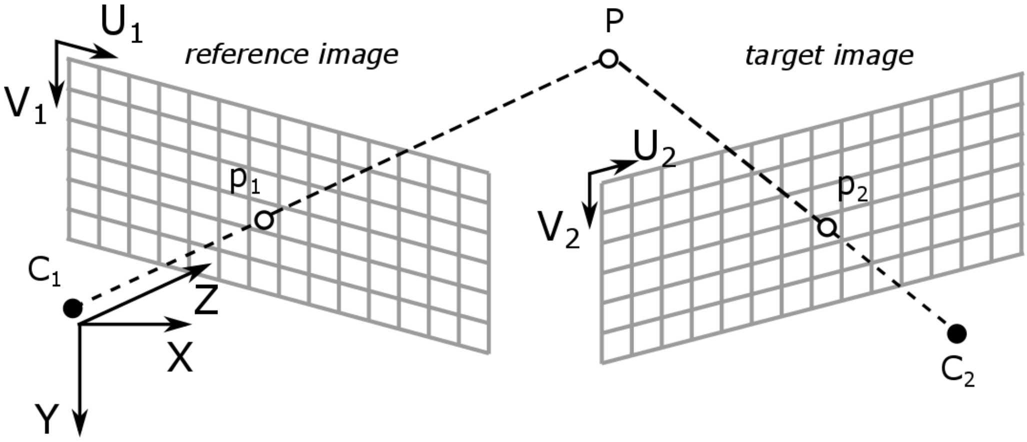


Figure 1.12 – Warping a pixel from a reference to a target image.

$P\ (x, y, z)$ as follows [Dar09]:

$$\lambda_1 p_1 = K_1 R_1 \begin{pmatrix} x \\ y \\ z \end{pmatrix} - K_1 R_1 C_1 \tag{1.11}$$

$$\lambda_2 p_2 = K_2 R_2 \begin{pmatrix} x \\ y \\ z \end{pmatrix} - K_2 R_2 C_2 \tag{1.12}$$

where $K_1$, $K_2$ and $R_1$, $R_2$ are respectively the $3 \times 3$ intrinsic camera parameters matrix and the $3 \times 3$ orthogonal rotation matrix for each camera. $\lambda_1$ and $\lambda_2$ are the homogeneous scaling factors.

From Equation 1.11, point $P$ can be expressed as:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = (K_1R_1)^{-1}(\lambda_1 p_1 + K_1R_1C_1) \tag{1.13}$$

By replacing Equation 1.13 in Equation 1.12 we obtain:

$$\lambda_2 p_2 = K_2R_2(K_1R_1)^{-1}(\lambda_1 p_1 + K_1R_1C_1) - K_2R_2C_2 \tag{1.14}$$

Assuming the first camera is also the origin of the world coordinate system and looking along the $Z$ direction, then the scaling factor $\lambda_1 = z$, $C_1 = O_3$, $R_1 = I_3$ and Equation 1.14 becomes:

$$\lambda_2 p_2 = K_2R_2K_1^{-1}zp_1 - K_2R_2C_2 \tag{1.15}$$

Furthermore, if we suppose that the cameras are identical and rectified, then $K = K_1 = K_2$, $R_2 = I_3$ (no rotation angle between cameras), $\lambda_2 = z$ and $C_2 = (c_x, 0, 0)^T$ (camera two is located on the $X$ axis of the real-world coordinate system). In this case Equation 1.15 is re-written as:

$$p_2 = p_1 + K \begin{pmatrix} \frac{c_x}{z} \\ 0 \\ 0 \end{pmatrix} \tag{1.16}$$

and $u_2$, $v_2$ can be expressed as:

$$u_2 = u_1 + \frac{f \cdot c_x}{z} \text{ and } v_2 = v_1 \tag{1.17}$$

where $f$ is the focal length of the cameras, $u_2 - u_1 = \frac{f \cdot c_x}{z}$ is also referred to as disparity and $c_x$ is the distance between cameras or baseline.

For a detailed review of DIBR based synthesis algorithms and methods designed to complement this technique see Sections 2.1 and 2.2. A discussion of some of the main issues that affect this technique is available in Section 3.1.2.

## 1.5 360° video: an emerging technology

Another approach towards immersive multimedia is the use of the so-called *Divergent Multiview Video* [BLTW16] (DMV) where the videos are acquired by cameras located around a position in the scene. These videos can be used to create a 360 degree video and a user can select a section of the field of view for displaying. Multiple 360° videos acquired by pairs of cameras mimicking the position of human eyes, allows the user to experience 3D 360° video. Of course, such formats also require specialized displaying tools that allow for separate representations for each eye and a way of changing the point of view by rotating around a fix point, thus limiting the field of view to a reasonable amount. Having multiple 360° videos may also allow a user to change the point of view by switching between the positions from where the videos where recorded in a similar fashion to MVD formats. Figure 1.14 shows a typical camera setup for recording a 3D 360° video. Each pair of cameras (red and yellow) are used to record a view for each eye. Stitching the views from all yellow or red cameras will produce a full 360° video. As the desired result is a seamless 360° video, it is reasonable to assume that the more cameras our system contains the better quality we can expect. However, due to practical limitations, only a limited number of cameras can be used. In order to overcome these limitations view synthesis techniques are employed to generate a so-called *virtual view* by interpolating the existing views and rendering a new point of view in between them. Figure 1.14 shows how a completely new position can be synthesized for the 360° video. $a$ and $c$ are the centers of the camera system and $b$ and $d$ represent a virtual position that is synthesized in case depth information is available for only the right or the left views of each pair.

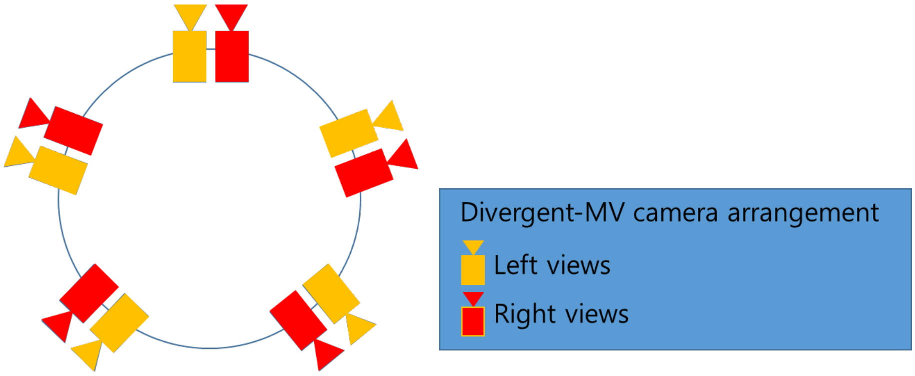


Figure 1.13 – Divergent-MV camera setup. [BLTW16].

Due to the flexibility provided by 360° video and the high number of possible

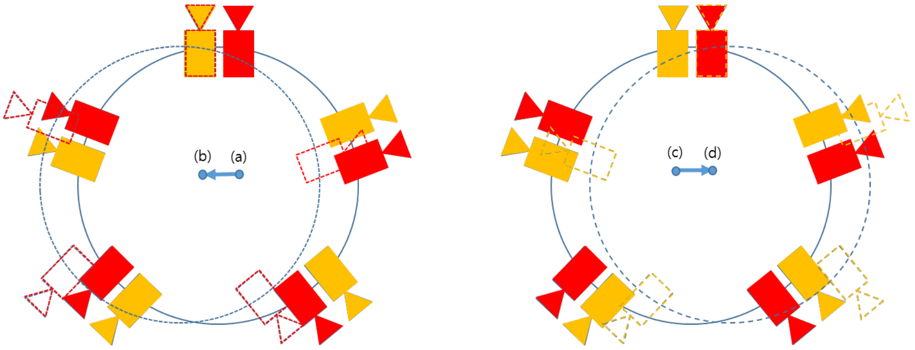


Figure 1.14 – Divergent-MV camera setup for synthesized views to the left or right of the original view. [BLTW16].

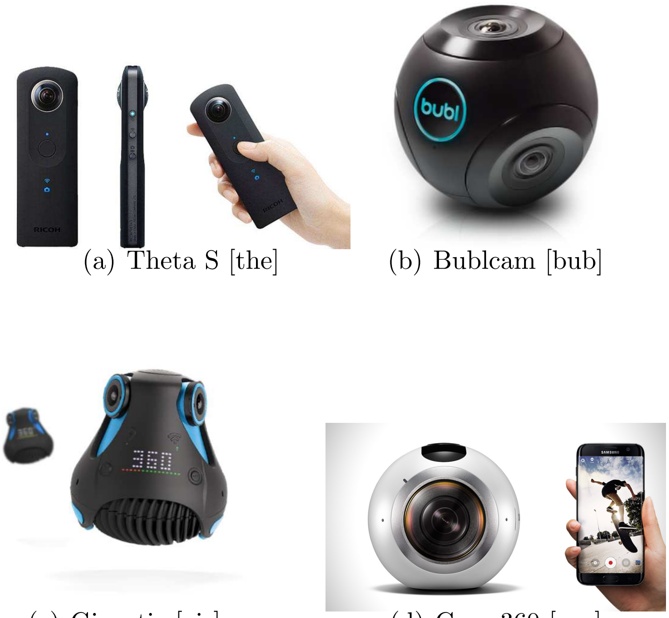
applications, a lot of interest has been expressed by large companies such as Google or Facebook. The Moving Pictures Experts Group (MPEG) also initiated a new set of exploration experiments [BLTW16] designed to evaluate the available technologies for 3D 360 video. A key point, as noted by MPEG, is that the user experience is highly dependent on the view synthesis algorithms. Because 360° videos require stitching multiple sequences together and also additional positions can be generated for 3D 360° video view synthesis algorithms are critical to obtaining good results.

Another aspect that also influence the quality of the 360° video is the photometric consistency of the multiple views. Since, each camera has a different field of view, variations in scene illumination, different camera auto exposure or auto white balance, may lead to the same object having a different color and brightness in two views. Furthermore, depending on the acquisition system our views can be desynchronized in time. This is also an aspect which should be taken into account.

### 1.5.1 360° video cameras

In this section we present a short review of the available technology for recording and displaying 360° video. As this is a new trend in multimedia and most companies are still releasing new products and finding new ways of using 360° videos there is no standard way to do it. We will not focus on a single solution but rather try to provide a short overview of the existing and upcoming solutions. One thing to note is that some steps are always present in generating a 360° video, such as: stitching that uses view synthesis or other methods to match the different views; fisheye effect correction; camera alignment; view blending and photometric correction.

Some of the simplest solutions to generate 360° videos are in the form of small portable devices with two or more opposing fisheye lens cameras. In this category we have the Ricoh Theta S, Bublcam, Giroptic or Samsung Gear 360. These devices are designed to be easy to use and can usually connect with modern smartphones or stream video directly on television sets. In general this type of devices have two (Theta S and Gear 360) to three cameras (Giroptic and Bublcam) and can record videos from full HD resolution up to 2K at 30 frames per second. The videos can be retrieved either from each camera and processed afterwards or a simple fast stitching algorithm is performed onboard. The goal is to provide a quick solution to acquire 360° videos often at the expense of quality.



(a) Theta S [the] (b) Bublcam [bub]

(c) Giroptic [gir] (d) Gear 360 [gea]

Figure 1.15 – 360° portable cameras. Ricoh Theta S and Bublcam on the top row and Giroptic and Gear 360 on the bottom row.

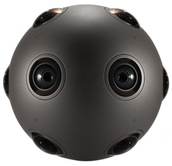


Figure 1.16 – Nokia Ozo profesional 360° camera [nok].

More complex solutions are provided by Nokia in the form of Nokia Ozo camera (Figure 1.16). This device contains eight synchronized 2K×2K sensors each equipped with a 195 degree angle of view. Another solution is offered by GoPro in the form of camera rigs as shown in Figure 1.17.

Figure 1.17 – GoPro camera rig for 360° video [gop].

Another system that was launched last year was developed by Facebook. The solution provides a camera system and an associated software for post processing and camera control. The system can create 3D 360 videos with resolutions of up to 8K for each eye.

As discussed earlier there are a lot of challenges to overcome in order to produce 360 content. Facebook divides these challenges in 3 major components:

1. The hardware
2. The camera control software
3. The stitching and rendering software

We will further describe only the third component as this contains the main algorithms used in producing the 360 videos. During this step the sequences recorder by each camera are combined to produce the final video. As stated by Facebook the main steps in the stitching and rendering component are:

Figure 1.18 – Facebook Surround 360° camera rig [fac].

1. Convert the raw images to gamma-corrected RGB.
   (a) Mutual camera color correction
   (b) Anti-vignetting

   (c) Gamma correction

   (d) Sharpening

   (e) Pixel demosaicing

2. Image correction to remove the lens distortion and re-project the image into a polar coordinate system.

3. Compensate for slight misalignments in camera orientation.

4. Compute optical flow between pairs of cameras to compute left-right stereo disparity.

5. Synthesize new views for each view direction.

6. Composite final pixels of left and right flows.

The main difference of this approach is the use of optical flow to compute the disparity between cameras in order to synthesize any number of intermediary views. As noted by Facebook this is an ill-posed inverse problem due the presence of occlusions.

### 1.5.2 Remarks

Even though 3D 360° video is a new technology still under development there are already various acquisition solutions available on the market. However, to take full advantage of this technology, new devices are required to view the content. Mobile devices and personal computers are limited as they usually cannot display 3D 360° content. A new generation of devices dedicated towards virtual reality and 3D 360° content viewing is becoming more and more popular. The so-called virtual reality headsets. Even though, this is not a new concept, due to the increase in 360° content and the appearance of main stream devices that can record 360° video, the VR headsets are a topic of interest.

As can be seen, the key software component in obtaining high quality 360° content is the stitching and rendering stage. As Facebook noted, using optical flow and view synthesis methods will generally produce higher quality results than other fast algorithms which aim at overlapping common edges of the field of view.

# Chapter 2

# View synthesis exploiting temporal prediction

## Contents

In this chapter we investigate the use of temporal correlations in view synthesis for MVD formats. Several techniques are introduced to address this challenge from a robust sub-pixel precision warping method for both texture and MVFs to full frame temporal rendering of frames in a virtual view. Furthermore, based on our findings we propose a combination of encoding and synthesis that significantly increases the quality of the default rendering software included in 3D-HEVC test model [ZTWY13].

## 2.1 Context and chapter overview

### 2.1.1 Context

Recent advances in video acquisition, compression and transmission technologies have brought significant market potential for immersive communications. Common examples [DPPC13] [TTFY11] include immersive teleconference systems, 3D video, holography and Free Viewpoint Television (FTV). A typical format for some of these applications is the MultiView Video composed of a set of N video sequences representing the same scene, referred to as views, acquired simultaneously by a system of N cameras positioned under different spatial configurations. An alternative representation is the Multiview-Video-Plus-Depth format [MSMW07], where the depth information is used in addition to texture for each viewpoint. This allows for a less costly synthesis of much more virtual views, using for example Depth-Image-Based-Rendering (DIBR) methods [Feh03].

View synthesis is the process of extrapolating or interpolating a view from other available views. It is a popular research topic in computer vision, and numerous methods have been developed in this field over the past four decades. View synthesis techniques can be mainly classified in three categories [SK00]. The methods in the first category, like DIBR, require explicit geometry information such as depth or disparity maps to warp the pixels in the available views to the correct position in the synthesized view [ZWPSxZy07] [CLLY08]. Methods in the second category require only implicit geometry, like some pixel correspondences in the available and synthesized view, that can be computed using optical flow [DCPP14] [KMW95] for instance. Finally, methods in the third category require no geometry at all. They appropriately filter and interpolate a pre-acquired set of samples (examples of tools in this category include light field rendering [LH96], lumigraph [BBMG01], concentric mosaics [SH99]). A common problem in view synthesis are areas that are occluded in the available views but should be visible in the virtual ones. These areas appear as holes in virtual views, also referred to as disocclusions . This problem is currently resolved by using inpainting algorithms such as the ones described in [DPP10] and [GM14]. Two of the most popular inpainting algorithms were developed by Bertalmio and Sapiro [BS00] and Criminisi *et al.* [CPT04].

Recently, the Moving Pictures Experts Group (MPEG) expressed a significant interest in MVD formats for their ability to support 3D video applications. This new activity is mainly focused on developing a 3D extension of the HEVC [HEV13] video coding standard, after a first standardization activity finalized with Multiview Video Coding (MVC) [CWU$^+$09]. An experimental framework was developed as well, in order to conduct the evaluation experiments [EXP10]. This framework defined a View Synthesis Reference Software (VSRS) as part of the 3D-HEVC test model [ZTWY13], which would later become an anchor to several new rendering techniques. Furthermore, establishing whether encoding all views or synthesizing some from coded views is better for multiview video sequences is still an open matter.

Traditionally, view synthesis methods, and VSRS in particular, only use inter-view correlations to render virtual views. However temporal correlations can also be exploited to improve the quality of the synthesis. In general, this type of methods synthesize or improve the synthesis of a frame by extracting additional information from different time instants, as opposed to DIBR methods which only use adjacent views at the same time instant. For instance in [SK10] the authors use Motion Vector Fields (MVFs) between frames of the intermediate views to improve the view synthesis in MVC standard. Chen *et al.* [CTL$^+$10] use MVFs computed through block-based ME in the reference views and then warp both the start and end point of the vectors in the synthesized view. The MVs are then used to retrieve information about dis-occluded regions from other frames. Sun *et al.* [SAX$^+$12] and Kumar *et al.* [KGV13] use adjacent views to extract background information from multiple time instants, used for hole filling in a DIBR synthesis. In [Siu12] the authors use the information from the current and other frames of the synthesized video to fill hole regions. Other studies use the inter-view correlations directly during coding, view-synthesis prediction (VSP) [MBXV06] [MFdW07] [YV09] or take advantage of multiview format redundancies to deal with network packet loss [LCCJ12]. Yuan *et al.* [YLLL12] use Wiener filter to improve the synthesis by eliminating distortions caused by coding. A comprehensive review of the state-of-the-art of view synthesis techniques is presented in Section 2.2.

### 2.1.2 Chapter overview

Section 2.3 presents the epipolar constraint model we impose between frames of different views at multiple time instants. Based on this formulation the following sections propose several view synthesis approaches that exploit temporal correlations:

- Section 2.4 introduces a sub-pixel precision warping and filtering (Wf) technique that can be used for both MC and DIBR warping. This technique is used to warp the reference adjacent views, and then to reversely motion compensate pixels from a past or future frame using derived MVs (based on the epipolar constraint) from the left or right base views. This allows us to partly fill disocclusions with real background information from other temporal instants - Temporal Hole Filling (THF). The remaining holes can be filled with any inpainting algorithm.

- In Section 2.5 we extend the THF idea (Section 2.4) and use full frame temporal predictions. More specifically, we use forward motion prediction with MVFs computed in the reference views and warped in the synthesized view to obtain up to four temporal predictions which are blended together with the DIBR predictions using either an average (P+Bavg) or adaptive approach (P+Badapt).

- The last approach (Section 2.6 ) proposes a modification of the 3D-HEVC coding scheme in order to further enhance the synthesized frames. The previous ideas are combined and additional tools are introduced in order to integrate them with 3D-HEVC. Two synthesis schemes are studied: a Direct and Hierarchical one, employed both on the temporal and inter-view axes. Furthermore, an adaptive fusion method (AF) that further extends the ideas of P+Badapt is introduced to judiciously select between temporal or inter-view prediction in case of ME failure. This method will be referred as View Synthesis Exploiting Temporal prediction in 3D-HEVC or simply VSTP.

### 2.1.3 Remarks

The approaches proposed in this chapter are designed to complement the synthesis method used in the 3D-HEVC standardization process in order to improve visual quality. We use optical flow to derive dense MVFs between frames in the adjacent views, then warp them at the level of the intermediate view. This allows us to build different temporal predictions from left and right adjacent views using reference frames at two time instants (past and future). Other ME techniques that are less computationally intensive can also be used at the cost of prediction accuracy [KKS$^+$00] [CHKK07] [WZHT10]. However, we prefer using an optical flow ME technique, since it offers a more accurate prediction [Mor14a].

## 2.2 State of the art of view synthesis techniques

In this state-of-the-art, we focus on the first class of view synthesis methods, also referred to as DIBR techniques. We first discuss the rendering technique used in the reference software for view synthesis, and in the rendering software used by the Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V) [Tec]. Then, an overview of other rendering techniques found in the literature is presented.

### 2.2.1 Reference software

#### 2.2.1.1 View Synthesis Reference Software

VSRS inputs two texture views and their two associated depth maps, along with intrinsic and extrinsic camera parameters. The output is a synthesized intermediate view. VSRS allows synthesizing frames using two operational modes: a general mode and a 1D mode, respectively used for non-parallel (e.g. cameras aligned in an arc) and 1D-parallel (cameras are aligned in a straight line perpendicularly to their optical axes) camera settings.

Figure 2.1 illustrates the rendering process in the general mode of VSRS. First, the left and right reference depth maps ($s_{D,l}$ and $s_{D,r}$) are warped to the virtual view position, giving $s'_{D,l}$ and $s'_{D,r}$. The occlusions are handled by the highest depth value (closest to the camera), usually the depth values are reversed quantized from 0 to 255 such that the highest value in the depth map corresponds to the lowest depth of the scene [DPPC13]. $s'_{D,l}$ and $s'_{D,r}$ are then median filtered to fill small holes, giving $s''_{D,l}$ and $s''_{D,r}$. A binary mask is maintained for each view to keep track of larger holes caused by disocclusions. $s''_{D,l}$ and $s''_{D,r}$ are then used to warp the texture views $s_{T,l}$

and $s_{T,r}$ to the virtual view position, giving $s'_{T,l}$ and $s'_{T,r}$ (this reverse warping process wherein the depths are warped first and then used to warp the texture is reported to give a higher rendering quality [EXP10]). Holes in one of the warped views are filled with collocated non-hole pixels from the other warped view, if available. This gives $s''_{T,l}$ and $s''_{T,r}$, which are then blended together to form a single representation. The blending can be a weighted average according to the distance of each view to the virtual view point (Blending-On mode), or it can simply consist in taking the closest view to the virtual view point, and discarding the other (Blending-Off mode). The binary masks of each view are merged together at this stage and the remaining holes are filled at the final stage of the algorithm by propagating the color information inward from the region boundaries.

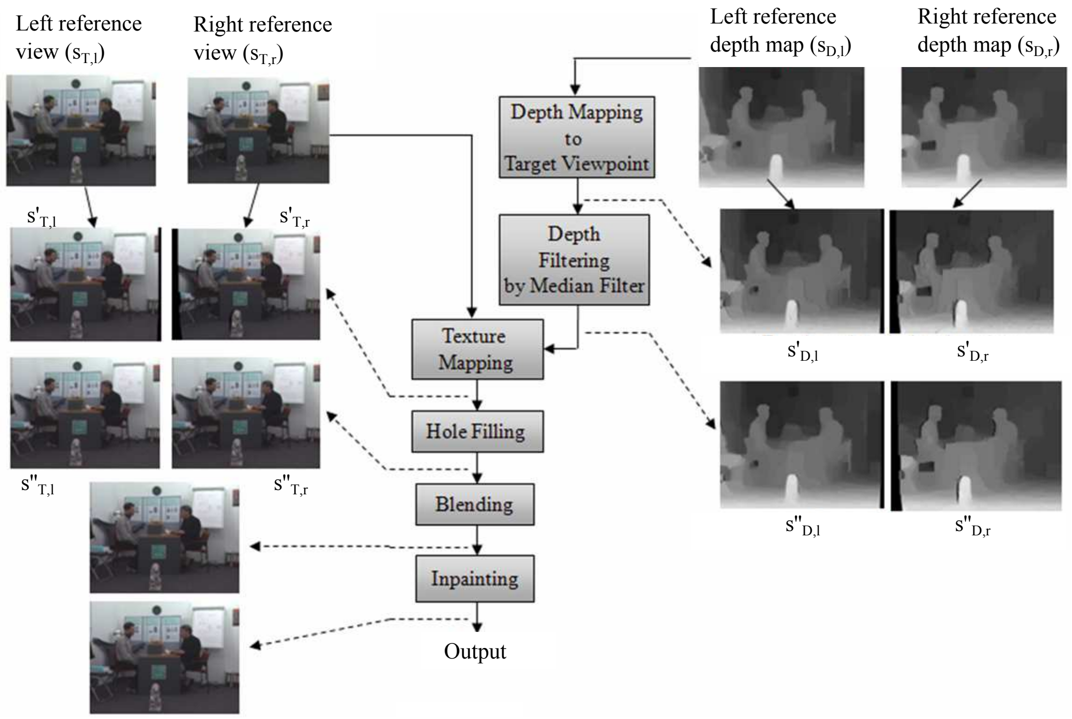

Left reference view ($s_{T,l}$)
Right reference view ($s_{T,r}$)
Left reference depth map ($s_{D,l}$)
Right reference depth map ($s_{D,r}$)
Depth Mapping to Target Viewpoint
Depth Filtering by Median Filter
Texture Mapping
Hole Filling
Blending
Inpainting
Output
$s'_{T,l}$
$s'_{T,r}$
$s''_{T,l}$
$s''_{T,r}$
$s'_{D,l}$
$s'_{D,r}$
$s''_{D,l}$
$s''_{D,r}$

Figure 2.1 – Flow diagram for View Synthesis Reference Software (VSRS) general mode [EXP10].

The 1D mode of VSRS works a bit differently. In this mode, the camera setup is assumed to be 1D parallel. This allows to make a number of simplifications to the warping process which is reduced to a simple horizontal shift. First, the color video is up-sampled for half-pixel or quarter pixel accuracy. A “CleanNoiseOption” and “WarpEnhancementOption” avoid warping unreliable pixels. The process gives two warped images, two warped depth maps and two binary masks from the left and right reference views. Each pair is then merged together. When a pixel gets mapped from both the left and the right reference views, the final pixel value is either the pixel closest to the camera or an average of the two. Remaining holes are filled by propagating the background pixels into the holes along the horizontal row. Finally, the image is downsampled to its original size.

#### 2.2.1.2 View Synthesis Reference Software 1D Fast

Each contribution to the 3D-HEVC standardization that proposes to modify the coding of dependent views or depth data, is required to present coding results on synthesized views. The software used for synthesizing the intermediate views is a variant of VSRS, called View Synthesis Reference Software 1D Fast (VSRS-1DFast). This software is included in the HTM package, and is documented in the 3D-HEVC test model [ZTWY13]. VSRS-1DFast allows inputting two or three texture and depth views along with their corresponding camera parameters, and synthesize an arbitrary number of intermediate views. Just like the 1D mode of VSRS, VSRS-1DFast assumes that the camera setup is 1D parallel. Figure 2.2 illustrates the different steps of the rendering algorithm used in VSRS-1DFast. The texture views $s_{T,l}$ and $s_{T,r}$ are first upsampled to obtain $\widehat{s}_{T,l}$ and $\widehat{s}_{T,r}$: the luma component is upsampled by a

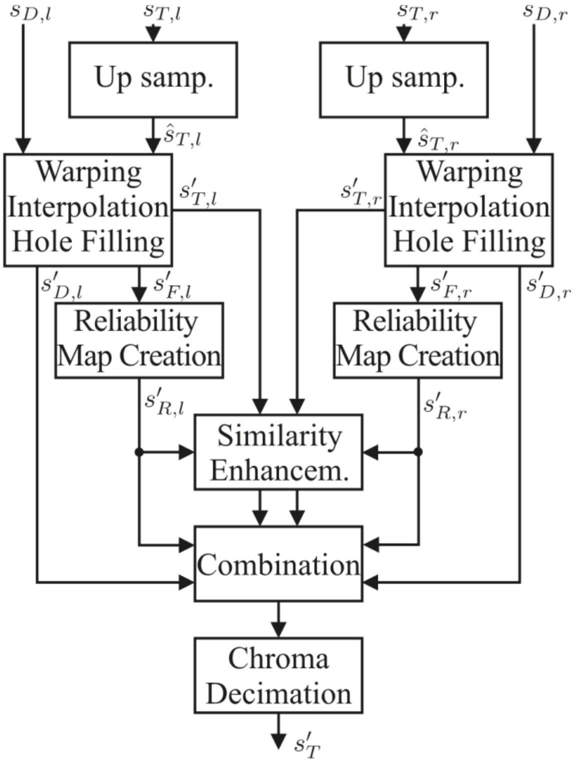


Figure 2.2 – Flow diagram for View Synthesis Reference Software 1D Fast (VSRS-1DFast) [ZTWY13].

factor of four in the horizontal direction, and the chroma by a factor of eight in horizontal direction and two in vertical direction, thus yielding the same resolution for all components. The warping, interpolation and hole filling are carried out for $\widehat{s}_{T,l}$ and $\widehat{s}_{T,r}$ line-wise. This gives two representations of the synthesized frame: $s'_{T,l}$ and $s'_{T,r}$. Then, two reliability maps $s'_{R,l}$ and $s'_{R,r}$ are determined indicating which pixels correspond to disocclusions (reliability of 0). A similarity enhancement stage then adapts the histogram of $s'_{T,l}$ to the one of $s'_{T,r}$. Finally, $s'_{T,l}$ and $s'_{T,r}$ are combined. If the "interpolative rendering" option is activated, the combination depends on the warped depth maps and the two reliability maps created. If not, the synthesized view is mainly rendered from one view and only the holes are filled from the other view. The resulting combination is later down-sampled to the original size of the texture views.

### 2.2.2 Rendering techniques in literature

In [FLG13], a rendering technique called View Synthesis using Inverse Mapping (VSIM) is introduced. It operates at full-pel accuracy and assumes a 1D-parallel camera setting. The left and right texture views are warped to the synthesized view position using simple horizontal shifts, also called column shifts. A table is maintained for the left and right interpretations of the synthesized frame which records the column shift of each pixel. Holes in these two tables are filled using a median filter. Then, the two representations are merged and the remaining holes are filled by checking the collocated value in the tables, and inverse mapping the pixel back to its original value in the left or right view. Residual holes are filled by simply assuming that their depth is the same as the depth of the collocated pixels in the original views. VSIM outperforms VSRS, on average, by 0.41 dB at quarter-pel accuracy and by 1.35 dB at full-pel accuracy on 5 sequences. However, the rendering runtime is not provided, making it difficult to assess the complexity of the method.

In [LE10], the depth maps are pre-processed with an adaptive smoothing filter in order to reduce holes after synthesis. The filter is only applied to edges in the depth map (corresponding to an abrupt transition in depth values) since these are the main cause for holes. The method is thus less complex than methods which apply a symmetric or asymmetric smoothing filter to the entire depth map. Furthermore, if hole regions correspond to vertical edges, an asymmetric Gaussian smoothing filter is used to further pre-process the depth map. No objective gains are reported, but a perceptual improvement is noticed on some synthesized sequences.

A technique that does not require pre-processing the depth map is introduced in [WLS+11]. A hole in the synthesized texture image is filled by the color of the neighboring pixel (between the 8 direct neighboring pixels) with the smallest depth value in the synthesized depth map (this is referred to as Horizontal, Vertical and Diagonal Extrapolation (HVDE)). The two warped texture images are complemented (holes in one are filled with available pixel values in the other), and later blended, giving a final image $W$. The same process (HVDE, complementation, and blending) can also be performed in case the depth maps were pre-processed with a bi-lateral smoothing filter, giving an image $A$, which would then be used to fill remaining holes in $W$. This technique is reported to outperform basic DIBR by 1.78 dB on one sequence.

Another method for improving the quality of the synthesis is to apply a non-linear transformation to the depth maps [WZ11]. Specifically, the depth range of points in the background is compressed, such that these points would have the same or slightly different depths. This reportedly reduces holes in the synthesis. The transformation depends on the depth map histogram. Objective gains are not presented but a visible improvement is noticed on the shown images.

Another desired feature is the possibility to freely change the quality of a synthesized view. Since the quality of DIBR rendering depends on the actual synthesis process, additional boundary artifact processing can be used to adjust the quality of the synthesis. Zhao *et al.* analyze and reduce the boundary artifacts from a texture-depth alignment perspective in [ZZC+11]. In [CVO11] Cheung *et al.* tackle the problem of bit allocation for DIBR multiview coding. The authors use a cubic distortion model based on DIBR properties and demonstrate that the optimal selection of QPs for texture and depth maps is equivalent to the shortest path in a specially constructed 3D trellis. Xiao *et al.* [XHT+14] propose a scalable bit allocation scheme, where a single ordering of depth and texture packets is derived. Furthermore, depth packets are ordered based on their contribution to the reduction of the synthesized view distortion.

Other works also exploit pixel-based processing with dense MVFs with an end goal of improving the synthesis at the decoder side. Li *et al.* compute dense MVFs on texture in [LLZ+14]. Time consuming optical flow computations are limited only around the edges of objects. Additional depth predictors are obtained by mapping the MVs computed on texture to depth. The depth map improvement is reflected in a high increase of quality for synthesized views.

### 2.2.3 Remarks

The rendering techniques used in the reference softwares, and in most contributions in literature, are all based on 3D image warping using depth maps. Pixels from reference views are mapped to pixels in the virtual view using the disparity information that the depth maps convey. However, we show that the synthesis can be improved by extending DIBR to the temporal axis.

## 2.3 Epipolar constraint

As previously discussed, most view synthesis algorithms warp the texture of a given frame using the associated depth maps to compute disparity vectors (DVs). However, temporal correlations in a video sequence, in the form of MVF, could be used to further improve it. The challenge is to obtain a MVF that can be used in the synthesized view. Directly computing the MVF between synthesized frames may provide a bad estimation as the reference and predicted frames are affected by synthesis artifacts. A possible solution when dealing with MVD sequences is to use inter-view correlations to link the MVFs of different views.

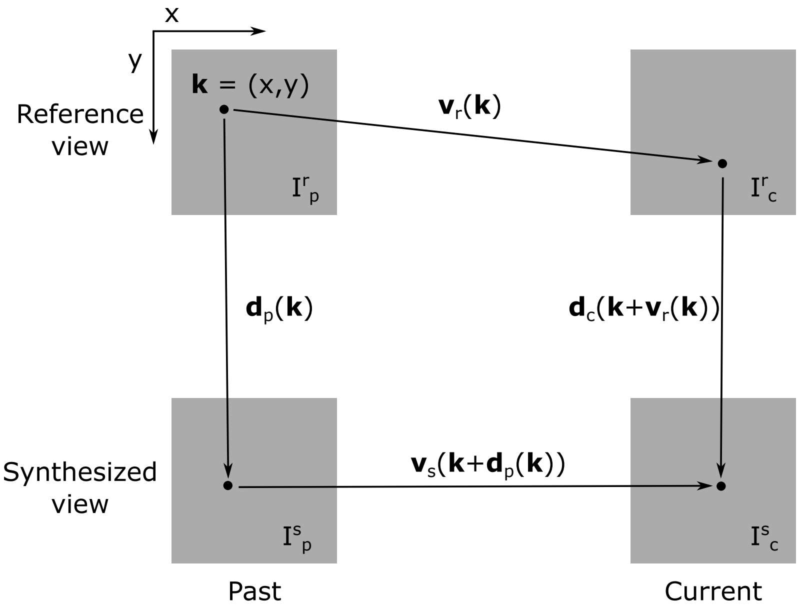

x
y
Reference view
$\mathbf{k} = (x,y)$
$\mathbf{v}_r(\mathbf{k})$
$I^r_p$
$I^r_c$
$\mathbf{d}_p(\mathbf{k})$
$\mathbf{d}_c(\mathbf{k}+\mathbf{v}_r(\mathbf{k}))$
Synthesized view
$\mathbf{v}_s(\mathbf{k}+\mathbf{d}_p(\mathbf{k}))$
$I^s_p$
$I^s_c$
Past
Current

Figure 2.3 – Epipolar constraint, the relation between the disparity vector fields (DVFs) $\mathbf{d}_p$ and $\mathbf{d}_c$ at two time instants $c$ and $p$ respectively, and the MVFs in the synthesized and reference view $\mathbf{v}_s$ and $\mathbf{v}_r$ respectively for a position $\mathbf{k}$ in the reference frame $I^r_p$.

Figure 2.3 shows the relation between the positions of a real-world point projection in different views and at different time instants. Let us consider $I_p^r$, $I_c^r$, $I_p^s$, $I_c^s$ which are, respectively, the reference ($r$) view frames and the synthesized ($s$) view frames at a past $p$ and current $c$ time instant. Let $M \times N$ be the size of the image with $M$ being the height and $N$ the width. Let $\mathbf{k} = (x, y)$ be a point in $I_p^r$, $\mathbf{v}_r(\mathbf{k})$ its associated MV ($I_c^r$ is the reference frame for $I_p^r$), pointing to a corresponding point in $I_c^r$, and $\mathbf{d}_p(\mathbf{k})$ its associated DV, pointing to a corresponding point in $I_p^s$. Let $\mathbf{v}_s(\mathbf{k} + \mathbf{d}_p(\mathbf{k}))$ be the MV of the projection of $\mathbf{k}$ in $I_p^s$ and $\mathbf{d}_c(\mathbf{k} + \mathbf{v}_r(\mathbf{k}))$ the DV of the projection of $\mathbf{k}$ in $I_c^r$. If the point is not occluded, there is only one projection of $\mathbf{k}$ in $I_c^s$, so the two vectors will point to the same position. This defines a so-called epipolar constraint [DMPP10] on $\mathbf{k}$, which can be written as:

$$\mathbf{v}_r(\mathbf{k}) + \mathbf{d}_c(\mathbf{k} + \mathbf{v}_r(\mathbf{k})) = \mathbf{d}_p(\mathbf{k}) + \mathbf{v}_s(\mathbf{k} + \mathbf{d}_p(\mathbf{k})) \tag{2.1}$$

Note the sense of the MVFs, in this formulation every pixel in image $I_p^r$ has an associated MV and DV. Thus, all pixels in $I_p^r$ can be predicted from $I_c^r$, while $I_c^r$ cannot be fully predicted from $I_p^r$ using $\mathbf{v}_r$. Similarly, the same statement can be said about frames $I_p^r$ and $I_p^s$ and DVF $\mathbf{d}_p$.

## 2.4 Temporal hole filling and sub-pixel precision warping

Traditional hole filling algorithms approximate missing information from surrounding pixels. The goal of this method is to fill the disoccluded areas in synthesized views using real scene information from different time instants.

### 2.4.1 Temporal hole filling

In general, disocclusions in the synthesized view can be classified in two categories depending whether the area in the reference view is a border or non-border occlusion with respect to the image [HKA13]. Border occlusions occur due to the reference image missing portions of the field of view that should be visible in the synthesized view. This types of occlusions are resolved by performing a synthesis from a left and a right reference view. The non-border occlusions are caused by objects in the foreground that obscure parts of the background that should be visible in the synthesis. Due to the motion of the foreground objects and camera, this types of occlusions vary over time and produce different holes at different time instants in the synthesized view. Thus, a part of the missing information may be available in frames at different time instants. By exploiting the temporal correlation in the video sequence it is possible to retrieve this information and reduce the size and the number of holes in the synthesis.

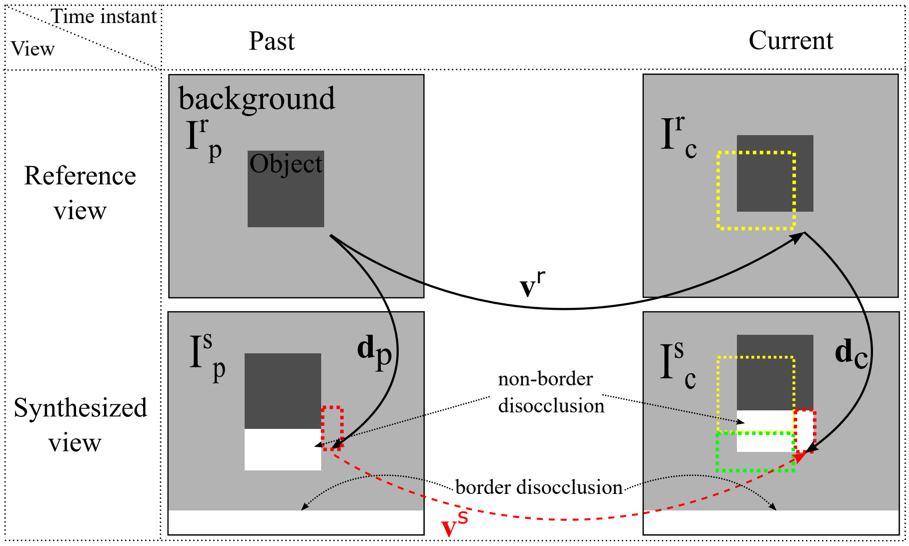


Figure 2.4 – Temporal retrieval of disoccluded area. Yellow dotted squares mark the position of the object in the previous frame, the green dotted square shows the disocclusions in the previous frame and red dotted squares mark the disoccluded area that was visible in a previous frame.

In Fig. 2.4, a foreground object is represented in two views at two different time instants, black arrows represent the MVF in the reference view ($r$) and DVFs for a past ($p$) and current ($c$) time instant ($\mathbf{v}^r$, $\mathbf{d}_p$, $\mathbf{d}_c$) and the red dashed arrow represents the MVs in the synthesized view, which can be used to retrieve information about the disoccluded area. Yellow and green dotted lines show the position of the object and disoccluded area respectively, in the past frame. It can be observed that a part of the disoccluded area in the current frame was visible in a past frame due to the motion of the object (this is shown in the figure with a dotted red line).

In Fig. 2.5 we show the relation between MVF and disparity maps for three views of a MVD sequence. Let us consider two base views, left ($L$) and right ($R$), and an intermediate view, which is synthesized at the decoder side using classic DIBR algorithms. $I^r_{pL}$, $I^r_{cR}$ and $I^s_f$ denote frames from the left, right and synthesized views respectively at a past, current or future time instant $(p, c, f)$. $\mathbf{v}$ and $\mathbf{d}$ are the MVF and disparity maps respectively.

Using the epipolar constraint as defined in Section2.3, the projection of a real world point in the synthesized view can be modeled using disparity maps. Thus, considering a point $\mathbf{k} = (x, y)$ in frame $I^r_{pL}$. The epipolar constraint for quadrant 0 can be written as:

$$\mathbf{v}^r_{pL}(\mathbf{k}) + \mathbf{d}_{cL}(\mathbf{k} + \mathbf{v}^r_{pL}(\mathbf{k})) = \mathbf{d}_{pL}(\mathbf{k}) + \mathbf{v}^s_{pL}(\mathbf{k} + \mathbf{d}_{pL}(\mathbf{k})) \tag{2.2}$$

Based on Eq. 2.2, $\mathbf{v}^s_{pl}$ can be derived from $\mathbf{v}^r_{pL}$, $\mathbf{d}_pL$ and $\mathbf{d}_cL$ as:

$$\mathbf{v}^s_{pL}(\mathbf{k} + \mathbf{d}_{pL}(\mathbf{k})) = \mathbf{v}^r_{pL}(\mathbf{k}) + \mathbf{d}_{cL}(\mathbf{k} + \mathbf{v}^r_{pL}(\mathbf{k})) - \mathbf{d}_{pL}(\mathbf{k}) \tag{2.3}$$

In other words $\mathbf{v}^s_{pL}$ warped with the disparity map $\mathbf{d}_{pL}$ is equal to $\mathbf{v}^r_{pL}$ with the motion intensity adjusted with the difference in disparity for point $\mathbf{k}$ at current and past time instants. This approach can be applied for past and future time instants using either the left or right view. A dense MVF in the base views can be obtained from the current frame and either a future or past one with an optical flow algorithm [Liu]. Assuming we are dealing with a 1D arrays of rectified cameras, the disparity maps only have an $x$ component, which is easily computed from the corresponding depth maps of each base view [DPPC13] as:

$$\mathbf{d}(\mathbf{k}) = f \cdot B \left[ \frac{Z(\mathbf{k})}{255} \left( \frac{1}{Z_{min}} - \frac{1}{Z_{max}} \right) + \frac{1}{Z_{max}} \right] \tag{2.4}$$

where $Z(\mathbf{k})$ is the inversed depth value of point $\mathbf{k}$, $Z_{min}$ and $Z_{max}$ are the minimum and maximum depth values respectively, $f$ is the focal length of the camera and $B$ is the baseline between the synthesized and base views.

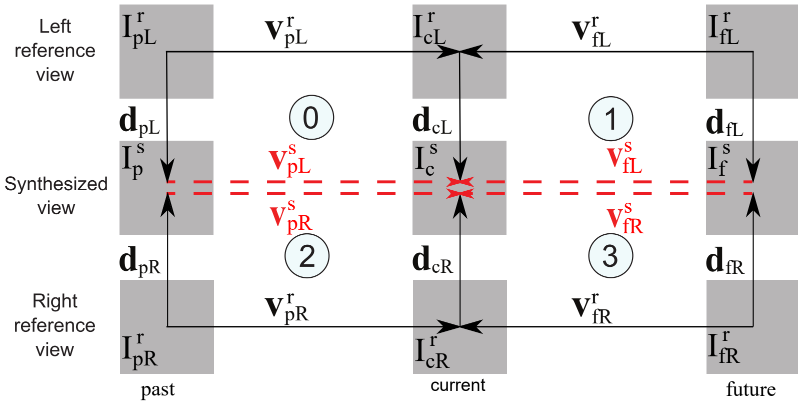


Figure 2.5 – Temporal hole filling scheme, for two base views and an intermediary synthesis, using past and future synthesized frames to retrieve information.

If we decompose Eq. 2.3 for the $x$ and $y$ components, we have:

$$\begin{array}{l}\mathbf{v}^s_{pL,x}(x+\mathbf{d}_{pL,x}(x,y),y)=\mathbf{v}^r_{pL,x}(x,y)+\mathbf{d}_{cL,x}(x+\mathbf{v}^r_{pL,x}(x,y),y+\mathbf{v}^r_{pL,y}(x,y))-\mathbf{d}_{pL,x}(x,y)\\ \mathbf{v}^s_{pL,y}(x+\mathbf{d}_{pL,x}(x,y),y)=\mathbf{v}^r_{pL,y}(x,y)\end{array} \tag{2.5}$$

#### 2.4.1.1 Remarks

Note that the MVs always point from past or future frames ($I^r_p$, $I^r_f$) to the current frame ($I^r_c$). In order to retrieve the information on the disoccluded areas of $I^s_c$ we would need to perform a backward MC.

The MVF ($\mathbf{v}^s_{pl}$) may contain holes after warping it at the level of the synthesized view. Note that $\mathbf{v}^s_{pl}$ is defined in all positions $\mathbf{k}+\mathbf{d}_{pL}(\mathbf{k})$ when $\mathbf{k}\in I^r_{pL}$. Since the synthesis is performed by warping $I^r_c L$ with $\mathbf{d}_{cL}$, $I^s_c$ will be defined in all positions $\mathbf{k}+\mathbf{d}_{cL}(\mathbf{k})$ when $\mathbf{k}\in I^r_{cL}$. The backward MC ensures that holes in the MVF do not match the disoccluded areas of $I^s_c$. Disocclusions at different time instants do not necessary coincide and so additional information can be obtained for the current frame.

Finally, we are able to obtain 4 MVFs as shown in Fig. 2.5 and multiple temporal predictions of the same disoccluded area are averaged. Since, integer rounding of motion or disparity predictors is inefficient, we propose a sub-pixel precision warping to be used in parallel with this method.

### 2.4.2 Sub-pixel precision warping

In addition to warping the side views as traditionally done in view synthesis, our method requires a warping of the MVFs in the adjacent views and a backward MC of past or future frames. In practice our hole filling method requires a number of different warping operations, namely: typical DIBR image warping, a warping of the dense MVFs, a MC of disparity and a backward MC to retrieve disoccluded areas. In order to take full advantage of dense MVFs and depth computed disparity we propose a simple technique for sub-pixel precision warping and backward MC.

To better describe our method let us consider the MVF $\mathbf{v}_s(\mathbf{k})$, the DVF $\mathbf{d}_c(\mathbf{k})$, the images $I^s_p$,$I^r_c$ and a warped image defined on possibly fractional positions $I^{fg}$. $\mathbf{u}=(x,y)$ represents a set of coordinates in $I^{fg}$. Each position $\mathbf{k}=(c,r)$ in the image, MVF or DVF will correspond to a position $\mathbf{u}$ in $I^{fg}$ through the function $\tau$

as shown in Eq. 2.6:

$$\tau(\mathbf{k}) = \mathbf{u}, \qquad \tau(\mathbf{k}) = (c/\alpha, r/\alpha) \tag{2.6}$$

where $\alpha$ is defined as $1/t$ and $t \in \mathbb{N}$ is used to indicate the precision of the warping. Considering that $\mathbf{v}_s$ and $\mathbf{d}_s$ contain fractional values, the goal is to perform a sub-pixel warping of $I_c^r$ with the DVF $\mathbf{d}_c$ and to backward motion compensate $I_p^s$ image using the derived MVF. A first step is to quantize the values in $\mathbf{d}$ in function of the precision parameter $\alpha$ as shown in Eq. 2.7:

$$\Phi_\alpha(x, y) = (\lfloor \frac{x}{\alpha} + \alpha \rfloor \alpha, \lfloor \frac{y}{\alpha} + \alpha \rfloor \alpha) \tag{2.7}$$

where $\Phi_\alpha$ is a rounding operation and "$\lfloor\rfloor$" indicates a floor operation. The quantized values of disparity and motion vectors are obtained by applying $\Phi$ over the two vector fields. The actual synthesis is performed in three steps, a warping of the inter-view reference image $I_c^r$ in $I^{fg}$, a filtering step and a temporal hole filling.

The $I_c^r$ image is warped in $I^{fg}$ as shown in Eq. 2.8:

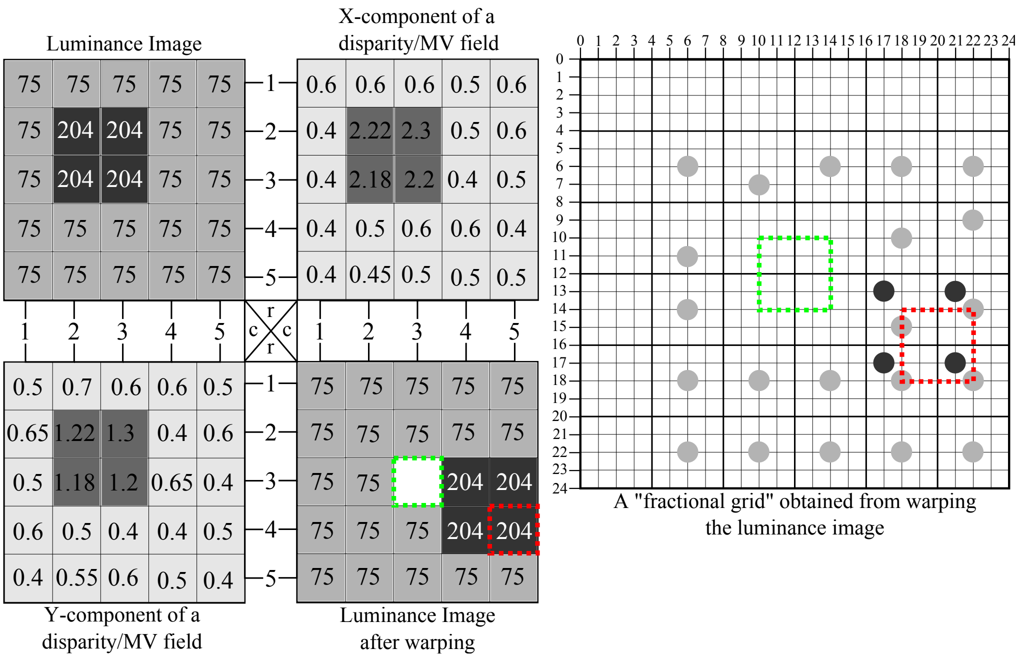
$$I^{fg}(\tau(\mathbf{k} + \Phi_\alpha(\mathbf{d}_c(\mathbf{k})))) = I_c^r(\mathbf{k}) \tag{2.8}$$

Overlapping values in $I^{fg}$ will be dealt with by using the disparity information, which relates to depth as shown in Eq. 2.4. High disparity indicates an object in the foreground and should be considered over a point with low disparity value. Nevertheless, overlaps should be marked and both values should be considered in the filtering step described in what follows.

In Fig. 2.6 we show an example of sub-pixel precision warping using our proposed method. We have a luminance matrix (top-left) with the corresponding disparity or MV field for $X$-axis (top-right) and $Y$-axis (bottom-left). On the right side of the image a fractional grid is displayed after displacing the pixels from the luminance image using Eq. 2.8. With dotted lines we represent 2 examples of filtering windows. Green indicates a hole and red an overlapping between foreground and background. The final luminance image (bottom-right) is obtained by centering a filtering window in each position $u = \tau(\mathbf{k})$. The output of the filter is obtained in two steps. First we identify the foreground luminance values by creating a list of pixels found in the filtering window and ordering them with respect to their associated depth in the reference image $I_c^r$. All $\{s, .., n\}$ positions in our list are then interpolated to obtain the final value, $s$ is obtained as the smallest value that satisfies $\Delta(s) > \beta$ and $\Delta$ is defined as:

$$\begin{aligned} \mathcal{L} &= \{d_1, .., d_i, .., d_n\} \\ \mathcal{L}_{dif} &= \{\delta_1, \delta_2, .., \delta_{n-1}\} \\ \Delta(i) &= \frac{\delta_i - \delta_{i-1}}{\delta_{i-1}} \end{aligned} \tag{2.9}$$

where $d_i$ are depth values, $\mathcal{L}$ is the list, $\delta_i = d_{i+1} - d_i$ and $\beta$ is an empirically determined threshold. Finally, we apply the temporal hole filling algorithm for unknown areas. We use derived motion vectors from the adjacent views as shown in Sec. 2.4.1 to backward motion compensate a past or future synthesized frame and extract additional information about the disoccluded area in the current frame. Note that past and future motion reference frames do not have an associated depth map, in this case when we derive a vector from the left or right MVFs to $\mathbf{v}_s$ we retain the corresponding depth from left and right, future and past frames, see Fig. 2.5. Additional unfilled disocclusions are marked for inpainting.


Luminance Image
X-component of a disparity/MV field
Y-component of a disparity/MV field
Luminance Image after warping
r
c
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
A "fractional grid" obtained from warping the luminance image

Luminance Image

| r \ c | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 75 | 75 | 75 | 75 | 75 |
| 2 | 75 | 204 | 204 | 75 | 75 |
| 3 | 75 | 204 | 204 | 75 | 75 |
| 4 | 75 | 75 | 75 | 75 | 75 |
| 5 | 75 | 75 | 75 | 75 | 75 |

X-component of a disparity/MV field

| r \ c | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0.6 | 0.6 | 0.6 | 0.5 | 0.6 |
| 2 | 0.4 | 2.22 | 2.3 | 0.5 | 0.6 |
| 3 | 0.4 | 2.18 | 2.2 | 0.4 | 0.5 |
| 4 | 0.4 | 0.5 | 0.6 | 0.6 | 0.4 |
| 5 | 0.4 | 0.45 | 0.5 | 0.5 | 0.5 |

Y-component of a disparity/MV field

| r \ c | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0.5 | 0.7 | 0.6 | 0.6 | 0.5 |
| 2 | 0.65 | 1.22 | 1.3 | 0.4 | 0.6 |
| 3 | 0.5 | 1.18 | 1.2 | 0.65 | 0.4 |
| 4 | 0.6 | 0.5 | 0.4 | 0.4 | 0.5 |
| 5 | 0.4 | 0.55 | 0.6 | 0.5 | 0.4 |

Luminance Image after warping

| r \ c | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 75 | 75 | 75 | 75 | 75 |
| 2 | 75 | 75 | 75 | 75 | 75 |
| 3 | 75 | 75 |  | 204 | 204 |
| 4 | 75 | 75 | 75 | 204 | 204 |
| 5 | 75 | 75 | 75 | 75 | 75 |

Figure 2.6 – A simple sub-pixel precision warping example with our proposed technique. Dotted lines represent filtering windows and the corresponding result in the warped image; green indicates a hole and red a case of foreground and background overlapping.

### 2.4.3 Experimental results

We test our method on four multiview sequences defined in the Common Test Conditions (CTCs) for conducting experiments with the reference software of 3D-HEVC [RMV13]: Balloons, Kendo, Newspaper and PoznanHall2. For each sequence we consider two non-adjacent reference views and we synthesize a middle view with our method and the reference VSRS1D-Fast rendering used in 3D-HEVC [ZTWY13] experiments. In order to have a fair comparison, the remaining disocclusions in our synthesis use the same filling as the reference. Each of the tested sequences is encoded using the configuration described in the CTCs. Four different QPs (25 30 35 40) are used for the texture encoding, the depth maps are encoded using corresponding QPs (34 39 42 45) as indicated by the CTCs. For more details on the sequences see [CFP11].

We evaluate the PSNR of the synthesis against original views for each sequence at each of the tested QPs. The encoding is performed with 3D-HEVC, the left view is set as base view, and the right as dependent view. The GOP size is set to 8, and the first frame of each GOP is used as a reference frame for temporal hole filling of the other frames of the GOP, inside the synthesized view. These reference frames are synthesized using the Wf technique described above without THF. In our experiments we set $\beta$ parameter to 1/10 and $\alpha$ to 1/4, and the size of the filtering window to 5, we found these values to provide best gains. The dense MVFs are computed using the optical flow algorithm in [Liu] between frames of the reference views. The optical flow parameters used in our experiments along with more details can be found in [Liu09].

| **Sequence** | **VSRS1D-Fast PSNR (dB)** | | | | **Wf+THF PSNR (dB)** | | | | **Gain (dB)** | | | | **Avg. Gain (dB)** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| QPs | 25 | 30 | 35 | 40 | 25 | 30 | 35 | 40 | 25 | 30 | 35 | 40 | |
| Balloons | 34.41 | 34.12 | 33.47 | 32.45 | 34.45 | 34.19 | 33.57 | 32.55 | 0.04 | 0.08 | 0.1 | 0.1 | 0.08 |
| Kendo | 35 | 34.53 | 33.79 | 32.77 | 35.4 | 34.92 | 34.17 | 33.1 | 0.4 | 0.39 | 0.38 | 0.32 | 0.37 |
| Newspaper | 29.2 | 29.05 | 28.78 | 28.31 | 29.83 | 29.71 | 29.4 | 28.84 | 0.63 | 0.66 | 0.62 | 0.53 | 0.61 |
| PoznanHall2 | 36.25 | 35.87 | 35.36 | 34.55 | 36.35 | 36.03 | 35.62 | 34.78 | 0.11 | 0.15 | 0.26 | 0.23 | 0.18 |

Table 2.1 – Average PSNR and gain for each sequence and each QP.

| **Sequence** | **VSRS1D-Fast PSNR (dB)** | | | | **THF PSNR (dB)** | | | | **Gain (dB)** | | | | **Holes (%)** | **Avg. Gain (dB)** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| QPs | 25 | 30 | 35 | 40 | 25 | 30 | 35 | 40 | 25 | 30 | 35 | 40 | | |
| Balloons | 24.73 | 24.89 | 24.77 | 24.27 | 26.08 | 26.01 | 25.86 | 25.3 | 1.34 | 1.12 | 1.09 | 1.03 | 0.11 | 1.14 |
| Kendo | 25.51 | 25.73 | 25.99 | 26.03 | 26.51 | 26.72 | 26.52 | 26.41 | 1 | 0.99 | 0.53 | 0.38 | 0.08 | 0.72 |
| Newspaper | 18.83 | 19.13 | 18.98 | 19.5 | 19.57 | 19.74 | 20.01 | 20.13 | 0.74 | 0.61 | 1.04 | 0.63 | 0.3 | 0.755 |
| PoznanHall2 | 28.68 | 27.85 | 28.52 | 28.67 | 30.94 | 29.77 | 28.67 | 29.71 | 2.26 | 1.92 | 0.15 | 1.04 | 0.04 | 1.34 |

Table 2.2 – Average PSNR and gain for disoccluded areas for each sequence and each QP.

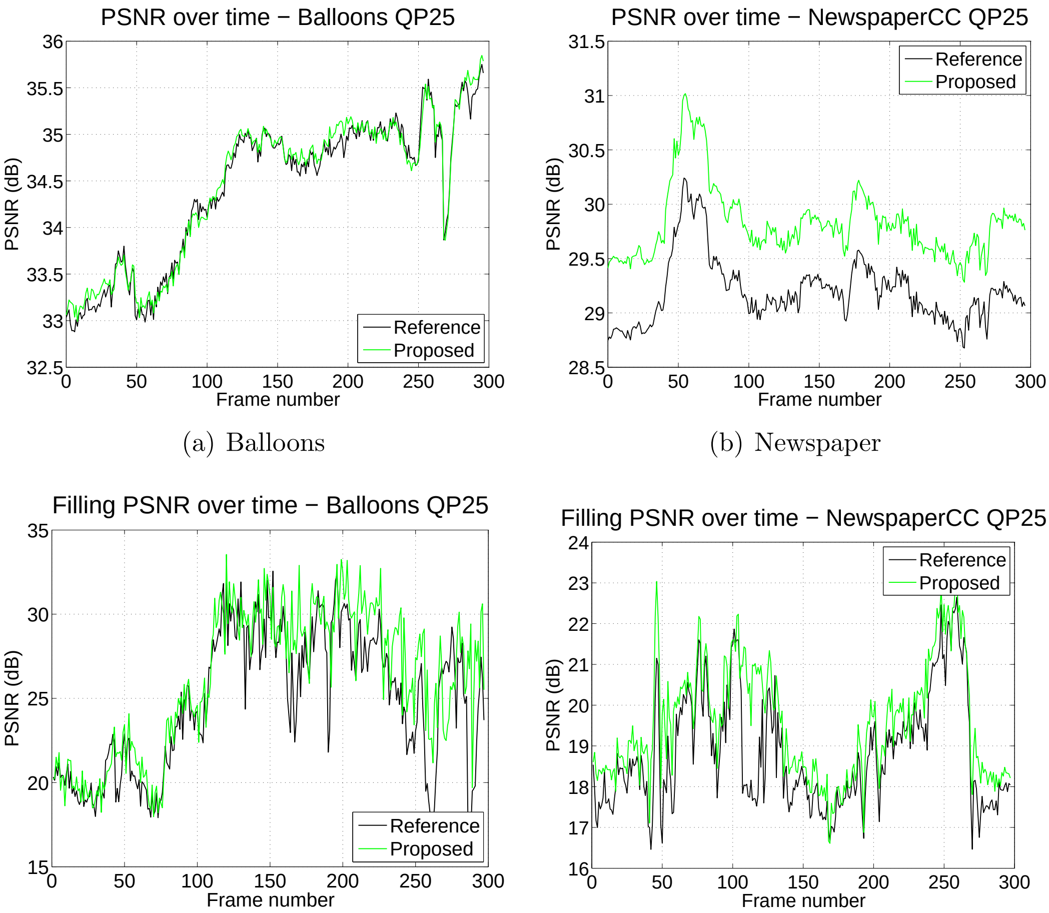
Tab. 2.1 shows the PSNR results for our method and the reference, for each tested sequence and QP. We can see that the proposed method outperforms the reference on all tested sequences, obtaining and overall average gain of 0.31dB. Using a different metric like SSIM will yield similar results for the proposed (0.9283) and reference (0.9276) methods, on average over tested sequences.

Tab. 2.2 shows the PSNR results on disoccluded areas. The same filling was used for both methods for remaining holes. These results reflect the improvement achieved only through temporal hole filling. We can see that even though only a part of the disoccluded areas is completed with temporal predicted pixels (as described in Sec. 2.4.1, see Fig. 2.4) we are able to achieve a good PSNR improvement. Note that these gains only reflect disoccluded areas, which represent a small percentage of the image, as shown in the table. The gain obtained for the entire frame comes from both temporal hole filling and proposed warping.

In Fig. 2.7 we show the PSNR comparison between our proposed method and the reference one for Balloons and Newspaper sequences. Out of the four tested sequences our method has the lowest gain on Balloons sequence and the highest gain on Newspaper sequence. For brevity reasons, we only show the result for QP25, the behavior is similar across all QPs. In Figs. 2.7(a) and 2.7(b) the PSNR is computed over the entire frame and in Figs. 2.7(c) and 2.7(d) the PSNR is computed over the disoccluded areas. We can see that our method outperforms the reference throughout the sequences on both full frame and disocludded areas. In Fig. 2.8 we show an example of the difference between the absolute errors of VSRS1D-Fast and our proposed synthesis on frame 15 of Newspaper sequence. Green to red colors indicate our method has a lower error. We can see high error pixels around the edges of object from both our method (red) and the reference (blue), however, it is easily noticeable that for most areas of the image our method offers a better prediction with a lower error.

### 2.4.4 Conclusions

In this section, we presented a temporal hole filling method based on motion derivation and a sub-pixel precision warping technique that can be applied for both DIBR warping and motion compensation. Real information on disoccluded areas is retrieved from previously synthesized past or future frames in order to reduce holes in the synthesis. The method is very robust and can be used with any motion estimation technique and a variety of schemes for the reference past and future frames. Gains of up to 0.31dB PSNR in average over the VSRS1D-Fast rendering software in 3D-HTM



(a) Balloons

(b) Newspaper

(c) Hole filling Balloons

(d) Hole filling Newspaper

Figure 2.7 – PSNR variation of the middle synthesized view over time for the reference and proposed method at QP 25 in Balloons and Newspaper sequences. 2.7(a), 2.7(b):full frame; 2.7(c), 2.7(d):disoccluded areas.

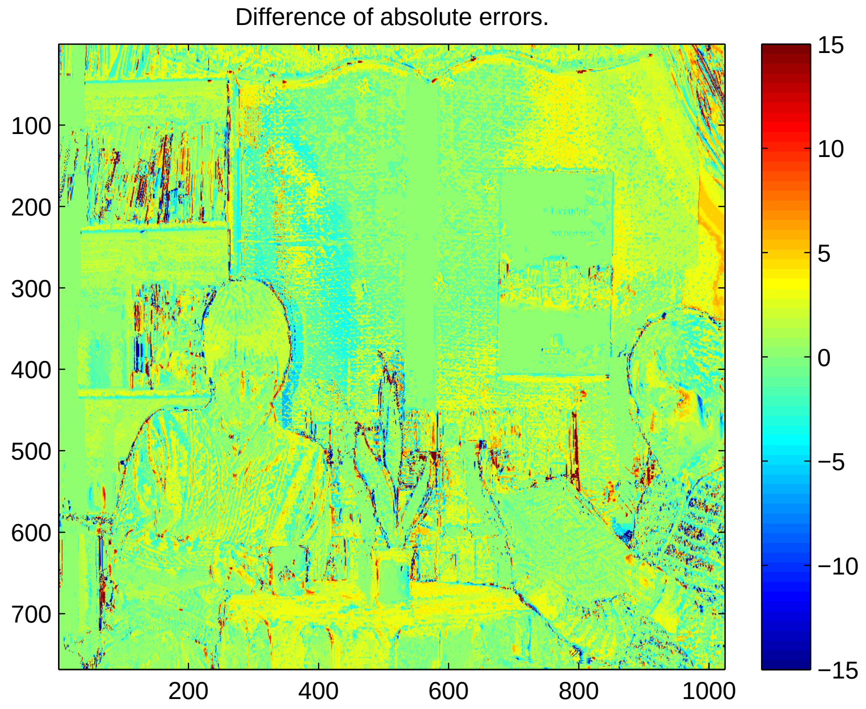


Figure 2.8 – Difference between absolute errors for frame 15 from Newspaper sequence.

for several test sequences. However, note that the size of the holes is generally small when using both a left and right reference for the synthesis. Furthermore, the THF approach is not guaranteed to fill all disocclusions as it depends on motion in the scene and motion estimation precision. As such the impact of THF is only marginal and the majority of gains are a result of the Wf technique.

## 2.5 Temporal prediction based view synthesis

In the previous section, we used temporal correlations to retrieve disoccluded areas from different time instants. While we were able to reduce the size of disoccluded areas the overall impact on the image was only marginal with most of the gains coming from the Wf technique. Our objective now is to address the entire frame and investigate the possibility of replacing or combining inter-view and temporal prediction.

As before, the first step in achieving this goal, is to obtain usable MVFs at the level of the synthesized view. Previously, we used reversed MVFs and backward MC in order to obtain different disocclusions when warping from multiple time instants. Now, the interest is to obtain a full temporal prediction of the frame. Indeed, the disocclusions can be addressed after the process using any inpainting algorithm. In order to obtain full temporal predictions of a frame we will compute forward MVFs in the prediction sens (i.e. frame $I_c^r$ is predicted from frame $I_p^r$).

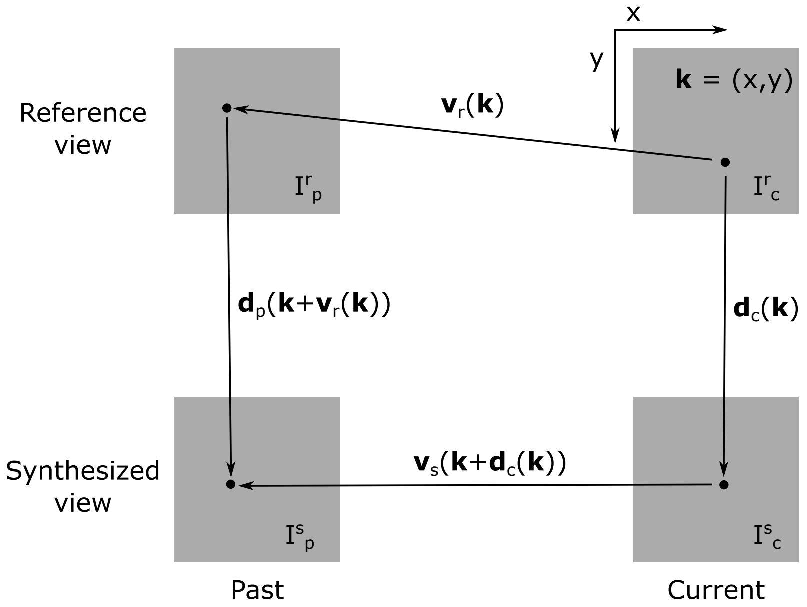

x
y
k = (x,y)
Reference view
Synthesized view
$\mathbf{v}_r(\mathbf{k})$
$I^r_p$
$I^r_c$
$\mathbf{d}_p(\mathbf{k}+\mathbf{v}_r(\mathbf{k}))$
$\mathbf{d}_c(\mathbf{k})$
$\mathbf{v}_s(\mathbf{k}+\mathbf{d}_c(\mathbf{k}))$
$I^s_p$
$I^s_c$
Past
Current

Figure 2.9 – Epipolar constraint when using forward prediction.

In Fig. 2.9, we depict the epipolar constraint when using forward MVFs in a similar fashion to Section 2.3. Note that the constraint is now expressed for a point $\mathbf{k} = (x, y)$ in frame $I_c^r$:

$$\mathbf{v}_r(\mathbf{k}) + \mathbf{d}_p(\mathbf{k} + \mathbf{v}_r(\mathbf{k})) = \mathbf{d}_c(\mathbf{k}) + \mathbf{v}_s(\mathbf{k} + \mathbf{d}_c(\mathbf{k})) \tag{2.10}$$

Using Eq. 2.10 we can derive the dense MVF in the synthesized view as:

$$\mathbf{v}_s(\mathbf{k} + \mathbf{d}_c(\mathbf{k})) = \mathbf{v}_r(\mathbf{k}) + \mathbf{d}_p(\mathbf{k} + \mathbf{v}_r(\mathbf{k})) - \mathbf{d}_c(\mathbf{k}) \tag{2.11}$$

Note that when using this formulation of the epipolar constraint we obtain the MVF for all positions $m \in \mathcal{M}$ where $\mathcal{M} = \{\mathbf{k} + \mathbf{d}_c(\mathbf{k}) \mid \mathbf{k} \in I_c^r\}$. As a consequence, our MVF in the synthesized view ($\mathbf{v}_s$) will have holes matching the disoccluded areas in a DIBR warping. Therefore, using this MVF to MC will result in the same holes as the DIBR synthesis.

### 2.5.1 Temporal and view prediction

As discussed in Section 2.4, we can use a past and a future reference frame for the MVF computation in the reference views. Thus, we can obtain four temporal predictions for each pixel, from a past or future reference frame using the left or the right available views.

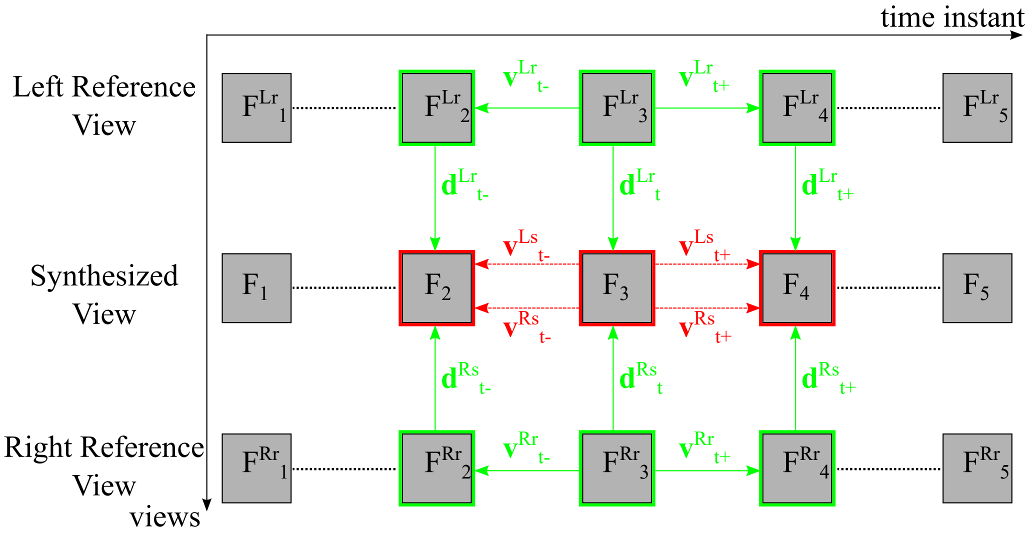


Figure 2.10 – Scheme for two reference views using past and future time instants $(t_-, t_+)$. Green: MVF warping step; red: the MC step.

In Fig. 2.10, we show the general scheme of the proposed method. Considering a left $(Lr)$ and a right $(Rr)$ reference view, with their associated depth maps, we aim at synthesizing a middle view. With green we represent the MVF warping and the required inputs: the MVFs in the reference views for a past $(t_-)$ and future $(t_+)$ time instant $(\mathbf{v}^{Lr}_{t_-}, \mathbf{v}^{Lr}_{t_+}, \mathbf{v}^{Rr}_{t_-}, \mathbf{v}^{Rr}_{t_+})$ and the six DVFs $(\mathbf{d}^{Lr}_{t_-}, \mathbf{d}^{Lr}_{t}, \mathbf{d}^{Lr}_{t_+}, \mathbf{d}^{Rr}_{t_-}, \mathbf{d}^{Rr}_{t}, \mathbf{d}^{Rr}_{t_+})$.

With red in Fig. 2.10, we show the MC step in which four predictions of the current frame are obtained using the four MVFs. The red and green scheme can then be iterated through all the frames of the synthesized view. Note that the temporal distance between the prediction and reference in the ME process is constant and set to 1 in Fig. 2.10). As each frame has different temporal references, the algorithm requires an initial DIBR synthesis.

The final steps in order to obtain the synthesized image are the blending of the temporal predictions and the inpainting of remaining holes. Note that during the blending step, the two inter-view predictions (from left and right) can also be taken into account. This aspect is discussed in the following section.

### 2.5.2 View synthesis

As discussed in Section 2.5.1 we obtain four temporal and two inter-view predictions of a frame. A first solution to obtain the synthesized frame is to combine these predictions in a similar manner as VSRS-1DFast, by computing the average or median of the values for each pixel. This works well for sequences that have low intensity motion and provide good results as can be seen in Section 2.5.3. However, when dealing with high intensity motion, the temporal predictions can contain artifacts due to the ME failures. In this situation, we should use only the inter-view predictions which are invariant with respect to the motion intensity. The challenge here is to determine when to use only inter-view prediction. Because there is no prior information about the texture, the accuracy of the six available predictions, four temporal and two inter-view, cannot be determined. However, we can reasonably assume that the ME artifacts may vary at different time instants, which implies that four temporal predictions with matching values (very close values) probably indicate a good result of the ME. The same reasoning can be applied for DIBR: two matching values predicted with DIBR indicate most likely a good prediction. Thus, we are interested in using the inter-view predictions when they have similar values and the temporal predictions are relatively different from each other and inter-view.

In order to better formulate this problem, let us consider four temporal ($\widehat{i}_1^t$, $\widehat{i}_2^t$, $\widehat{i}_3^t$, $\widehat{i}_4^t$) predictions, two inter-view ($\widehat{i}_1^{iv} < \widehat{i}_2^{iv}$) predictions and the vectors $\mathbf{p}_t = [\widehat{i}_1^t, \widehat{i}_2^t, \widehat{i}_3^t, \widehat{i}_4^t]$ and $\mathbf{p}_{iv} = [\widehat{i}_1^{iv}, \widehat{i}_2^{iv}]$. When the temporal predictions are very close to the inter-view predictions or contained in the interval $[\widehat{i}_1^{iv}, \widehat{i}_2^{iv}]$ there are no reliable assumptions that can be made about the accuracy of each prediction type. Therefore, we should use the average or median of the six predictions ($[\mathbf{p}_t, \mathbf{p}_{iv}]$). However, when some or all of the temporal predictions are outside of this interval and there is a relatively high difference between the two prediction types, we should use only the inter-view predictions. Based on this situations we can formulate the selection process as follows:

$$\widehat{i} = \begin{cases} \text{mean}(\mathbf{p}_{iv}) & \text{if } \text{mean}(\mid [\mathbf{p_t}, \mathbf{p_{iv}}] - \text{mean}([\mathbf{p_t}, \mathbf{p_{iv}}]) \mid) > \\ & \qquad \text{mean}(\mid \mathbf{p_{iv}} - \text{mean}(\mathbf{p_{iv}}) \mid) \\ \text{mean}([\mathbf{p_t}, \mathbf{p_{iv}}]) & \text{otherwise} \end{cases} \tag{2.12}$$

For empirical reasons we decided to use the average over the median in the blending process as it provides slightly better results. The selection method described above is designed to use only inter-view prediction when the temporal one is unreliable. Note that the reference frames used in MC are also synthesized, therefore we never use only the average of temporal predictions.

### 2.5.3 Experimental results

Similarly to Section 2.4.3 we test this approach using the test model designed for 3D-HEVC (3D-HTM 7.0). Details can be found in the Common Test Conditions (CTCs) for conducting experiments with the reference software of 3D-HEVC [RMV13]. The video sequences used in our tests are: Balloons, Kendo, NewspaperCC and PoznanHall2. The first three sequences have a resolution of 1024 × 768 with 30 frames per second while PoznanHall2 has a resolution of 1920 × 1088 with 25 frames per second, additional details can be found in [CFP11]. We use the full sequences for our tests (300 frames for the first three sequences and 200 for the later). The left and right reference views used in the synthesis are 1&5 for the first two sequences, 2&6 and 5&7 for the NewspaperCC and PoznanHall2. The synthesized views are 3, 3, 4 and 6, respectively. We test the synthesis using different quality encoding for the reference texture and depth sequences. Each sequence is encoded using four QPs: 25 30 35 40, for the texture and corresponding QPs for the depth maps: 34 39 42 45, as indicated by the CTCs. The reference synthesis we compare against is performed with VSRS-1DFast [ZTWY13].

For comparison purpose, we also include the results of Wf approach described in Section 2.4.2. For fairness of comparison, all methods use the same hole filling technique as that of VSRS-1DFast. Two different blending options are used, first an averaging of the predictions (avg) and second the blending described in Section 2.5.2. As shown in Section 2.5.1, each frame in the synthesized view is motion compensated from a past and future reference frame. While the past reference frame is available, the future reference frame needs to be synthesized only from inter-view predictions before the MC step. The MVFs are computed using the optical flow implementation in [Liu] and the MC is performed with sub-pixel precision. The parameters used and additional details about the optical flow method can be found in [Liu09].

For each sequence and each QP, we synthesize the intermediate views using the reference software (VSRS-1DFast), Wf, predictions average blending (P+Bavg) and the prediction adaptive blending (P+Badapt). We evaluate the PSNR of each synthesis using the original uncompressed sequences.

Tab. 2.3 shows the average PSNR for the reference methods and ours. We can see that our method provides a better synthesis. However, on Kendo sequence the averaging of predictions does not provide good results due to high intensity motion. This issue is resolved by the adaptive blending and the quality of the synthesis is

| **Sequence** | VSRS-1DFast<br>PSNR (dB) | | | | Wf<br>PSNR (dB) | | | | P+Bavg<br>PSNR (dB) | | | | P+Badapt<br>PSNR (dB) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| QPs | 25 | 30 | 35 | 40 | 25 | 30 | 35 | 40 | 25 | 30 | 35 | 40 | 25 | 30 | 35 | 40 |
| Balloons | 34.37 | 34.07 | 33.43 | 32.41 | 34.39 | 34.14 | 33.52 | 32.51 | 34.72 | 34.44 | 33.78 | 32.73 | 34.74 | 34.45 | 33.8 | 32.72 |
| Kendo | 34.98 | 34.51 | 33.77 | 32.75 | 35.37 | 34.9 | 34.15 | 33.08 | 34.86 | 34.42 | 33.75 | 32.79 | 35.37 | 34.87 | 34.13 | 33.06 |
| NewspaperCC | 29.2 | 29.05 | 28.78 | 28.31 | 29.81 | 29.69 | 29.39 | 28.83 | 29.91 | 29.8 | 29.5 | 28.95 | 29.85 | 29.74 | 29.44 | 28.9 |
| PoznanHall2 | 36.24 | 35.87 | 35.36 | 34.55 | 36.35 | 36.02 | 35.51 | 34.77 | 36.44 | 36.11 | 35.7 | 34.88 | 36.49 | 36.2 | 35.7 | 34.86 |
| Average | 33.70 | 33.37 | 32.83 | 32 | 33.98 | 33.69 | 33.14 | 32.3 | 33.98 | 33.69 | 33.18 | 32.34 | 34.11 | 33.82 | 33.27 | 32.38 |
| ΔPSNR | - | - | - | - | 0.28 | 0.32 | 0.31 | 0.3 | 0.28 | 0.32 | 0.35 | 0.34 | 0.41 | 0.45 | 0.44 | 0.38 |

Table 2.3 – Average PSNR for all methods and sequences at each QP.

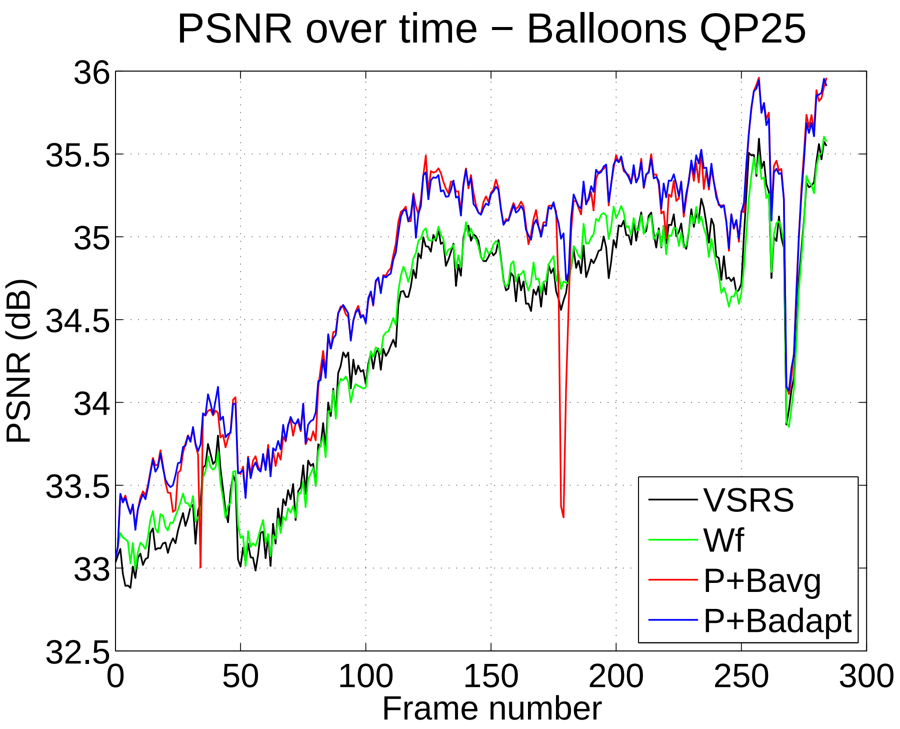
highly increased compared to the averaging blend. The last rows show the average and ΔPSNR values of Wf, P+Bavg and P+Badapt over the VSRS-1DFast reference. We can see that all methods provide a gain over VSRS-1DFast (0.42dB in average, with P+Badapt), while our proposed methods manages to outperform Wf (0.1dB in average). It can be noticed that the gain depends on the sequence.

In Figs. 2.11 and 2.12, we show the PSNR variation over time for the four test sequences. For brevity reasons, we only show the QP 25 results as the behavior is similar across QPs. Black, green, red and blue colors indicate the methods VSRS-1DFast, Wf, P+Bavg and P+Badapt. We can see that the proposed methods outperform VSRS-1DFast and Wf, we obtain gains of up to 0.65dB and 0.37dB on NewspaperCC and Balloons sequences respectively. Fig. 2.13 shows some details of the synthesis with the tested methods. From left to right we show the original uncompressed, the VSRS-1DFast, Wf and P+Badapt. Red squares mark areas containing distortions. In 2.13(a) we can see distortions around the contours of the balloons and in 2.13(b) around the edge of the head. It is noticeable that in the images on the right these artifacts are diminished.

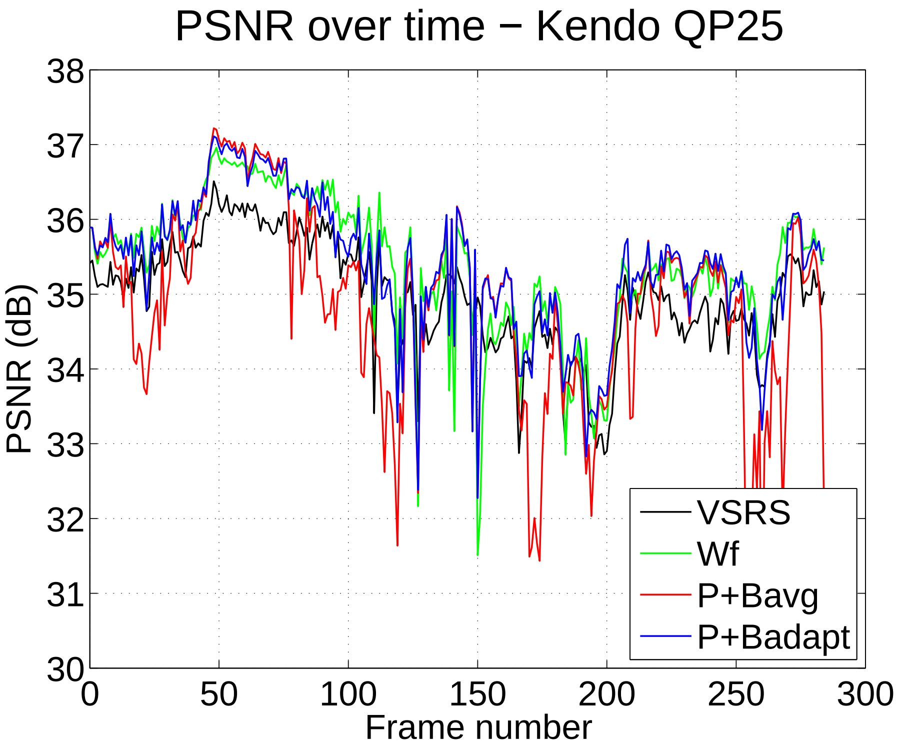
As can be seen in our experiments, the P+Bavg method provides a better quality synthesis when compared to VSRS-1DFast and Wf on most test sequences. However, this method uses temporal correlations in a video sequence and is dependent on the quality of the ME technique used. While it is able to obtain a very good gain on Balloons sequence it falls behind on Kendo sequence due to ME failure caused by high intensity motion. This problem is corrected using the adaptive blending presented in Section 2.5.2. In Fig. 2.11(a) and 2.11(b) we can see some drops in quality on some frames with P+Bavg which are corrected by P+Badapt.

### 2.5.4 Conclusions

The approach presented in this section uses temporal predictions of a frame to improve the inter-view prediction. The method outperforms both VSRS-1DFast and

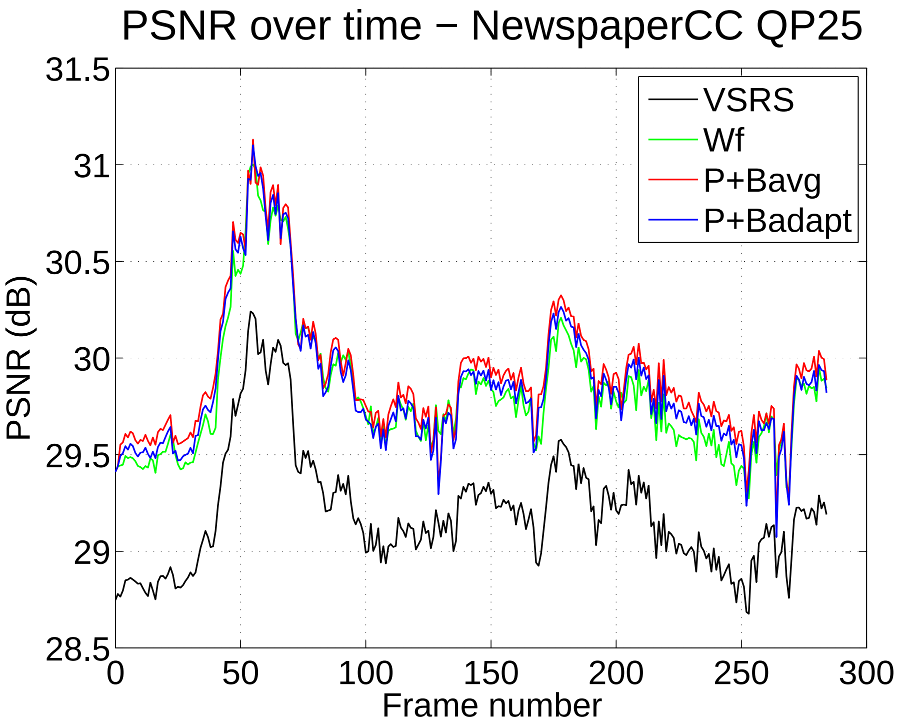


(a) Balloons

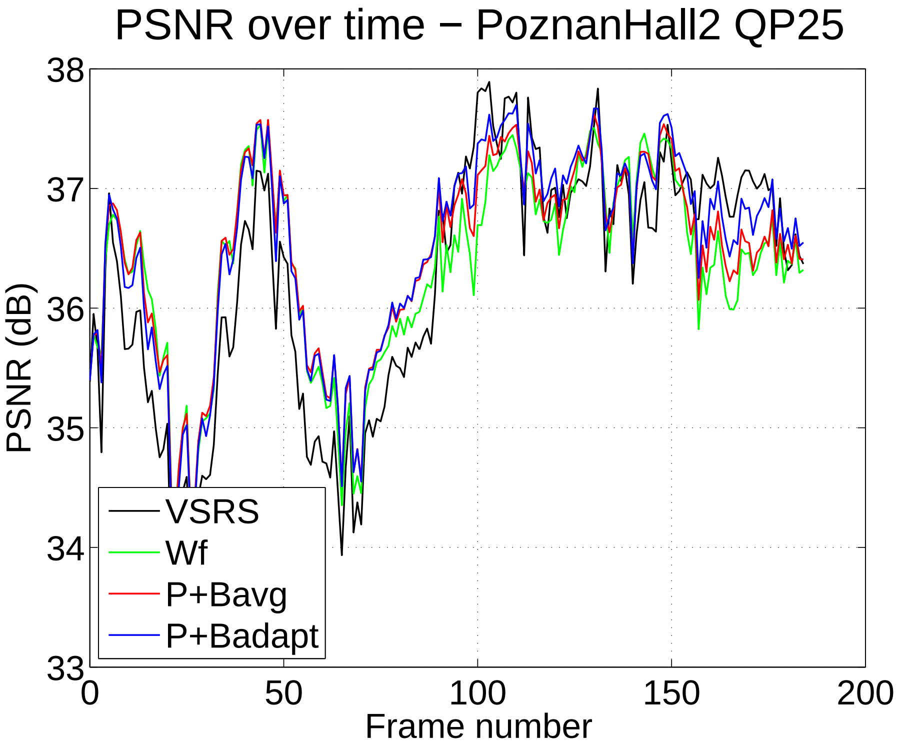


(b) Kendo

Figure 2.11 – PSNR variation of the synthesized view over time for the reference and proposed methods at QP 25 in Balloons 2.11(a), Kendo 2.11(b), NewspaperCC 2.12(a) and Poznan Hall2 2.12(b) sequences.

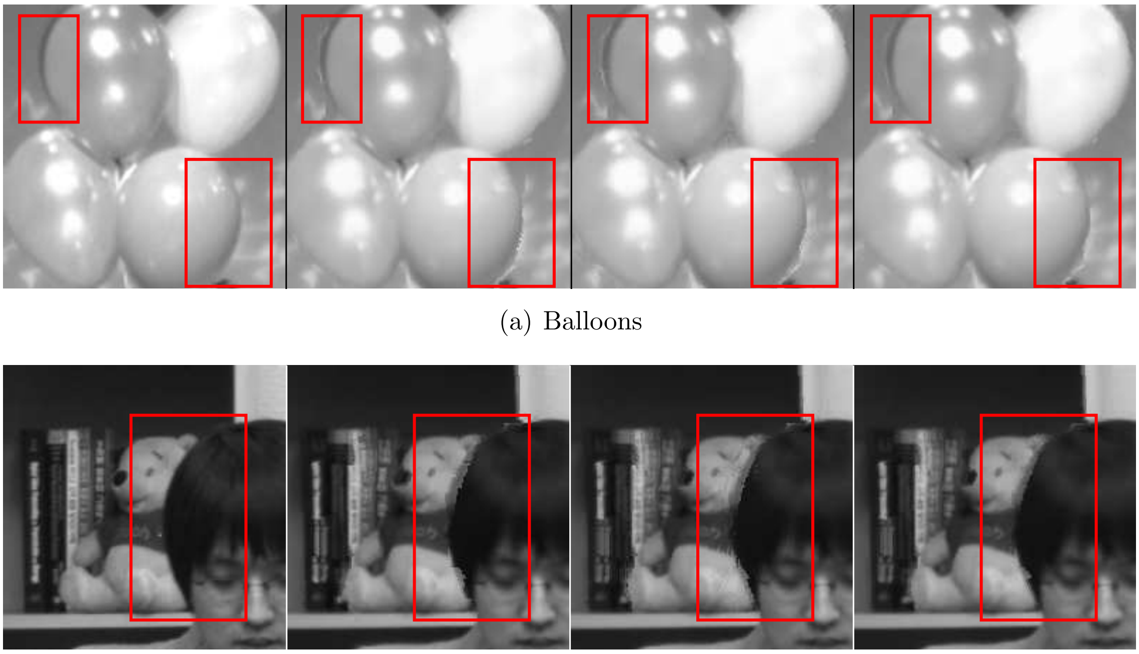


(a) NewspaperCC



(b) Poznan Hall2

Figure 2.12 – PSNR variation of the synthesized view over time for the reference and proposed methods at QP 25 in NewspaperCC 2.12(a) and PoznanHall2 2.12(b) sequences.



(a) Balloons

(b) NewspaperCC

Figure 2.13 – Details in Balloons 2.13(a) and NewspaperCC 2.13(b) sequences on frame 38. From left to right: original uncompressed, VSRS-1DFast, Wf, P+Badapt. Red squares show distortions in the synthesis.

Wf. However, the reference frames used for temporal prediction are also synthesized thus their quality is limited.

## 2.6 3D-HEVC view synthesis exploiting temporal prediction

In Sections 2.4 and 2.5 we showed how temporal correlations can be used to improve view synthesis. However, since the reference frames used for temporal prediction are also synthesized, gains are somewhat limited. Considering that some of the views that are reconstructed via synthesis at the decoder side are actually available at the encoder side, we could maximize the effectiveness of temporal prediction by sending additional information about the synthesized view. Mainly, we send one additional intra frame per GOP for the synthesized view. We will refer to this synthesis method as View Synthesis exploiting Temporal Prediction (VSTP).

Similarly to THF and P+Badapt methods, we need to find a suitable way of obtaining the MVFs in the synthesized view. Considering that the temporal reference in the synthesis is always the first frame of the GOP, we can no longer use a constant temporal prediction distance as we did for P+Badapt. In order to also take advantage of THF we want to use reverse MVFs and backward motion compensations. Thus, the epipolar constraint formulation in Section 2.3, Eq. 2.1 is used to obtain the MVFs in the synthesized view. As discussed in Section 2.4.1 up to four MVFs can be obtained for each frame of the synthesized view using Eq. 2.3 in the four quadrants of Figure 2.5.

There will be holes in $\mathbf{v}_s$ that coincide with disocclusions created when warping $I_p^r$ with the $\mathbf{d}_p$ DVF. If two or more positions in $I_p^r$, $\mathbf{k}_1$ and $\mathbf{k}_2$ for instance, are warped to the same position $\mathbf{k}_3$ in $I_p^s$ (occlusion), the vector $\mathbf{v}_s(\mathbf{k}_3)$ retained is the one which corresponds to the pixel with the highest depth value, as shown in Equation 2.13, the motion vectors for occluded points of the scene are thus ignored.

$$\mathbf{v}_s(\mathbf{k}_3) = \begin{cases} \mathbf{v}_r(\mathbf{k}_1) + \mathbf{d}_t(\mathbf{k}_1 + \mathbf{v}_r(\mathbf{k}_1)) - \mathbf{d}_p(\mathbf{k}_1) \\ \qquad \text{if } Z_p^r(\mathbf{k}_1) > Z_p^r(\mathbf{k}_2) \\ \mathbf{v}_r(\mathbf{k}_2) + \mathbf{d}_t(\mathbf{k}_2 + \mathbf{v}_r(\mathbf{k}_2)) - \mathbf{d}_p(\mathbf{k}_2) \\ \qquad \text{otherwise} \end{cases} \tag{2.13}$$

Using the motion vector field $\mathbf{v}_s$ and $I_p^s$, a prediction of $I_c^s$ can be made, although it will contain holes due to disoccluded areas in $\mathbf{v}_s$ and also areas disoccluded due to the motion of foreground objects (a consequence of backward motion compensation). The four predictions are then merged into a single one $\widetilde{I}_c^s$, where the value of each pixel equals the average of the non-disoccluded pixel values in the four predictions as shown in the following equation. When all four predictions contain the same disocclusion, the pixel value is computed by inpainting. While the four predictions contain disocclusions, the majority of these holes are not the same as they depend on both motion and disparity at different time instants and views.

Unlike P+Badapt approach where the forward MC reference was a synthesized frame we are now relying on a decoded frame with backward MC. In general, frames encoded from original views provide better quality than synthesis. Preliminary tests showed that a blend of temporal predictions in this case provides better results than DIBR. Therefore, we create a full temporal prediction of the frame without averaging with DIBR as we did in Section 2.5.2:

$$\widetilde{I}_c^s(\mathbf{k}) = \begin{cases} \dfrac{\sum_{i=0}^{A(\mathbf{k})} \mathcal{P}^{(i)}\left(I_c^s(\mathbf{k})\right)}{A(\mathbf{k})} & \text{if } A(\mathbf{k}) \neq 0 \\ \text{inpainted} & \text{if } A(\mathbf{k}) = 0 \end{cases} \tag{2.14}$$

where $A(\mathbf{k})$ is the number of existing predictions for position $\mathbf{k}$. Disocclusions ($A(\mathbf{k}) = 0$) are filled using the same inpainting method used in VSRS-1DFast, which is a simple line-wise interpolation.

Figure 2.14 illustrates the steps of VSTP algorithm. In order to generate a temporal prediction, the algorithm inputs two frames of the reference view at two time instants, i.e., a current and a future or past time instants, denoted by $I^r_{c,L}$ and $I^r_{p,L}$ respectively in the figure, and computes a dense MVF between the two ($\mathbf{v}_{r,p,L}$). The dense MVF is then warped at the level of the synthesized view using the corresponding disparity maps ($\mathbf{d}_{c,L}$ and $\mathbf{d}_{p,L}$). We also retain a disparity map corresponding with the new MVF ($\mathbf{d}'$). Thus, each pixel has an associated MV and DV. The next step is the backward MC in which we use a key frame ($I^s_p$) as reference in order to obtain a first temporal prediction, in case of overlapping values we use $\mathbf{d}'$ to select the foreground pixel. $\widehat{I^s_{p,R}}$, $\widehat{I^s_{f,L}}$, $\widehat{I^s_{f,R}}$ are obtained using the same steps in the right reference view at the same time instant and at a future time instant in the left and right reference views respectively, as shown in Figure 2.5. The final synthesis is obtained by performing a simple merge between the four temporal predictions or an inter-view/temporal fusion as described in Section 2.6.2. The inter-view prediction is denoted by $\widehat{I^i}$ in Figure 2.14.

### 2.6.1 Prediction schemes in a GOP

The synthesized view is rendered GOP-wise in our algorithm. The GOP structure is the one used to code the left and right reference views. In addition to the reference views (as required by VSRS-1DFast) we send a first frame per GOP of the synthesized view (at the encoder side we require this view, it can be either original or synthesized from uncompressed adjacent views if not available) in the bitstream. These frames, referred to in the rest of this work as key frames, are efficiently coded using 3D-HEVC with the left view serving as inter-view reference (the base view). The rest of the frames are synthesized using our method with one of the temporal prediction schemes described below. For the first frame actually synthesized in a GOP, the key frame of the current GOP and the one of the future GOP respectively are the past and future reference frames, $I^s_p$ and $I^s_f$ respectively.

Figure 2.15 shows the difference between the two temporal prediction schemes. The “Direct” scheme uses the key frame of the current GOP and the one of the next GOP as past and future reference frames for all remaining frames to synthesize in the GOP. This results in an asymmetric prediction, with two different temporal distances between each of the two key frames and the current frame. The temporal distance

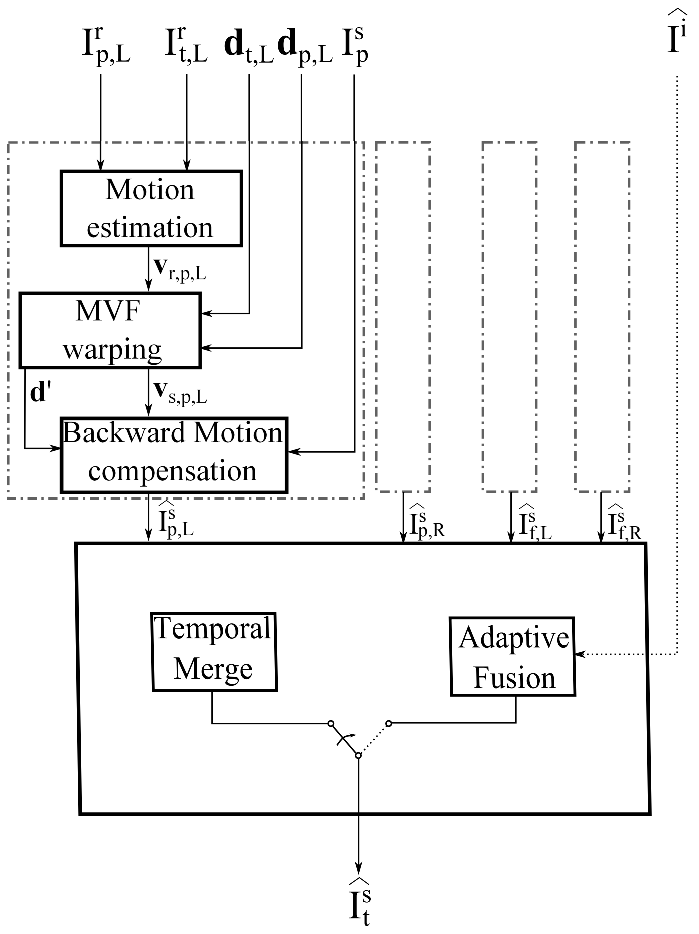
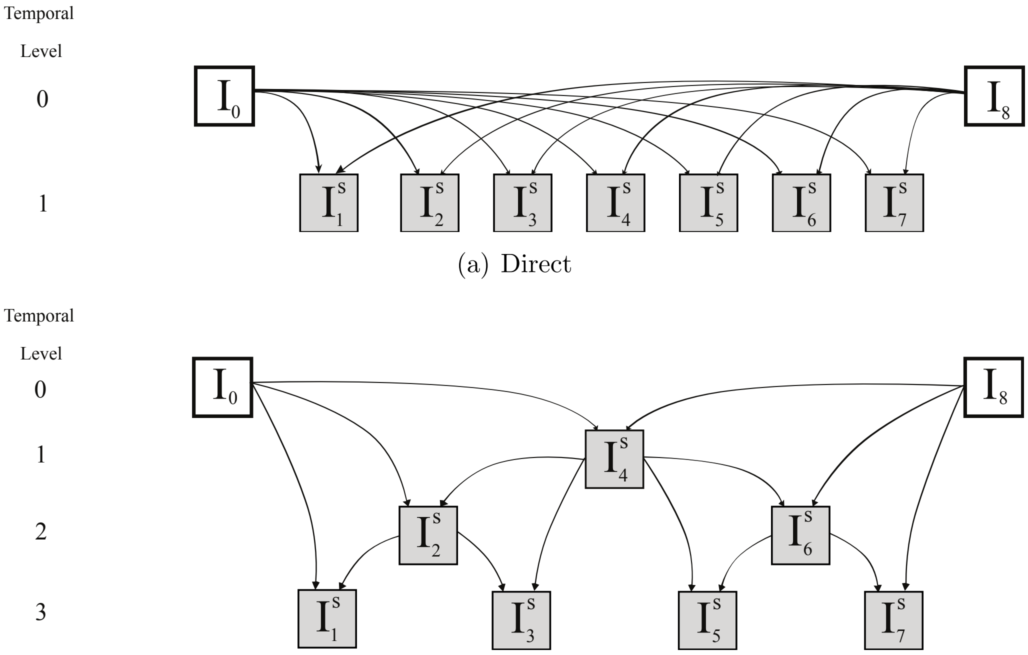


Figure 2.14 – Flow diagram for View Synthesis exploiting Temporal Prediction (VSTP). The dotted dash line is the temporal prediction block, which is applied four times, i.e. past and future time instants ($p$ and $f$) in the left and right ($L$ and $R$) reference ($r$) views.



(a) Direct

(b) Hierarchical

Figure 2.15 – Temporal prediction schemes inside a GOP of the synthesized view.

can be as high as the GOP size minus one, and an optical flow computation with such large temporal distances can give imprecise MVFs thus making the "Direct" scheme inefficient. An alternative scheme, called the "Hierarchical" scheme, can be used, in which temporal layers are used to perform symmetric predictions (with equal temporal distances). In each layer, the past and future references for the current frame are either the key frames or already synthesized frames in lower layers. The maximal temporal distance in this scheme equals half of the GOP size.

### 2.6.2 Adaptive Fusion

In the proposed method the synthesized frame is obtained by merging our four temporal predictions as described in Equation (2.14). When dealing with fast moving objects, the optical flow computation between frames with high temporal distance may give imprecise MVFs which lead to an inconsistent positioning of the objects in the four temporal predictions. In this case, a simple average-based merging would result in a bad representation of objects with high motion intensity. In the following, we refer to the traditional disparity based synthesis used in VSRS-1DFast as the inter-view prediction. We introduce a different merging algorithm called "Adaptive Fusion" which uses the inter-view prediction and our temporal prediction alternatively for different parts of the image. The idea of this method is to generate a binary fusion map in which we mark the bad pixels from the temporal prediction, to be replaced by the inter-view prediction. The first step of this algorithm is to estimate which areas will select the inter-view prediction and which ones will select the temporal prediction. The next step is the actual fusion, where each pixel value is computed as an average between either the temporal or inter-view predictions , depending on the previously computed binary map.

In order to describe our selection process for a pixel, let us consider: $\widehat{i}^t_{pL}$, $\widehat{i}^t_{fL}$, $\widehat{i}^t_{pR}$, $\widehat{i}^t_{fR}$ four temporal predictions of a pixel at position $\mathbf{k}$ and $\widehat{i}^i$ the blend between the left and right inter-view predictions obtained from VSRS-1DFast. It is safe to assume that good temporal predictions of a pixel are similar, i.e., the values are close to each other (have a low spread). On the contrary, imprecise MVFs might lead to dissimilar values that span over a large range (have a wide spread) and in this case inter-view prediction should be used. Note that in some cases $\widehat{i}^i$ is worse than the temporal prediction even if we have a wide spread. The challenge is to remove artifacts in the temporal prediction without introducing new ones from the inter-view prediction. By comparing the value of $\widehat{i}^i$ to our four temporal predictions we can identify four cases. In the following, the maximum and minimum value of the temporal predictions are denoted by $\widehat{i}^t_{max}$ and $\widehat{i}^t_{min}$ respectively :

**Case 1:** Wide spread and $\widehat{i}^i \in [\widehat{i}^t_{min}, \widehat{i}^t_{max}]$

**Case 2:** Wide spread and $\widehat{i}^i \notin [\widehat{i}^t_{min}, \widehat{i}^t_{max}]$

**Case 3:** Low spread and $\widehat{i}^i \in [\widehat{i}^t_{min}, \widehat{i}^t_{max}]$

**Case 4:** Low spread and $\widehat{i}^i \notin [\widehat{i}^t_{min}, \widehat{i}^t_{max}]$

We consider **Case 1** and **Case 4** as typical situations in which we should select inter-view and temporal predictions respectively. Indeed, in **Case 1**, wide spread means there is a bad match between the four temporal predicted values, which indicate an imprecise optical flow computation. An inter-view prediction inside this range is probably the best value. **Case 4** indicates a good temporal prediction and we should use the average of the four points. In **Case 2**, the inter-view predicted value is either good or very bad depending on how far away it is from $\widehat{i}^t_{min}$ or $\widehat{i}^t_{max}$. In **Case 3**, the two prediction values are close and we prioritize the temporal one. When dealing with disocclusions, the number of available temporal or inter-view predictions for a pixel can vary, i.e., a certain position $(x, y)$ can be a disocclusion in one or more temporal or inter-view predictions. In situations when only one type of prediction is available we select it, and if we have no prediction at all, we mark the pixel to be filled later.

Considering the vectors $\mathbf{p_t} = [\widehat{i}^t_{pL}, \widehat{i}^t_{fL}, \widehat{i}^t_{pR}, \widehat{i}^t_{fR}]$ and $\mathbf{p_{t\&i}} = [\widehat{i}^t_{pL}, \widehat{i}^t_{fL}, \widehat{i}^t_{pR}, \widehat{i}^t_{fR}, \widehat{i}^i]$, the selection between inter-view and temporal prediction for a pixel is done as follows:

$$\widehat{i} = \begin{cases} \widehat{i}^t & \text{if mean}(\mid \mathbf{p_t} - \text{mean}(\mathbf{p_t}) \mid) \\ & -\text{mean}(\mid \mathbf{p_{t\&i}} - \text{mean}(\mathbf{p_{t\&i}}) \mid) < \alpha \\ \widehat{i}^i & \text{if mean}(\mid \mathbf{p_t} - \text{mean}(\mathbf{p_t}) \mid) \\ & -\text{mean}(\mid \mathbf{p_{t\&i}} - \text{mean}(\mathbf{p_{t\&i}}) \mid) > \alpha \end{cases} \tag{2.15}$$

where $\widehat{i}^t = \text{mean}(\mathbf{p_t})$ and $\alpha$ is a threshold used to control the selection process (by increasing $\alpha$ we favor the temporal prediction). Adding an outlying value to the $\mathbf{p_t}$ vector will increase its mean absolute deviation, on the contrary an inlying value will maintain a similar mean absolute deviation. In our model, we select temporal prediction when $\widehat{i}^i$ is an outlier, this corresponds to **Case 4**. For **Case 2** and **Case 3**, we favor the temporal prediction and for **Case 1** we favor the inter-view prediction. The value for $\alpha$ used in this work was empirically found to be optimal at 0.5.

From this process, we deduce a binary selection map:

$$B(\mathbf{k}) = \begin{cases} 0 & \text{if } \widehat{i} = \widehat{i}^t \\ 1 & \text{if } \widehat{i} = \widehat{i}^i \end{cases} \tag{2.16}$$

which indicates the selected prediction type for each pixel.

### 2.6.3 Discussion on the method

In dense camera rig systems, a high number of views are available at the encoder side. Typically, only a subset is coded and sent in the bitstream, the rest being synthesized at the receiver side [ZTWY13]. Our prediction method uses the synthesized view at the encoder side, since one frame per GOP of that view is transmitted in the bitstream. Indeed, synthesizing the intermediate views instead of sending them is a more efficient alternative as show in [Mor14b]. Our method can be seen as in between these two scenarios: we only send some information on the synthesized views, which we exploit to improve the synthesis. Consequently, in this work, we do not only propose a rendering method, but also a change in the design of the transmission stage. Note that we could have proposed a method where the key frames in the synthesized view are rendered with the left and right reference views using VSRS for instance, but then the rendering artifacts created in these key frames would be propagated to the rest of the frames in the MC stage.

In comparison to other pixel-based methods such as [LLZ+14] which improve the encoding of the depth map using dense MVFs computed on texture, our method warps the dense MVFs at the level of the intermediate view and uses them to MC texture images as shown in this section. Boundary artifacts reduction methods such as [ZZC+11] can be used in parallel with VSTP. Since, our final synthesis is a blend between DIBR rendering and the temporal predictions, reducing the artifacts in the DIBR synthesis will increase the quality of our method. Also, a better texture-depth alignment can benefit the warping of the dense MVFs. However, our method also gives the possibility to adjust the QP of the key frames which will in turn affect all frames inside a GOP or modify the frequency of the key frames which will reduce or increase the temporal distance of the prediction resulting in a higher quality rendering and a variation of the transmission rate.

Furthermore, our method provides new possibilities to control the rate and distortion in comparison to VSRS-1DFast: modifying the QP of key frames or adjusting their frequency. The bit allocation optimization scheme for DIBR multiview coding presented in [CVO11] can be employed with our method as-well. However, a study towards the integration of the additional rate and distortion control options provided by VSTP within such schemes should be performed. For simplicity reasons in our experiments we will use the recommended depth and texture QPs for 3D-HEVC testing, as discussed in section 2.6.4.1.

### 2.6.4 Experimental results

#### 2.6.4.1 Experimental setting

The algorithm takes as input two coded left and right views with their associated depth videos and camera parameters, and one frame per GOP of the intermediate view, and outputs the whole intermediate view after synthesizing the rest of the frames. The synthesis results are compared against the original intermediate sequences to measure the PSNR. We thus consider a five-view scenario in these experiments in which we code two views (left and right) and key frames from 1/2 view and synthesize three intermediary views at 1/4, 1/2 and 3/4 positions between the two base views. We assume that one of the three intermediary views is available at the encoder side(1/2). The coding configuration described in the Common Test Conditions (CTCs) defined by JCT-3V for conducting experiments with the reference software of 3D-HEVC [RMV13] is used for coding the left and right views. The recommended texture and depth QPs are 25, 30, 35, 40 and 34, 39, 42, 45 respectively. The optical flow algorithm used in our method can be downloaded from [Liu], the configuration parameters are reported in Table 2.4 and more details can be found in [Liu09].

| **Parameter** | **Description** | **Value** |
|---|---|---|
| Alpha | Regularization weight | 0.012 |
| Ratio | Downsampling ratio | 0.4 |
| MinWith | Width of the coarsest level | 20 |
| nOuterFPIterations | Number of outer fixed point iterations | 7 |
| nInnerFPIterations | Number of inner fixed point iterations | 1 |
| nSORIterations | Number of Successive Over Relaxation iterations | 30 |

Table 2.4 – Optical flow parameters

The method is tested on four sequences of the test set in the CTCs: Balloons, Kendo, Newspaper and PoznanHall2. Each sequence is composed of three real views and we also consider two virtual views. The CTCs indicate to use the middle view as base view, and the left and right views as dependent views. However, here we want the left and right views to be decodable without the middle view because only the first frame in each GOP of that view will be sent in the bitstream. We thus set the left view as base view, and the others as dependent views. Also, we code roughly 10 seconds of video of each sequences. Note that the number of frames is lower in PoznanHall2 because its frame rate is lower as well (cf. Table 2.5).

| **Class** | **Sequence** | **Frames per second** | **Number of frames** |
|---|---|---|---|
| class A (1920 × 1088) | PoznanHall2 | 25 | 200 |
| class C (1024 × 768) | Balloons | 30 | 300 |
| | Kendo | 30 | 300 |
| | Newspaper | 30 | 300 |

Table 2.5 – Sequences used in our experiments

We compare our synthesis method to the reference VSRS-1DFast in 3D-HEVC test model, HTM. We evaluate the performance of the reference and the proposed methods using the Bjontegaard delta-PSNR (BD-PSNR) [Bjo01] metric on the synthesized views. The PSNR is evaluated against the original intermediate views.

Evaluating our synthesis against frames synthesized from uncompressed views, as indicated by the CTCs, would penalize the lack of artifacts that arise from disparity warping, which are present in both compressed and uncompressed VSRS synthesis. The rate in the reference method is the sum of the rates needed to code the left and right views with their associated depth videos. The same rate is considered in the proposed method, to which is added the rate needed to code the first frame in each GOP of the intermediate view. We use the BD-PSNR metric to measure the improvement (see Figure 2.17).

#### 2.6.4.2 Synthesis results

Table 2.6 gives the BD-PSNR values obtained with the two prediction schemes with simple merging ("Direct" and "Hierarchical") and "Adaptive Fusion" applied in the "Hierarchical" scheme ("HierarchicalAF") when considering only the PSNR of the 1/2 intermediary view synthesized with VSTP. In Table 2.7 we show the BD-PSNR for the 3 intermediary views. Here, the PSNR is computed as the average between the 3 (1/4, 3/4 synthesized with VSRS-1DFast and 1/2 with VSTP). A positive value in this table indicates a gain. On average, our method brings 0.53dB, 0.59dB and 0.87dB BD-PSNR increase with "Direct" and "Hierarchical" schemes with simple temporal predictions merging, and the "Hierarchical" scheme with the "Adaptive Fusion" method respectively, compared to the reference VSRS-1DFast method. In the last column of the table (HierAF+HierSynth) we show the BD-PSNR obtained if we synthesize the 1/4 and 3/4 virtual views from left base view and our VSTP synthesis, and from VSTP synthesis and the right base view respectively. The depth map for the 1/2 view is synthesized from right and left base views. By employing this hierarchical synthesis we take advantage of the higher quality of our rendering method to improve the 1/4 and 3/4 views without modifying the rate. The delta-PSNR between reference and ours for 1/4 and 3/4 views is -0.09dB, -0.01dB, 1.58dB for Balloons, Kendo and Newspaper sequences in average over all QPs. As expected these results are consistent with the BD-PSNR reported in Table 2.7(HierAF+HierSynth compared to HierarchicalAF), since the rate is not modified. Note, that the 5 view test case scenario no longer contains the Poznan Hall2 sequence. This is due to using original views as reference for evaluating the PSNR of the 1/4 and 3/4 views which in the case of Poznan Hall2 sequence are not available. As discussed in Section 2.6.3 synthesis is proven to be more efficient. However, the quality of an encoded view is always higher than that of a synthesis, we obtained 38.50dB PSNR compared to 35.81dB PSNR and 32.99dB PSNR for direct 3D-HEVC encoding, VSTP synthesis and VSRS-1DFast synthesis, respectively, in average over all sequences and all QPs.

| Sequence | BD-PSNR (in dB) | | |
|---|---|---|---|
| | Direct | Hierarchical | HierarchicalAF |
| Balloons | 1.94 | 1.84 | 2.45 |
| Kendo | -1.12 | -0.56 | 0.93 |
| Newspaper | 4.70 | 4.80 | 5.28 |
| PoznanHall2 | 2.17 | 1.99 | 2.32 |
| **Average** | **1.92** | **2.01** | **2.74** |

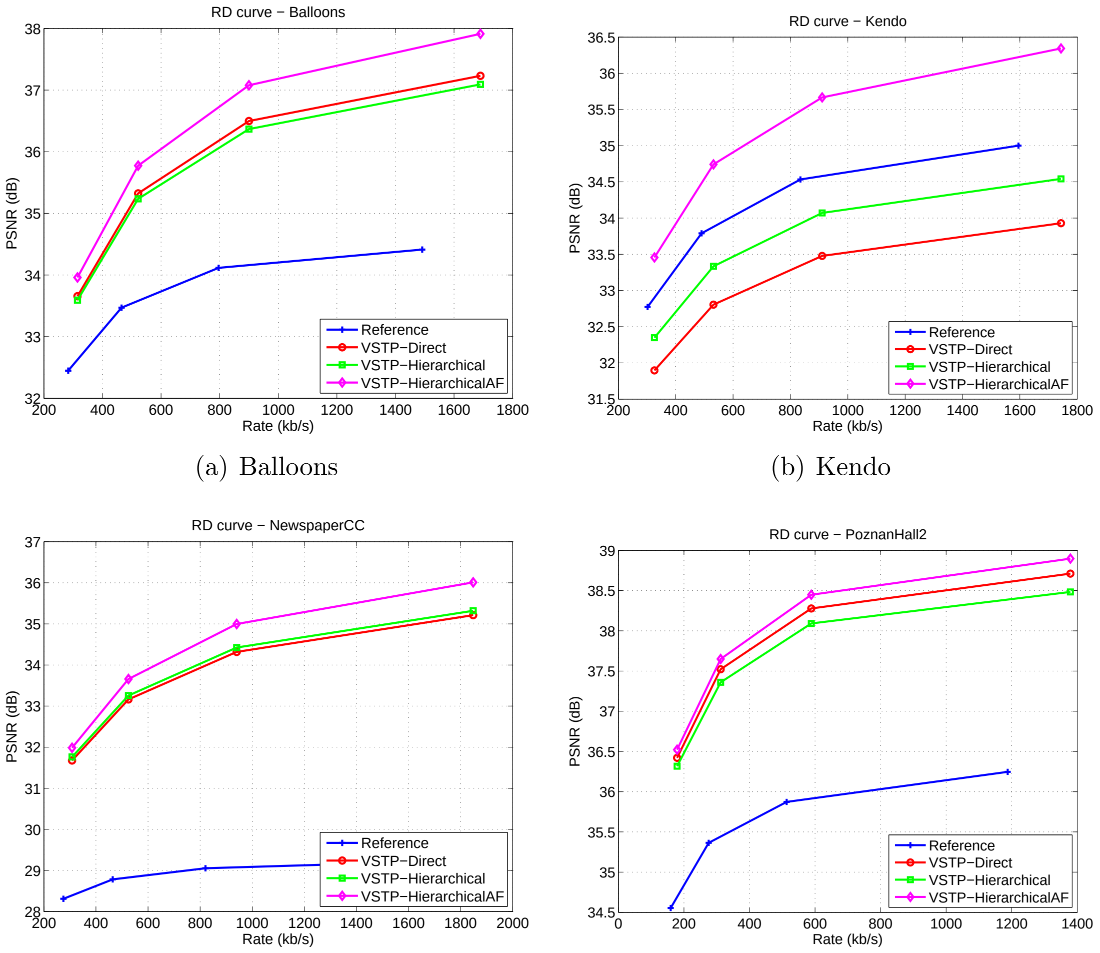
Table 2.6 – BD-PSNR values for a 3 view test case, obtained with both prediction schemes and adaptive fusion in the proposed method compared with the reference VSRS-1D fast method.

| Sequence | BD-PSNR (in dB) | | | |
|---|---|---|---|---|
| | Direct | Hierarchical | HierarchicalAF | HierAF +<br>HierSynth |
| Balloons | 0.52 | 0.49 | 0.69 | 0.64 |
| Kendo | -0.45 | -0.27 | 0.22 | 0.22 |
| Newspaper | 1.52 | 1.55 | 1.71 | 2.78 |
| **Average** | **0.53** | **0.59** | **0.87** | **1.21** |

Table 2.7 – BD-PSNR values for a 5 view test case, obtained with both prediction schemes, adaptive fusion and hierarchical synthesis in the proposed method compared with the reference VSRS-1D fast method.

The Rate Distortion (RD) curves on the 3 view test case for the reference and the proposed method (for both schemes and merging methods) are given in Figure 2.17, while the 5 view test scenario RD curves are shown in Figure 2.16. We can see that while both schemes with simple merging outperform the reference method for Balloons and Newspaper, our method outperforms the reference only with the "Hierarchical" scheme with adaptive fusion in Kendo. This is also represented in BD-PSNR values for this sequence which are only positive in the "Hierarchical" scheme with adaptive fusion, as shown in Table 2.6. Using the "Adaptive Fusion" method with the "Hierarchical" scheme brings high additional gains for Kendo sequence and moderate additional gains for Balloons, Newspaper sequences. This is expected because the fusion method was designed with the main goal of correcting bad temporal predictions caused by high intensity motion as is the case of Kendo sequence.

To better evaluate our method we perform an additional test. Since VSTP synthesis requires information to be sent through the bitstream, mainly one frame per GOP, we perform a direct comparison between the encoding of a dependant



(a) Balloons

(b) Kendo

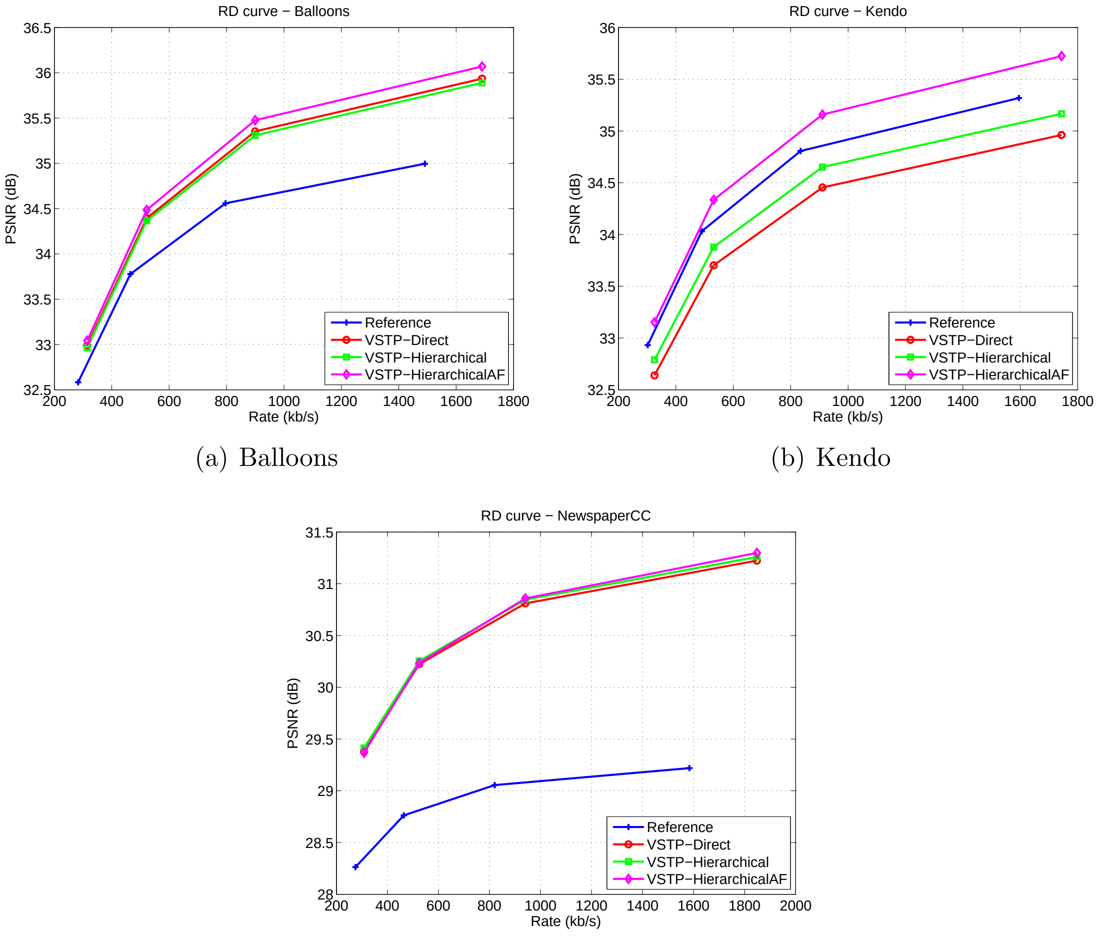
(c) Newspaper

(d) PoznanHall2

Figure 2.16 – RD curves of the reference and proposed method on 3 view test scenario for the Balloons, Kendo, NewspaperCC and PoznanHall2 sequences.

view and our VSTP synthesis. The results indicate we are able to outperform the encoding at low bitrates. This is possible due to encoding errors at low bitrates having a greater impact on the quality of the image as compared to synthesis errors; while, at the same time synthesis provides better rate. The tests were performed on Balloons, Kendo and Newspaper sequences for QPs ranging from 35 to 50 and we obtained: 1.33, 1.06, 0.62 dB BD-PSNR gain, over 3D-HEVC, respectively for each sequence.

Figures 2.18 and 2.19 show, for the four tested sequences, the variation of the PSNR of the synthesized view over time with the reference and the proposed method (both schemes and “Hierarchical” scheme with “Adaptive Fusion”). Only one QP (25) is represented for simplicity as the behavior of any curve is similar across all QPs. In the proposed method and for all sequences, we notice periodic peaks in the



(a) Balloons

(b) Kendo

(c) Newspaper

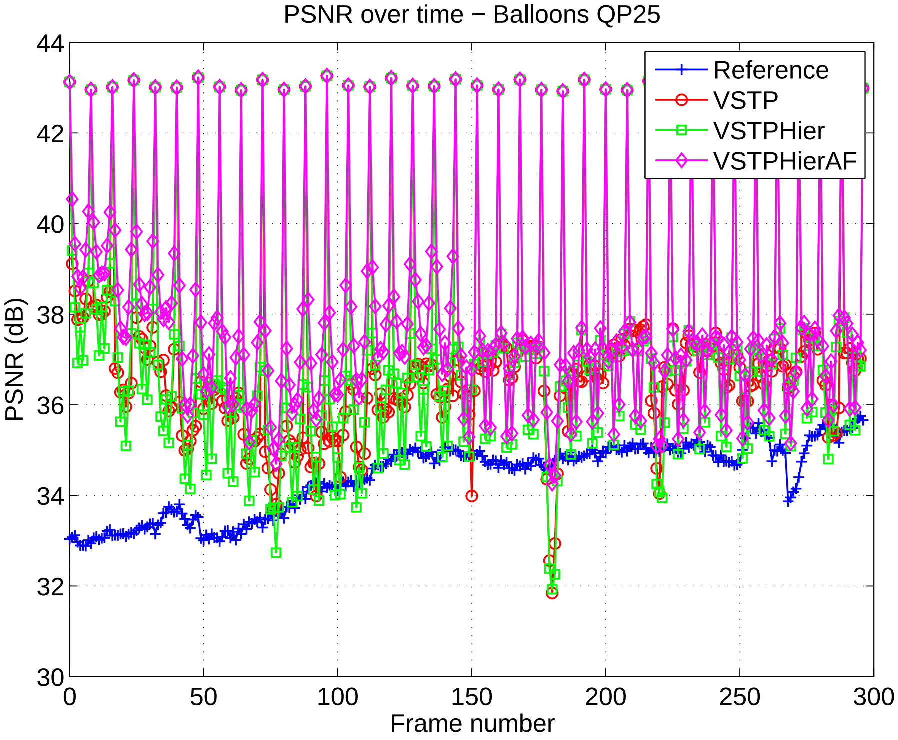
Figure 2.17 – RD curves of the reference and proposed method on 5 view test scenario for the Balloons, Kendo and NewspaperCC sequences.

synthesized view PSNR, which correspond to the first frame of each GOP. Since these frames are not synthesized but rather decoded, their PSNR is higher than any other frames in the GOP. For the Balloons, NewspaperCC and PoznanHall2 sequences, the proposed method outperforms the reference VSRS-1DFast rendering for most frames. For the Kendo sequence, our method is better only in certain parts. Figure 2.20 shows two side-by-side examples of ideal, real and color coded fusion maps for Kendo and NewspaperCC sequences. The ideal fusion map displayed here is only showing, in white, the pixels that if replaced by inter-view prediction, would have their absolute error decreased by at least 5 (we ignore small gains). We can see that our map is consistent with the ideal map for correcting high errors. The color coded maps indicate with green and blue a correct selection of temporal or inter-view prediction, respectively. Black and red indicate incorrect selections of temporal and inter-view prediction, respectively. This is also shown in Figure 2.21 where we display the difference between the absolute error of temporal and inter-view prediction for the same frame of Kendo sequence. Positive values indicate inter-view prediction is better and we can see a correspondence between high values and our fusion map.

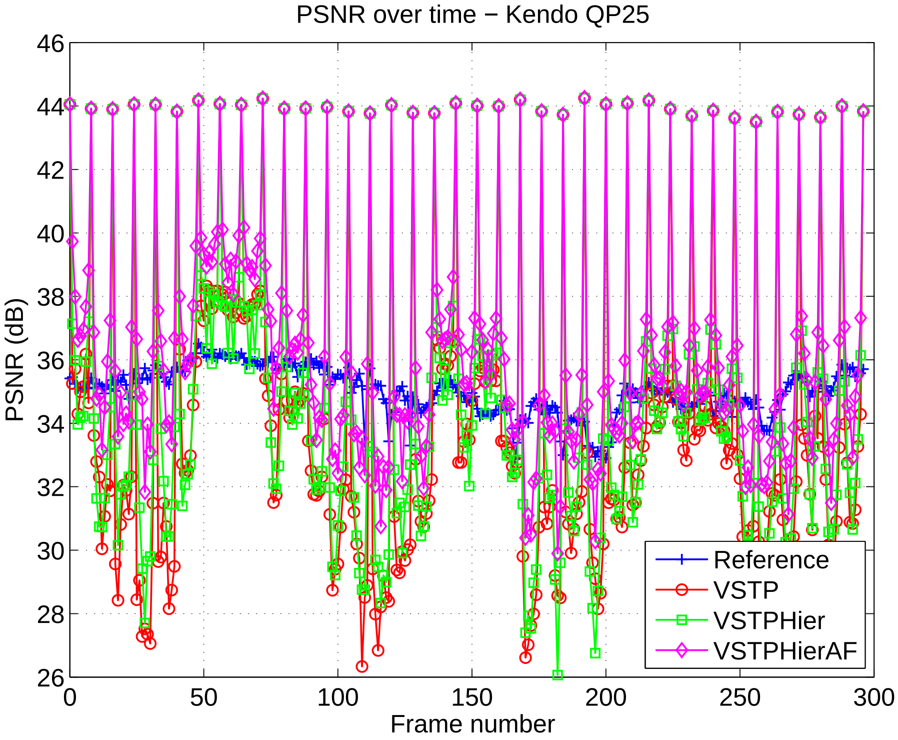
Figure 2.22 shows parts of frames synthesized using the reference and the proposed method with hierarchical scheme and Figure 2.23 shows parts of frames using the proposed method with and without adaptive fusion. For fairness of comparison, for our method, we show frames that are actually synthesized and not decoded. We can notice a clear improvement in the synthesis quality with our method: the artifacts obtained with VSRS-1DFast (highlighted in red in the figures) are efficiently removed and also artifacts in our method are removed when using the adaptive fusion.

#### 2.6.4.3 Results interpretation

The “Adaptive Fusion” method with the “Hierarchical” scheme brings significant gains in BD-PSNR. To better describe our results we will refer to an ideal case where we use the original frames to create a fusion map in which we mark all the pixels that have a lower error in the inter-view prediction compared to the temporal one, for simplicity we will only test 3 seconds from each sequence. As a mean of verifying the quality of our obtained fusion map we compute the difference between the mean absolute error (MAE) of pixels marked by a fusion map, for temporal and inter-view predictions, referred to as ΔMAE as shown in the following equation, where $\widehat{I}$ is either the temporal or inter-view prediction, $B$ is the binary fusion map and $\widehat{I_t}$, $\widehat{I_i}$

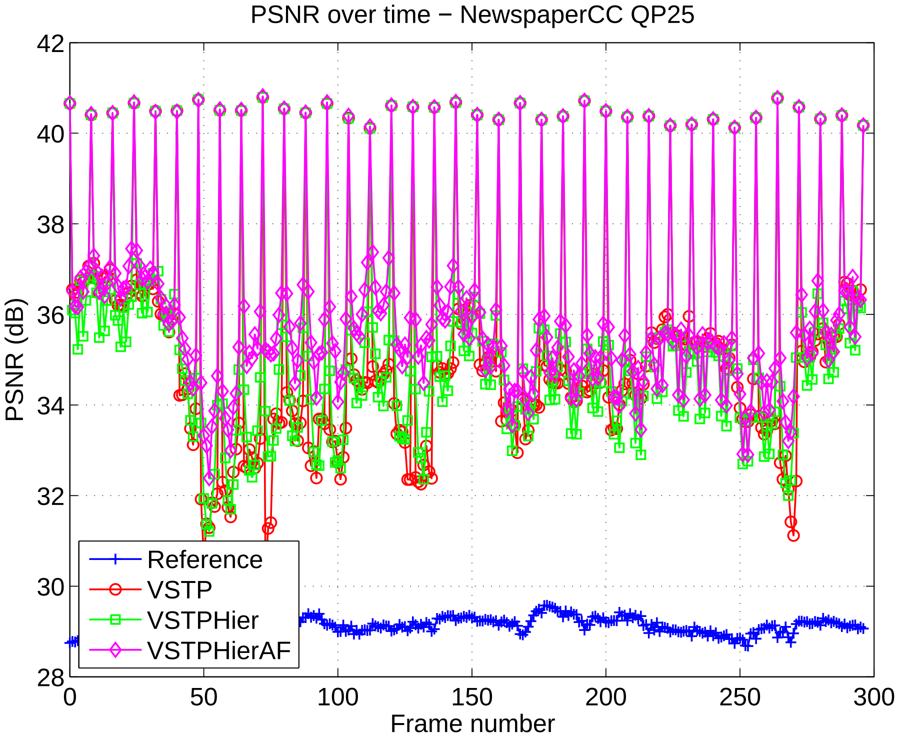


(a) Balloons

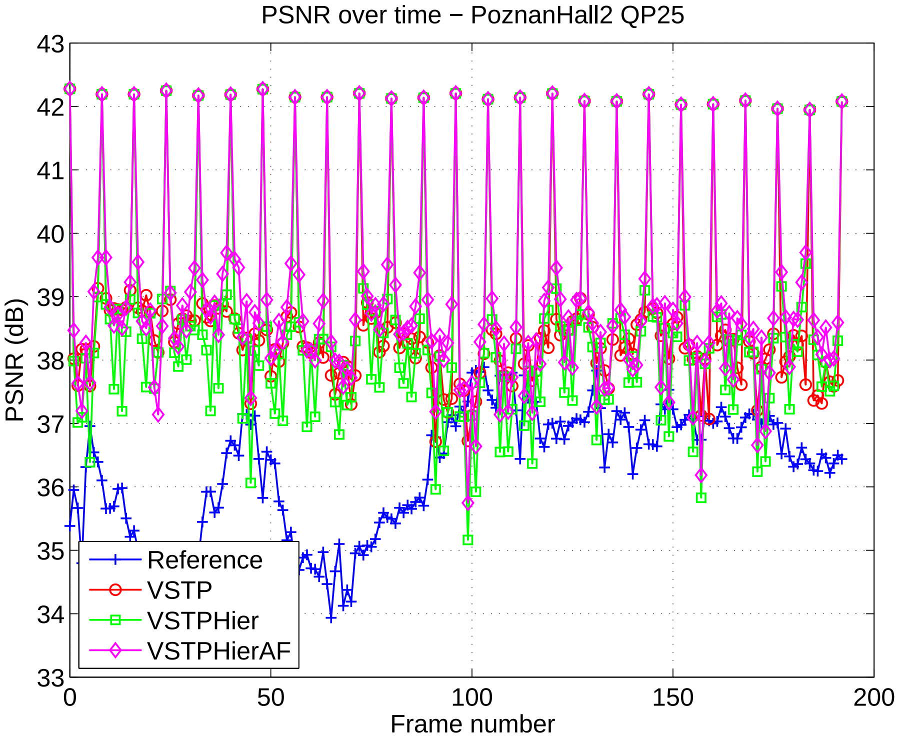


(b) Kendo

Figure 2.18 – PSNR variation over time of the middle synthesized view for the reference and proposed methods at QP 25 on Balloons and Kendo sequences.

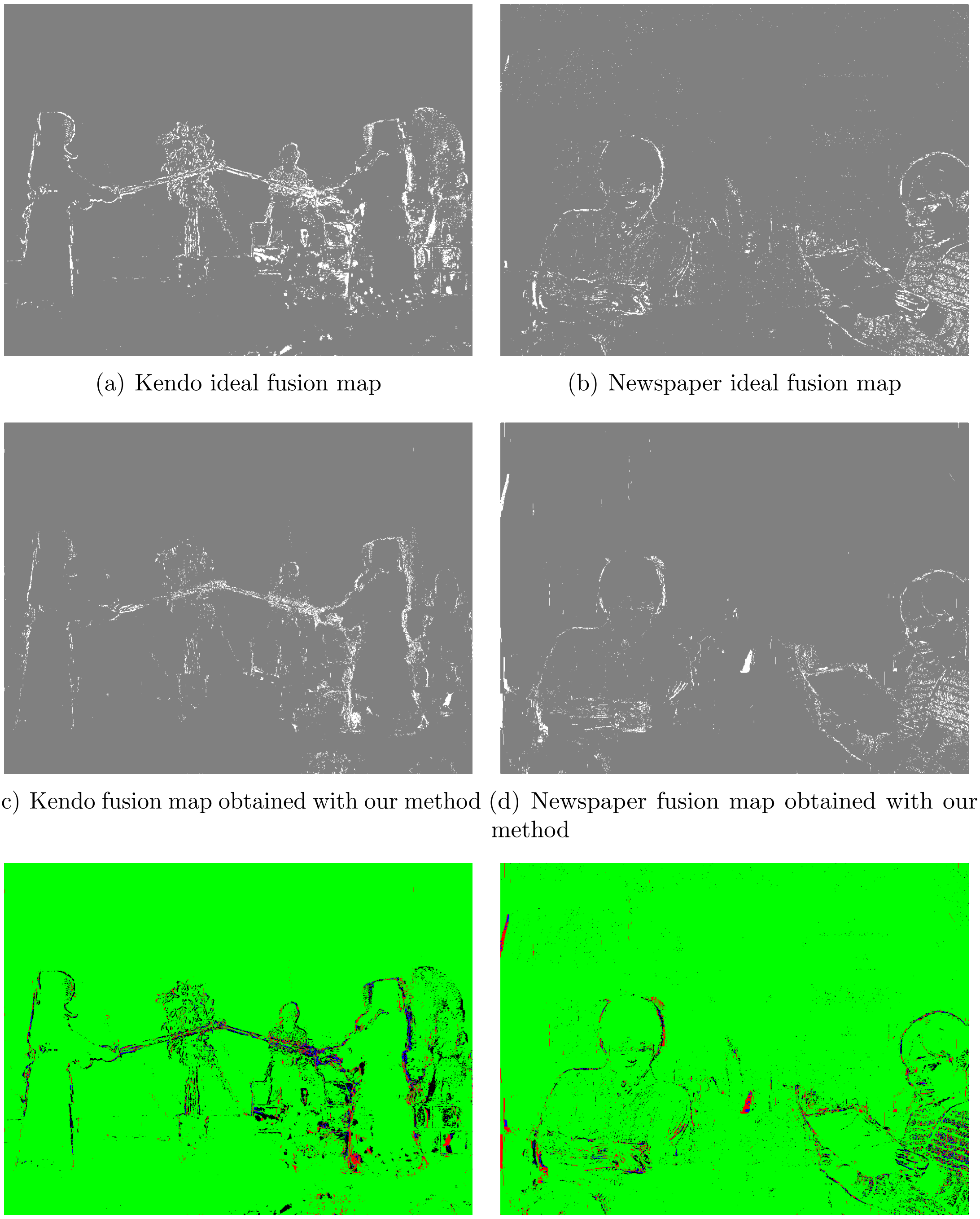


(a) Newspaper



(b) PoznanHall2

Figure 2.19 – PSNR variation over time of the middle synthesized view for the reference and proposed methods at QP 25 on NewspaperCC and PoznanHall2 sequences.



(a) Kendo ideal fusion map

(b) Newspaper ideal fusion map

(c) Kendo fusion map obtained with our method

(d) Newspaper fusion map obtained with our method

(e) Kendo color coded fusion map obtained with our method

(f) Newspaper color coded fusion map obtained with our method

Figure 2.20 – Fusion maps for frame 4 in Kendo and Newspaper sequences, at QPs 30 and 25 respectively. White indicates inter-view prediction. Figures 2.20(a) and 2.20(b) are the ideal maps, inter-view prediction is only selected if it corrects high temporal errors (the original view was used for this computation). Figures 2.20(c) and 2.20(d) are obtained with the "Adaptive Fusion" method. Figures 2.20(e) and 2.20(f) are color coded fusion maps. Green is a correct selection of temporal prediction while black is an incorrect one. Blue is a correct selection of inter-view prediction while red is an incorrect one.

and $I$ are the temporal and inter-view predictions and the original frame respectively.

$$\mathrm{MAE}(\widehat{I}, B) = \begin{cases} 0, & \text{if } B(x,y) = 0 \quad \forall \quad x, y \\ \frac{\sum\limits_{x=1}^{M}\sum\limits_{y=1}^{N} B(x,y)|\widehat{I}(x,y)-I(x,y)|}{\sum\limits_{x=1}^{M}\sum\limits_{y=1}^{N} B(x,y)}, & \text{otherwise} \end{cases} \tag{2.17}$$

$$\Delta\mathrm{MAE}(\widehat{I_t}, \widehat{I_i}, B) = \mathrm{MAE}(\widehat{I_t}, B) - \mathrm{MAE}(\widehat{I_i}, B)$$

Table 2.8 shows the percentages of replaced pixels and the MAE reduction for our method and the ideal case. In the last column we have the ratio between the delta sum of absolute differences (ΔSAD) in our method and the ideal case, as shown in Equation (2.18) where $B_{ideal}$ is the ideal fusion map.

$$\begin{aligned} \mathrm{SAD}(\widehat{I}, B) &= \sum_{x=1}^{M}\sum_{y=1}^{N} B(x,y)|\widehat{I}(x,y) - I(x,y)| \\ \Delta\mathrm{SAD}(\widehat{I_t}, \widehat{I_i}, B) &= \mathrm{SAD}(\widehat{I_t}, B) - \mathrm{SAD}(\widehat{I_i}, B) \\ \mathrm{SADR}(\widehat{I_t}, \widehat{I_i}, B_{AF}, B_{ideal}) &= \frac{\Delta\mathrm{SAD}(\widehat{I_t}, \widehat{I_i}, B_{AF)}}{\Delta\mathrm{SAD}(\widehat{I_t}, \widehat{I_i}, B_{ideal})} \end{aligned} \tag{2.18}$$

The values in Table 2.8 are the averages for all QPs. For example let us consider the Kendo sequence at QP 25. In average for this case 25.39% of the pixels in a frame are better predicted with inter-view prediction, our method selects 3.48% of the pixels to be replaced by inter-view prediction, out of which 1.6% is a bad selection (temporal prediction was actually giving better results and we replaced it with inter-view prediction). Note that the 25.39% ideally selected pixels include predicted areas which are better only by a small margin. Our selection however focuses on correcting

| **Sequence** | **Inter-view predicted pixels (%)** | | **ΔMAE** | |
|---|---|---|---|---|
| | Real | Ideal | Real | Ideal |
| Balloons | 2.74 | 30.58 | 0.50 | 2.57 |
| Kendo | 3.67 | 27.13 | 2.20 | 3.48 |
| Newspaper | 3.16 | 28.39 | 0.58 | 7.35 |
| PoznanHall2 | 0.81 | 26.05 | -0.55 | 2.34 |
| **Average** | **2.60** | **28.03** | **0.68** | **2.65** |

Table 2.8 – Adaptive Fusion results: percentage of replaced pixels and MAE gains for our method and the ideal case in which the fusion map is determined using the original view.

high errors. Even though parts of our replaced areas are actually worse predictions and increase the MAE, overall we still obtain a positive ΔMAE which shows we are correcting the high errors, as also shown in Figures 2.20 and 2.21. For the Balloons and Newspaper sequences where the introduction of "Adaptive Fusion" brings a small additional increase in BD-PSNR we have a smaller percentage of replaced pixels with a small ΔMAE in contrast to the Kendo sequence where this method brings a high additional increase in BD-PSNR. For the PoznanHall2 sequence we have a similar result in BD-PSNR, the "Direct" and "Hierarchical" schemes already provide a very good result due to low intensity motion. Here the "Adaptive Fusion" method corrects some small temporal prediction errors but also introduces inter-view prediction errors, this explains why we have a negative ΔMAE over the replaced pixels in this sequence. Note that the number of replaced pixels is smaller compared to the other sequences, only 0.81% of a frame on average, thus the quality of the entire image is affected only by a small margin.

The results of Table 2.6 and the RD curves in Figure 2.17 show that the "Hierarchical" scheme outperforms the "Direct" scheme, which was expected, since the temporal prediction distances are shorter in the first scheme. Note that in a GOP of 8 frames, the fifth frame is synthesized in the same way in both shemes, which is why the curves of Figures 2.18 and 2.19 corresponding to the two schemes, intersect not only in the first frame of each GOP but also in the fifth frame. Figures 2.18 and 2.19 also shows that the proposed method sometimes does not perform well on some series of frames, especially in the Kendo sequence. This is due to the dense motion estimation process in the reference view which gives incorrect MVs when there is high intensity motion. On the contrary, the "Adaptive Fusion" method brings a good increase of PSNR on these frames. Tweaking the optical flow parameters can account for this and would thus solve the problem but that would imply an additional rendering complexity and coding overhead if those parameters are to be sent for each frame.

Our method improves the quality of the synthesis on three levels: first, it accounts for a difference in illumination between the coded reference views and the synthesized view, which rendering techniques such as VSRS-1DFast cannot do. Indeed, while VSRS-1DFast cannot warp a different illumination level from the reference views into the synthesized view, our method propagates the correct illumination level of the sent key frames accross the rest of the frames using motion compensation. Second, our method fills holes due to disocclusions more efficiently than VSRS-1DFast. Indeed, these holes are filled using inpainting in the latter, hence creating artifacts such as the

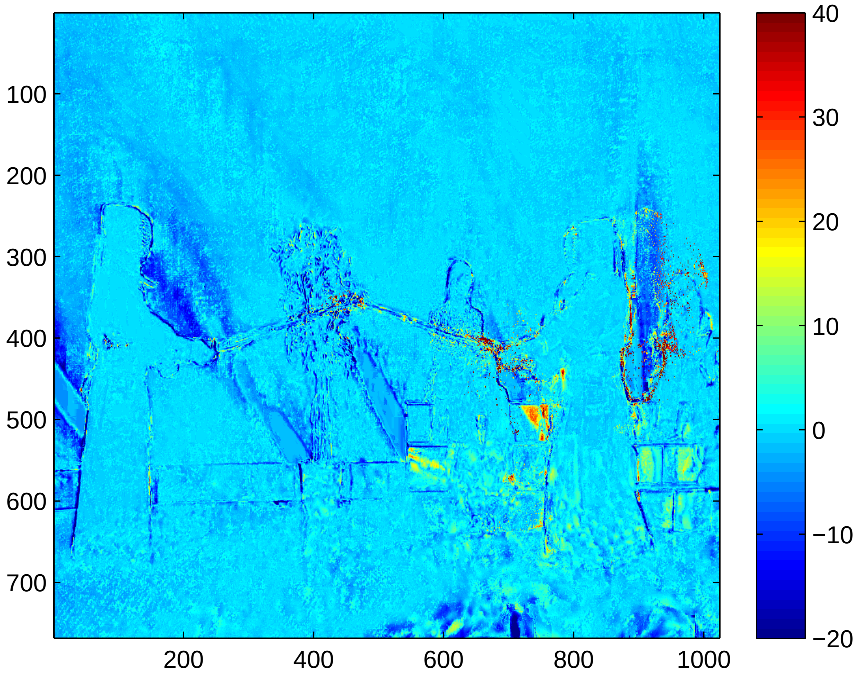
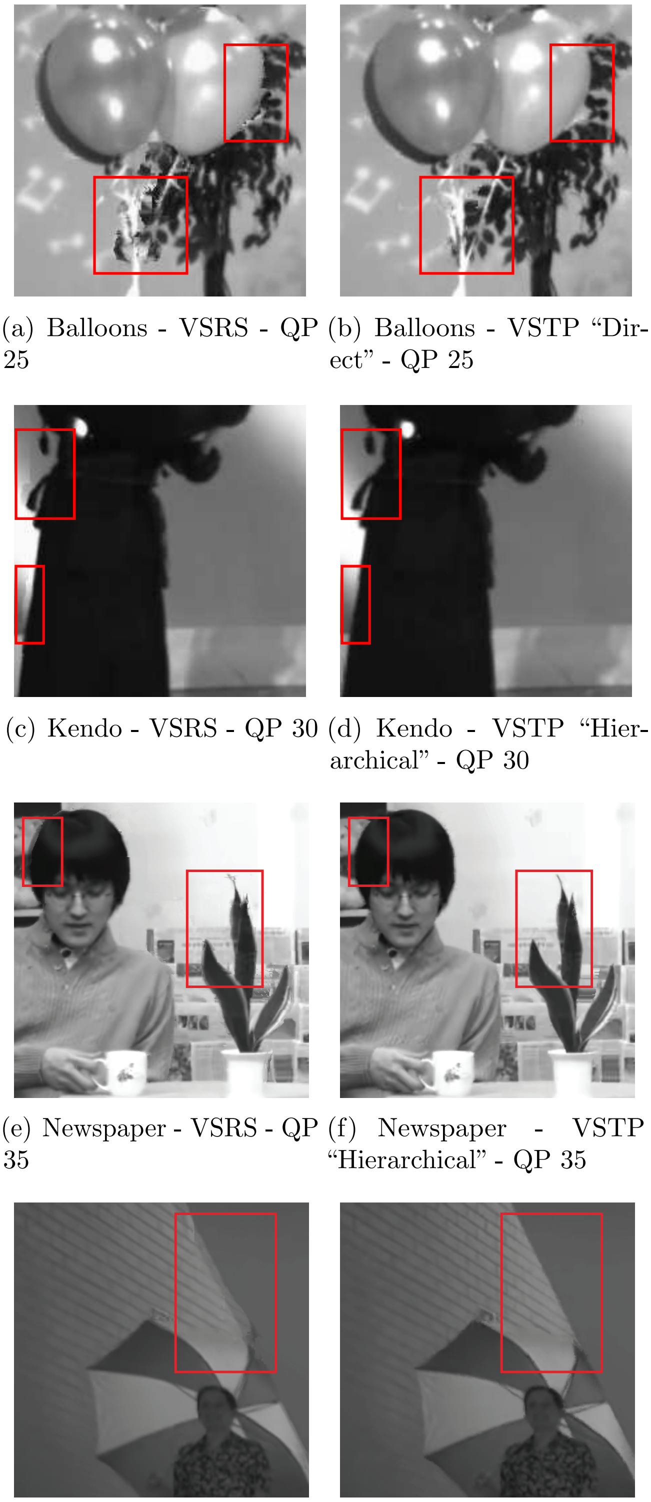


Figure 2.21 – Difference between inter-view and temporal prediction error ($\Delta$MAE) on frame 4 in Kendo sequence, QP 30.

ones highlighted in Figure 2.22. In our method, the disocclusion areas can be found in previously synthesized frames. Third, foreground objects are better rendered because the method is less sensitive to depth distortions. We use DVFs to warp dense MVFs rather than directly warping the texture (cf. Figures 2.22(e), 2.22(f), 2.22(g), 2.22(h)). In addition, VSTP brings texture information from different time instants that cannot be obtained from inter-view prediction. The fusion between the two prediction types will reduce the chance of having residual holes in the final synthesis. This explains how our method efficiently removes the aforementioned artifacts, as shown in Figure 2.22. Also, subjective viewing[1] of the sequences has shown that there are no flickering effects with our method.

The method is inherently more complex than VSRS-1DFast due to the dense motion estimation/compensation stage. Shortcuts that can reduce the complexity of our method, at the price of loosing some prediction accuracy, include block-based

[1] A synthesis example (raw YUV: 2 × 135MB) can be downloaded for viewing at the following links:
`http://perso.telecom-paristech.fr/~cagnazzo/vsrs.zip`
`http://perso.telecom-paristech.fr/~cagnazzo/vstp.zip`
for VSRS-1DFast and VSTP respectively.



(a) Balloons - VSRS - QP 25

(b) Balloons - VSTP "Direct" - QP 25

(c) Kendo - VSRS - QP 30

(d) Kendo - VSTP "Hierarchical" - QP 30

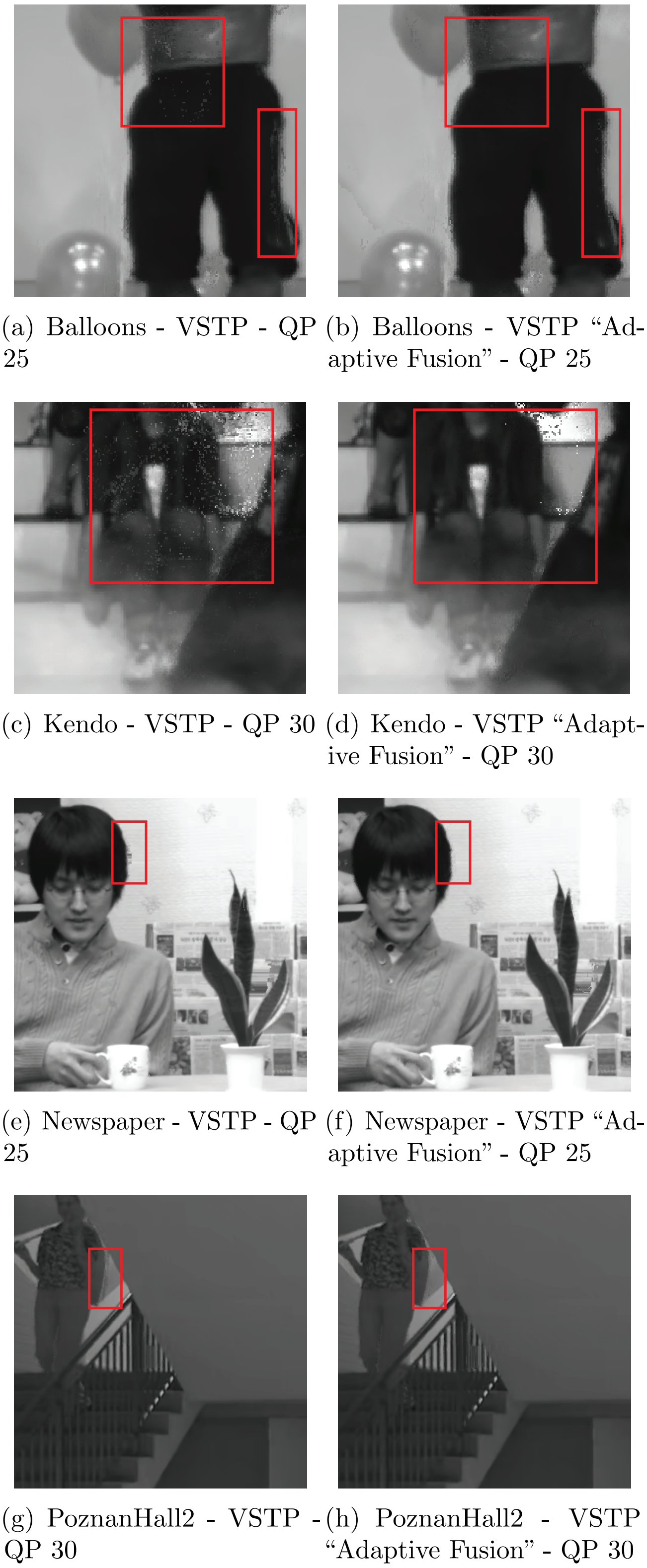
(e) Newspaper - VSRS - QP 35

(f) Newspaper - VSTP "Hierarchical" - QP 35

(g) PoznanHall2 - VSRS - QP 30

(h) PoznanHall2 - VSTP "Adaptive Fusion" - QP 30

Figure 2.22 – Parts of frames synthesized with the reference VSRS-1DFast and the proposed method. Highlighted artifacts in VSRS-1DFast (Figures 2.22(a), 2.22(c), 2.22(e) and 2.22(g)) are efficiently removed in our method (Figures 2.22(b), 2.22(d), 2.22(f) and 2.22(h)).



(a) Balloons - VSTP - QP 25

(b) Balloons - VSTP "Adaptive Fusion" - QP 25

(c) Kendo - VSTP - QP 30

(d) Kendo - VSTP "Adaptive Fusion" - QP 30

(e) Newspaper - VSTP - QP 25

(f) Newspaper - VSTP "Adaptive Fusion" - QP 25

(g) PoznanHall2 - VSTP - QP 30

(h) PoznanHall2 - VSTP "Adaptive Fusion" - QP 30

Figure 2.23 – Parts of frames synthesized with and without "Adaptive Fusion". Highlighted artifacts after merging the temporal predictions (Figures 2.23(a), 2.23(c), 2.23(e) and 2.23(g)) are efficiently removed when using "Adaptive Fusion" (Figures 2.23(b), 2.23(d), 2.23(f) and 2.23(h)).

motion estimation/compensation and uni-predictive MC (predict using only a past frame, or only a future frame).

## 2.7 Summary

In the first parts of this chapter (Sections 2.4 and 2.5), we presented several view synthesis techniques designed to exploit temporal prediction in order to improve the quality of the synthesis. A first approach uses reversed MVFs computed in the reference views and warped at the level of the synthesis by imposing an epipolar constraint between frames in different views and time instants. Using these MVFs we show how information can be extracted from different time instants. As the method requires a backward MC we propose a robust sub-pixel precision warping and filtering technique that further increases the quality of the synthesis. This contributions have been published in [PMPP$^+$15].

A second approach uses full frame temporal predictions to improve the synthesis. We use forward ME and MC in order to generate four temporal predictions of the frame. Along with the two inter-view predictions we, we use two blending methods. A simple averaging or an adaptive blending approach that selects between the average of all six predictions or the inter-view ones. Gains over the first approach and VSRS-1DFast are reported. This work is published in [PCPP$^+$16].

Based on our findings, the final part of this chapter combines the approaches and integrates them with 3D-HEVC resulting in a method that can be viewed as in between coding and synthesis [PMC$^+$16]. Namely, some key frames of the synthesized view are encoded in the bitstream, and the rest are interpolated using MC with vectors warped from reference views. Four temporal predictions are used to synthesize a frame. Two prediction schemes referred to as “Direct” and “Hierarchical” have been presented in this work. The first synthesizes frames using only with key frames as references, while the other motion compensates from previously synthesized frames. We also introduced a prediction merging method referred to as “Adaptive Fusion” that selects between inter-view and temporal prediction. Our method brings 0.53dB and 0.59dB BD-PSNR increase with the “Direct” and “Hierarchical” schemes respectively and 0.87dB BD-PSNR with “Hierarchical” scheme and “Adaptive Fusion” in average for several test sequences over the state-of-the-art VSRS-1DFast software under 3D-HEVC standards.

Furthermore, the MVF precision on frames with high intensity motion can be improved by using a better motion estimation technique or using an adaptive GOP size with respect to motion intensity. The “Adaptive Fusion” method can be further improved by finding a better inter-view/temporal selection criterion. Additional adjacent views that are not available at the encoder side can be further improved by deriving the vector fields required to directly predict the frames from the key frames. Finally, the frequency at which key frames are sent in our method can be modified: lower frequencies allow bitrate savings but they imply motion estimation between distant frames, which decreases prediction accuracy. Finding a good trade-off for this parameter is an interesting future research subject.

# Chapter 3

# Region-of-interest based quality evaluation of view synthesis techniques

## Contents

The large number of factors which affect the quality of synthesized images complicates the problem of objectively measuring or comparing the performance of view synthesis algorithms. View synthesis methods introduce localized artifacts when creating new virtual views. Therefore, evaluating these methods requires a different approach in order to identify and emphasize synthesis artifact prone areas, while diminishing the impact of other types of artifacts, such as those produced by quantization during the video coding. In this chapter, we investigate the use of a Region-Of-Interest approach to evaluate the quality of DIBR based synthesis methods. Based on the assumption that certain areas determined by the geometrical properties of the scene are prone to distortions, we select a ROI by analyzing multiple DIBR methods together.

## 3.1 Introduction and problem overview

### 3.1.1 Quick reminder of view synthesis

We begin this chapter with a quick review of view synthesis techniques and their usage. The process of generating a video sequence or an image from existing sequences or images, as if acquired from a new point of view, is known as view synthesis. Several methods exist in the literature and can be mainly divided into three categories based on the use of geometrical information [SK00]:

***i)*** Methods that use explicit scene geometry in the form of depth maps to warp pixels from one view into a virtual one [ZWPSxZy07] [CLLY08], also known as DIBR methods.

***ii)*** Methods that use implicit geometry such as pixel correspondences computed with optical flow or any other motion estimation technique [DCPP14] [DCPP14] [KMW95].

***iii)*** Methods that do not require geometrical information and use interpolation and filtering to synthesize new views. Some of the most popular ones include light field rendering [LH96], concentric mosaics [SH99] or lumigraph [BBMG01].

The first category received great interest as it provides a fast and efficient way of generating multiple views. Applications such as 2D to 3D automatic conversion or free view point television (FTV) [TTFY11], immersive teleconference systems, medical applications and gaming [DPPC13], generally rely on this category of view synthesis methods.

### 3.1.2 Error sources in view synthesis

The quality of DIBR generated virtual views is greatly affected by multiple factors. For an easier understanding we propose to classify the error sources as follows:

**Geometrical limitations:** Some areas in the virtual view are not visible in the reference views. As no information is available, they manifest as holes in the synthesized image. These areas are also known as disocclusions. They can be divided in two types based on their location [HKA13]:

- Border disocclusions are produced by the displacement of the field of view and are located on the sides of the images. In order to avoid them it is usually preferred to merge two synthesized views from a left and a right reference view.
- Non-border disocclusions appear around foreground object edges. Even when using a left and right reference for synthesis, parts of the non-border disocclusions may coincide in the merged views. Traditionally this problem is resolved using inpainting algorithms such as [DPP10] [GM14] [CPT04]. Other methods propose a preprocessing of the depth maps in order to reduce the size of disocclusions [LE10], [WZ11]. When working on video sequences, temporal correlations can also be exploited to retrieve information on disoccluded areas, as discussed in Chapter 2.

**Precision related errors:** This type of error sources can be further divided depending on how they affected the final synthesis:

- Directly: This type of errors are caused by processes that directly affect the texture. For example the precision of the warping process. Furthermore, all synthesized views are also affected by the encoding quality of the reference views or depth maps. When using encoded reference views, the pixels that are warped in the synthesized view are subject to an absolute quantization error of up to half the quantization step.
- Indirectly: Errors caused by the quality of the depth map. Like texture, encoded depth maps will also be subjected to quantization errors, especially since they are usually encoded using higher QPs. However, the quantization

errors in depth maps will impact the synthesized view in a different manner. A small error in the depth map results in warping the pixel to a slightly different position in the virtual view. While this is not a big issue for pixels located in areas with uniform texture, it can create very high distortions on the edges of objects (consider a scene with a black object on white background). The precision of depth maps is also affected by the quantization of real depth values to, usually, 256 levels. Another common problem is the texture-depth alignment which may lead to pixels belonging to a foreground object to be warped as if they are part of the background or vice-versa. In general, these problems appear in areas where depth maps are not uniform (i.e. foreground/background separation).

**Illumination errors:** The source of these errors is the inability of view synthesis algorithms to correctly reproduce variations in illumination of the scene. A quick example would be a scene containing a mirror. As the surface of the mirror has a relatively uniform depth, the object is displaced as a whole without accounting for the change in reflexion. The same can be said about shadows or other illumination variations.

### 3.1.3 Discussion and chapter overview

Because the artifacts produced by synthesis are inherently different from those of encoding, evaluating the quality of synthesis in systems using DIBR rendering is not a trivial matter. Especially, considering the final goal of such systems is to provide a 3D experience. Measures such as Peak-Signal-to-Noise-Ratio (PSNR) provide a good objective evaluation but fail to emphasize the errors caused by object distortions. Evaluation methods that take into account the structure of the image have been created, one of the most popular being the structural similarity based metric (SSIM) [WBSS04] (see Sec. 3.2.2.1). While SSIM takes into account the structural distortions of an image, small differences in background color reproduction might mask the impact of important artifacts. As discussed above, the majority of high errors in view synthesis are mostly located close to the edges of foreground objects.

The Video Quality Expert Group (VQEG) created the 3DTV Work Group, which is now part of the Immersive Media Group [VQE], to conduct experiments on the quality of 3D media. Numerous studies were made to address the problem of synthesized video evaluation. Tikanmaki *et al.* [TGM08] studied the assessment of

3-D encoded video and the authors also considered the synthesized view quality. Bosc *et al.* [BPc+11] studied the quality of DIBR synthesis and proposed two approaches based on a region of interest (ROI) evaluation. A first method analyzes the contours shifts in the synthesized view and a second one focuses on evaluating the mean SSIM score over disoccluded areas.

Our goal is to comparatively evaluate multiple view synthesis methods. As most evaluation methods we consider the reference to be known. Although this is untrue for a virtual view, for the purpose of evaluation it is generally preferred to synthesize an existing view of an MVD video sequence. The first part of this chapter proposes a new view synthesis evaluation technique, based on SSIM, which focuses on comparing view synthesis artifacts around sensitive, error prone, areas of the image. Two different methods are used for selecting the areas of interest in the evaluation of two synthesis methods. Firstly, we analyze the distribution of errors and separate high synthesis errors from quantization ones. A second approach is focused on directly evaluating the areas predicted differently by the two tested methods. We show this technique to bring a better differentiation of synthesis methods with respect to the impact of synthesis artifacts on the image quality. Also, additional information can be extrapolated on the spatial localization of distortions when compared to an SSIM or PSNR evaluation.

The second part of this chapter further extends these ideas and several possible enhancements are discussed. Furthermore, we perform an in depth analysis of the proposed technique and compare it to the work of Bosc *et al.* [BPc+11]. A publicly available view synthesis subjective evaluation database is used in order to validate our assumptions.

## 3.2 A distortion evaluation framework for view synthesis

### 3.2.1 View synthesis methods used in this study

We evaluate three different view synthesis methods. The DIBR implementation of VSRS-1DFast [ZTWY13], a method that uses the filtering technique described in Section 2.4.2, [PMPP+15] and the blend of temporal prediction with DIBR synthesis described in [PCPP+16].

All methods use depth maps to compute disparity. Usually, depth maps are given with inversed quantized values between $[0 \quad 255]$. The tests were performed on MVD sequences acquired with 1D arrays of rectified cameras. The disparity was computed using Eq. 2.4.

All methods use two texture and depth views and synthesize an intermediary view. The tests were performed using a factor of four sub-pixel precision. The same line-wise hole-filling method is used in all three view-synthesis methods, albeit the holes can differ in size. Additional details on the methods are available in Chapter 2.

### 3.2.2 Synthesis evaluation and ROI selection

As discussed in Sec. 3.1, view synthesis evaluation methods should also take into account the structure of the image. While some metrics, such as the structural similarity index, take into account structure, they do so in a local and low level sense by means of correlation. However, the main issues in synthesis are the disoccluded areas and the distortion of foreground objects and other artifacts caused by texture-depth misalignment or imprecise depth maps. Thus, when comparing different methods, areas around foreground object edges and disocclusions should be emphasized in the evaluation. Smooth background areas typically have low errors (see Sec. 3.2.3). It is reasonable to assume that methods which bring only small corrections in these areas will not have a significant visual impact, even though PSNR gains can be achieved. In what follows, we will describe the SSIM metric and show how a selection of artifact prone areas can be achieved for better evaluating view synthesis methods.

#### 3.2.2.1 Structural Similarity index (SSIM)

Wang *et al.* [WBSS04] assume that the human visual system is highly focused on the perception of structural information of a scene. The proposed measure is designed to asses the degradation of structural information. SSIM separates the similarity measurement in three components: luminance, contrast and structure.

The general form of SSIM index is given by:

$$SSIM(r,d) = [l(r,d)]^{\alpha} \cdot [c(r,d)]^{\beta} \cdot [s(r,d)]^{\gamma} \tag{3.1}$$

where, $r$ and $d$ refer to windows in the reference and distorted images respectively. $\alpha$, $\beta$ and $\gamma$ are usually equal to 1. The functions $l$, $c$ and $s$ correspond to luminance, contrast and structure comparisons and are defined as following:

$$l(r,d) = \frac{2\sigma_r\sigma_d + C_1}{\sigma_r^2 + \sigma_d^2 + C_1} \tag{3.2}$$

$$l(r,d) = \frac{2\mu_r\mu_d + C_2}{\mu_d^2 + \mu_r^2 + C_2} \tag{3.3}$$

$$s(r,d) = \frac{\mu_{rd} + C_3}{\mu_r\mu_d + C_3} \tag{3.4}$$

A condensed form of the SSIM is given by:

$$SSIM(r,d) = \frac{(2\mu_r\mu_d + C_1)(2\sigma_{rd} + C_2)}{(\mu_r^2 + \mu_d^2 + C_1)(\sigma_r^2 + \sigma_d^2 + C_2)} \tag{3.5}$$

where $\mu_r$ and $\mu_d$ are the means of $r$ and $d$, $\sigma_r$ and $\sigma_d$ are the standard deviations and $\sigma_{rd}$ is the correlation coefficient between $r$ and $d$. $C_1$ and $C_2$ are two variables used to stabilize the division with small denominator.

The index computation is performed on windows centered around a position $(x, y)$. The SSIM computation between the reference and distorted windows ($r(x, y)$ and $d(x, y)$) will return an SSIM index for the position $(x, y)$ in an image. The score of an image can then be obtained by centering the windows in each pixel and averaging the SSIM index, this is known as mean SSIM index or MSSIM:

$$MSSIM(I_r, I_d) = \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} SSIM(r(x,y), d(x,y)) \tag{3.6}$$

where $I_r$ and $I_d$ are the reference and distorted images.

Note that, in general the mean SSIM index is simply referred to as SSIM, in order to avoid confusion with Multi scale SSIM. In the rest of this work we will refer to mean SSIM index simply as SSIM.

#### 3.2.2.2 Histogram based area selection

When testing two view synthesis methods, a first way of selecting the areas prone to synthesis errors would be to look for pixels which have a relative high absolute error. This can provide a good indication on the quality of the synthesis methods. Errors produced by the quantization during the encoding of the reference views and errors caused by depth quantization or the interpolation process are usually uniformly spread and do not necessarily depend on the structure of the scene or the view synthesis method employed. This can also be observed in Fig. 3.1 where two binary masks are shown. Black indicates pixels that have an absolute error larger than twice the mean absolute error. Fig. 3.1(a) shows the mask for a frame encoded with 3D-HEVC at QP 25 and Fig. 3.1(b) is obtained from the same frame synthesized with VSRS-1DFast from non-encoded reference views. It is easily noticeable that in the case of encoding, high errors are spread across the image. In the case of synthesis, impactful errors are concentrated and their spatial positioning is dependent on the structure of the scene. Focusing the synthesis evaluation on these areas can provide a better indication of the method's quality for object distortion, while ignoring other less impactful error sources, like quantization errors produced by encoding.

The threshold used in generating the mask should be selected in such a way that is able to separate the large errors coming from synthesis. In order to do this, we depict the distribution of absolute errors for a synthesized view. In Fig. 3.2, as expected, we find a large percentage of pixels with small errors. This is normal for encoded sequences as errors are normally distributed around zero. However, in Fig. 3.2 we also find an increased error density around a larger value, marked with a red line in the figure. This is caused by the synthesis process. As discussed, the synthesis will introduce high distortions compared to the quantization errors especially for low QPs. Quantization errors are bounded in absolute value by half the quantization interval, while synthesis errors can be higher. The threshold can be determined by finding this value where higher errors are concentrated. Let us consider two vectors $\mathcal{E} = [\epsilon_1, \epsilon_2, .., \epsilon_n]$ and $\mathbf{P} = [p_1, p_2, .., p_n]$. $\mathcal{E}$ contains absolute error values such that $\epsilon_x > \epsilon_{x+1}$ and $\epsilon_x - \epsilon_{x+1} = constant$. $\mathbf{P}$ is the percentage of pixels with an absolute error between $\epsilon_x$ and $\epsilon_{x+1}$. The threshold can be expressed as:

$$\mathcal{T} = \mathcal{E}(\min(\{x|\Delta(x) > 0\}) + 1) \tag{3.7}$$

where $\Delta(x)$ is:

$$\Delta(x) = p_{x+1} - p_x \tag{3.8}$$

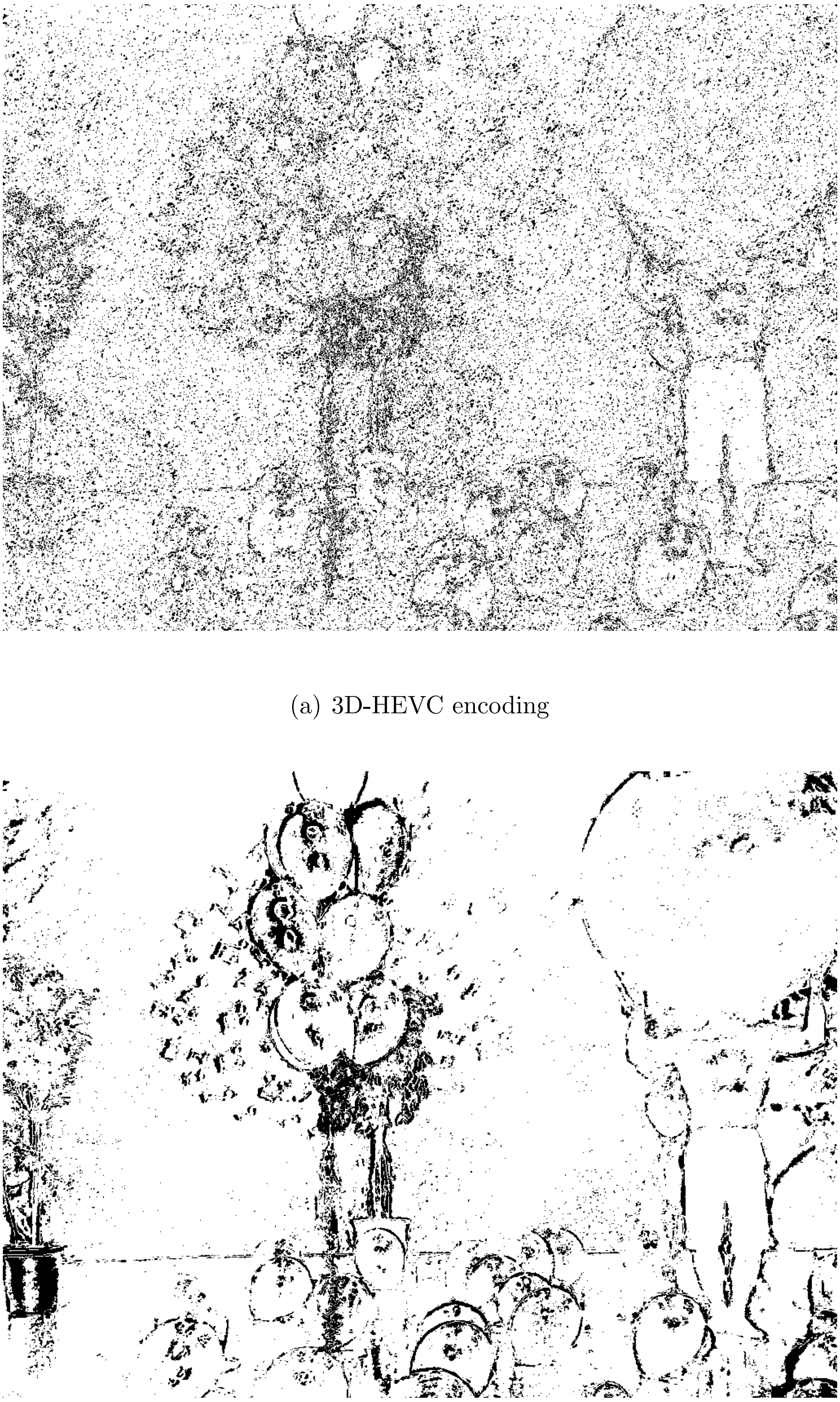
The binary mask used for synthesis distorted area selection can then be computed as:

$$B(x,y) = \begin{cases} 0 & \text{if } \mid (I_r(x,y) - I_d(x,y)) \mid < \mathcal{T} \\ 1 & \text{if } \mid (I_r(x,y) - I_d(x,y)) \mid \geq \mathcal{T} \end{cases} \tag{3.9}$$

where $I_r$ and $I_d$ are the reference and distorted images.

However, this approach can produce different masks for two evaluated synthesis methods ($B_{d_1}$ and $B_{d_2}$). In order to assure a consistent evaluation in both compared methods, the SSIM index should be computed using a single mask. This can be achieved by performing the evaluation in the locations obtained by merging the two masks as shown in Eq. 3.10.

$$B_{hist}(x,y) = B_{d_1}(x,y) \vee B_{d_2} \tag{3.10}$$



(a) 3D-HEVC encoding

(b) Synthesis

Figure 3.1 – Binary masks on Balloons sequence frame 1, black indicates pixels with high absolute errors. 3.1(a) was obtained from a 3D-HEVC encoding at QP 25 and 3.1(b) from the same view synthesized from non-encoded reference views.

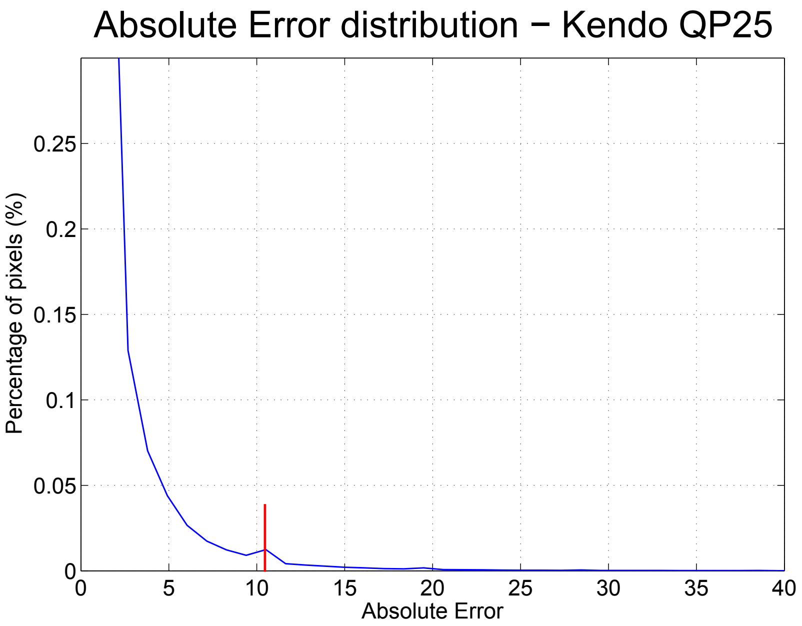


Figure 3.2 – Absolute error distribution for a synthesized frame in Kendo sequence, at QP=25.

where $\vee$ is the logical *or* operation.

The score of each method can then be obtained by averaging the SSIM index over all pixels selected with the binary mask:

$$\begin{aligned} SSIM_{hist}^{d1}(I_r, I_{d_1}, I_{d_2}) = & \frac{1}{\sum\limits_{x=1}^{M}\sum\limits_{y=1}^{N} B_{hist}(x,y)} \\ & \sum_{x=1}^{M}\sum_{y=1}^{N} SSIM(r(x,y), d1(x,y)) \times B_{hist}(x,y) \end{aligned} \tag{3.11}$$

where $d_1$ and $d_2$ refer to the two distorted images obtained by different synthesis methods and $M$, $N$ are the width and height of the image.

#### 3.2.2.3 Error prone area selection

Another option for selecting relevant spatial locations that need to be evaluated when comparing synthesis methods, is to look directly at the differences between methods. We can select these areas by generating a new selection mask containing all areas which were rendered differently by the two methods as shown in Eq. 3.12:

$$B_{epas}(x,y) = \begin{cases} 0 & \text{if } \mid (I_{d_1}(x,y) - I_{d_2}(x,y)) \mid < \mathcal{T} \\ 1 & \text{if } \mid (I_{d_1}(x,y) - I_{d_2}(x,y)) \mid \geq \mathcal{T} \end{cases} \tag{3.12}$$

where ($\mathcal{T}$) is a threshold.

When comparing two synthesis methods, we are interested in their behavior in areas where pixels are predicted differently. Evaluating areas where both methods provide similar pixel predictions will not offer a good comparison of the methods. Establishing a selection threshold in this case is easier. Since we are interested in relative large differences, the mean absolute error can provide a good threshold.

### 3.2.3 Experimental results

In order to verify our evaluation method we use the 3D-HEVC test model (3D-HTM). The encoder and renderer configurations follow the Common Test Conditions (CTCs) for conducting experiments with 3D-HEVC [RMV13]. The tested video sequences are: Balloons, Kendo, NewspaperCC and PoznanHall2. The first three sequences have a resolution of 1024 × 768 with 30 fps and a total of 300 frames. The later has a resolution of 1920 × 1088 with 25 fps and a total of 200 frames. For each sequence, we use two encoded reference views with their associated depth maps and synthesize an intermediate view. For Balloons and Kendo sequences, we use views 1&5 as reference and synthesize view 3. Views 2&6 and 5&7 are used as reference for NewspaperCC and PoznanHall2 sequences respectively, while views 4 and 6 are synthesized. The encoding is performed with 3D-HEVC using four QPs for texture: 25, 30, 35, 40. Different QPs are used for the depth maps, as recommended by the CTCs: 34, 39, 42, 45.

We evaluate the synthesis methods as detailed in Section 3.2.2. For VSRS-1DFast we use the CTCs recommended configuration. The similarity enhancement and sub pixel precision options are active. The warping and filtering technique (Wf) presented in Section 2.4.2 [PMPP$^+$15] uses a filtering window of size 7 and a sub pixel precision factor of 1/4. The method based on temporal and inter-view prediction blending (P+Badapt) detailed in Section 2.5 [PCPP$^+$16] uses an optical flow implementation for motion estimation [Liu] and a temporal prediction distance of two. Each method is evaluated using PSNR and SSIM. SSIM$_{hist}$ and SSIM$_{epas}$ are used to evaluate and compare Wf and P+Badapt with VSRS-1DFast.

Table 3.1 shows the PSNR results for the three tested methods: VSRS-1DFast, Wf and P+Badapt. On the bottom of the table we can see the average result across sequences and the last row shows the gain obtained by the later two methods. As can be seen both Wf and P+Badapt outperform VSRS-1DFast while the best results are obtained by P+Badapt. Another aspect of interest is that the gain remains relatively stable across QPs. Using the SSIM metric shows similar results, as can be observed in Table 3.2.

| **Sequence** | VSRS-1DFast<br>PSNR (dB) | | | | Wf<br>PSNR (dB) | | | | P+Badapt<br>PSNR (dB) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| QPs | 25 | 30 | 35 | 40 | 25 | 30 | 35 | 40 | 25 | 30 | 35 | 40 |
| Balloons | 34.37 | 34.07 | 33.43 | 32.41 | 34.39 | 34.14 | 33.52 | 32.51 | 34.74 | 34.45 | 33.8 | 32.72 |
| Kendo | 34.98 | 34.51 | 33.77 | 32.75 | 35.37 | 34.9 | 34.15 | 33.08 | 35.37 | 34.87 | 34.13 | 33.06 |
| NewspaperCC | 29.2 | 29.05 | 28.78 | 28.31 | 29.81 | 29.69 | 29.39 | 28.83 | 29.85 | 29.74 | 29.44 | 28.9 |
| PoznanHall2 | 36.24 | 35.87 | 35.36 | 34.55 | 36.35 | 36.02 | 35.51 | 34.77 | 36.49 | 36.2 | 35.7 | 34.86 |
| Average | 33.70 | 33.37 | 32.83 | 32 | 33.98 | 33.69 | 33.14 | 32.3 | 34.11 | 33.82 | 33.27 | 32.38 |
| $\Delta$PSNR | - | - | - | - | **0.28** | **0.32** | **0.31** | **0.3** | **0.41** | **0.45** | **0.44** | **0.38** |

Table 3.1 – Average PSNR for all tested methods and sequences at each QP.

| **Sequence** | VSRS-1DFast<br>SSIM | | | | Wf<br>SSIM | | | | P+Badapt<br>SSIM | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| QPs | 25 | 30 | 35 | 40 | 25 | 30 | 35 | 40 | 25 | 30 | 35 | 40 |
| Balloons | .9583 | .9541 | .9460 | .9322 | .9571 | .9530 | .9450 | .9313 | .9597 | .9556 | .9479 | .9343 |
| Kendo | .9635 | .9593 | .9528 | .9430 | .9631 | .9590 | .9526 | .9429 | .9638 | .9600 | .9538 | .9444 |
| NewspaperCC | .8965 | .8898 | .8771 | .8573 | .9004 | .8939 | .8802 | .8590 | .9020 | .8957 | .8824 | .8621 |
| PoznanHall2 | .9352 | .9322 | .9272 | .9190 | .9358 | .9330 | .9281 | .9198 | .9370 | .9340 | .9290 | .9208 |
| Average | .9384 | .9339 | .9258 | .9129 | .9391 | .9347 | .9265 | .9133 | .9406 | .9363 | .9283 | .9154 |
| $\Delta$ SSIM | - | - | - | - | **.0007** | **.0009** | **.0007** | **.0004** | **.0022** | **.0025** | **.0025** | **.0025** |

Table 3.2 – Average SSIM for all tested methods and sequences at each QP.

Table 3.3 shows the results obtained when evaluating Wf against VSRS-1DFast with the proposed methods: SSIM$_{hist}$, SSIM$_{epas}$. We can see that losses or gains are slightly increased and better differentiated in comparison to SSIM results. Also, when computing the difference between the average values we no longer have a gain at low QPs. This indicates that while the Wf method improves the overall image in comparison to VSRS-1DFast, it does not provide any benefits toward reducing the object boundary distorsions. The PSNR and SSIM gains provided by this method are given by a reduction in small errors. This is expected since the method proposes a sub-pixel precision warping technique with high accuracy, without tackling the structural aspect of the scene.

The comparison results of P+Badapt and VSRS-1DFast are reported in Table 3.4. A significant increase in $\Delta$ values can be observed in comparison to SSIM. SSIM$_{hist}$

| **Sequence** | VSRS-1DFast | | | | | | | | Wf | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| QPs | 25 | | 30 | | 35 | | 40 | | 25 | | 30 | | 35 | | 40 | |
| Method | hist | epas | hist | epas | hist | epas | hist | epas | hist | epas | hist | epas | hist | epas | hist | epas |
| Balloons | .8546 | .9186 | .8550 | .9120 | .8593 | .8970 | .8396 | .8708 | .8464 | .9112 | .8473 | .9052 | .8535 | .8905 | .8356 | .8649 |
| Kendo | .8798 | .9279 | .8743 | .9192 | .8613 | .9058 | 0.8407 | .8852 | .8717 | .9239 | .8678 | .9149 | .8568 | .9019 | .8375 | .8819 |
| NewspaperCC | .6062 | .8309 | .5592 | .8173 | .5958 | .8050 | .5763 | .7801 | .6194 | .8358 | .5753 | .8243 | .6078 | .8113 | .5866 | .7841 |
| PoznanHall2 | .7517 | .8922 | .7359 | .8834 | .7254 | .8699 | .7041 | .8485 | .7466 | .8932 | .7345 | .8857 | .7300 | .8739 | .7095 | .8527 |
| Average | .7731 | .8924 | .7561 | .8830 | .7605 | .8694 | .7402 | .8462 | .7710 | .8910 | .7562 | .8825 | .7621 | .8694 | .7423 | .8459 |
| $\Delta$ | - | - | - | - | - | - | - | - | **-.0021** | **-.0014** | **.0001** | **-.0004** | **.0016** | **0** | **.0021** | **-.0002** |

Table 3.3 – VSRS-1DFast and Wf evaluation for all QPs with SSIM$_{hist}$ and SSIM$_{epas}$.

| **Sequence** | VSRS-1DFast | | | | | | | | P+Badapt | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| QPs | 25 | | 30 | | 35 | | 40 | | 25 | | 30 | | 35 | | 40 | |
| Method | hist | epas | hist | epas | hist | epas | hist | epas | hist | epas | hist | epas | hist | epas | hist | epas |
| Balloons | .8615 | .9138 | .8585 | .9080 | .8624 | .8943 | .8415 | .8697 | .8702 | .9195 | .8667 | .9138 | .8692 | .9008 | .8480 | .8771 |
| Kendo | .8844 | .9238 | .8779 | .9159 | .8611 | .9035 | .8423 | .8842 | .8789 | .9237 | .8743 | .9166 | .8583 | .9055 | .8402 | .8879 |
| NewspaperCC | .6073 | .8265 | .5514 | .8121 | .5876 | .7996 | .5955 | .7749 | .6245 | .8366 | .5735 | .8254 | .6062 | .8134 | .6143 | .7891 |
| PoznanHall2 | .7494 | .9077 | .7312 | .9034 | .7230 | .8967 | .7023 | .8812 | .7630 | .9122 | .7478 | .9087 | .7389 | .9021 | .7163 | .8876 |
| Average | .7756 | .8929 | .7548 | .8849 | .7585 | .8735 | .7454 | .8525 | .7842 | .8980 | .7656 | .8911 | .7682 | .8805 | .7547 | .8604 |
| $\Delta$ | - | - | - | - | - | - | - | - | **.0085** | **.0051** | **.0108** | **.0063** | **.0096** | **.0069** | **.0093** | **.0079** |

Table 3.4 – VSRS-1DFast and P+Badapt evaluation for all QPs with SSIM$_{hist}$ and SSIM$_{epas}$.

focuses on high synthesis errors which are most likely caused by object boundary distortions, as discussed in Section 3.1. We can conclude that P+Badapt improves the synthesis from a structural point of view. This is also visible in Figure2.13 in Section 2.5.3, where P+Badapt shows noticeable improvements on object edges.

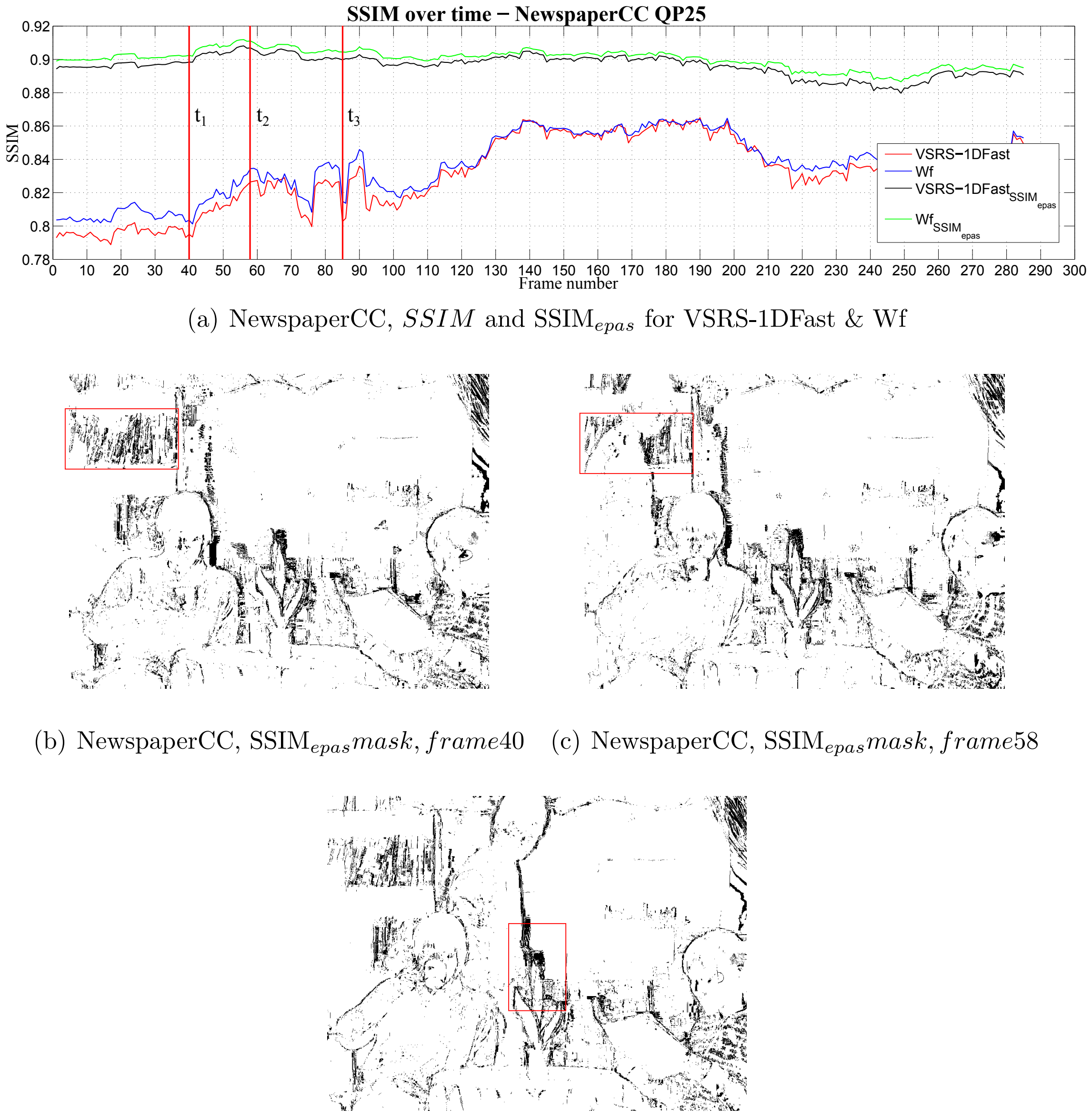
Another interesting aspect is the behavior of $\Delta$SSIM$_{hist}$ and $\Delta$SSIM$_{epas}$ across QPs. While the $\Delta$SSIM and $\Delta$PSNR report similar values across QPs, we can see that in Table 3.3 SSIM$_{hist}$ has a tendency to increase at lower QPs. This behavior can be explained by the threshold selection process described in 3.2.2.2. As the QP increases the quantization errors are in turn increased and they become closer to the high synthesis errors. Thus, the evaluated areas may contain more artifacts caused by quantization, reflecting the better overall warping precision of Wf over VSRS-1DFast and masking the structural distortions. However, definitive conclusions should be drawn from a more thorough evaluation of SSIM$_{hist}$ and SSIM$_{epas}$, using smaller QP steps.

Furthermore, in Fig. 3.3(a) we show the behavior over time of SSIM and our proposed evaluation technique. While SSIM score is relatively similar across frames, variations can be observed for SSIM$_{hist}$ and SSIM$_{epas}$ scores. Additional information about a methods strengths or weaknesses can be extrapolated from analyzing these variations. Let us look for example at three time instances marked in Fig. 3.3(a) with vertical red lines ($t_1$, $t_2$ and $t_3$, frames 40, 58 and 85 respectively). We can clearly notice an increase in SSIM$_{epas}$ at $t_2$ in comparison to $t_1$. This is consistent with SSIM, however, it is hardly noticeable. Let us look at the SSIM$_{epas}$ masks for the two time instances in Fig. 3.3(b) and 3.3(c) to identify the cause. We can see the error prone area marked with a red square in Fig. 3.3(b). In Fig. 3.3(c) this area is obstructed by a person walking in front of it and the errors are concealed. Also, observe that $\Delta$SSIM$_{epas}$ is smaller between frames 50 and 70. This points to Wf method achieving higher quality than VSRS-1DFast in this area. Obviously when the area is obstructed the gain is reduced.

At $t_3$ we can see a sudden drop in SSIM$_{epas}$ which is not noticeable in SSIM. Looking at the selection mask we can observe the person approaching another foreground object which is identified as an error prone area by the selection mask. This is marked with a red square in Fig. 3.3(d). To better understand why we have quality loss on this frame, let us look at the texture. In Fig. 3.4 we can see the reference frame and the VSRS-1dFast and Wf synthesized frames respectively. It is easily noticeable that both methods will have new artifacts in this area at $t_3$. This type of artifact appears due to the proximity of the two objects in the foreground. The area in-between them is not visible in the left or right base views (i.e. disoccluded area). This additional information on the tested methods, in terms of structural configuration of the scene and error prone areas, cannot be easily extrapolated by using only SSIM or PSNR.

### 3.2.4 Conclusions

In this section, we presented a distortion evaluation technique for view synthesis methods based on the SSIM metric. We compute the SSIM index on areas which are prone to synthesis errors such as object boundaries and complex textures. The area selection is performed either through a separation between structural artifacts caused by synthesis and quantization errors from the encoding process of the left and right base views, or by directly selecting areas which are predicted differently by two evaluated synthesis methods. The evaluation was performed on three view synthesis methods and four multiview sequences, using 3D-HEVC encoding at four QPs 25, 30, 35 and 40, against PSNR and SSIM results. The proposed technique was shown to provide a better differentiation between synthesis methods. Also, additional information can be extrapolated about the scene structure and spatial positioning of artifacts, while providing a good indication of the impact of synthesis errors.

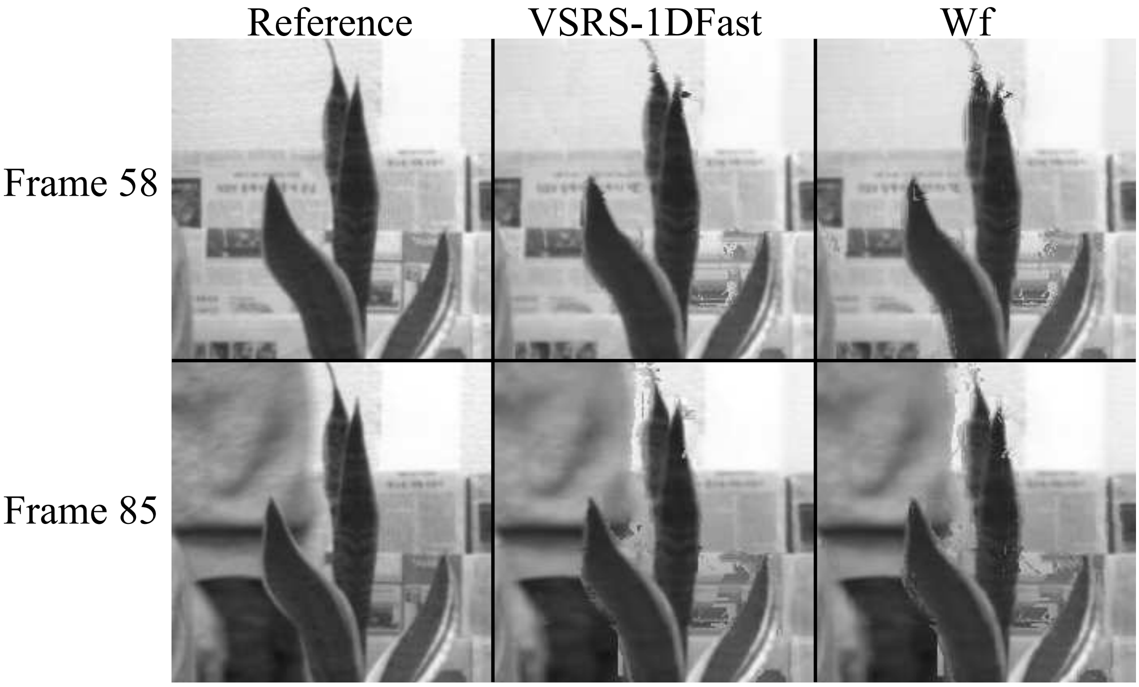


(a) NewspaperCC, $SSIM$ and SSIM$_{epas}$ for VSRS-1DFast & Wf

(b) NewspaperCC, SSIM$_{epas}mask, frame40$ (c) NewspaperCC, SSIM$_{epas}mask, frame58$

(d) NewspaperCC, SSIM$_{epas}$ mask, frame 85

Figure 3.3 – Figure 3.3(a) - SSIM and SSIM$_{epas}$ over time. Figures 3.3(b), 3.3(c) and 3.3(d) show the selection masks for SSIM$_{epas}$.



(a) NewspaperCC details

Figure 3.4 – Details of NewspaperCC sequence corresponding to time indexes: t2 and t3 in Figure 3.3(a).

## 3.3 Towards a region of interest evaluation

### 3.3.1 Comparing multiple synthesis methods

As discussed in Sec. 3.1 synthesized videos can have multiple types of artifacts which affect the quality of the image in different ways. DIBR synthesis methods compute pixel disparity, from depth map sequences, and then warp the images from the reference view into a new view. Depth maps are usually stored as video sequences and the values are inversely quantized to 256 levels with respect to real scene depth. Because depth maps are subjected to distortions from the acquisition device or transmission systems, the synthesized image can be subjected to geometrical distortion of foreground objects and also poor reproduction of complex textures. As noted in the previous section and by other studies [BPc$^+$11] [YHFK08], traditional metrics such as PSNR or SSIM may not be the best way to asses the quality of synthesized images. This behavior can be explained by the strong correlation between scene geometry and position of highly distorted areas.

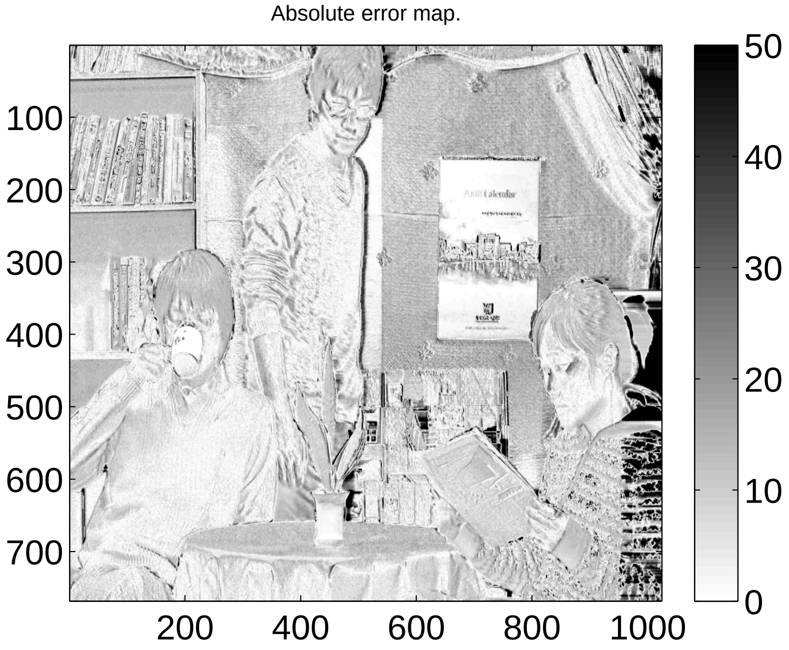


Figure 3.5 – Absolute error gray scale map for frame 93 of Newspaper sequence. View 6 synthesized from view 4 using [Feh03].

Figure 3.5 depicts a gray scale representation of the absolute errors of frame 93 of Newspaper sequence synthesized using [Feh03]. Black indicates an absolute error higher than 50 while white represents an absolute error of 0. It is easily noticeable that the absolute errors are not uniformly distributed throughout the image and are concentrated in certain critical areas. In this example view 6 was synthesized from view 4. We can see a large concentration of high errors on the left side of the image. This is consistent to a border disocclusion which was filled with an inpainting algorithm. Furthermore, highest errors are concentrated around foreground objects and there exists a high correlation between scene geometry and high distortions. Areas that have the same depth and uniform textures are usually represented without distortions, while foreground object edges and more complex textures have a high distortion. Also, we can notice that not all contours are equally distorted. In this example right most edges of objects tend to have a higher distortion. This behavior can be attributed to the direction of the synthesis from view 4 to view 6, which results in holes on one side of the foreground objects. This type of spatial error distribution is usually similar in most DIBR methods. Because of this, using a ROI when evaluating the quality of synthesis methods may provide a better indication of a method's performance as shown in Section 3.2.

Given the goal of comparing multiple synthesis methods the ROI can be selected as discussed in Section 3.1 by thresholding the absolute error or analyzing contours. Another possibility which may provide good results is to look at areas that are rendered differently by the methods which we want to compare (see Section 3.2.2.3). This is a reasonable assumption as background areas with non complex texture are usually identical in most synthesis methods and do not affect the quality of the image. Also areas that are rendered identically by multiple methods do not provide any differentiation between the tested DIBR algorithms.

Consider a number of distorted images $\mathbf{I_1^d}, \mathbf{I_2^d}, .., \mathbf{I_n^d}$. Each image is a synthesis of the same view using the same reference and one of $n$ methods. We define $\mathcal{P}$ as:

$$\mathcal{P}(x,y) = \mathrm{std}([\mathbf{I_1^d}(x,y), \mathbf{I_2^d}(x,y), .., \mathbf{I_n^d}(x,y)]) \tag{3.13}$$

where $(x,y)$ denotes a position in the image and std is the standard deviation.

The binary mask of the ROI can be expressed as:

$$B(x,y) = \begin{cases} 1 & \text{if } \mathcal{P}(x,y) > \tau \cdot \mathrm{mean}(\mathcal{P}) \\ 0 & \text{if otherwise} \end{cases} \tag{3.14}$$

where $\tau$ is a coefficient used to balance the selection and mean is the average value of $\mathcal{P}$.

As the ground truth is also available when computing the ROI, it is possible to include it in the computation. Including the ground truth in the computation doesn't provide useful information for differentiating the methods. However, it may lead to a more balanced selection of critical areas by taking into account not only regions which differ in the tested methods but also regions that have a relatively high distortion in all methods. This way, the score will also reflect the global quality of a synthesized image instead of only with respect to the tested methods.

$$\mathcal{P}(x, y) = \text{std}([\mathbf{I^d_1}(x, y), .., \mathbf{I^d_n}(x, y), \mathbf{I^r_1}(x, y), .., \mathbf{I^r_m}(x, y)]) \tag{3.15}$$

where $I^r$ is the reference used to compute the metric and $m$ is the number of times we add the ground truth. Due to a variable number of methods that can be evaluated in parallel, the ground truth needs to be weighted. In our experiments we used a weight of 1/6 (i.e. the ground truth was added once). However, in this case, the mask will have a lot of noise in the form of localized pixels selected for evaluation. Because the artifacts depend on the structure of the scene it is best to remove single pixels and also consider pixels on the edge of critical areas. This can be achieved by performing an erosion and dilation operation on the binary mask. In order to extend the initial ROI, the dilation operation should use a larger morphological structuring element. In our tests we used a $2 \times 2$ square element for the erosion and a $7 \times 7$ square element for the dilation. This values were empirically found to provide good results.

Note that the approach presented in Section 3.2.2.2 can also be extended in a similar manner for comparing multiple synthesis methods.

### 3.3.2 Subjective evaluation database used in our experiments

In order to validate this technique we use a view synthesis subjective evaluation database available at [dat]. The tests were performed using Absolute Categorical Rating with Hidden Reference Removal (ACR-HR) [ACR97] with 32 subjects. The evaluation was performed on synthesized views using a 2D display, rather than showing both the synthesis and reference on a stereoscopic 3D display. Three multiview video sequences were used: Book arrival, Lovebird, Newspaper. Sequence details are reported in Table 3.5. For each sequence there are three views used in the experiments: a left, center and right view indicated in Table 3.5. Four synthesized views are generated for each sequence: left→right, right→left, left→center, right→center. The reference views are original uncompressed. Each synthesis is then performed using the seven methods described below:

**A1:** based on [Feh04]. Depth map preprocessed by a low pass filter, borders are

| **Sequence** | **Resolution** | **Frames per second** | **Number of frames** | Views |
|---|---|---|---|---|
| Book arrival | 1024 × 768 | 15 | 100 | 8 9 10 |
| Lovebird | 1024 × 768 | 30 | 150 | 6 7 8 |
| Newspaper | 1024 × 768 | 30 | 200 | 4 5 6 |

Table 3.5 – Sequences used in our experiments

cropped and the image is resized to the original resolution.

**A2:** based on [Feh04] with inpainting algorithm proposed by Telea [Tel04]

**A3:** Tanimoto *et al.* [TFK$^+$], View Synthesis Reference Software (VSRS).

**A4:** Muller *et al.* [MSD$^+$08], depth aided inpainting

**A5:** Ndjiki-Nya *et al.* [NNKD$^+$11], hole-filling using a patch-based texture synthesis.

**A6:** Koppel *et al.* [KNND$^+$10], synthesis is improved in disoccluded areas using depth temporal information

**A7:** the disoccluded areas are not filled

Additional details on the database and an extensive study can be found in [BLCMP12].

### 3.3.3 Experimental Results

In this section we report our findings using the ROI evaluation technique described in Section 3.3.1 and use the subjective evaluation database to validate the results. The first part of this section describes the testing methodology while the results are presented in the second part.

#### 3.3.3.1 Testing methodology

In order to validate the results obtained with the proposed technique we want to evaluate all sequences and views, synthesized with each method. However, as the authors of [dat] also notice there are some outliers in the methods. Method **A1** has the highest scores in the subjective tests while all objective metrics indicate this method is by far the worst. This is due to the method not using any inpainting algorithms to fill the disoccluded areas. The borders are cropped and the image is rescaled. The non-border disocclusions are avoided by performing a low-pass filtering of the depth map. While the final result is an image with no localized impactful artifacts, it cannot be used for 3D viewing, as the geometry of the scene no longer corresponds to the reference. These results also point out to the subjects inclination to notice localized artifacts more easily than a global change in the frame which further motivates the use of ROI evalution in synthesis methods. Since we analyze view synthesis for its capability of producing 3D content, we will not use this method in our results.

In our tests we use three quality evaluation metrics: Structural SIMilarity index (SSIM) [WBSS04], Peak-Signal-to-Noise-Ratio (PSNR) and Multi-scale SSIM (MSSIM) [WSB04]. For each metric we apply the region of interest we described in Section 3.3.1 and the one proposed by Bosc *et al.* in [BPc$^+$11]. For our method we use multiple variants: proposed mask (P) without erosion/dilation (e/d) or ground truth (GT); P with e/d and P with both e/d and GT. To measure the performance of each metric we compute the average values across frames for each sequence/view/method ($3 \times 4 \times 6$). In [BPc$^+$11] the authors selected four critical points (subjective vs objective results) to evaluate the method. Our tests will be performed on all points using the Difference Mean Opinion Score (DMOS). The performance indicators we use are Pearson Correlation Coefficient (PCC), Spearman's Rank Order Correlation Coefficient (SROCC) and the Root-Mean-Squared-Error (RMSE). Before computing the PCC we will perform a fitting of the results using the recommended nonlinear function from VQEG Phase I final report [Vid00]:

$$Y = \beta_2 + \frac{\beta_1 - \beta_2}{1 + \exp^{-\frac{X - \beta_3}{|\beta_4|}}} \tag{3.16}$$

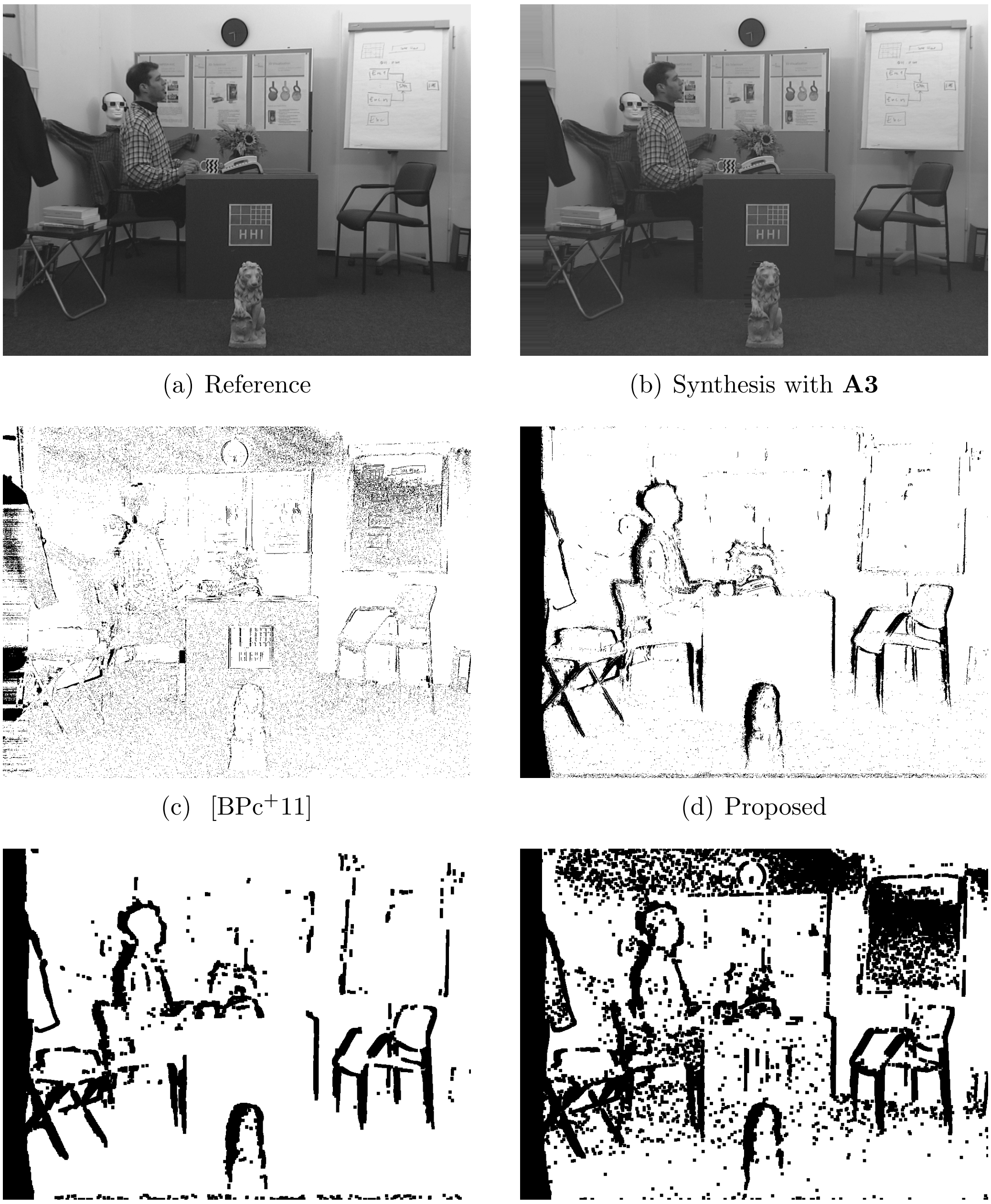
where $\beta_1$, $\beta_2$, $\beta_3$, $\beta_4$ are parameters, $Y$ are the predicted values and $X$ are the objective results.

#### 3.3.3.2 Results and discussion

In Figure 3.6 we show an example of masks for frame 10 of Book arrival sequence, view 10 synthesized from 8. Figures 3.6(a) and 3.6(b) show the reference and the synthesized frame with method **A3**. The filled dissoccluded areas are easy to notice on the left side of foreground objects and also on the left border of the image. An additional source of errors which is harder to notice is the slight displacement of certain textures on the foreground object w.r.t. the reference (e.g. the desk). Another source of errors is caused by a slight difference in luminance. This is common with DIBR synthesis methods. While they are able to warp objects to their new position in the virtual view, changes in luminance between views are not accounted for. While this type of distortions are not visually impactful, as they are difficult to notice, they can have an impact on the results of objective metrics and are relevant to this study.

Figures 3.6(c), 3.6(d), 3.6(e) and 3.6(f) show the binary masks for [BPc+11], P, P+e/d and P+GT+e/d, respectively. When comparing 3.6(c) and 3.6(d) we can see that our mask is less noisy and better adjusted to the scene geometry. Also, the right side of the image, which corresponds to a border disocclusion is completely selected, as opposed to [BPc+11]. Furthermore, the texture details of the desk are not selected in our mask, because this area has a uniform depth and is rendered similarly with all DIBR methods. Although there is a slight displacement which will result in high errors, they are hard to notice and are not critical in differentiating the evaluated methods. Performing the e/d operation will reduce the isolated patches/pixels selected in the map while, increasing solid areas. Finally, adding the ground truth in the mask computation will lead to an increased selection. We can notice that additional textures are selected: the desk, the white board and the area surrounding the clock. In this example, the percentages of selected pixels are: 7.5%, 11.44%, 17.21%, 33.2% for Bosc [BPc+11], P, P+e/d and P+GT+e/d, respectively. This behavior is similar on other sequences/views/methods, however, for brevity reasons we only discuss this example.

In Figure 3.7 we show the scatter plots for SSIM and ROI SSIM with the binary masks [BPc+11], P, P+e/d and P+GT+e/d, respectively. Each point represents the DMOS against the average of the objective score over all frames of a sequence/view/method. An improvement can be observed when using our proposed approach. This is also reflected in the numerical results reported in Table 3.6. Our methods outperforms [BPc+11] on all test cases. When compared to the Non-ROI scores, we are able to outperform SSIM with all proposed ROIs, while P+GT+e/d show similar performance to PSNR and MSSIM. A loss is observed with PSNR-P and MSSIM-P. This behavior can be explained by the use of e/d and GT. As discussed above the masks will have a larger number of selected pixels. Also, SSIM is already computed using a pixel's neighborhood, thus performing the e/d operation will allow PSNR-P+GT+e/d to account for the original's ROI neighborhood. However, the SSIM score will decrease in this case as pixels which are further away from the ROI are evaluated. Another interesting aspect is the actual implementation for a ROI evaluation with different metrics. For MSSIM the tests were performed by rescaling the ROI. However, it is also possible to recompute the ROI using the rescaled images. Furthermore, additional metrics can be computed with respect to a ROI, though, in



(a) Reference

(b) Synthesis with **A3**

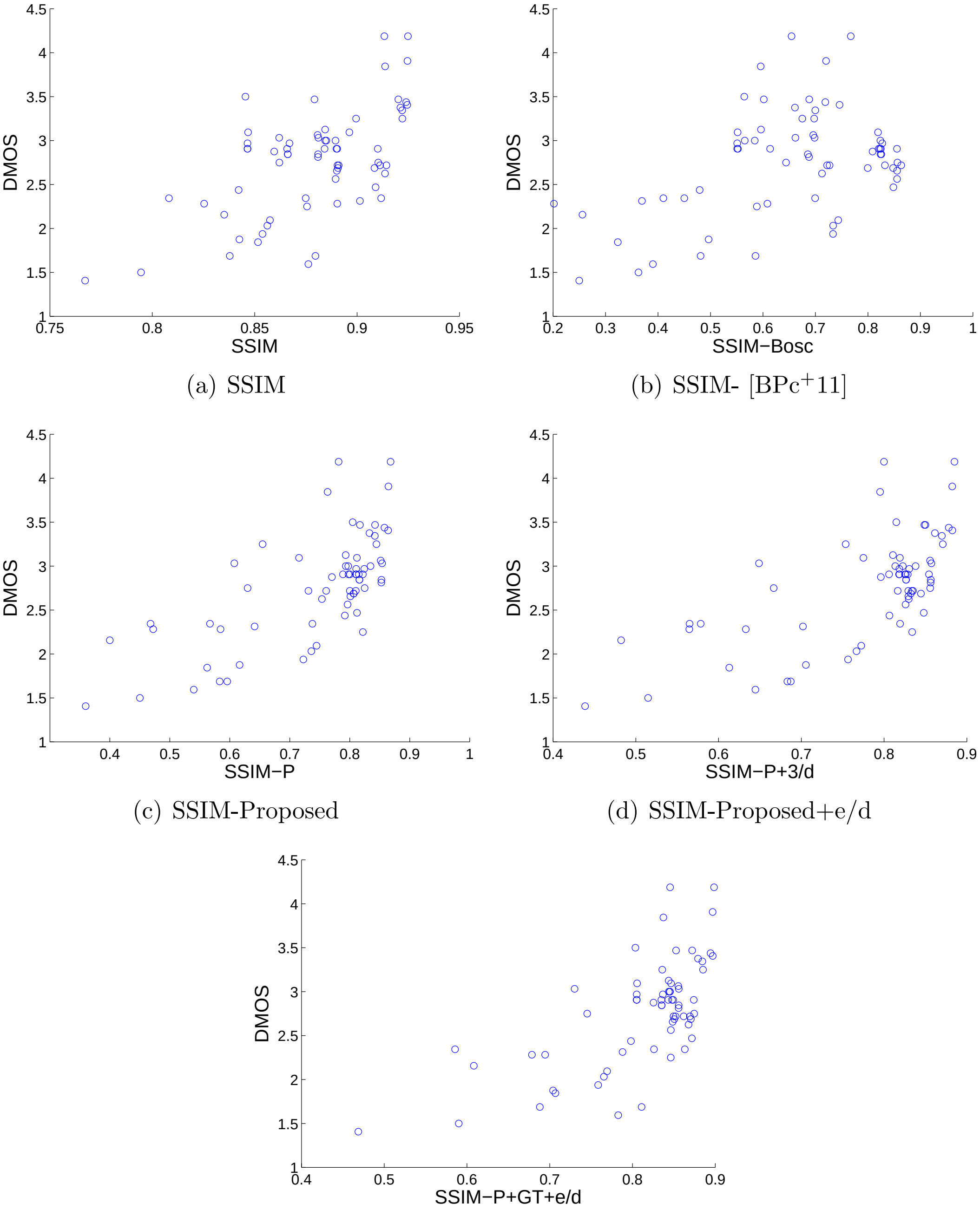
(c) [BPc+11]

(d) Proposed

(e) Proposed+e/d

(f) Proposed+GT+e/d

Figure 3.6 – Book arrival sequence view 10 synthesized from view 8 with method **A3**. Luminance frames and binary masks for the proposed methods and [BPc+11]. Black pixels are selected for evaluation.



(a) SSIM

(b) SSIM- [BPc+11]

(c) SSIM-Proposed

(d) SSIM-Proposed+e/d

(e) SSIM-Proposed+GT+e/d

Figure 3.7 – Scatter plots of objective results for SSIM with tested ROIs. Each point is the DMOS against the average objective score over all frames for a sequence, synthesis and method.

| **Metric** | Non-ROI | | | [2] | | | P | | | P+e/d | | | P+GT+e/d | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PCC | SROCC | RMSE | PCC | SROCC | RMSE | PCC | SROCC | RMSE | PCC | SROCC | RMSE | PCC | SROCC | RMSE |
| SSIM | 60.85 | 49.94 | 47.16 | 61.29 | 58.64 | 47.08 | **70.18** | **65.28** | **42.34** | 69.00 | 56.63 | 43.02 | 68.88 | 55.46 | 43.09 |
| PSNR | **85.97** | 77.57 | **30.36** | 68.52 | 32.55 | 43.29 | 71.66 | 67.18 | 41.45 | 74.31 | 68.32 | 39.77 | 82.26 | **79.20** | 33.79 |
| MSSIM | **80.10** | 65.89 | **35.58** | 68.67 | 38.35 | 43.21 | 73.86 | **70.69** | 40.07 | 72.11 | 67.4 | 41.18 | 77.18 | 67.81 | 37.79 |

Table 3.6 – PCC, SROCC and RMSE for Non-ROI, [BPc+11] and our proposed methods using SSIM, PSNR and MSSIM

the case of perceptual based metrics the way to perform such an evaluation becomes more difficult.

### 3.3.4 Conclusions

In this section we presented a study on the use of ROI in the evaluation of DIBR based synthesis methods. We extended the ideas presented in Section 3.2 and proposed a ROI generation method that can be used with traditional metrics, such as SSIM, PSNR and MSSIM. The technique was validated using a publicly available subjective evaluation database, for view synthesis methods, and showed to improve the objective results of SSIM, while maintaining similar results for PSNR and MSSIM when compared to subjective scores. However, it should be noted that most quality evaluation methods showed a rather low correlation with subjective results. Furthermore, as discussed in Section 3.3.3.1, significant inconsistencies can be identified between subjective and objective results for synthesized views. Future directions may include finding a better threshold for the ROI selection by taking into account perceptual aspects or finding ways to use a ROI for perceptual metrics. Another study direction is to perform extensive subjective tests for view synthesis using more methods and also encoded reference views and depth maps.

## 3.4 Summary

In this chapter we investigated the objective evaluation of view synthesis algorithms. Firstly, two ROI generation techniques were proposed that focus the evaluation of traditional metrics on areas which are prone to errors during synthesis. Unlike other ROI based techniques, we aimed at simultaneously evaluating two methods and comparing them rather than providing an absolute measure of quality. The first ROI generation technique relies on the observation of a secondary peak in the absolute error distribution to separate between synthesis and compression errors. The second ROI map is generated by simply selecting areas which are differently predicted. This approach led to some interesting results. By emphasizing synthesis errors we were able to extrapolate additional information such as the occurrence of synthesis artifacts at certain moments of time. This work was published in [PCPP$^+$15].

Secondly, we extended these ideas to compare any number of methods and we performed a study of multiple evaluation techniques and metrics using a publicly available view synthesis subjective evaluation database. We generate a pixel level selection map based on the thresholding of standard deviation of predictions of a pixel with or without the ground truth. Erosion and dilation operations are applied to reduce single pixel occurrences. This approach was published in [PVPPD16].

While we were able to show that ROI evaluation can improve the performance of traditional metrics for synthesized images, our findings indicate that objective quality evaluation on view synthesis is still an open subject. Considering the significant inconsistencies between objective and subjective results, for some synthesis methods (e.g. methods that provide a disoclussion free image but are not geometrically consistent) we are driven to conclude that the subjective evaluation standardization for multiview video sequences and view synthesis plays a critical role. However, since a delivery method of multiview content to the general public is not yet well defined and multiple options are still being explored, subjective evaluation conditions and the role of view synthesis may change drastically in the future.

# Chapter 4

# Video quality and resolution enhancement from multi-source compressed video

## Contents

This chapter tackles the problem of video reconstruction and resolution enhancement. The scenario is similar to the situation encountered in view synthesis where we had four temporal and two inter-view predictions of a frame without knowing the reference. In this case, we are dealing with multiple compressed descriptions of a video sequence. Each description can be subjected to a certain compression level with hybrid video coders. Furthermore, the videos can have different resolutions. In order to address this, we use the key features that govern hybrid video coders and model the problem as a convex optimization one. The rest of this chapter is organized as follows. In Section 4.1 we motivate the utility of this approach and present a state of the art of SR video reconstruction techniques. Section 4.2 states the problem and presents the mathematical model. Section 4.3 describes in detail the convex optimization method based on the mathematical foundation of [CP12]. In Section 4.4, we explain the adaptations performed to enable the use of HEVC encoding. Experimental results and an extended discussion on the proposed method's performance is available in Section 4.5. Finally, Section 4.6 concludes this chapter and presents future work possibilities.

## 4.1 Introduction and state of the art of SR and video reconstruction techniques

The continuous evolution of transmission systems, storage and video compression technology in the past decade provides the end user with easy access to video content. A varied number of distribution methods exist, from the classical DVD's to online streaming on the world wide web. High resolution, studio quality video sequences are usually down-sampled and compressed in order to match the requirements of certain applications and the limitations imposed by transmission and storage technologies. Large video databases such as YouTube or Netflix provide multiple resolutions and different encodings of the same video in order to account for user bandwidths and displays. This situation creates a lot of potential for resolution enhancement and compression artifact reduction techniques from single and multiple sources.

Super-resolution (SR) algorithms are post-processing techniques that infer a spatially High Resolution (HR) estimate from one or more Low Resolution (LR) images. Currently, SR is an active research field, a review of SR algorithms is available in [Mil10] while performance comparisons can be found in [NB12], [TSK10] and [KPS+11].

In general, SR algorithms can be divided in single-frame (SF-SR) or multi-frame (MF-SR) approaches. The later exploits the motion between succesive LR frames in order to extract unique information from each representation. The problem formulation in most cases assumes a high number of descriptions (5-30) is available which are subjected to different pixel shift operations (rotations, translations), blurring and sub-sampling. Some of the most popular MF-SR algorithms rely on a Bayesian probabilistic formulation and employ various SR priors such as smoothness with Total Variation (TV) [BMK11] or the Simultaneous Auto Regressive (SAR) image model [MNCM01], $l_1$ based priors [VVB+13] or non-stationary image prior combinations [SMRA14]. This type of MF-SR approaches are best suited to tackle the problem of image acquisition, were a high number of descriptions is available with simple motion and a similar blurring.

In the recent work of Liu and Sun [LS14] the Bayesian approach is extended to videos. In this scenario, the descriptions are consecutive frames of a video sequence. As noted by the authors this problem is inherently more difficult as real world videos have complex motion rather than a simple parametric form. This work proposes a practical SR framework where optical flow [HS81], blur kernel [KH96] and noise levels [LSK+08] are simultaneously estimated. Gains of up to 3 dBs are reported over bicubic up-sampling, when super-resolving using 15 forward and backward frames. The degradation was synthetically added as Gaussian blurring, sub-sampling and Gaussian white noise; tests were performed on real world video sequences. However, as reported by the authors it can take 2 hours to super-resolve one frame. In the work of Ma *et al.* [MJW15], motion blur is taken into consideration and improvements over [LS14] are reported on real world sequences in a similar set-up where 30 frames are used in the computation.

As discussed in the beginning of this section, videos nowadays are mostly available in compressed form. Segall *et al.* investigate the problem of SR on compressed video in [SKMM02]. They show that compression artifacts complicate the SR reconstruction and suggest that a model of compression should be employed. In [GAM04], a Bayesian maximum a posteriori probability formulation is proposed that takes into account the quantization in frequency domain. The method is shown to provide improvements over spatial domain methods on frequency quantized images, when exact motion information is known. Wang *et al.* [WYT09] tackle the problem of compressed video enhancement from a different perspective. The authors propose a practical framework that enhances the quality of video by combining different encodings of the same sequence. In this scenario real world videos are encoded using MPEG-2 in two configurations. The proposed algorithm is able to combine the two decoded videos. Gains of up to 1.5 dB are reported, however, no resolution enhancement is performed.

In the recent work of Kappeler *et al.* [KYDK16], compressed video SR is achieved by means of Convolutional Neural Networks. They perform an extensive test of SR algorithms on a real world sequence (Myanmar at 960 × 540 resolution). The LR descriptions are created with ffmpeg and then compressed with MPEG's H.264/AVC encoder (4 different compression levels were tested). The proposed method shows gains of up to 4 dB over bicubic interpolation, albeit the algorithm was trained on the same sequence and 14 hour were needed for 3 frame input training and up to 1 min to super resolve due to the motion compensation (motion information was computed before training with optical flow). Furthermore, the tests showed that one of the best performing methods out of multiple SF-SR and MF-SR including [LSK$^+$08] [MJW15] is the exemplar based learning method of Timofte *et al.* [TSG15].

In this chapter we extend the ideas in [GPPC13] and build a convex optimization approach adapted to video SR. We propose a practical framework that is able to reconstruct or enhance a HR video sequence from multiple video sequences encoded with any Hybrid Video Coder. We model the down-sampling process to account for polyphase filters. Each description can be encoded with its own encoder and subjected to a different down-sampling method. The model accounts for the particularities of video compression and can be adapted to any hybrid Video Coder (VC). The minimization process is performed using a modern and efficient proximal dual-splitting algorithm [CP12] that leaves room for parallel implementations.

The effectiveness of the method is shown on multiple video sequences encoded with HEVC. A generic Matlab implementation of a VC is used to perform preliminary tests on specific scenarios. Gains of up to 6 dBs can be obtained over bicubic up-sampling and 3 dBs over [TSG15]. Furthermore, the method can be used to improve a HR compressed sequence from a LR one. The proposed framework can be used as a refinement method on the output of other SR techniques.

## 4.2 Modeling the SR problem in the compressed domain

### 4.2.1 Problem statement

We depict a model of the super resolution problem in Fig. 4.1. Starting from an original video sequence, we apply different degradation models which consist of sub-sampling ($L$) and compressing the source with a VC. Four essential operations are traditionally involved in a video coder: prediction, transform, quantization and entropy coding. The prediction step allows efficient compression of the redundancies present in the source signal. Then, a linear transform aims at further reducing the correlations in the residual signal and compacts the energy in a limited number of coefficients. The transform coefficients are mapped to a finite countable set of codewords during quantization. Finally, entropy coding exploits the remaining statistical redundancies in the resulting codewords and generates the binary representation of the video signal. In the remaining of this section, we further detail these essential building blocks and formalize a sub-sampling and compression model that will be used as an anchor for the proposed SR approach.

#### 4.2.1.1 From pixels to transform coefficients

Since our work features SR from multiple observations, let us consider a set of $M$ encoded data streams - providing views of the same scene - stemming from different video coders. Each video coder has its own configuration (resolution, bitrate, etc.).

We denote by $\overline{x} = [\overline{x_1}, ..., \overline{x_K}]$, with $\forall i \in [1, K]$, $\overline{x_i} \in \mathbb{R}^N$ the original high resolution (HR) sequence. In a compression scheme with no prediction, for every $m \in \{1, \ldots, M\}$, the $m$-th coder generates a vector of coefficients $z_{m,i} \in \mathbb{R}^{P_{m,i}}$ which corresponds to a quantized version of the output of a linear transform $T_{m,i}$ applied to $L_{m,i}\overline{x_i}$. More specifically we have $\forall i \in [1, K]$ :

$$\overline{y}_{m,i} = T_{m,i} L_{m,i} \overline{x_i} \tag{4.1}$$

$$z_{m,i} = \mathcal{Q}_{m,i}(\overline{y}_{m,i}) \tag{4.2}$$

where $\mathcal{Q}_{m,i}$ is the vector quantizer employed by the $m$-th encoder for image $i$.

The above formulation does not account for the hybrid nature of video coders. Indeed, video coders do not apply directly the transforms to pixel blocks, but to a residual obtained by differentiating the observation with a prediction. We therefor

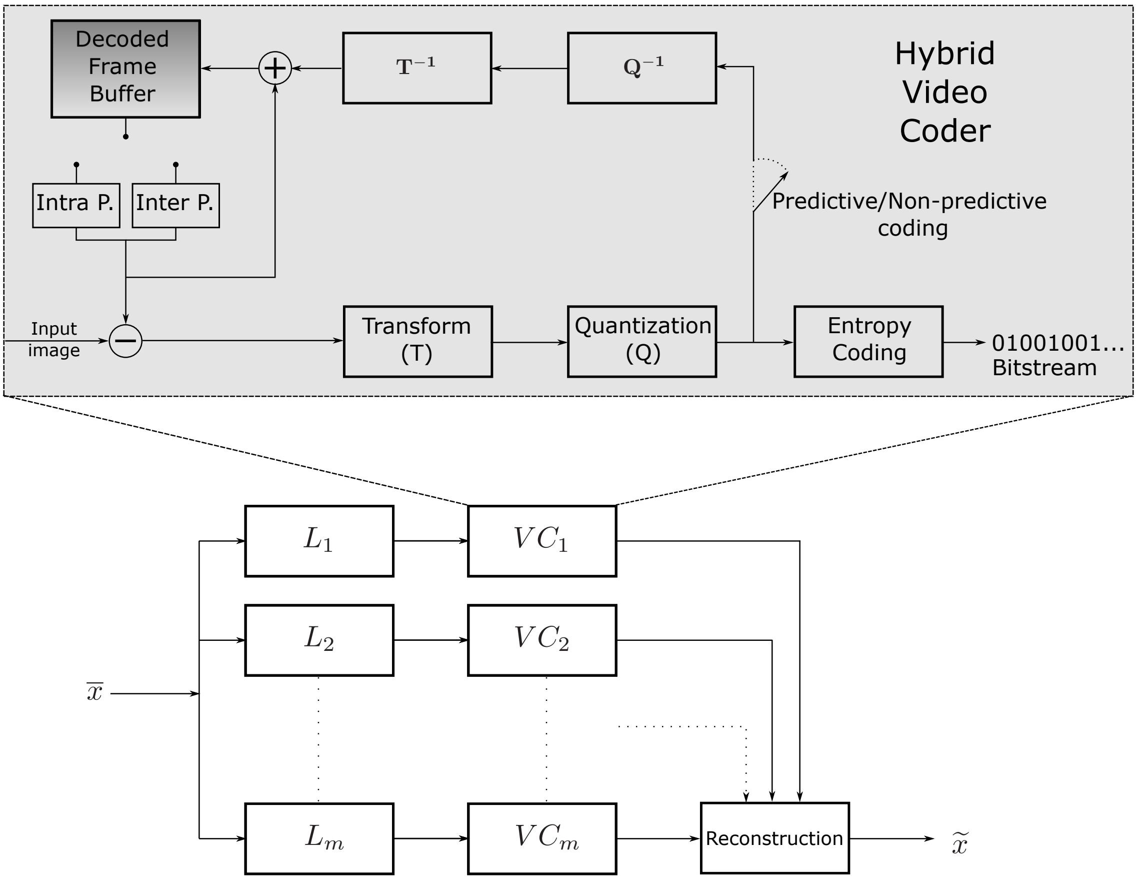


Figure 4.1 – A generic model for multiple video sources sub-sampling, compression and reconstruction.

denote by $\widetilde{x_{m,i}}$ the predicted image of the $m$-th encoder for image $i$. Eq. (4.1) can thus be rewritten as :

$$\overline{y}_{m,i} = T_{m,i}(L_{m,i}\overline{x_i} - \widetilde{x_{m,i}}) \tag{4.3}$$

Obtaining the predicted image $\widetilde{x_{m,i}}$ typically depends on the video coder used, and more details about its computation will be given later on in this section.

#### 4.2.1.2 Modeling the quantization process

We further detail the quantization process in this section to introduce useful notions and notations. For the sake of clarity, we voluntarily remove indexes related to the coder and the image being processed ($m$ and $i$ in the previous section). Let us assume that $\mathcal{Q}$ performs a scalar quantization with $n_Q$ quantization levels $r_1, \ldots, r_{n_Q}$ and decisions levels $d_0, \ldots, d_{n_Q}$ such that $d_0 < \cdots < d_{n_Q}$ as shown in Figure 4.2. With

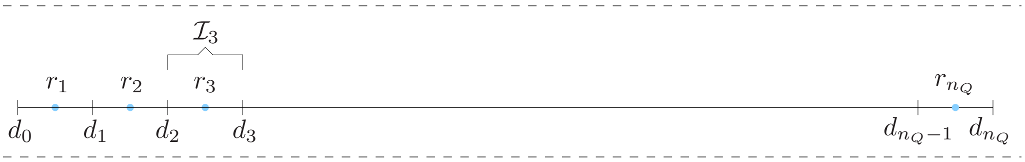


Figure 4.2 – Quantization model: interval limits and reconstruction values.

these notations, the relation between a quantized coefficient $z^{(k)}$ and the original coefficient $\overline{y}^{(k)}$ follow the subsequent quantization rule:

$$\forall i \in [1, n_Q] \qquad z^{(k)} = r_i \;\; \Leftrightarrow \;\; \overline{y}^{(k)} \in \mathcal{I}_i \tag{4.4}$$

where $\mathcal{I}_i$ is the interval defined as

$$\mathcal{I}_i = \begin{cases} [d_{i-1}, d_i[ & \text{if } i < n_Q \\ [d_{n_Q-1}, d_{n_Q}] & \text{if } i = n_Q. \end{cases} \tag{4.5}$$

We now denote by $(i^{(k)})_{1 \leq k \leq P}$ the quantization index selected for $z^{(k)}$. Then it can be deduced that $\overline{y}$ belongs to the following closed convex set

$$C = \left\{ y = (y^{(k)})_{1 \leq k \leq P} \in \mathbb{R}^P | \; (\forall k \in \{1, \dots, P\}) \;\; d_{i^k-1} \leq y^{(k)} \leq d_{i^{(k)}} \right\} \tag{4.6}$$

Note that the closure of $\mathcal{I}_{i^{(k)}}$ instead of $\mathcal{I}_{i^{(k)}}$ itself has been considered in order to make $C$ closed (*i.e.* $d_{i^{(k)}} \in C$). The projection onto $C$ can then be straightforwardly defined:

$$\left(\forall y = (y^{(k)})_{1 \leq k \leq P} \in \mathbb{R}^P\right), \mathcal{P}_C(y) = (p^{(k)})_{1 \leq k \leq P} \tag{4.7}$$

where $(\forall k \in \{1, \dots, P\})$

$$p^{(k)} = \begin{cases} d_{i^{(k)}-1} & \text{if } y^{(k)} < d_{i^{(k)}-1} \\ d_{i^{(k)}} & \text{if } y^{(k)} > d_{i^{(k)}} \\ y^{(k)} & \text{otherwise.} \end{cases} \tag{4.8}$$

#### 4.2.1.3 Modeling the re-sampling process

In the present work, $L_{m,i}$ corresponds to a sub-sampling process (with or without some prefiltering), but it could also account for a registration error, after some suitable linearization. Thus, we model the sub-sampling process to account for the most common methods used in video coding: a polyphase filter followed by a decimation.

This model accounts for the subsampling procedure used in the scalable video coding extensions of H.264/AVC (SVC [WSR+07]) or HEVC (SHVC [BYCR15]) video standards. In the following formulation we use the Fourier transform to express the frequency response of a filter $\widetilde{l}[r]$ :

$$\widetilde{L}(f) = \sum_{r=0}^{R-1} \widetilde{l}[r] e^{-i2\pi fr} \tag{4.9}$$

where $R$ is the kernel size of the filter. Note that other transforms can be used depending on the coding method that is modeled. For example wavelet transforms can be used to model JPEG 2000 compression [ISO00]. If we consider the polyphase components of the filter as:

$$\widetilde{e_q}[p] = \widetilde{l}[pQ + q] \tag{4.10}$$

where $Q$ is the number of phases or components, the filter can now be expressed as a sum of phase components as:

$$\widetilde{L}(f) = \sum_{q=0}^{Q-1} \sum_{p=0}^{P-1} \widetilde{e_q}(p) e^{-i2\pi f(pQ+q)} \tag{4.11}$$

where $P$ is the number of taps for each phase (*i.e.* the kernel size of a single phase filter). Each phase ($L_q$) can easily be obtained by fixing $q$ in the above equation and summing over $p$. For convenience the above formulation assumes that $R$ is a multiple of $Q$, if not, $\widetilde{l}[r]$ can be extended by zero-padding. Using this formulation, we can compute any number of phases and filter an image. However, in order to obtain the downsampled version of an image, we want to combine only certain phases and decimate them with respect to the downsampling scale.

In Fig. 4.3, we depict a simple example of downsampling and upsampling with a factor of 1/2 and 2 respectively. Here, $x$ represents four adjacent pixels in an image row. $U$ denotes an image expansion with zeros, while $D$ is a decimation. More precisely, the downsampling process uses only 1 phase (0.5), thus, the image is expanded by a factor of 2. The same operation will also be applied on the filter in order to match the zero values in the image. Once the filter is applied, decimation is used to extract the pixels at positions 1.5 and 3.5. Note that the decimation process needs to account for both the phase decimation and the initial expansion of the image. The low resolution (LR) representation is denoted by $y$, while the downsampling operator $L$ is defined as:

$$L(x) = D_L(U_L(\widetilde{e}_{q_2}) * U_L(x)) \tag{4.12}$$

The up-sampling process defined by $H$ follows the same logic and is defined as:

$$H(x) = D_H(U_H(\widetilde{e}_{q_1,q_3}) * (U_H(y))) \tag{4.13}$$

However, in this case two phases are involved $q_1$ and $q_3$. The image is expanded with zeros by a factor of 3 and the filter $\widetilde{e}_{q_1,q_2}$ is defined as:

$$\widetilde{e}_{q_1,q_2} = [w_1^{q_2}, w_1^{q_1}, w_2^{q_2}, w_2^{q_1} ..., w_T^{q_2}, w_T^{q_1}] \tag{4.14}$$

In this case the filter is expanded by inserting one zero value in-between consecutive pairs $w_t^{q_2}, w_t^{q_1}$ such that phases 1 and 3 can be computed in a single application of the filter. The decimation process will be used to remove the original pixels of $y$. We aim to perform a single matrix multiplication in the transform domain in order to easily model the adjoint operator and reduce computational time. Thus, the final filter will be a 2-D version of the current one and the adjoint operator required for our solver is easily expressed as the Hermitian transpose (*i.e.* the complex conjugate of the transpose). The weights are determined using any popular interpolation method,

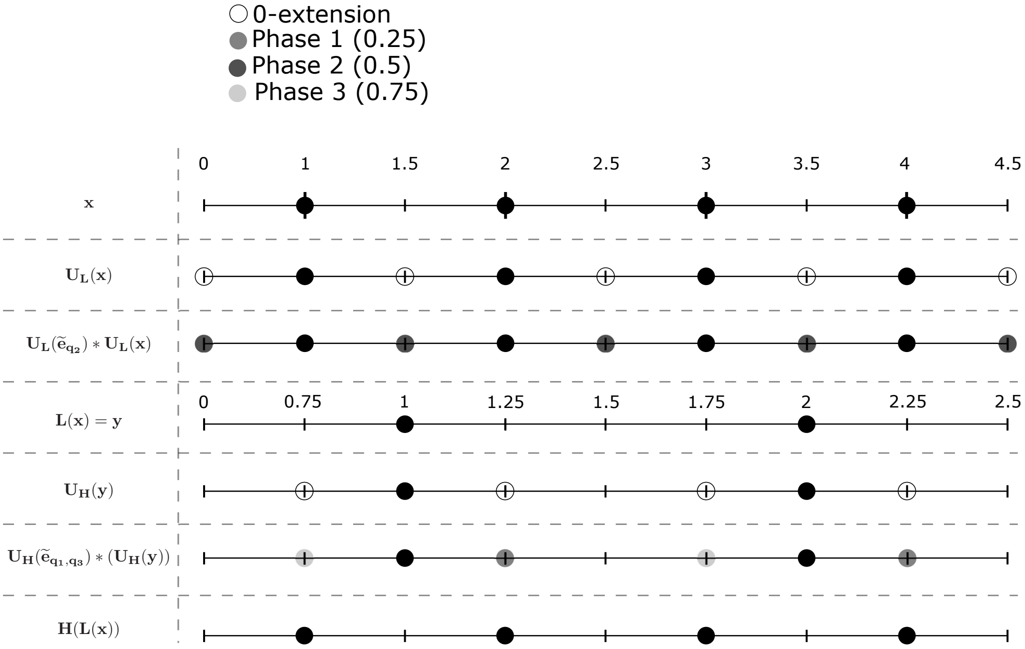


Figure 4.3 – Down-sampling and up-sampling operators.

such as: Lanczos resampling [TG90], Bicubic [Key81] or filters proposed for SVC [WSR+07] or SHVC [CBYH15].

## 4.2.2 Modeling the SR process

### 4.2.2.1 A data-fidelity measure in the compressed domain

We propose to evaluate the fidelity of an observation in the transform domain. In absence of additional clues, reconstruction levels represent the best quality reference (which minimizes the error) for the solution in each transform domain. We opt for the reasonable choice of minimizing the sum of distances between the projections of the sub-sampled solution onto the transform bases and the corresponding quantized transforms observed in the compressed bitstream, according to a suitable metric $\phi_m$. Eventually, to account for the unequal reliability of the reconstruction levels for each encoded version, we use an additional parameter $\alpha_m$ :

$$J_{\mathrm{DF}}(\mathbf{x}) = \sum_{i=1}^{K} \sum_{m=1}^{M} \alpha_m \phi_m \left( T_{m,i}(L_{m,i}\widehat{x}_i - \widetilde{x_{m,i}}) - z_{m,i} \right). \tag{4.15}$$

### 4.2.2.2 Exploitation of available data

The above objective function measures a distance to reconstruction levels in the compressed domain. We propose to strengthen the modeling of the SR problem using all available information in the compressed bitstream. In particular, we know the reconstruction levels and the associated quantization intervals for each quantized coefficient in the bitstream. Since quantization constraints are in the form of a closed convex set $C_{m,i}$ (See Eq. 4.6), the latter constraints can be directly used in the formulation of the optimization problem. Therefor we enforce the following admissibility condition to the solution:

$$\text{Find } \widehat{x} : \forall m \in [1, M], \forall i \in [1, K],\ T_{m,i}(L_{m,i}\widehat{x}_i - \widetilde{x}_i) \in C_{m,i} \tag{4.16}$$

### 4.2.2.3 *A Priori* knowledge

Encompassing *a priori* information into the reconstruction problem is a common choice in the literature. We first enforce the solution to have pixel values belonging to a specific *range*, typically known given the application domain. This condition can be expressed as follows:

$$\text{Find } \widehat{x} : \forall i \in [1, N], \forall m \in [1, M], x_{\min}^{(m,i)} \leq \widehat{x}^{(i)} \leq x_{\max}^{(m,i)} \,. \tag{4.17}$$

Moreover, a typically adopted choice is to enforce the smoothness of the solution by limiting its discontinuities according to a suitable metric. This is necessary in order to deal with the noise and artifacts introduced at the compression stage. We opt here for the classical Total Variation (TV) [CP04] to measure the discontinuity of the solution. In order to avoid over-smoothing, the TV will not be introduced in the minimization criterion, but rather limited by means of an additional constraint:

$$\text{find } \widehat{x} : \forall i \in [1, K],\ \text{TV}(\widehat{x_i}) \leq \eta_i. \tag{4.18}$$

Obviously, the choice of the bound $\eta_i$ is critical, and its computation will be detailed in the experiments section.

For the super-resolution case the sole TV constraint may be insufficient. In particular, we compute data fidelity w.r.t. the reconstruction levels only using the LR (and transformed) versions of the solution. As a matter of fact, among all the possible solutions providing the desired minimum distance, there is still no guarantee that unlikely ones will not be picked. Among them, some may be particularly noisy, in which case even the activation of the TV constraint can only lead to poor results.

To cope with this problem, we propose to balance the minimization criterion with an additional *super-resolution prior*. To this aim, let us consider a set of up-sampling operators $H_{m,i}$, which can be chosen as to optimally adapt/compensate the corresponding sub-sampling operators $L_{m,i}$. The super-resolution prior is here defined as the distance of the solution $\widehat{x}$ from its subsequently sub-sampled and up-sampled versions, according to a suitable metric per description $\psi_m$, namely:

$$J_{\text{SR}}(\mathbf{x}) = \sum_{i=1}^{K} \sum_{m=1}^{M} \psi_m \Big( (\text{Id } - H_{m,i} L_{m,i}) \widehat{x_i} \Big) \,. \tag{4.19}$$

In this way, a preference is expressed in favor of solutions which “look like” the results of proper up-sampling processes, which can be, in our case, adapted to the down-sampling counterparts that generated the observations. Thus, the choice of $H$ for this constraint is critical as it assumes that $H$ is the good solution to reverse $L$. For example in the case of bicubic down-sampling, $H$ can be easily defined as the bicubic up-sampling process as described in Section 4.2.1.3. In fact, this constraint can be interpreted as a correction of the solution w.r.t. the artifacts introduced by a subsequent application of matched up-sampling and down-sampling operators.

The proposed framework can also be used to refine the solution of another super resolution algorithm by changing the initialization. When the initialization is obtained using a combination of the linearly up-sampled observations ($H_{m,i}(obs_{m,i})$) the constraint will help in balancing the solution. However, when a more complex method is used to provide a better initialization, this constraint might not take advantage of the additional information. For instance, example based super resolution methods introduce new information not contained in the observation due to their learning process. In this case, the above constraint will not take advantage of this, as it assumes that $H$ is the “proper” way to reverse $L$ and the new information is regarded as a distortion and corrected. This problem can be overcome by defining $H_{m,i}$ as the algorithm used for generating the initialization. However, this is not a feasible solution as it would require modeling the algorithm as a linear process and can also lead to a high increase in computational time. Therefore, when using an initialization based on a complex super resolution algorithm rather than a filter based up-sampling this constraint should be disabled.

#### 4.2.2.4 Wrapping up the SR model

Based on the convex constraints and the objective functions detailed previously, let us now formally define the considered optimization problem. To this aim, we now denote $\iota_C$ the characteristic function of a closed convex set $C$, defined by:

$$\iota_C(y) = \begin{cases} 0 & \text{if } y \in C \\ +\infty & \text{otherwise.} \end{cases} \tag{4.20}$$

We propose then to minimize the following criterion, with a parameter $\beta \in [0; +\infty[$ allowing to balance the cost functions:

$$\text{Find } \widehat{x} \in \underset{x\in\mathbb{R}^{K\times N}}{\operatorname{argmin}} \Bigg( J_{\text{DF}}(x) + \beta J_{\text{SR}}(x) + \sum_{i=1}^{K}\sum_{m=1}^{M} \Big( \iota_{C_{m,i}}(T_{m,i}(L_{m,i}x_i - \widetilde{x_{m,i}}))\Big) + \sum_{i=1}^{K}\sum_{m=1}^{M} \Big( \sum_{s=1}^{S} \iota_{D_{s(m,i)}}(F_s x_i)\Big)\Bigg) \tag{4.21}$$

Note that the $F_s$ introduces the range and smoothness constraints from Eq. (4.17) and Eq. (4.18) into the problem formulation, hence in our case $S = 2$. The range constraint is directly applied to the image:

$$F_1 = \mathrm{Id}\,,$$

$$D_1(m,i) = \{x \in \mathbb{R}^N : x^{(k)} \in [x^{m,i}_{\mathrm{min}}, x^{m,i}_{\mathrm{max}}]\, \forall k \in [1, N]\} \quad (4.22)$$

For the isotropic TV-based smoothness constraint, the image gradient needs to be computed (with $\nabla_h, \nabla_v$ being the horizontal and vertical gradient operators respectively):

$$F_2 = (\nabla_h, \nabla_v), D_2(m,i) = \{x \in \mathbb{R}^N : \sum_{k=1}^{N} \sqrt{\nabla_h^2 x^{(k)} + \nabla_v^2 x^{(k)}} \leq \eta_i\}, \quad (4.23)$$

Furthermore, if frames are compressed without the use of predictive coding, as is the case of intra frames in older coders that do not employ intra-prediction, the data fidelity criterion and the quantization interval based constraint can be easily adapted by replacing $L_{m,i}x_i - \widetilde{x_{m,i}}$ with $L_{m,i}x_i$. Also, note that, $\widetilde{x_{m,i}}$ is given as a constant for each $x_i$. We could allow the prediction to vary with respect to its reference:

$$\widetilde{x_{m,i}} = M_{m,i}L_{m,i-1}\widehat{x}_{i-1} \quad (4.24)$$

where $M_{m,i}$ denotes a motion compensation operation. In the case of intra frames that use intra-prediction we would need to define a new operator that models the intra-prediction process in the video coder which was used. However, using such a formulation will introduce non linear operators which complicate the optimization problem. Furthermore, the coefficients of the residual are computed with respect to a certain prediction at the decoder side. Using a different prediction, albeit a better one, might lead to overall worse results when summing with the residual. Therefore, we recommend using a fixed prediction. This is achieved by computing it from the compressed observations before solving the problem. As such, the optimization process can be applied for each frame ($x_i$) independently and the summation over $i$ can be removed from the model.

## 4.3 A convex optimization solver

In this section, we tackle the problem of solving Eq. (4.21). As discussed in Sec. 4.2.2.4 each frame can be optimized independently. As such, for the sake of simplicity we remove the frame index coefficient $i$ in the following description. Considering our problem is based on linear operators, our choice of solver falls on the primal-dual algorithm proposed by Combettes *et al.* in [CP12], known as Monotone Lipschitz Forward-Backward-Forward (M-LFBF) algorithm. This algorithm, unlike other similar methods, assures a lower computational complexity for problems involving linear operators as it does not require any matrix inversion [CP12]. Furthermore, the block iterative structure of the algorithm allows for efficient parallel implementations on multi-core architectures.

In the following section we will further detail some properties of the proximity operators which are used in this work.

### 4.3.1 Proximity operators

We begin by defining the proximity operator [Mor65] in a real Hilbert space $\mathcal{H}$ with norm $\|\cdot\|$ for a function $\varphi \in \Gamma_0(\mathcal{H})$. Here, $\Gamma_0(\mathcal{H})$ denotes the class of proper lower semi-continuous convex functions from $\mathcal{H}$ to $]-\infty, +\infty]$. This gives the following definition:

$$\operatorname{prox}_{\varphi} \colon \mathcal{H} \to \mathcal{H} \colon u \mapsto \underset{v \in \mathcal{H}}{\operatorname{argmin}} \frac{1}{2} \|v-u\|^2 + \varphi(v). \tag{4.25}$$

A useful property which allows us to deal with the reconstructed coefficients in the transform domain $(z_m)$ states the following: If $\psi = \varphi(\cdot - v)$, where $v \in \mathcal{H}$, then

$$(\forall u \in \mathcal{H}) \qquad \operatorname{prox}_{\psi} u = v + \operatorname{prox}_{\varphi}(u-v). \tag{4.26}$$

Based on this we can compute the proximity for the data fidelity term $J_{\mathrm{DF}}(\mathbf{x})$. Let us consider $\Phi_m \overset{\triangle}{=} \phi_m(\cdot - z_m)$. As such, the data fidelity term can be expressed as $\Phi((T_m(L_m\hat{x} - \widetilde{x_m})))$ and by applying Eq. (4.26), we obtain the following expression for the proximity operator:

$$prox_{\Phi} u = z_m + prox_{\phi}(u - z_m) with \quad u \mapsto T_m(L_m\hat{x} - \widetilde{x_m}) \tag{4.27}$$

Another property of interest is the relation between the projection and proximity operators for characteristics functions of closed convex sets. If $\psi = \iota$ and $C$ is a closed convex set on $\mathcal{H}$, then:

$$(\forall u \in \mathcal{H}) \qquad \operatorname{prox}_{\psi} u = \operatorname{prox}_{\iota_C} u = \mathcal{P}_C(u), \tag{4.28}$$

### 4.3.2 Algorithm

Using the properties above, the algorithm in [CP12] can be adapted for solving the problem of Eq. (4.21). As discussed in Sec. 4.2.1 we need to account for frames which use predictive coding (intra or inter prediction) and also frames for which only transform coding is employed. As the prediction is a constant during the iterative process, we only need to compute it once. Furthermore, if the initialization differs from $H_m(obs)$ (for example a state-of-the-art SR method is used) the SR prior given by Eq. (4.19) will be disabled.

Taking all of the above into consideration will lead to the pseudo-code algorithm described in Alg. 1.

**Algorithm 1** Proposed M-LFBF based algorithm.

```
Initialization of primal and dual variables: $x_n$ and $v_{m,n}$ for $n = 0$ and $m = 1..M$
if Predictive coding then
    Compute $\widetilde{x_m}$ for $m = 1..M$
else
    Set $\widetilde{x_m} = 0$ for $m = 1..M$
end if
for $n = 0, 1, \ldots$ do
    if $H_m(obs) \quad init$ then
        $d = 3M$
        $y_{1,n} = x_n - \gamma \left( \sum_{m=1}^{M} \left( L_m^\top \left( T_m^\top v_{m,n} + \widetilde{x_m} \right) + \right. \right.$
 $\left. \left. + L_m^\top \left( T_m^\top v_{m+M,n} + \widetilde{x_m} \right) + (\mathrm{Id} - L_m^\top H_m^\top) v_{m+2M,n} \right) \right.$
 $\left. + \sum_{s=1}^{S} F_s^\top v_{s+d,n} \right)$
    else
        $d = 2M$
        $y_{1,n} = x_n - \gamma \left( \sum_{m=1}^{M} \left( L_m^\top \left( T_m^\top v_{m,n} + \widetilde{x_m} \right) + \right. \right.$
 $\left. \left. + L_m^\top \left( T_m^\top v_{m+M,n} \right) \right) + \sum_{s=1}^{S} F_s^\top v_{s+d,n} \right)$
    end if
    for $m = 1, \ldots, M$ do
        $y_{2,m,n} = v_{m,n} + \gamma T_m \left( L_m x_n - \widetilde{x_m} \right)$
        $p_{2,m,n} = y_{2,m,n} - \mathrm{prox}_{\frac{\alpha_m \Phi_m}{\gamma}} \left( \frac{y_{2,m,n}}{\gamma} \right)$
        $q_{2,m,n} = p_{2,m,n} + \gamma T_m \left( L_m y_{1,n} - \widetilde{x_m} \right)$
        $v_{m,n+1} = v_{m,n} - y_{2,m,n} + q_{2,m,n}$

        $y_{2,m+M,n} = v_{m+M,n} + \gamma T_m \left( L_m x_n - \widetilde{x_m} \right)$
        $p_{2,m+M,n} = y_{2,m+M,n} - \mathcal{P}_{C_m} y_{2,m+M,n}$
        $q_{2,m+M,n} = p_{2,m+M,n} + \gamma T_m \left( L_m y_{1,n} - \widetilde{x_m} \right)$
        $v_{m+M,n+1} = v_{m+M,n} - y_{2,m+M,n} + q_{2,m+M,n}$

        if $H_m(obs) \quad init$ then
            $y_{2,m+2M,n} = v_{m+2M,n} + \gamma(\mathrm{Id} - H_m L_m) x_n$
            $p_{2,m+2M,n} = y_{2,m+2M,n} - \gamma \, \mathrm{prox}_{\frac{\beta_m \psi_m}{\gamma}} \left( \frac{y_{2,m+2M,n}}{\gamma} \right)$
            $q_{2,m+2M,n} = p_{2,m+2M,n} + \gamma(\mathrm{Id} - H_m L_m) y_{1,n}$
            $v_{m+2M,n+1} = v_{m+2M,n} - y_{2,m+2M,n} + q_{2,m+2M,n}$
        end if
    end for
    for $s = 1, \ldots, S$ do
        $y_{2,s+d,n} = v_{s+2M,n} + \gamma F_s x_n$
        $p_{2,s+d,n} = y_{2,s+d,n} - \gamma \mathcal{P}_{D_s} \left( \frac{y_{2,s+d,n}}{\gamma} \right)$
        $q_{2,s+d,n} = p_{2,s+d,n} + \gamma F_s y_{1,n}$
        $v_{s+d,n+1} = v_{s+d,n} - y_{2,s+d,n} + q_{2,s+d,n}$
    end for
```

```
  if $H_m(obs) \quad init$ then
      $q_{1,n} = p_{1,n} - \gamma \Big( \sum_{m=1}^{M} \big( L_m^\top \big( T_m^\top p_{m,n} + \widetilde{x_m} \big) +$
$+ L_m^\top \big( T_m^\top p_{m+M,n} + \widetilde{x_m} \big) + (\mathrm{Id} - L_m^\top H_m^\top) p_{m+2M,n} \big)$
$+ \sum_{s=1}^{S} F_s^\top p_{2,s+d,n} \Big)$
  else
      $q_{1,n} = p_{1,n} - \gamma \Big( \sum_{m=1}^{M} \big( L_m^\top \big( T_m^\top p_{m,n} + \widetilde{x_m} \big) +$
$+ L_m^\top \big( T_m^\top p_{m+M,n} + \widetilde{x_m} \big) \big) + \sum_{s=1}^{S} F_s^\top p_{2,s+d,n} \Big)$

  end if
  $x_{n+1} = x_n - y_{1,n} + q_{1,n}$
end for
```

### 4.3.3 Discussion

The algorithm relies on successive computation of the criterion, constraints and their adjoint denoted by $^\top$ and the projection and proximity operators associated to each. As the transform and resampling operations are linear, their adjoint operators are easily computed as discussed in 4.2.1.3.

The explicit expression for computing the projection onto $C_m$ set is given in Eq. (4.7) and (4.8). In a similar fashion, considering the expression in Eq. (4.22), the projection on the range constraint set $D_1$ is achieved by setting all out-of-range pixels to the closest bound of interval $[x_{min}, x_{max}]$. For the smoothness constraint, the projection is not available in closed form, but several approaches exist in the literature to compute it [vdBF08], [CPPPP12]. The iterative technique described in [CPPPP12] is employed in our algorithm.

The computation of proximity operators is based on the explicit expressions available for a large number of convex functions [CP10], [CCPW07].

Furthermore, in order to assure the convergence of the algorithm to an optimal solution according to [CP12], $\gamma$ in Alg. 1 is subjected to the following constraint:

$$\gamma \in [\epsilon, (1-\epsilon)/\xi] where, \quad \epsilon \in ]0, 1/(\xi+1)[ \quad and$$

$$\xi = \sqrt{\sum_{m=1}^{M} \Big(2\|T_m(L_m - \widetilde{M_m})\|^2 + \|\,\mathrm{Id}\, - H_m L_m\|^2\Big) + \sum_{s=1}^{S} \|F_s\|^2} \quad (4.29)$$

where $\widetilde{M_m}$ denotes the prediction operator. Note that if predictive coding is not used, $\|T_m(L_m - \widetilde{M_m})\|^2$ becomes $\|T_m(L_m)\|^2$. Furthermore, the term $\|\,\mathrm{Id}\, - H_m L_m\|^2$ is removed if the HR initialization is not based on the $H$ operator. The norm of the operators can be computed using the iterative algorithm in [CPCP09, Algorithm 4].

## 4.4 HEVC Integration

In this work we apply the SR and video reconstruction model to the latest video coding standard High Efficiency Video Coding (HEVC, 2013) [BHO+12]. It is to be highlighted that HEVC always computes a prediction for a coding unit (CU) (more specifically, for each PU in a CU), either by Intra or Inter prediction, before encoding the CU residual (Eq. 4.3). In particular, the predicted frame $\widetilde{x_{m,i}}$ can be built by concatenation of all the predicted units, without explicitly knowing the prediction mode used for each unit.

HEVC computes residual signals at the CU level, but these residuals are transformed at the TU level. TUs are square pixel units that can be recursively subdivided, so different transform sizes are specified in HEVC (4x4, 8x8, 16x16, 32x32). Due to complexity considerations, HEVC relies on finite approximations of well-known transforms: the Discrete Cosine Transform (DCT) and its inverse (IDCT). Moreover, a Discrete Sine Transform (DST) is specifically used for 4x4 Intra units. The transform matrices are fully standardized and can be found in [BFB14].

Given an input residual frame (obtained by subtracting the predicted frame from the source signal), the implementation of HEVC transforms $T_{m,i}$ requires the extraction of the following two elements from HEVC bitstreams : the frame type (to distinguish between DST and DCT for 4x4 units) and the TU partitioning.

HEVC quantization is performed at a TU level on the transformed residual. HEVC standard implements a scalar quantizer similar to the one presented in Section 4.2.1.2. The applicable quantizer is indicated by a Quantization Parameter (QP) ranging from 0 to 51 which serves as an integer index to derive the applicable step size $\Delta_q$. HEVC follows a logarithmic structure : the step size doubles when the QP increases by 6. The first six step sizes (for QP ranging from 0 to 5) are presented below, alongside with the formula allowing to infer the step-size at higher QPs.

$$\Delta_{q,0..5} = \{2^{-4/6}, 2^{-3/6}, 2^{-2/6}, 2^{-1/6}, 1, 2^{1/6}\} \tag{4.30}$$

$$\Delta_q(QP) = \Delta_{q,QP\,mod\,6} \cdot 2^{\lfloor QP/6 \rfloor} \tag{4.31}$$

Given an input frame of transformed coefficients, in addition to the information extracted in previous section (TU Partitioning, frame type), the implementation of HEVC quantizer ($Q_{m,i}$) thus only requires the extraction of the QP map (containing the QP of each TU) to compute the step-size for each unit.

## 4.4.1 Framework adjustments

### 4.4.1.1 Extracting the required HEVC information

Applying the SR model to HEVC encoded video streams relies on information we can extract during the decoding process (TU partitioning, QP map, etc.). An OpenHEVC decoder [ope] has been patched to output the required elements for the SR approach. In particular, the modified decoder generates the following informative streams: the reconstructed frames, the encoded coefficients (denoted as $z_{m,i}$ in Section 4.2) the predicted frames, the TU partitioning and TU types and the QP map.

#### 4.4.1.2 Encoding configurations

HEVC compliant video streams are generated using the reference software HM 15.0 [vtJoIMITV13]. The encoding configuration uses the default Random Access configuration file, with some slight modifications. First, CU-based multi-QP optimization is enabled by setting the parameter $MaxDeltaQP$ at 2. Second, since our SR model has not considered HEVC in-loop filters yet, both the deblocking and sample adaptive offset filters were turned off in the configuration file. HEVC supports two special coding modes for intra coding denoted as Pulse Code Modulation (PCM) and transform skipping mode. PCM, which consists in bypassing prediction, transform, quantization and entropy coding, and samples are coded by a predefined number of bits. In this work, the PCM mode is disabled (as it is by default in Random Access configuration file).

#### 4.4.1.3 A closer look on the HEVC residual skip coding tools

In a generic compression framework as the one denoted by VC in this work (Sec. 4.5.1.1), residual information is systematically transformed and quantized. However, HEVC may entirely skip the residual for a block, i.e. when the prediction is good enough given the target quality. This choice is made during Rate-Distortion Optimization (RDO) at the encoder side, and is most frequently indicated explicitly in the bitstream. Indeed, HEVC standard defines in the transform tree syntax, for each TU, a flag *cbf luma* which indicates if residual luminance information is present for the current TU (similar codewords *cbf cr* and *cbf cb* are used for chrominance residuals). Besides, the absence of residual is automatically inferred for 64x64 TUs in Inter frames. The case where TUs have no residual are not naturally modeled by our framework. First, the data-fidelity cost function (Eq. 4.15) relies on quantized coefficients observed in the bitstream, which are missing in this case. Considering the absence of reliable information, skipped TUs are not taken into account in the data-fidelity computation.

The solution validity (Eq. 4.16) relies on quantization intervals which cannot be extracted from the bitstream when the QP of a TU is not defined. This constraint may help to model the uncertainty of skipped TU residuals. Therefore, we relied on empirical testing to define the best selected QP for a skipped TU, by evaluating our SR model on one Inter frame of two HEVC encoded low-resolution observations generated with two different degradation operators: bicubic with anti-aliasing (BicAA) and without (BicNAA). We tested three different QPs to apply to skipped units: the

|  |  | QP of skipped units |  |  |  |  |  |
|---|---|---|---|---|---|---|---|
|  |  | 1 |  | Max. * |  | 51 |  |
| Sequence | Enc. QP | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| Akiyo | 20 | **36.61** | 0.9598 | 35.60 | **0.9612** | 35.29 | 0.9600 |
|  | 30 | 32.45 | 0.9176 | **34.17** | **0.9416** | 34.16 | 0.9415 |
|  | 40 | 27.95 | 0.8449 | **30.35** | **0.8802** | **30.35** | **0.8802** |
| Foreman | 20 | 33.69 | 0.9179 | **34.24** | **0.9287** | 32.90 | 0.9281 |
|  | 30 | 29.75 | 0.8530 | **31.25** | 0.8873 | 31.05 | **0.8875** |
|  | 40 | 26.11 | 0.7721 | **27.96** | **0.8190** | **27.96** | **0.8190** |
| Bus | 20 | **28.24** | **0.8608** | **28.24** | 0.8510 | 27.82 | 0.8221 |
|  | 30 | 25.25 | 0.7403 | 26.12 | 0.7637 | **26.24** | **0.7703** |
|  | 40 | 22.02 | 0.5498 | 23.36 | 0.5818 | **23.40** | **0.5857** |

Table 4.1 – Using different QP settings for skipped units. (*: Max. mode uses the frame maximum available QP)

QP range boundaries: 1 and 51, and the maximum available QP during the frame encoding, which depends on the $MaxDeltaQP$ parameter. The results are gathered in Tab. 4.1. When using QP 1, only very small modifications of the quantized coefficients are tolerated (smallest possible quantization interval). Interestingly, this choice is not always the worst at low QPs for static sequences (i.e. Akiyo), but huge quality drops are observed at higher QPs. On the opposite, using QP 51 is almost equivalent to removing the constraint for skipped units: it implies a low confidence on pixel values, and the very wide quantization intervals enable unit variations while preserving their validity. This option has been found slightly inferior (in terms of average quality and stability over sequences/QPs) than the solution based on the maximum encoding QP; we thus applied this latter solution in the remaining of this publication. Such a result is eventually quite intuitive: if the encoder RDO decides to skip a TU, it is most probably because no information is to be coded at the encoding QP, which is thus a natural candidate for modeling the degree of confidence we can have in the unit pixel values.

## 4.5 Experimental results

In this section, we report the main results of this work. We begin by discussing the experimental setup and defining the test bench architecture and algorithm parameters which are used throughout the main experiments. A series of small tests are performed in order to show the proposed method's behavior and better define the experimental framework. A discussion of the results concludes this section.

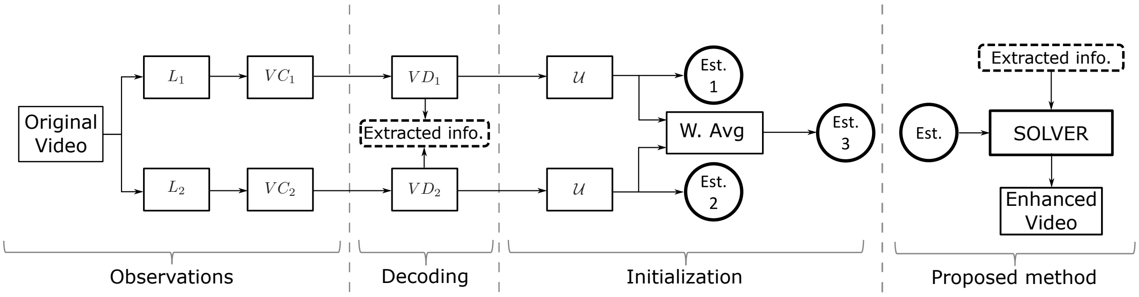


Figure 4.4 – A schematic view of the experimental setup. Two re-sampling operators are applied on an input sequence. Each observation is compressed and decompressed, and useful information is extracted. Decoded observations are up-sampled to their original resolution using either the reverse of the degradation operator or a SOA SR method. Then, the proposed framework is initialized with one high-resolution estimate and the information extracted during the decoding.

The proposed framework enables SR from one observation. In the preliminary work [BPPDB16], it was showed that the convex optimization SR approach is efficient on one HEVC Intra coded observation, albeit State Of the Art (SOA) approaches (as the learning-based Anchored Neighborhood Regression proposed by Timofte *et al.* [TSG15]) could exhibit higher quality improvements in certain cases.

## 4.5.1 Experimental setup and preliminary tests

### 4.5.1.1 Quick presentation of the experimental setup

In Fig. 4.4 we depict our test bench architecture. Two observations are generated by applying two different degradation operators denoted by $L_1$ and $L_2$ followed by a compression step with a video coder ($VC$).

In this work we use two VCs. A Matlab implementation of a generic hybrid video coder that matches the scheme in Fig. 4.1. For the sake of simplicity we do not apply any form of lossless coding on the transform coefficients and measure the rate using the entropy of the transform coefficients. A second choice of video coder falls on HEVC, using the configuration described in Section 4.4.1.2. The bitstream is obtained using the reference software HM 15.0 [vtJoIMITV13].

As discussed in Section 4.2 and 4.3 the solver requires some information to be extracted for applying $T_m$, determining the convex set $C_m$ and computing the prediction ($\widetilde{x_m}$). For both coders the extraction of the required information is performed at the time of decoding. In the case of HEVC encoder, a patched OpenHEVC decoder is used (see Section 4.4.1).

The decoded observations are re-sampled to their original resolution. The up-sampling ($\mathcal{U}$) is performed using the $H$ operator which reverses the down-sampling process of operator $L$ applied to the original sequences (see Section 4.2.1.3). Other HR estimates can be obtained using state-of-the-art (SOA) SR methods which usually provide higher quality than the polyphase filter up-sampling. We selected the work from Timofte *et al.* [TSG15] as an SOA reference, since it is one of the best performing approaches (see Section 4.1). The two up-sampled descriptions are then combined by weighted averaging with weights $\alpha_m$. The choice of initialization may influence the end result as the algorithm may converge towards a different local minimum. Furthermore, as discussed in Section 4.3, if the up-sampled observation introduces new information the, SR prior $I_d - H_m L_m$ will be disabled. Each up-sampling method generates three natural initialization candidates: $\uparrow_{\mathcal{U}} (Obs_1)$, $\uparrow_{\mathcal{U}} (Obs_2)$, $W.Avg._{\alpha_m}(\uparrow_{\mathcal{U}})$, where $\mathcal{U}$ denotes the up-sampling method ($H_m$, $SOA$). Of course, any number of observations can be used if required by a certain scenario and other $SOA$ methods can be combined. However, different test architectures are left as a future study direction. A comparison of different initializations is discussed in Section 4.5.1.4.

#### 4.5.1.2 Parameter selection

The application of the proposed framework relies on the definition of some parameters and metrics. First, as presented in Sections 4.2.2.1 and 4.2.2.3, the data-fidelity cost function $J_{\mathrm{DF}}$ and SR “prior” $J_{\mathrm{SR}}$ depend on the suitable metrics $\phi_m$ and $\psi_m$. As we use the Peak Signal to Noise Ratio (PSNR) to evaluate the results quality, consequently, we relly on the $l^2$ norm for $\phi_m$ and $\psi_m$.

The parameter $\alpha_m$ accounts for the unequal reliability of the observations. This parameter may not be easily estimated, since the quality of the observations is not measurable w.r.t. the unavailable original sequence at the decoder side. In the remaining of this work (unless when explicitly indicated), we simply set all $\alpha_m$ to $1/M$ which implies equal importance of each observation.

The constraint imposed on the TV of the result (Section 4.2.2.3, Eq. (4.18)) has to be defined. In order to obtain an adequate smoothness level, we impose a content dependent boundary on the TV. Namely, we measure for a video frame $i$ the TV of the high-resolution initialization denoted by $x_i^0$. The TV boundary used for the final result is derived according to:

$$\text{find } \hat{x} : \quad \mathrm{TV}(\hat{x_i}) \leq \eta \quad where \quad \eta = \eta_0 \times \mathrm{TV}(x_i^0) \tag{4.32}$$

where $\eta_0$ is used to weight the smoothness of the result. As such, a value close to 1 will lead to a similar smoothness level as the observation whereas smaller $\eta_0$ values increase the result smoothness. The $\eta$ parameter (unless when explicitly indicated) is empirically determined and set to 0.95 in the remaining of this work.

Finally, the $\beta$ parameter is used to weight the super resolution prior $J_{\mathrm{SR}}$ (Equation 4.19). Different values could be assigned for individual observations which should reflect the performance of subsequent application of down-sampling and up-sampling ($H_m L_m$) w.r.t. the level of compression. However, in order to preserve the generality of the method we set this parameter to a value of 0.15 in all tests, which, provides overall good results on all tested scenarios.

#### 4.5.1.3 Choice of $L_m$

In order to select the down-sampling operators used in our main experiments, we perform a preliminary test with different choices for the $L$ and $H$ operators (see Section 4.2.1.3). Our goal is to study the behavior of our algorithm, the $H$ based up-sampling and the SOA anchor [TSG15] w.r.t. $L$. We thus limit the test to a certain set of conditions that emphasize this behavior. We select the Foreman sequence in a full intra mode coding configuration. The choice of video coder falls on the generic VC compression model (see Section 4.5.1.1 and Figure 4.1). To minimize the impact of compression on the performance of the filters we use a QP of 1, the quantization step is given by Equation 4.31. Furthermore, we relax the TV constraint and set $\eta_0 = 1.05$. In this scenario we use only 1 description generated with various $L$ models. We test 2 popular interpolation functions: Bicubic (*Bic*) and Lanczos3 ($Lanc_3$). The Bicubic and Lanczos3 functions are defined on the intervals $[-2, 2]$ and $[-3, 3]$. Thus, the phase used in a down-sampling of scale 1/2, with each filter, has 4 and 6 taps, respectively. Furthermore, we also combine the filter with an anti-aliasing effect by stretching the functions, resulting, in a larger number of taps for each phase. The results are reported in Table 4.2. *Bic* interpolation filters use 4, 8 and 12 taps while $Lanc_3$ has 6, 10 and 14 taps. From the start we can notice that the SOA method denoted by $\uparrow_{SOA}$ provides a significant improvement over filter based up-sampling using the same interpolation function as $L$ denoted by $\uparrow_H$. An interesting observation is that the SOA method exhibits a non-uniform performance behavior w.r.t. the number of taps. Best $SOA$ performance is achieved for $\downarrow Bic8$ and $\downarrow Lanc_3 10$. Our method, however, shows an increase in performance with the number of taps. Overall, we obtain comparable quality with $SOA$ for $Bic4\&8$ and $Lanc_3 6\&10$ and outperform it on the other two filters. Using $SOA$ as initialization further improves the result.

| $\downarrow\uparrow$ method | $\uparrow_H$ | $\uparrow_{SOA}$ | $\uparrow_{Prop.}$ $(\uparrow_H)$ | $\uparrow_{Prop.}$ $(\uparrow_{SOA})$ |
|---|---|---|---|---|
| $\downarrow_L$ $(Bic4)$ | 32.22 | 32.44 | 32.44 | 33.22 |
| $\downarrow_L$ $(Bic8)$ | 31.86 | 33.35 | 33.38 | 33.75 |
| $\downarrow_L$ $(Bic12)$ | 30.35 | 31.11 | 34.33 | 34.66 |
| $\downarrow_L$ $(Lanc_3 6)$ | 31.95 | 31.34 | 32.12 | 32.51 |
| $\downarrow_L$ $(Lanc_3 10)$ | 32.58 | 33.27 | 33.16 | 33.85 |
| $\downarrow_L$ $(Lanc_3 14)$ | 30.88 | 31.24 | 34.39 | 34.91 |

Table 4.2 – Comparing the PSNR (dB) of up-sampling methods $\uparrow_H$, $\uparrow_{SOA}$, $\uparrow_{Prop}$ $(\uparrow_H)$, $\uparrow_{Prop}$ $(\uparrow_{SOA})$ w.r.t different down-sampling filters.

Thus, for fairness of comparison, in our main tests we decide to select the best performing case for $SOA$, $Bic8$ and $Bic4$, as $L_1$ and $L_2$ operators. We will refer to these choices as BicAA and BicNAA.

#### 4.5.1.4 Initialization of the proposed method

As discussed in our previous work [BPPDB16], the initialization can have significant impact on the final result quality. We conduct a preliminary test on Foreman sequence to discuss this phenomenon (CIF, SRx2, 10 HEVC Intra frames, QP 25). Two observation are generated using BicAA and BicNAA (see Section 4.5.1.3) degradation operators. After compression and decompression, 6 HR estimates are generated : three from the reverse polyphase filters -and their average- and three other generated using the SOA SR method. In Table 4.3, we detail the quality of the high-resolution estimates we can derive from the observations, and gather the results obtained by our framework using each estimate as the initialization.

Table 4.3 highlights typical behavior of the proposed framework: amongst all available high-resolution estimates, the best option is to select the one with highest quality. This namely justifies the use of single-image SR to generate the high-resolution estimate. In this particular example where both observations are encoded with similar compression settings, decoded frames are of comparable quality and to use their average exhibits the best results (with either polyphase up-sampling or SOA SR). In general, observations may be of very different quality, in which case a weighted average may be a better option. Yet, in real-world scenarios at a decoder side, one cannot compute the quality w.r.t. the unavailable original sequence. Thus, looking for the optimal initialization is in general a difficult problem.

| $\uparrow Met.$ | $\uparrow_f\ (Obs_1)$ | $\uparrow_f\ (Obs_2)$ | $Avg(\uparrow_f)$ |
|---|---|---|---|
| $\uparrow_H$ | 31.11 | 31.55 | 31.56 |
| $\uparrow_{Prop.}\ (\uparrow_H)$ | 33.27 | 33.21 | 33.36 |
| $\uparrow_{SOA}$ | 32.01 | 31.72 | 32.51 |
| $\uparrow_{Prop.}\ (\uparrow_{SOA})$ | 33.64 | 33.58 | 33.97 |

Table 4.3 – PSNR (dB) comparisons when using different initializations for the proposed method. $\uparrow_f$ denotes the up-sampling with method $f$. The columns correspond to the up-sampled observations and their average.

## 4.5.2 Comparison with reference and SOA.

### 4.5.2.1 SR from two low-resolution observations

In a first scenario, we consider the issue of SR from two low-resolution observations. The experimental setup follows Figure 4.4, and the two observations are generated by down-sampling (by a factor of 2 in each dimension) a given input sequence using BicAA and BicNAA. HEVC was used to compress wach LR observation. Two coding configurations are specifically analyzed, denoted by II and IP. II mode corresponds to a full Intra configuration: each frame of the sequence is treated as an independent Intra frame, without motion estimation and compensation tools. However, in the case of HEVC, I-frames use Intra prediction. On the opposite, IP configuration exploits P frames to improve the coding efficiency, and in this case, a GOP size of 8 was picked. Evaluations are carried out on 6 CIF sequences (352x288).

As discussed in previous section, multiple high-resolution estimates can be generated from two observations. In this scenario, low-resolution observations are of comparable quality, thus using the average between the observations is a coherent choice. In Tables 4.4 and 4.5, we detail results obtained for each sequence at different QP values. For each QP, three values are presented: the one denoted by *Ref* represents the average between up-sampled decoded observations using polyphase filters. Similarly, the column denoted by *SOA* measures the quality obtained when averaging the up-sampled decoded observations using the single image SR work from Timofte et. al. [TSG15]. Finally, *Prop.* column stands for the proposed algorithm initialized with the SOA result (i.e. the average between decoded observations up-sampled with SOA SR).

Tables 4.4 and 4.5 highlights the efficiency of the proposed framework in the tested scenario. First, we show a significant improvement over the reference obtained by averaging the bi-cubic up-samplings. Namely, the PSNR gain can be superior to 5 dBs for low QP encoding, and up to 0.5dB gain is achieved at QP 35. This result tends to demonstrate that the complementary information gathered from each observation is advantageously used by the proposed framework. PSNR gains superior to 3 dBs can be observed at low QPs, when compared to SOA. SSIM results are also reported in this table. However, note that at high QPs the methods tend to exhibit similar performance, as the high compression level combined with the down-sampling operation lead to a highly unreliable description for inferring additional information. Furthermore, using the generic VC compression model leads to higher gains at high QPs. This is explained by the use of skipped blocks in HEVC compression. The estimated image is less reliable on skipped blocks (see Section 4.4).

| | | | QP1 | | | QP15 | | | QP25 | | | QP35 | | | Average | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Sequence | Mode | Ref | SOA | Prop. | Ref | SOA | Prop. | Ref | SOA | Prop. | Ref | SOA | Prop. | Ref | SOA | Prop. |
| PSNR (dB) | akiyo | II | 34.76 | 37.40 | **41.11** | 34.65 | 37.19 | **39.16** | 34.13 | 36.27 | **36.92** | 31.81 | 32.71 | **32.8** | 33.84 | 35.89 | **37.5** |
| | | IP | 34.76 | 37.4 | **40.97** | 34.62 | 37.1 | **38.84** | 34.16 | 36.24 | **36.87** | 32.17 | 33.13 | **33.14** | 33.93 | 35.97 | **37.46** |
| | foreman | II | 32.15 | 33.28 | **38.29** | 32.08 | 33.2 | **36.42** | 31.58 | 32.53 | **33.97** | 29.55 | 29.94 | **30.09** | 31.34 | 32.24 | **34.69** |
| | | IP | 32.15 | 33.28 | **38.06** | 32.01 | 33.12 | **36.03** | 31.25 | 32.07 | **32.99** | 28.78 | 29.06 | **29.06** | 31.05 | 31.88 | **34.03** |
| | bus | II | 26.84 | 27.93 | **31.91** | 26.82 | 27.91 | **30.48** | 26.62 | 27.70 | **28.83** | 25.23 | 25.99 | **26.16** | 26.38 | 27.38 | **29.35** |
| | | IP | 26.83 | 27.93 | **31.8** | 26.79 | 27.91 | **30.21** | 26.37 | 27.46 | **27.91** | 24.14 | 24.64 | **24.55** | 26.03 | 26.99 | **28.62** |
| | mobile | II | 22.69 | 23.7 | **27.6** | 22.68 | 23.69 | **26.54** | 22.61 | 23.61 | **25.05** | 21.97 | 22.83 | **23.16** | 22.49 | 23.46 | **25.59** |
| | | IP | 22.69 | 23.7 | **27.54** | 22.67 | 23.69 | **26.26** | 22.49 | 23.57 | **24.51** | 21.1 | 21.67 | **21.76** | 22.24 | 23.16 | **25.02** |
| | football | II | 28.01 | 29.85 | **33.27** | 27.99 | 29.82 | **31.83** | 27.72 | 29.46 | **30.37** | 26 | 27 | **27.18** | 27.43 | 29.03 | **30.66** |
| | | IP | 28.01 | 29.85 | **33.13** | 27.96 | 29.77 | **31.52** | 27.45 | 29.01 | **29.24** | 24.54 | 24.93 | **25.02** | 26.99 | 28.39 | **29.73** |
| | flower | II | 22.97 | 23.22 | **26.55** | 22.97 | 23.21 | **25.95** | 22.92 | 23.17 | **24.70** | 22.46 | 22.75 | **23.05** | 22.83 | 23.09 | **25.06** |
| | | IP | 22.97 | 23.22 | **26.51** | 22.96 | 23.21 | **25.77** | 22.84 | 23.15 | **24.29** | 21.92 | 22.09 | **22.18** | 22.67 | 22.92 | **24.69** |
| Average | | | **27.90** | **29.23** | **33.06** | **27.85** | **29.15** | **31.58** | **27.51** | **28.69** | **29.64** | **25.80** | **26.4** | **26.51** | **27.27** | **28.37** | **30.20** |
| SSIM | akiyo | II | .9642 | .978 | **.9816** | .9592 | .9729 | **.9743** | .9472 | .9595 | **.958** | .9066 | .9138 | **.9143** | .9443 | .9561 | **.957** |
| | | IP | .9642 | .978 | **.9813** | .959 | .9723 | **.9728** | .9484 | .9602 | **.9593** | .9136 | .9206 | **.921** | .9463 | .9578 | **.9586** |
| | foreman | II | .9402 | .9551 | **.9654** | .9356 | .9503 | **.9517** | .909 | .9205 | **.9213** | .851 | .8565 | **.8585** | .909 | .9206 | **.9242** |
| | | IP | .94 | .955 | **.9632** | .9319 | .9457 | **.9445** | .9002 | .9094 | **.9044** | .8327 | .8362 | **.8354** | .9012 | .9116 | **.9119** |
| | bus | II | .8524 | .8905 | **.9384** | .8498 | .8881 | **.9134** | .8272 | .8658 | **.8734** | .6996 | .728 | **.7277** | .8073 | .8431 | **.8632** |
| | | IP | .8523 | .8905 | **.9363** | .8472 | .8862 | **.9074** | .8093 | .8473 | **.8372** | .665 | .6886 | **.6859** | .7935 | .8281 | **.8417** |
| | mobile | II | .7873 | .8576 | **.9242** | .7857 | .8557 | **.8989** | .7756 | .8438 | **.8623** | .7136 | .7732 | **.7761** | .7656 | .8326 | **.8654** |
| | | IP | .7872 | .8575 | **.9227** | .7844 | .8539 | **.8906** | .7649 | .8318 | **.8341** | .6599 | .7036 | **.705** | .7491 | .8117 | **.8381** |
| | football | II | .8771 | .9145 | **.943** | .8748 | .9124 | **.924** | .849 | .8874 | **.8847** | .7029 | .7258 | **.7266** | .826 | .86 | **.8695** |
| | | IP | .877 | .9144 | **.9408** | .8724 | .9091 | **.9165** | .8358 | .8694 | **.8509** | .6507 | .6625 | **.662** | .809 | .8389 | **.8425** |
| | flower | II | .8292 | .8645 | **.9225** | .8276 | .8627 | **.9044** | .8192 | .8532 | **.8753** | .7782 | .8087 | **.8097** | .8135 | .8473 | **.878** |
| | | IP | .8291 | .8644 | **.9214** | .8265 | .8615 | **.8995** | .8129 | .8468 | **.8598** | .7562 | .7817 | **.7817** | .8062 | .8386 | **.8656** |
| Average | | | **.875** | **.91** | **.9451** | **.8712** | **.9059** | **.9248** | **.8499** | **.8829** | **.8851** | **.7608** | **.7833** | **.7836** | **.8392** | **.8705** | **.8846** |

Table 4.4 – PSNR comparison of the reference method, SOA [TSG15] and proposed framework, when two low resolution observations are available. These results were obtained using HEVC compression.

Additional results are provided in Figure 4.5 where rate-distorsion curves are shown for all sequences. All 6 QPs were used to generate the RD curves. The results show a similar trend with what was presented in Table 4.4, significant gains are obtained over the SOA SR, and best results are obtained at low QPs. The impact of the QP on the method efficiency is related to the extensive use of transformed coefficients -observed in the bitstream- and their respective quantization intervals. The reliability of these anchors highly depends on the encoding QP, which justifies the higher performance of the approach for high quality encodings. This test was also performed using generic VC, the results are shown in Figure 4.6. Note that the generic implementation of VC does not skip any blocks, therefore at high QPs the method can show a higher improvement when compared to HEVC encoding.

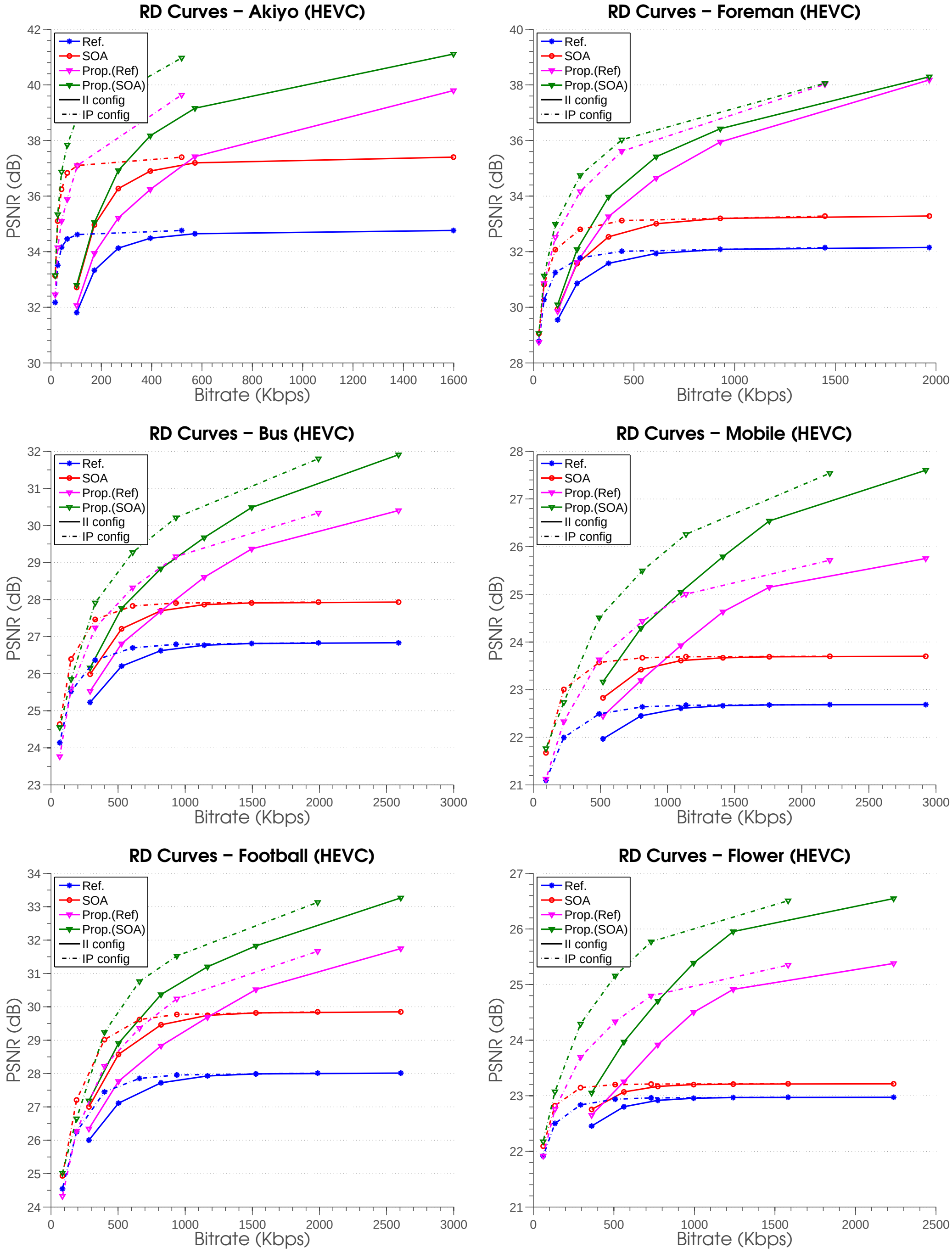


Figure 4.5 – Rate-distortion curves for the reference, SOA and proposed method initialized with the reference or SOA using HEVC encoding.

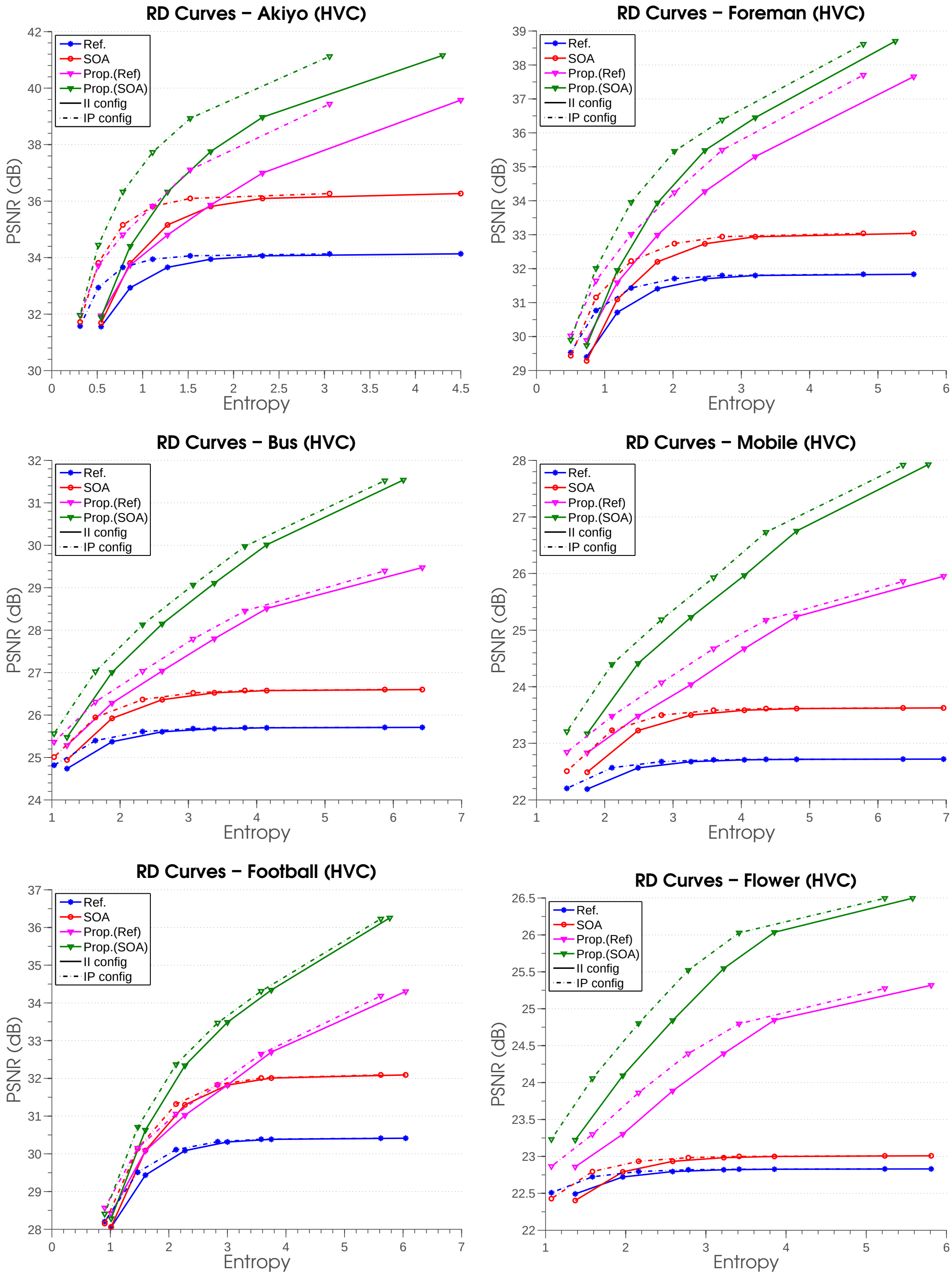


Figure 4.6 – Rate-distortion curves for the reference, SOA and proposed method initialized with the reference or SOA using generic VC encoding.

Furthermore, the PSNR over time (Figures 4.7 and 4.8), highlights the temporal stability of the approach. The proposed method achieves a relatively constant gain, in time, over the Ref and SOA.

#### 4.5.2.2 SR from one low-resolution observation and one high-resolution observation

In a second scenario, we consider the case where one observation is available at LR and the other one is available at the original resolution. Our framework is capable of combining these observations naturally, since each observation is modeled with its own degradation model. In general, HR coded streams exhibit higher quality than up-sampled low-resolution streams, encoded with similar parameters. This behavior leads to a large ΔPSNR between HR and LR descriptions. Intuitively, if the ΔPSNR is very large there is not a lot of information that can be extracted from a LR observation which is not already contained in the HR description. Therefore, we begin this scenario with a small test performed on a few frames of Bus sequence, with generic VC in full Intra mode. Our goal is to analyze the algorithms behavior w.r.t. the ΔPSNR of the two observations, denoted by $\Delta_{Obs}$ in Table 4.6. $\uparrow_H$ Obs 1, $\uparrow_{SOA}$ Obs 1 and Prop denote the up-sampled observation with $H$ and SOA methods and the result obtained by our proposed method. $\Delta$ is the improvement obtained with Prop over Obs 2. First column shows the QPs used in coding Obs 1 and Obs 2, respectively. In this test the initialization of M-LFBF solver was Obs 2. We can easily notice that higher gains are achieved when the descriptions are more similar in terms of quality. An interesting observation can be made for QPs 1/20 and 15/20. Even though, the quality of Obs 1 only increases with 0.02 and Obs 2 remains unchanged we can see a large difference in $\Delta$ (from 0.69 to 1.47). This behavior can be explained by the algorithms dependency on the information variety between descriptions, rather than their individual quality. Tests performed on other sequences reveals a similar behavior, however, for the sake of brevity we do not repeat this test for each encoder, configuration and sequence. As such, we decide to perform a complete set of tests using a QP combination that provides similar quality observations. QP 40 for the HR observation and QPs 1 and 15 for the LR observation. As we did in our previous experiment (Section 4.5.2.1), we report both PSNR and SSIM scores. The LR observation is obtained with BicAA down-sampling. As the quality of the observations is closer than in our preliminary tests we initialize the algorithm using the average. Ref and SOA in this case denote the average between Obs 2 and Obs 1 up-sampled with $H$ and SOA, respectively. The results are reported in Table 4.7. Our algorithm outperforms the reference and SOA methods on all sequences. Gains of up to 3 dBs and 1 dBs are obtained with respect to the reference and SOA methods, respectively. On average over all sequences and all QPs combinations, 1.95 dBs and 0.73 dBs gains are obtained over Ref and SOA.

Figures 4.7 and 4.8 show the PSNR variation over time of the two descriptions: down-sampled and encoded at QP15, full resolution at QP40. The two tested up-sampling methods are shown and the results obtained by combining the encoded and up-sampled versions. This test was performed using generic VC encoding over 50 frames. It can be observed that the proposed method is capable of efficiently combining the two descriptions and a stable gain over time is achieved over the other up-sampling methods or averages with the encoded frames(Ref, SOA).

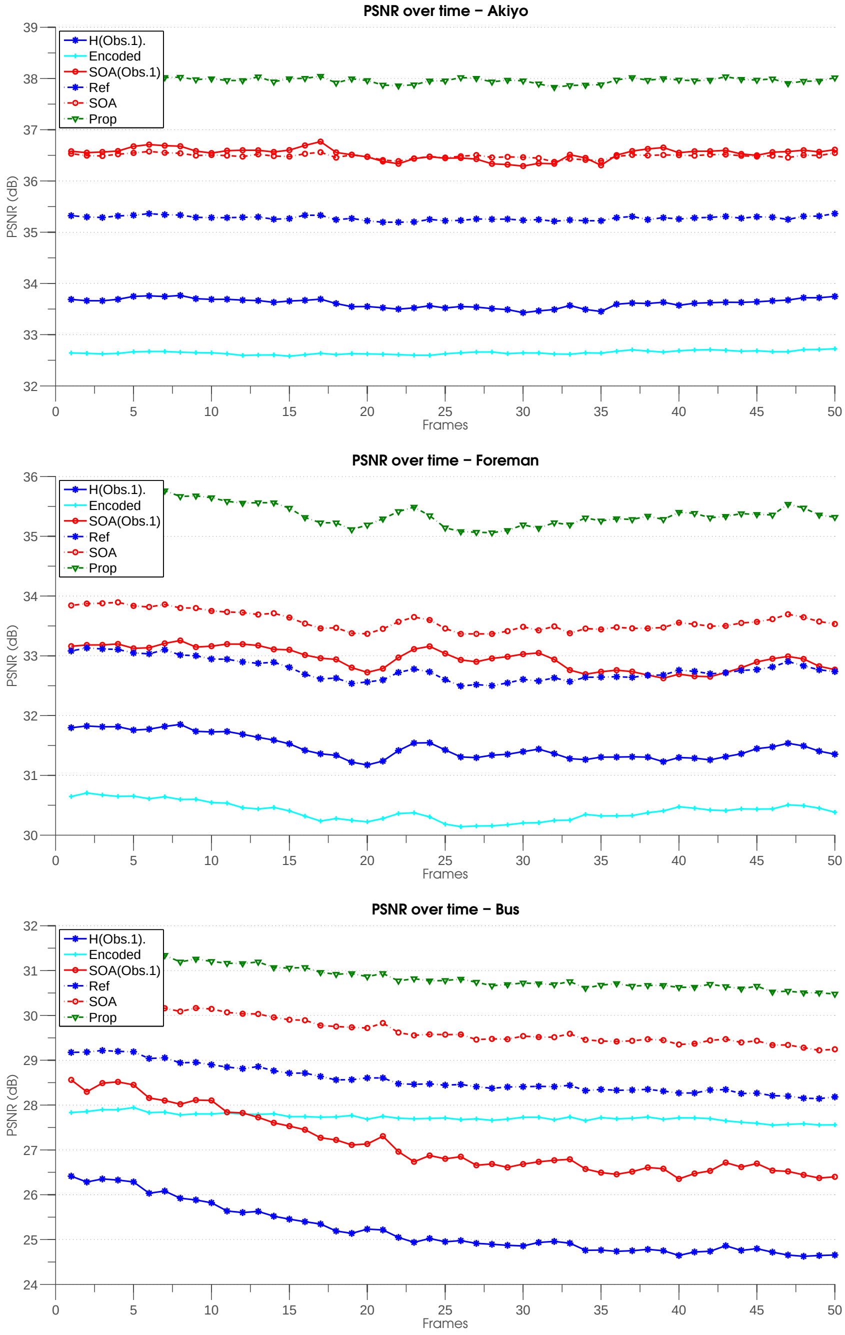


Figure 4.7 – PSNR over time on Akiyo, Foreman and Bus sequences. One down-sampled observation encoded at QP15 (Obs 1) and one observation at full resolution encoded at QP40 (Encoded). H(Obs 1) and SOA(Obs 1) denote the two up-sampling methods applied on Obs 1; Ref and SOA denote the weighted average of the encoded observation and the up-sampled Obs 1 and Prop is the proposed method.

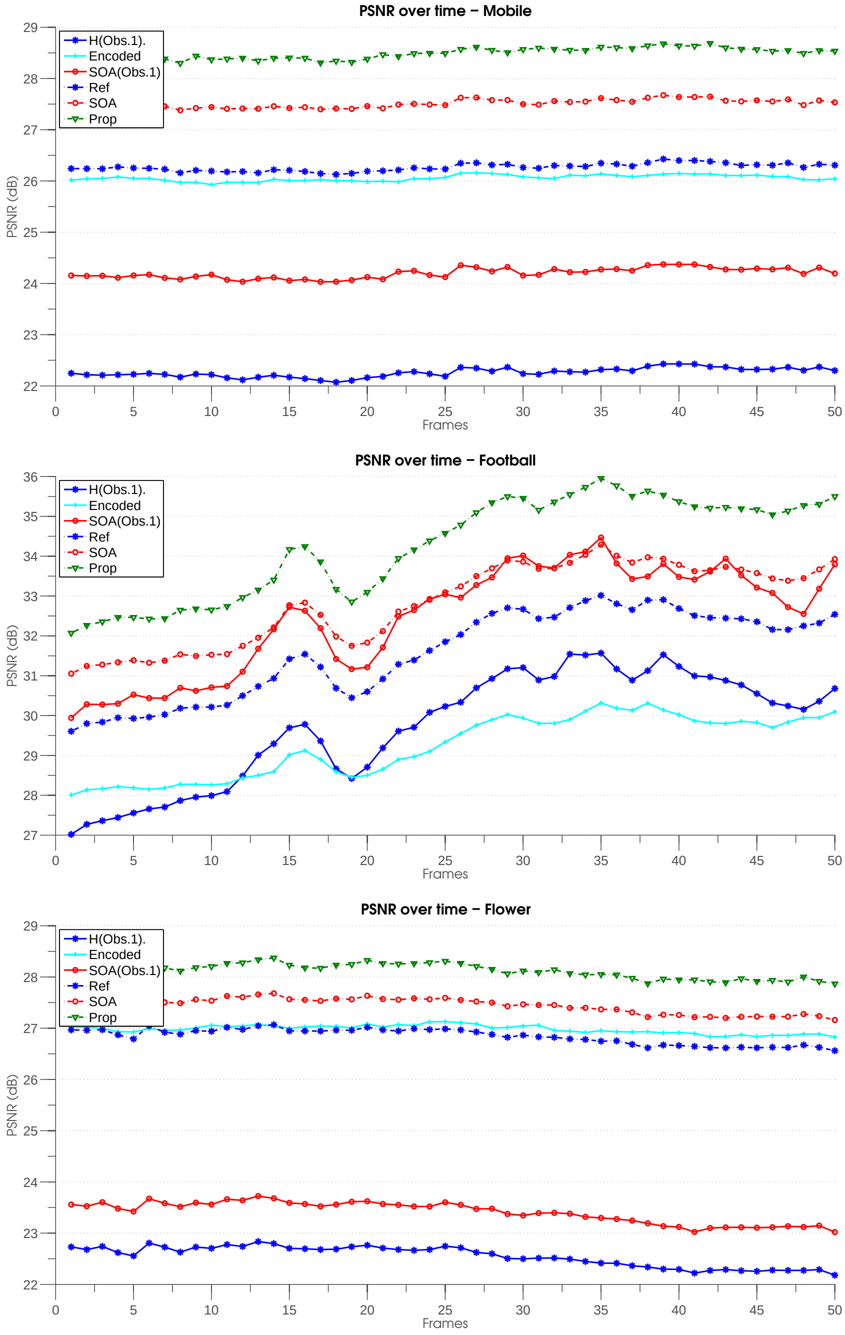
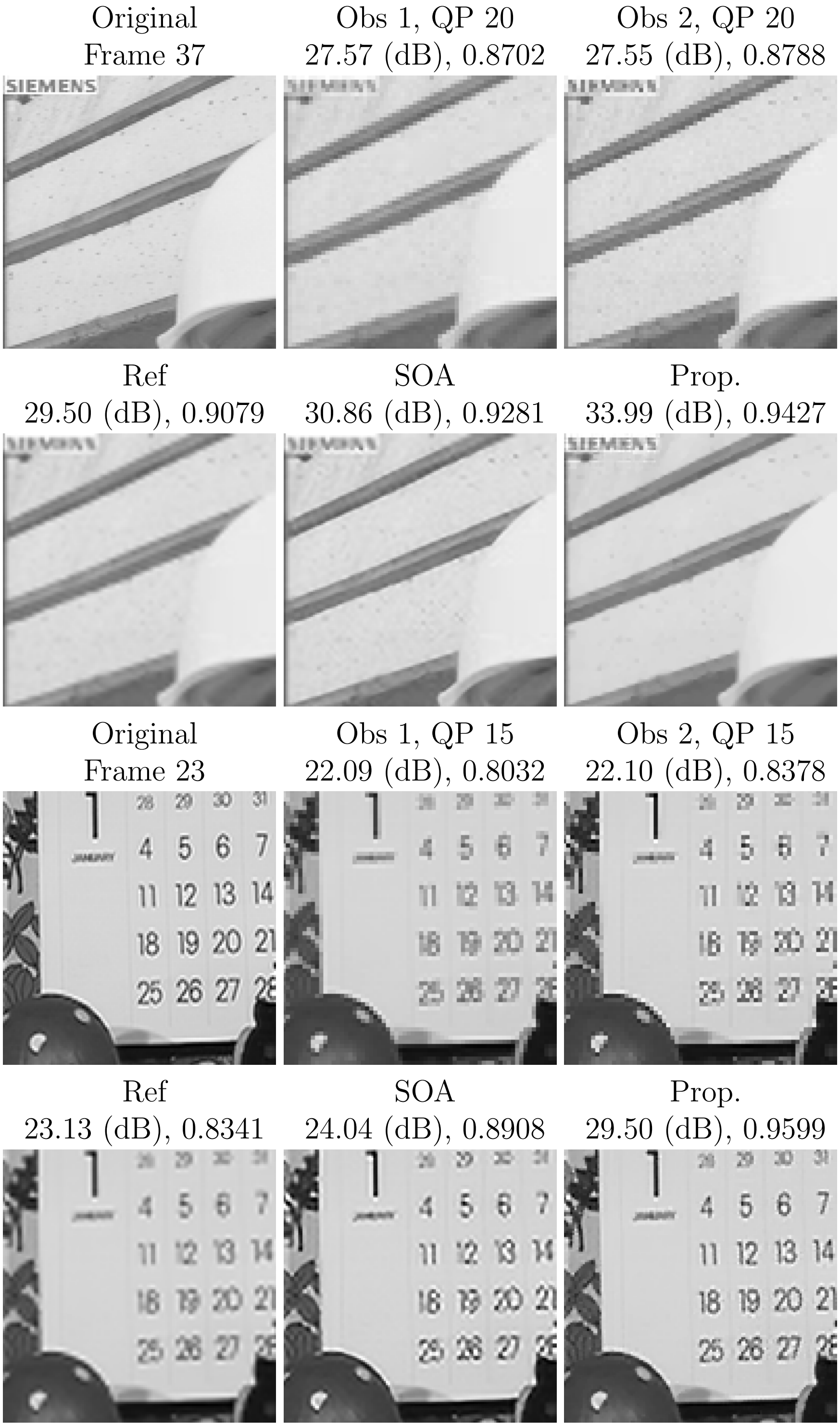


Figure 4.8 – PSNR over time on Mobile, Footbal and Flower sequences. One down-sampled observation encoded at QP15 (Obs 1) and one observation at full resolution encoded at QP40 (Encoded). H(Obs 1) and SOA(Obs 1) denote the two up-sampling methods applied on Obs 1; Ref and SOA denote the weighted average of the encoded observation and the up-sampled Obs 1 and Prop is the proposed method.

#### 4.5.2.3 Visual results

Visual results are available in Figure 4.9. Image details from Foreman and Mobile sequences are depicted for each observation and tested method. PSNR and SSIM results are reported for each image. It is easily noticeable that the proposed method provides the best results. The text which is almost unreadable in the Ref and SOA super resolved images is readable when using the proposed approach.

Figure 4.10 shows details from Bus and Flower sequences. It can be noticed that the proposed method better reconstructs high frequency components for complex textures. Less complex textures are depicted in Figure 4.11, in this case the differences between methods are harder to observe. The over smoothing effects caused by bicubic up-sampling are less impactfull on more uniform textures. However, objective metrics indicate that the proposed method and the SOA have a higher quality.



Original
Frame 37

Obs 1, QP 20
27.57 (dB), 0.8702

Obs 2, QP 20
27.55 (dB), 0.8788

Ref
29.50 (dB), 0.9079

SOA
30.86 (dB), 0.9281

Prop.
33.99 (dB), 0.9427

Original
Frame 23

Obs 1, QP 15
22.09 (dB), 0.8032

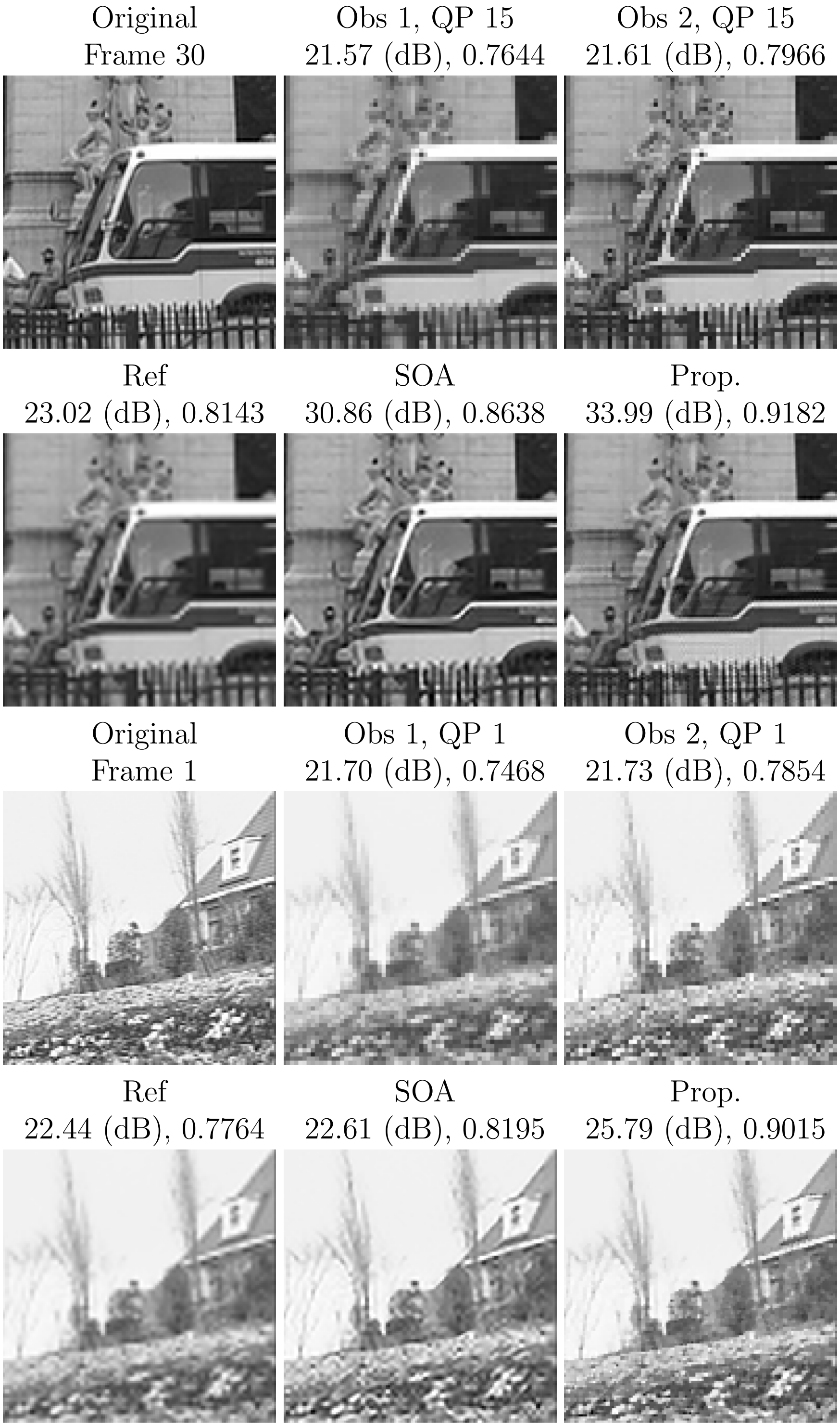
Obs 2, QP 15
22.10 (dB), 0.8378

Ref
23.13 (dB), 0.8341

SOA
24.04 (dB), 0.8908

Prop.
29.50 (dB), 0.9599

Figure 4.9 – Details of the up-sampled images and corresponding results of the super-resolution tested methods on Foreman and Mobile sequences. PSNR and SSIM values are computed on the compared image patch.



Original
Frame 30

Obs 1, QP 15
21.57 (dB), 0.7644

Obs 2, QP 15
21.61 (dB), 0.7966

Ref
23.02 (dB), 0.8143

SOA
30.86 (dB), 0.8638

Prop.
33.99 (dB), 0.9182

Original
Frame 1

Obs 1, QP 1
21.70 (dB), 0.7468

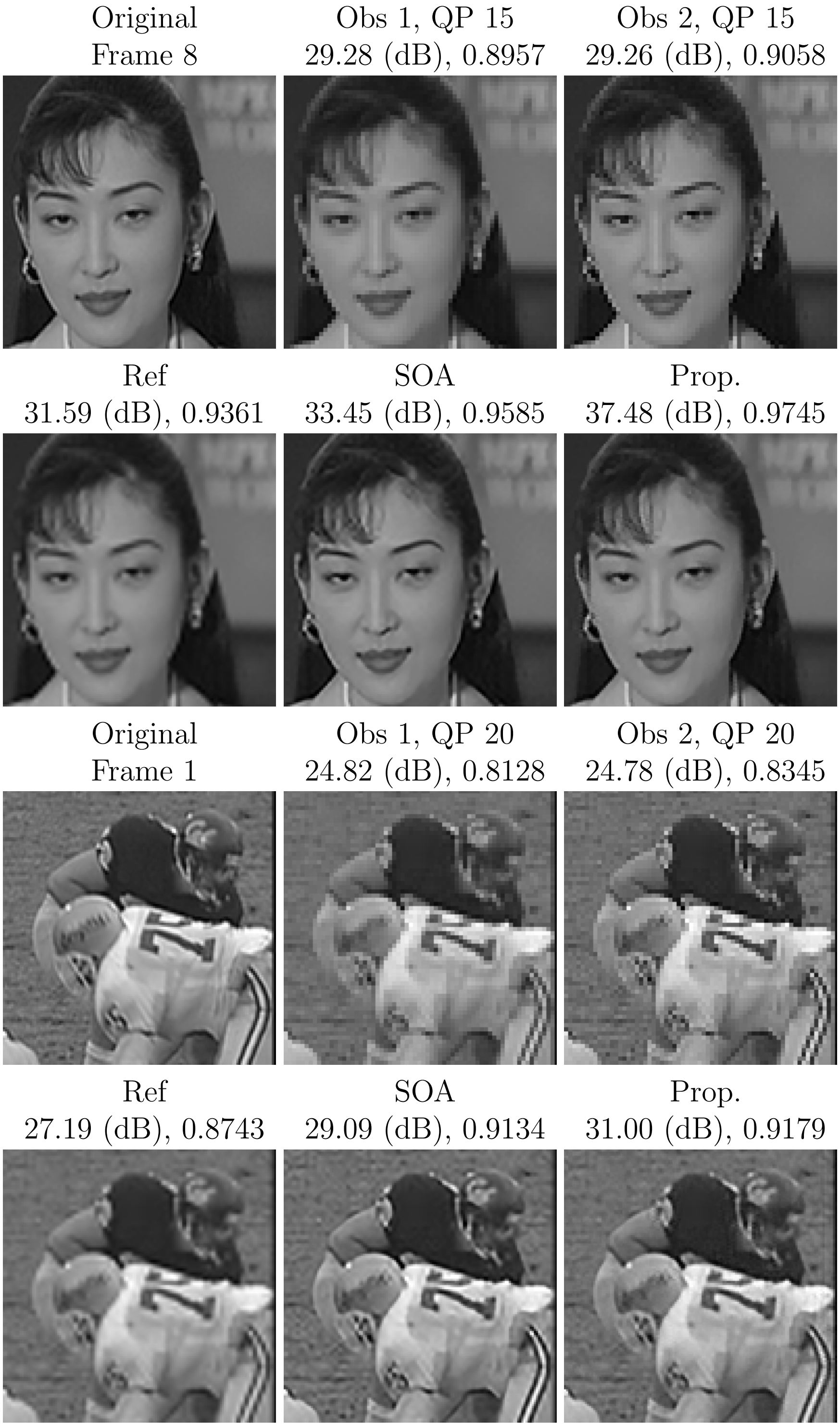
Obs 2, QP 1
21.73 (dB), 0.7854

Ref
22.44 (dB), 0.7764

SOA
22.61 (dB), 0.8195

Prop.
25.79 (dB), 0.9015

Figure 4.10 – Details of the up-sampled images and corresponding results of the super-resolution tested methods on Bus and Flower sequences. PSNR and SSIM values are computed on the compared image patch.



Original
Frame 8

Obs 1, QP 15
29.28 (dB), 0.8957

Obs 2, QP 15
29.26 (dB), 0.9058

Ref
31.59 (dB), 0.9361

SOA
33.45 (dB), 0.9585

Prop.
37.48 (dB), 0.9745

Original
Frame 1

Obs 1, QP 20
24.82 (dB), 0.8128

Obs 2, QP 20
24.78 (dB), 0.8345

Ref
27.19 (dB), 0.8743

SOA
29.09 (dB), 0.9134

Prop.
31.00 (dB), 0.9179

Figure 4.11 – Details of the up-sampled images and corresponding results of the super-resolution tested methods on Akiyo and Football sequences. PSNR and SSIM values are computed on the compared image patch.

### 4.5.3 Convergence speed

In Figure 4.12 we show the PSNR of the estimated HR image and the distance to the convex set $C$ at each iteration of the solver. The distance to the convex set $C$ in this case was computed as the absolute error between the transform of the down-sampled estimate and its projection on the convex set $C$. The test was performed on Akiyo sequence, frame 10, from two descriptions using BicAA and BicNAA down-samplings encoded with QP 20. For the sake of brevity, we do not show other examples, as the behavior is similar across different sequences and compression levels. In this case 97 iterations were performed before the algorithm was stopped. The stop criterion used in our experiments is:

$$\text{Stop if:} \quad \text{mean}\left(|x_i - x_{i-1}|\right) \leq 10^{-4} \tag{4.33}$$

where $||$ denotes the absolute value and mean is the average value of a pixel. *I.e.* the average variation of a pixel is less than $10^{-4}$. Of course, the number of iterations can be increased by modifying the threshold. However, we found that in most tests 80% of the maximum gain was obtained in the first 30 to 40 iterations. Note that the PSNR is still increasing when the distance to set $C$ is 0, as the cost function uses multiple constraints.

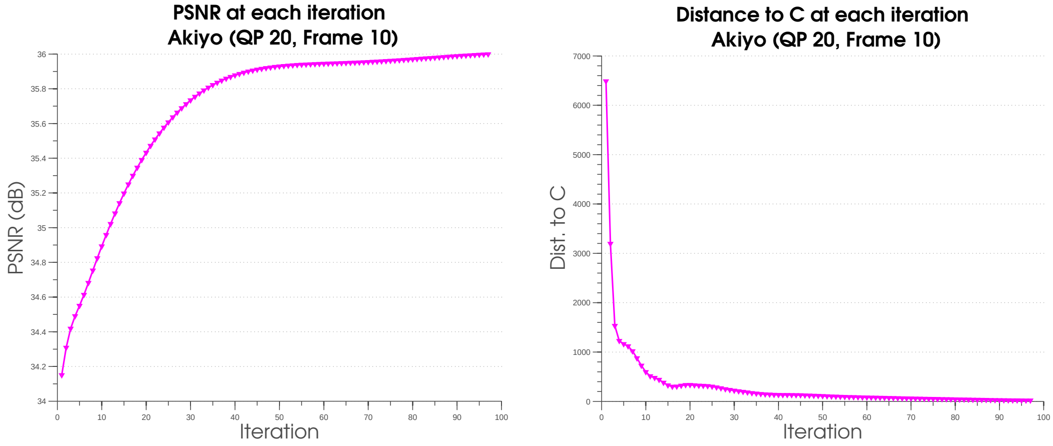


Figure 4.12 – PSNR and distance to the convex set $C$ for each iteration. The distance represents the absolute error between the transformed, LR, estimated image and its projection onto $C$.

The average time per iteration with a Matlab sequential implementation, for two LR descriptions, on a workstation with core I7-6700 processor was 0.5 seconds. Thus, one frame can be super-resolved in 25 to 50 seconds with 50 to 100 iterations.

Note that this time does not include the initialization with SOA. However, using $H$ to up-sample the observation has a negligible computation time. Depending on the usage scenario, the algorithm can be limited to a relatively small number of iterations (10-20) with a reduced gain and a lower computational time. A C++, optimized implementation would further reduce the run time. Furthermore, the algorithm can be easily parallelized for a multi-core implementation. Up to five times runtime reduction was reported for similar proximal optimization algorithms when parallelized [GCPP12].

| | | | QP1 | | | QP15 | | | QP25 | | | QP35 | | | Average | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Sequence | Mode | Ref | SOA | Prop. | Ref | SOA | Prop. | Ref | SOA | Prop. | Ref | SOA | Prop. | Ref | SOA | Prop. |
| PSNR (dB) | Akiyo | II | 34.13 | 36.27 | **41.16** | 34.06 | 36.09 | **38.97** | 33.65 | 35.16 | **36.33** | 31.56 | 31.69 | **31.88** | 33.35 | 34.80 | **37.08** |
| | | IP | 34.13 | 36.27 | **41.12** | 34.06 | 36.09 | **38.93** | 33.66 | 35.16 | **36.33** | 31.57 | 31.72 | **31.96** | 33.36 | 34.81 | **37.09** |
| | Foreman | II | 31.84 | 33.04 | **38.70** | 31.80 | 32.94 | **36.45** | 31.41 | 32.20 | **33.94** | 29.40 | 29.28 | **29.74** | 31.11 | 31.87 | **34.71** |
| | | IP | 31.84 | 33.04 | **38.61** | 31.80 | 32.94 | **36.38** | 31.43 | 32.22 | **33.95** | 29.53 | 29.44 | **29.90** | 31.15 | 31.91 | **34.71** |
| | Bus | II | 25.71 | 26.60 | **31.54** | 25.70 | 26.58 | **30.01** | 25.60 | 26.36 | **28.15** | 24.74 | 24.94 | **25.48** | 25.44 | 26.12 | **28.79** |
| | | IP | 25.71 | 26.60 | **31.52** | 25.70 | 26.58 | **29.98** | 25.61 | 26.37 | **28.13** | 24.82 | 25.01 | **25.56** | 25.46 | 26.14 | **28.80** |
| | Mobile | II | 22.72 | 23.63 | **27.93** | 22.72 | 23.61 | **26.75** | 22.68 | 23.50 | **25.23** | 22.19 | 22.49 | **23.17** | 22.58 | 23.31 | **25.77** |
| | | IP | 22.72 | 23.63 | **27.92** | 22.72 | 23.61 | **26.73** | 22.68 | 23.50 | **25.19** | 22.20 | 22.51 | **23.21** | 22.58 | 23.31 | **25.76** |
| | Flower | II | 22.83 | 23.01 | **26.50** | 22.83 | 23.00 | **26.04** | 22.79 | 22.93 | **24.84** | 22.49 | 22.40 | **23.22** | 22.74 | 22.84 | **25.15** |
| | | IP | 22.83 | 23.01 | **26.50** | 22.83 | 23.00 | **26.03** | 22.80 | 22.93 | **24.80** | 22.51 | 22.43 | **23.23** | 22.74 | 22.84 | **25.14** |
| | Football | II | 30.41 | 32.09 | **36.25** | 30.39 | 32.01 | **34.34** | 30.08 | 31.30 | **32.34** | 28.06 | 28.06 | **28.29** | 29.73 | 30.87 | **32.81** |
| | | IP | 30.41 | 32.09 | **36.23** | 30.39 | 32.01 | **34.31** | 30.11 | 31.32 | **32.37** | 28.20 | 28.15 | **28.41** | 29.78 | 30.89 | **32.83** |
| Average | | | **27.94** | **29.11** | **33.66** | **27.91** | **29.04** | **32.08** | **27.71** | **28.58** | **30.13** | **26.44** | **26.51** | **27.00** | **27.50** | **28.31** | **30.72** |
| SSIM | Akiyo | II | .9639 | .9779 | **.9839** | .9603 | .9725 | **.9737** | .9465 | .9514 | **.9524** | .8924 | .8799 | **.8837** | .9408 | .9454 | **.9484** |
| | | IP | .9639 | .9779 | **.9836** | .9603 | .9725 | **.9735** | .9465 | .9514 | **.9528** | .8927 | .8803 | **.8848** | .9409 | .9455 | **.9487** |
| | Foreman | II | .9343 | .9511 | **.9680** | .9318 | .9458 | **.9512** | .9087 | .9107 | **.9162** | .8271 | .8085 | **.8185** | .9005 | .9040 | **.9135** |
| | | IP | .9343 | .9511 | **.9675** | .9319 | .9459 | **.9508** | .9097 | .9112 | **.9172** | .8315 | .8121 | **.8225** | .9019 | .9051 | **.9145** |
| | Bus | II | .8483 | .8884 | **.9470** | .8471 | .8857 | **.9226** | .8344 | .8643 | **.8820** | .7534 | .7632 | **.7651** | .8208 | .8504 | **.8792** |
| | | IP | .8483 | .8884 | **.9467** | .8472 | .8858 | **.9217** | .8355 | .8648 | **.8824** | .7599 | .7674 | **.7693** | .8227 | .8516 | **.8800** |
| | Mobile | II | .7896 | .8580 | **.9334** | .7887 | .8557 | **.9007** | .7805 | .8374 | **.8542** | .7193 | .7407 | **.7412** | .7695 | .8230 | **.8574** |
| | | IP | .7896 | .8580 | **.9331** | .7887 | .8557 | **.8999** | .7805 | .8374 | **.8539** | .7205 | .7416 | **.7431** | .7698 | .8232 | **.8575** |
| | Flower | II | .8282 | .8633 | **.9259** | .8270 | .8611 | **.9110** | .8205 | .8493 | **.8820** | .7888 | .8049 | **.8193** | .8161 | .8446 | **.8846** |
| | | IP | .8282 | .8633 | **.9256** | .8270 | .8611 | **.9107** | .8206 | .8494 | **.8814** | .7902 | .8065 | **.8197** | .8165 | .8451 | **.8844** |
| | Football | II | .9131 | .9386 | **.9633** | .9115 | .9354 | **.9454** | .8955 | .9091 | **.9101** | .7894 | .7783 | **.7762** | .8773 | .8903 | **.8988** |
| | | IP | .9131 | .9385 | **.9630** | .9116 | .9354 | **.9451** | .8967 | .9094 | **.9110** | .7954 | .7799 | **.7784** | .8792 | .8908 | **.8994** |
| Average | | | **.8796** | **.9129** | **.9534** | **.8778** | **.9094** | **.9339** | **.8646** | **.8872** | **.8996** | **.7967** | **.7969** | **.8018** | **.8547** | **.8766** | **.8972** |

Table 4.5 – PSNR comparison of the reference method, SOA [TSG15] and proposed framework, when two low resolution observations are available. These results were obtained using generic VC compression.

| QPs | $\uparrow_H Obs_1$ | $\uparrow_{SOA} Obs_1$ | $Obs_2$ | $\Delta_{Obs}$ | Prop. | $\Delta$ |
|---|---|---|---|---|---|---|
| 15 20 | 26.41 | 28.56 | 43.76 | **17.35** | 44.45 | **0.69** |
| 1 20 | 26.43 | 28.63 | 43.76 | **17.32** | 45.23 | **1.47** |
| 1 25 | 26.43 | 28.64 | 39.33 | **12.9** | 41.45 | **2.12** |
| 1 30 | 26.43 | 28.63 | 35.15 | **8.72** | 37.63 | **2.48** |

Table 4.6 – PSNR (dB) comparison of different QP combinations for a low-resolution and a high-resolution description on Bus sequence with generic VC using II configuration.

| | | | QPs 1 & 40 | | | QPs 15 & 40 | | | Average | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Sequence | Mode | Ref | SOA | Prop. | Ref | SOA | Prop. | Ref | SOA | Prop. |
| PSNR (dB) | akiyo | II | 34.9 | 36.69 | **37.81** | 34.81 | 36.49 | **37.24** | 34.86 | 36.59 | **37.52** |
| | | IP | 35.29 | 37.09 | **38.3** | 35.18 | 36.84 | **37.42** | 35.24 | 36.97 | **37.86** |
| | foreman | II | 32.64 | 33.75 | **36.04** | 32.57 | 33.62 | **35.37** | 32.6 | 33.69 | **35.7** |
| | | IP | 32.28 | 33.25 | **35.19** | 32.16 | 33 | **34.21** | 32.22 | 33.12 | **34.7** |
| | bus | II | 28.1 | 29.38 | **30.31** | 28.08 | 29.33 | **30.19** | 28.09 | 29.36 | **30.25** |
| | | IP | 27.43 | 28.63 | **29.36** | 27.39 | 28.54 | **29.13** | 27.41 | 28.59 | **29.24** |
| | mobile | II | 25.23 | 26.54 | **27.3** | 25.22 | 26.52 | **27.34** | 25.23 | 26.53 | **27.32** |
| | | IP | 25.11 | 26.35 | **27.02** | 25.09 | 26.31 | **27.03** | 25.1 | 26.33 | **27.03** |
| | football | II | 28.93 | 30.59 | **31.69** | 28.9 | 30.53 | **31.53** | 28.91 | 30.56 | **31.61** |
| | | IP | 27.81 | 29.22 | **29.47** | 27.77 | 29.12 | **29.19** | 27.79 | 29.17 | **29.33** |
| | flower | II | 25.9 | 26.54 | **27.22** | 25.89 | 26.52 | **27.23** | 25.89 | 26.53 | **27.23** |
| | | IP | 25.34 | 25.91 | **26.53** | 25.32 | 25.88 | **26.55** | 25.33 | 25.9 | **26.54** |
| Average | | | **29.08** | **30.33** | **31.35** | **29.03** | **30.23** | **31.03** | **29.06** | **30.28** | **31.19** |
| SSIM | akiyo | II | 0.9517 | 0.9647 | **0.9726** | 0.9483 | 0.9597 | **0.9637** | 0.95 | 0.9622 | **0.9681** |
| | | IP | 0.9545 | 0.9664 | **0.9744** | 0.9511 | 0.9615 | **0.9645** | 0.9528 | 0.964 | **0.9695** |
| | foreman | II | 0.9127 | 0.9312 | **0.9509** | 0.9087 | 0.9246 | **0.9343** | 0.9107 | 0.9279 | **0.9426** |
| | | IP | 0.9072 | 0.9252 | **0.9378** | 0.9008 | 0.9144 | **0.9124** | 0.904 | 0.9198 | **0.9251** |
| | bus | II | 0.8278 | 0.8684 | **0.9029** | 0.8255 | 0.8647 | **0.8878** | 0.8267 | 0.8665 | **0.8953** |
| | | IP | 0.8204 | 0.8614 | **0.8914** | 0.8158 | 0.8539 | **0.8696** | 0.8181 | 0.8576 | **0.8805** |
| | mobile | II | 0.8494 | 0.8902 | **0.9065** | 0.8483 | 0.8883 | **0.9009** | 0.8488 | 0.8892 | **0.9037** |
| | | IP | 0.8565 | 0.8947 | **0.9046** | 0.8546 | 0.8912 | **0.8975** | 0.8556 | 0.893 | **0.901** |
| | football | II | 0.83 | 0.8743 | **0.9116** | 0.8279 | 0.8708 | **0.8953** | 0.829 | 0.8726 | **0.9035** |
| | | IP | 0.8081 | 0.8551 | **0.8788** | 0.8043 | 0.8486 | **0.8542** | 0.8062 | 0.8518 | **0.8665** |
| | flower | II | 0.8849 | 0.9056 | **0.9167** | 0.8836 | 0.9033 | **0.9104** | 0.8842 | 0.9045 | **0.9135** |
| | | IP | 0.8783 | 0.8992 | **0.9089** | 0.8763 | 0.8954 | **0.9012** | 0.8773 | 0.8973 | **0.905** |
| Average | | | **0.8735** | **0.903** | **0.9214** | **0.8704** | **0.898** | **0.9076** | **0.8719** | **0.9005** | **0.9145** |

Table 4.7 – PSNR comparison of the reference method, SOA and proposed framework, when one low resolution and one high resolution observations are available. These results were obtained using HEVC compression.

## 4.6 Conclusions

This work presents a model-based SR approach specifically designed for compressed video streams, and focuses on scenarios where multiple observations are available. The proposed model makes explicit use of the available compressed syntax (encoded coefficients, unit sizes, etc.) and builds a heterogeneous cost function combining data-fidelity objectives and a priori constraints. The resulting minimization problem, efficiently solved via convex optimization, embeds the SR result into a domain that closely fits the given compressed observations. Experimental results demonstrate that in most cases, combining the complementary information available in the different observations allows very efficient SR, significantly outperforming the capabilities of single image SR [TSG15]. Indeed, quality improvements superior to 5dB w.r.t. one of the best performing learning-based single image SR method can be observed for high-quality encodings, which has a noticeable impact on the visual quality of the reconstructed video sequence. The flexibility of the proposed framework is also to be highlighted. First, an arbitrary number of observations can be considered. Second, each observation is modeled with its own degradation model, allowing to combine observations at different resolutions and smoothness characteristics. Such an explicit modeling avoids a typical pitfall of learning-based approaches whose performance may dramatically vary depending on the re-sampling used to generate the observation. Third, the approach is independent of the video coder used, which is illustrated in the present work using both a generic coding model VC and the HEVC standard. Extend the framework application to other compression schemes (JPEG, JPEG2000, VC9, etc.) is straightforward. Yet, short-term research focuses on discussing more thoroughly the complexity and real-time capabilities of the proposed framework, requiring the implementation and optimization of the convex solver on parallel processing platforms.

# Chapter 5

# A railroad detection algorithm for infrastructure surveillance using enduring airborne systems

## Contents

Infrastructure surveillance is an important requirement for many companies. With the advancement of technology, drones can now provide an efficient tool for such applications. A possible future scenario is the automated surveillance of railroads. Numerous algorithms that provide railroad detection exist. However, they are not well suited for this particular scenario. Some algorithms provide railroads extraction functionality for satellite images while others are better suited for small, low altitude drones and ground level acquired pictures. In this work we propose a railroad detection algorithm suited for larger enduring drones. We use Hough Transform to detect lines and perform a line clustering in the $\rho$ and $\theta$ space. A score model is proposed in

order to identify the railroad. We test our method on several sequences supplied by Airbus Defense & Space and show our algorithm to provide a detection rate of 93.23% in average.

## 5.1 Introduction

With the advancement of technology, new opportunities arise in the field of video surveillance. As drones are no longer limited to military applications and are even available as entertainment devices that can be controlled through modern mobile phones, automatic video surveillance of infrastructures is a real possibility. This is also the goal of the SURICATE project (SUrveillance de Reseaux et d'InfrastruCtures par des systemes AeroporTes Endurants), which proposes the use of Unmanned Aerial Vehicles (UAV) for the surveillance of infrastructures such as railroads or electrical lines. This work is centered around these ideas and tackles a specific scenario: the surveillance of railroads using enduring drones.

Railroad and road detection is a known problem in image processing and a large number of methods exists that propose solutions for various usage scenarios. A first use case scenario is that of roads and railroads detection in satellite images. Radu Stoica *et al.* propose an algorithm based on a Monte Carlo dynamics for finite point processes [SDZ04]. Mohammadzadeh *et al.* use a few samples from road surface and apply a particle swarm optimization to a fuzzy-based mean calculation system in order to obtain road mean values in each band of high resolution satellite color images. However, this type of scenarios are inherently different from detecting railroads or roads in images or videos acquired by drones. Our scenario requires a less complex approach and it is desired to be as close as possible to real time usage, as the algorithm will be used for tracking and detection purposes, either for tracking the railroad with the on board camera or providing additional data that can be used for drone orientation and flight control.

Other types of algorithms use feature extraction in order to detect railroads in pictures [TLZ16]. A large number of methods for generating features exist, some of the more popular include Histogram of Gradients (HOG) [NB05] or Scale-invariant feature transform (SIFT) [Low99]and [BAS08]. However, object detection usually requires the use of learning algorithms such as Support Vector Machines (SVM) [PRDD10].

Pali *et al.* [PMTB14] propose to use Probabilistic Hough Transformation (PHT) [KEB91] to determine the vanishing point of the railroad. They use this method to guide a small drone along a railroad. However, in our scenario the vanishing point cannot be determined as the images are acquired from a higher altitude.

In this work we propose a Hough Transform (HT) based algorithm to detect railroads. We perform a clustering with respect to $\rho$ and $\theta$ in the HT. The cluster selection is performed using a scoring technique that takes into account the geometrical properties of the railroad and the length of the detected lines. We test our method using several test video sequences acquired by Airbus Defense & Space in the framework of SURICATE project. The rest of this chapter is organized as follows: Section 5.2 describes the proposed algorithm, in Section 5.3 we show and discuss our results and Section 5.4 concludes this chapter.

## 5.2 Method description

In this section we describe our proposed algorithm. As previously discussed, we aim at providing a robust and fast railway detection method that can be used on board UAVs for tracking and orientation purposes during infrastructure surveillance. In order to achieve this, we use a sequential algorithm where each module's output is the input of the next. The whole algorithm can also be divided into two larger modules: a line detection block and a line selection one.

In Figure 5.1 we depict the general scheme of the proposed algorithm. Our method can be divided into 7 steps, starting from the input image and finalizing with the detected lines coordinates. The first step in the algorithm is an edge detection. There are numerous algorithms that can be used for edge detection. Some of the more popular ones, that have been proven over time are: Laplace, Sobel and Canny methods [SSA13]. The Laplace and Sobel methods are based on a gradient filter edge detector. They rely on the fact that the second derivative of a function is zero when the first derivative is at a maximum. The Laplacian method searches for zero crossings in the second derivative which can be approximated by a convolution with the following mask:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

The Sobel algorithm performs a convolution with two masks. A horizontal and a vertical one:

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

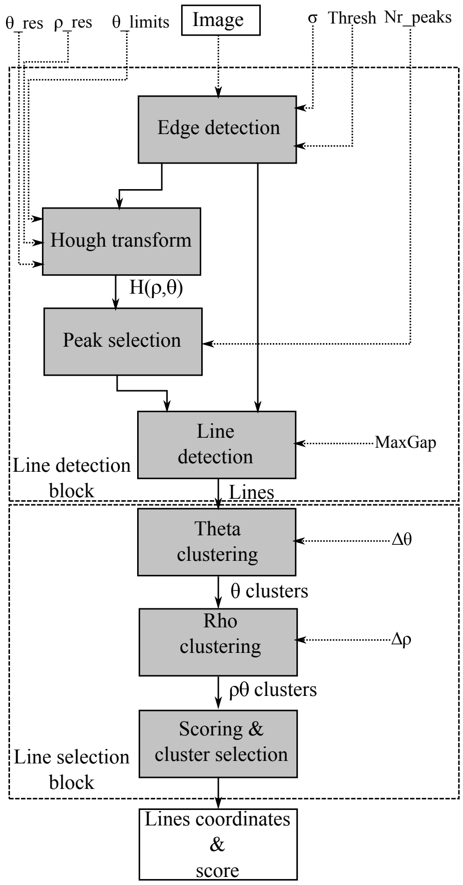


Figure 5.1 – Algorithm general scheme. Dotted lines indicate input data.

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Unlike the previous methods Canny uses multiple steps. It applies the same masks as Sobel method after performing a smoothing of the image with a Gaussian filter. The final step consists in edge thinning and thresholding. This method is less susceptible to noise and for this reason it is preferred over the first two. However, the selection of $\sigma$ and the threshold parameter Thresh parameters plays an important role and should be carefully balanced. Too much smoothing may lead to a loss of useful information, while no smoothing will lead to noise in the edge detection step. The Thresh parameter has a similar behavior, as increasing it might lead to loss of contours and having a very small value may take into account insignificant edges.

The next step of the algorithm is the Hough Transform (HT) [Hou62] [DH72]. This is a well known method for detecting straight lines in images. Each pixel in the image is mapped into a parameter space to a set of lines potentially passing through it:

$$x \cdot sin(\theta) + y \cdot cos(\theta) = \rho \tag{5.1}$$

where $\rho$ is the distance from the origin of the image's coordinate system and the line and $\theta$ is the angle of the line with respect to the $x$ axis. The method is very robust and is known to work even with noisy data. Several variants of the transform exist such as those described by Leavers in [Lea93] or [KEB91]. Fernandes and Oliveira propose in [FO08] a real time implementation for HT through an improved voting scheme. Frame rates of up to 52.63 are reported.

The next two steps in the pipeline identify the lines in the image. Firstly, a selection of peaks is performed in the transform space and then lines are identified for each pair of $(\rho, \theta)$. The MaxGap parameter is used to set the maximum accepted discontinuity, in the binary edge image, when identifying a line (i.e. the binary edge image may have discontinous lines at $(\rho, \theta)$, discontinuities smaller than MaxGap pixels are filled). Once a set of lines is identified with corresponding start/end points and associated $(\rho, \theta)$ pairs, the list is passed to the next block which performs the line selection that best characterizes a railroad in the given context (UAV infrastructure surveillance).

The second part of our method is comprised of three modules. Two clustering modules for $\theta$ and $\rho$ and a scoring and cluster selection module. In what follows we will describe each module in detail and discuss the particularities and issues that can be encountered.

A first thing to notice is that the clustering is performed in two steps as opposed to running a clustering algorithm, for all $(\rho, \theta)$ pairs, such as k-means [KMN$^+$02]. The reason behind this is that each of the two parameters is bound by a specific condition. Performing this analysis separately allows us to identify the clusters efficiently by searching for a maximum with respect to the frequency of lines at each $\theta$ and $\rho$ interval.

In the case of $\theta$, we know that railroads are parallel so lines belonging to a railroad should have the same angle. Of course, due to the nature of the HT, several concurrent lines can be identified for each rail instead of two parallel lines for each rail. This is due to peaks in the HT transform that are not always isolated. These effects are caused by the quality of the image, the precision of the edge detection method or simply by the resolution with which the HT was computed. Therefore, a small variation should be allowed for lines that belong to a railroad. This is denoted in Figure 5.1 by $\Delta\theta$. Identifying the $\theta$ line clusters is now simply a matter of searching for $\theta$ intervals with a high frequency of lines in a histogram computed over a quantization of the $\theta$ search domain ($\theta_{limits}$). The minimum quantization step in this case is given by $\theta_{res}$ and the maximum step should not be higher than $\Delta\theta$. Once a cluster is identified the lines are suppressed and the procedure is repeated. The number of clusters should be limited manually and also automatically in order to avoid relatively small clusters with respect to the frequency of the peak intervals. A good form for this threshold is:

$$F(\theta_{min}^{C_k}, \theta_{max}^{C_k}) > \tau \cdot F(\theta_{min}^{C_1}, \theta_{max}^{C_1}) \tag{5.2}$$

where, $(\theta_{min}^{C_k}, \theta_{max}^{C_k})$ is the $\theta$ interval of the cluster, $F$ returns the number of lines in the interval, $C_1$ is the first identified cluster which has the highest line frequency and $\tau$ is a constant between 0 and 1.

Once a set of $\theta$ clusters is identified we can proceed to separating each one into multiple clusters with respect to $\rho$. The procedure is similar to the $\theta$ case and differs in the selection of $\Delta\rho$. As rails are equally spaced, $\Delta\rho$ can be empirically determined for a given scenario or estimated using the drone camera parameters and altitude. In a similar manner with $\theta$ clustering, a stopping criterion can be expressed as:

$$F(\rho_{min}^{C_k}, \rho_{max}^{C_k}) > \tau \cdot F(\rho_{min}^{C_1}, \rho_{max}^{C_1}) \tag{5.3}$$

The final step of the pipeline is an analysis of the clusters and a selection of the best matching ones. The first thing required is to define what makes a cluster of lines most likely to belong to a railroad. For this purpose we propose computing a score for each cluster depending on the length of the lines and the variation of $\theta$ and $\rho$ within each one. We will separate this score into three intermediary scores: $S_\theta$, $S_\rho$ and $S_{ll}$ (line lengths).

The first score $S_\theta$ should indicate the similarity of the lines angles within the cluster and also take into account the number of lines within the cluster. Even though, the angles are limited to an interval, less variation should indicate a better match. Let us consider the following formulation for a single line $\theta$ score:

$$s_\theta^{C_k}(j) = \frac{\Delta\theta + \sum\limits_{i=1}^{N} |\theta^{C_k}(i) - \theta^{C_k}(j)|}{N} \tag{5.4}$$

where, $k$ denotes the cluster, $N$ is the total number of lines within the cluster and $||$ is the absolute value. $\Delta\theta$ assures a non-zero score. This is an indication of how similar the angles are within the cluster (high value indicates increased angle variation). The

$S_\theta$ score for cluster $C_k$ can now be expressed as:

$$S_\theta^{C_k} = \sqrt{\frac{N \cdot \sum\limits_{i=1}^{N} ll^{C_k}(i)}{\sum\limits_{i=1}^{N} s_\theta^{C_k}(i) \cdot ll^{C_k}(i)}} \tag{5.5}$$

where, $ll^{C_k}(i)$ is the length of the line $i$ in cluster $C_k$. This value can be interpreted as the geometric mean between the number of lines and the inverse of the $s_\theta^{C_k}$ weighted average with the length of the lines. Longer lines should be given more weight and a high number of lines increase the reliability of the detection.

A similar set of operations can be performed for $\rho$ values in order to obtain $S_\rho^{C_k}$. Similarly to $\theta$ the lines should be relatively close to each other. We can define $s_\rho^{C_k}(j)$:

$$s_\rho^{C_k}(j) = \frac{\Delta\rho + \sum\limits_{i=1}^{N} |\rho^{C_k}(i) - \rho^{C_k}(j)|}{N} \tag{5.6}$$

and $S_\rho^{C_k}$:

$$S_\rho^{C_k} = \sqrt{\frac{N \cdot \sum\limits_{i=1}^{N} ll^{C_k}(i)}{\sum\limits_{i=1}^{N} s_\rho^{C_k}(i) \cdot ll^{C_k}(i)}} \tag{5.7}$$

The final component of the score should reflect the lengths of the lines in each cluster. Considering that all clusters contain a high number of small lines (this aspect will be further discussed in the experimental section) we are interested in evaluating only the longer lines as they will provide more information about the structure of the rail. We first select all lines with a length higher than the average line length of the cluster as:

$$\mathcal{L}(ll^{C_k}) = \{i | ll^{C_k}(i) > mean(ll^{C_k})\} \tag{5.8}$$

The line length score $S_{ll}^{C_k}$ can be defined as:

$$S_{ll}^{C_k} = \sqrt{\frac{\sum\limits_{i \in \mathcal{L}(ll^{C_k})} ll^{C_k}(i)}{M} \cdot N} \tag{5.9}$$

where $M$ is the number of elements in $\mathcal{L}(ll^{C_k})$. Finally, score can be written as the geometric average of $S_{ll}^{C_k}$, $S_{\rho}^{C_k}$ and $S_{\theta}^{C_k}$.

$$\mathcal{S} = \sqrt[3]{S_{\theta}^{C_k} \cdot S_{\rho}^{C_k} \cdot S_{ll}^{C_k}} \tag{5.10}$$

The cluster with the highest score is then selected as the railroad.

## 5.3 Experimental results

In this section we present our experimental results and discuss the methodology of the tests and the selection of parameters.

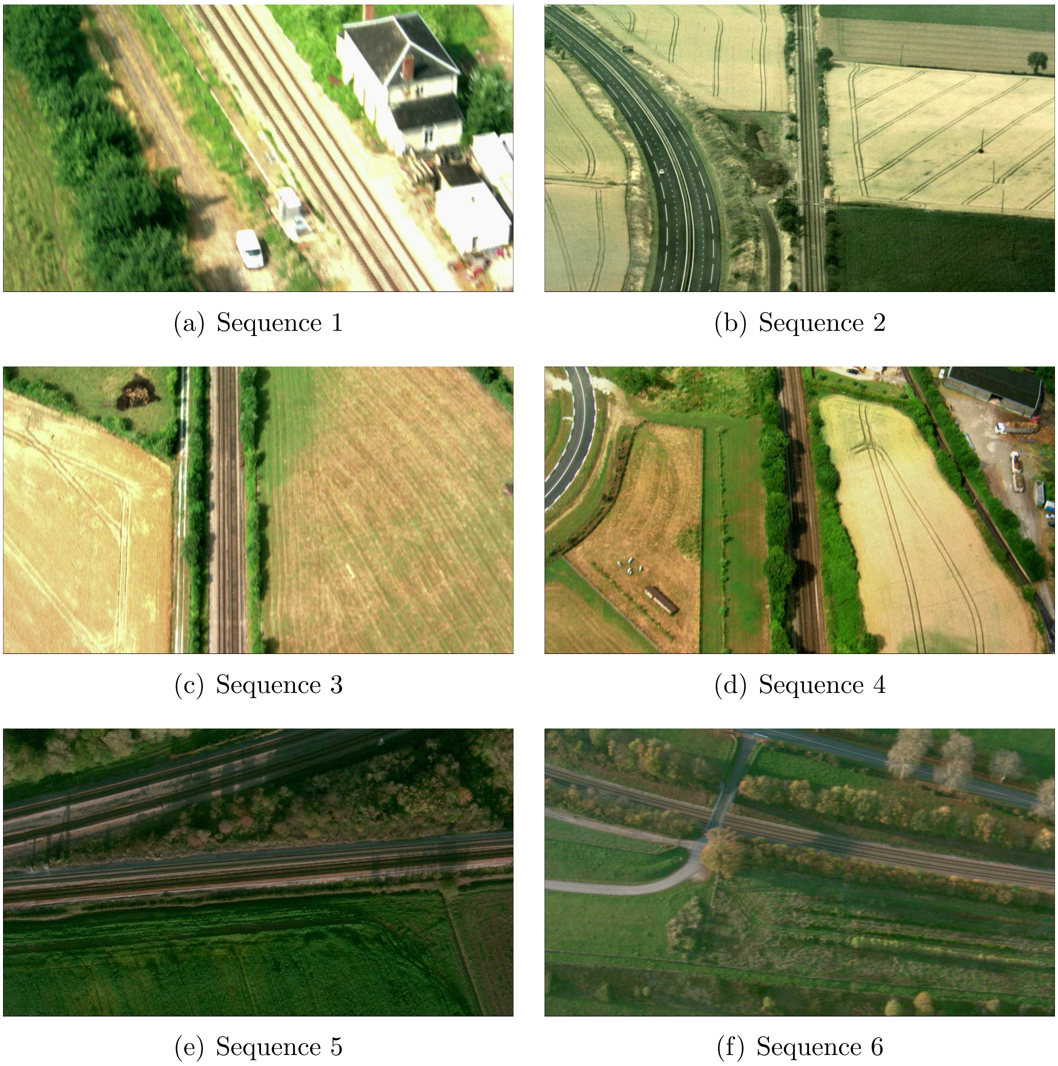
### 5.3.1 Testing data

In order to validate our approach we test the method on a set of video sequences acquired by Airbus Defence & Space in the framework of the SURICATE project. The video sequences were acquired in raw YUV format and contain recordings of railroads located in France. Due to the very large size of the data we extracted several smaller sequences, from various locations, with different content including roads or other geometrical structures similar to railroads. Each short sequence has 300 frames and a resolution of 1920 × 1080. Figure 5.2 shows a typical example of content found in each test sequence.[1]

### 5.3.2 Testing Methodology

For testing purposes we implement our algorithm in Matlab. However, in the future an on-board implementation will most likely be done in order to perform real time testing and calibration of parameters. Each frame is evaluated and the selected line cluster is drew over the texture. The resulting video sequences are then visually evaluated in order to determine the detection rate for each sequence. We consider the line positively detected if the line is located over the railroad and has the correct angle. If the railroad is not entirely detected (e.g. the detected lines cover only a part of the railroad), we consider this case also as a positive detection, as having the $\theta$ and $\rho$ intervals will provide a good indication of the railroad position and relative angle to the drone. All other cases when the detected lines fail to indicate the proper angle of the rail or are located over different structures in the image are considered

[1] We would like to thank Airbus Defense & Space for providing the test sequences used in our experiments.



(a) Sequence 1

(b) Sequence 2

(c) Sequence 3

(d) Sequence 4

(e) Sequence 5

(f) Sequence 6

Figure 5.2 – Test sequences representative frames. Each frame shows the type of content present in each of the test sequences[1].

as false detections. In addition, we will show a step by step run of the algorithm and the intermediary results.

### 5.3.3 Parameter calibration

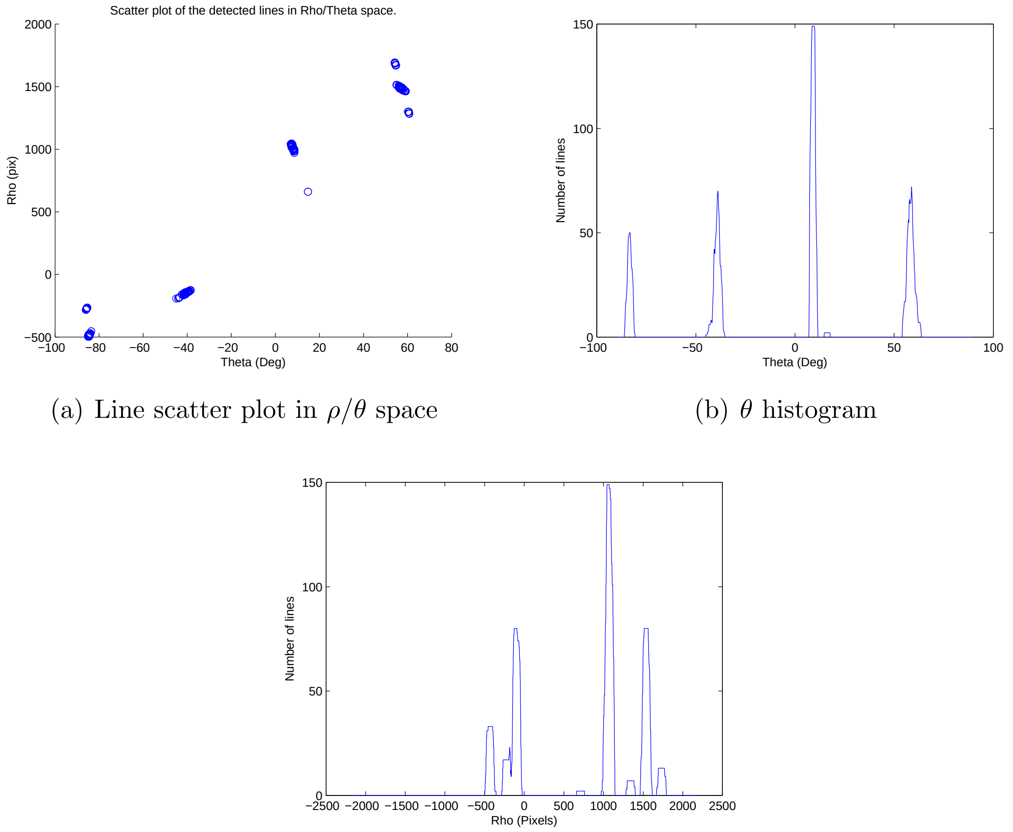
In our tests, we use the same parameters for all test sequences. Although, in the future an automatic calibration is preferred, as some of the parameters are strongly linked with the drone's camera and flight path. A large increase in speed can be obtained by reducing the search domain for $\theta$. In normal conditions, the UAV will have a predefined flight path in close proximity to the railroad. The relative angle of the railroad can be determined with respect to the aircraft by using the GPS and geographical information. This information can be used to drastically reduce the search angle limits ($\theta_{limits}$) and increase the speed and reliability of the detection. However, in our experiments we used the maximum angle span from -90 to 90 degrees, relative to the image x-axis. The HT resolution for $\theta$ and $\rho$ is also dependent on the zoom. Once the railroad is identified the camera may be zoomed in and our algorithm will indicate the relative position of the railroad in the image which can be used for tracking. The resolution in this case may be lowered as the railroad will have a larger size relative to the image size. Also, based on the degree of zoom in, camera parameters and drone altitude, the $\Delta\rho$ can be easily computed as railroads have constant widths. In Table 5.1 we report the parameters used in our tests, note that $\theta_{limits}$ upper limit is 89.6 degrees as 90 and -90 degrees indicate the same direction.

| **Parameter** | **value** | **Parameter** | **value** |
|---|---|---|---|
| $\theta_{res}$ | 0.4 | $\Delta\theta$ | 3 |
| $\rho_{res}$ | 1 | $\Delta\rho$ | 50 |
| $\theta_{limits}$ | [-90, 89.6] | $\sigma$ | 1.4 |
| Nr_peaks | 150 | Thresh | 0.15 |

Table 5.1 – Algorithm parameters used in our tests.

### 5.3.4 Results

In Table 5.2 we report our detection rate. As can be seen, we obtain a very good detection rate for the railroads. An example of line clusters and histograms w.r.t. $\rho$ and $\theta$ is depicted in Figure 5.3 while Figure 5.4 shows an example of the algorithm's behavior for frame 40 of Sequence 2. The detected clusters of lines are depicted, as-well as the edge detection step. In Figure 5.4(c) we show all the detected lines.



(a) Line scatter plot in $\rho/\theta$ space

(b) $\theta$ histogram

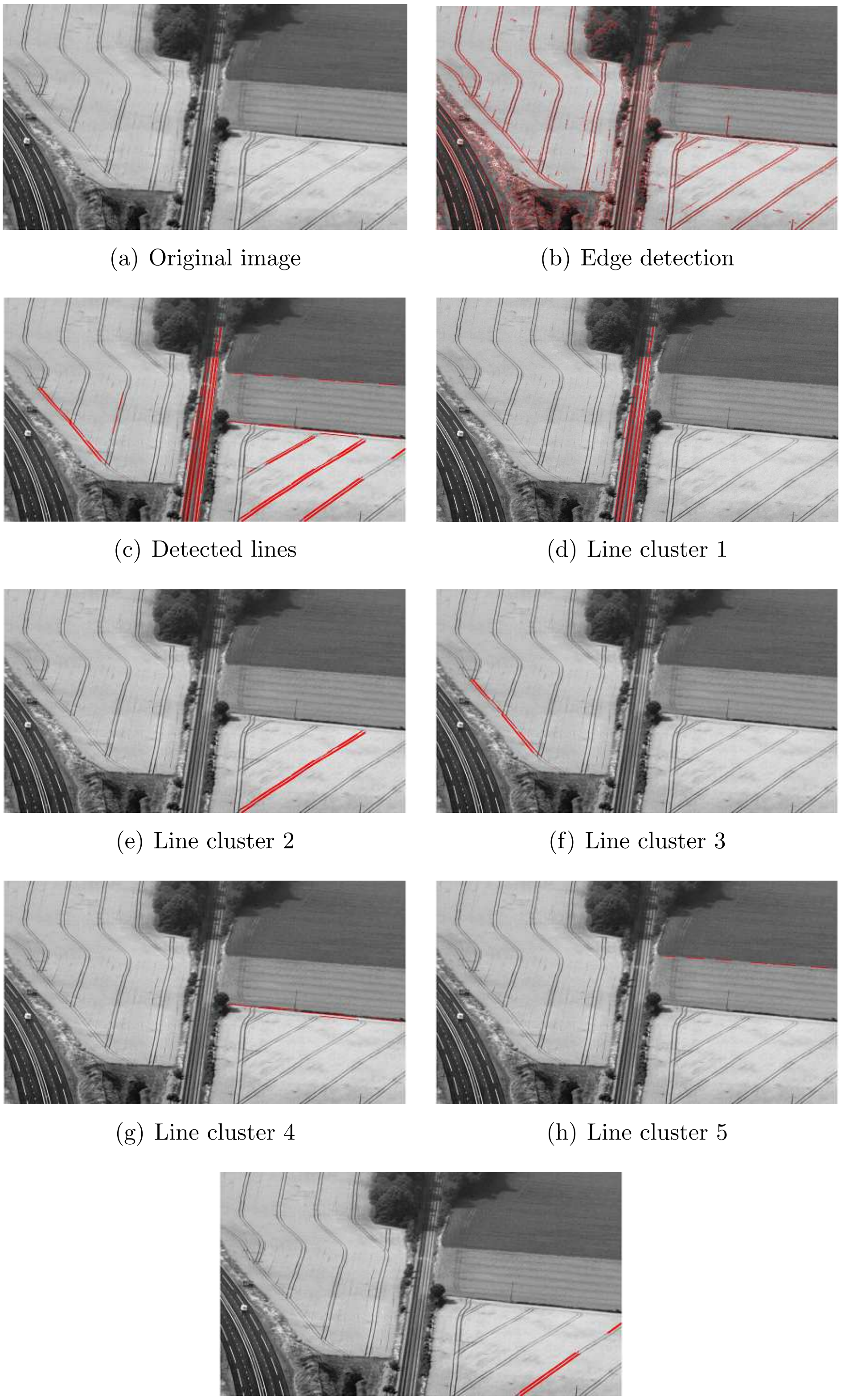
(c) $\rho$ histogram

Figure 5.3 – The line scatter plot in $\rho/\theta$ space 5.3(a) and the lines histograms with respect to Theta 5.3(b) and $\rho$ 5.3(c).

The reported score ($\mathcal{S}$) for the 6 clusters is: 9.0932, 5.9146, 5.9701, 4.6062, 3.6431 and 3.0416. As expected the first cluster which also contains the railroad has the highest score and is selected as the railroad detection.

| **Sequence** | **Det. rate(%)** | **Sequence** | **Det. rate( %)** |
|---|---|---|---|
| Seq. 1 | 99.6 | Seq. 4 | 72,6 |
| Seq. 2 | 96.6 | Seq. 5 | 96.3 |
| Seq. 3 | 94.3 | Seq. 6 | 100 |

Table 5.2 – Positive detection rates on tested sequences.



(a) Original image

(b) Edge detection

(c) Detected lines

(d) Line cluster 1

(e) Line cluster 2

(f) Line cluster 3

(g) Line cluster 4

(h) Line cluster 5

(i) Line cluster 6

Figure 5.4 – An example of the detected lines and clustering process for frame 40 of Sequence 2. White lines indicate the detected lines in the image for Figures 5.4(c)to 5.4(i). Figure 5.4(b) shows the detected edges with with lines.

## 5.4 Conclusions

In this work we presented a railroad detection algorithm for drone surveillance of infrastructure. The method can be used for railroad tracking with the UAV's camera and also for navigational purposes in the case of GPS or connection failure with the drone. We tested the proposed technique using a set of sequences supplied by Airbus, Defense & Space, acquired in the context of the SURICATE project which proposes infrastructure surveillance using UAVs. We were able to obtain a detection rate of 93.23 in average over all tested sequences. The algorithm was also integrated by Airbus Defense & Space with the existing flight control algorithms of a drone. Furthermore, additional improvements can be made by creating a parameter adjustment system with respect to the drone camera and flight information data. Other improvements can be made by taking into account the temporal aspect and estimating the position of the railroad in future frames thus eliminating possible erroneous detections.

# Conclusions and future perspectives

## Thesis objectives

The goal of this thesis is to propose new algorithms for view synthesis in MVD video compression systems and to tackle the video reconstruction problem from multiple compressed video sources. Three topics that answer these requirements were tackled. Firstly, DIBR based view synthesis can be further improved by taking advantage of temporal correlations in a synthesized view. As distortions produced during view synthesis are inherently different from those introduced by video compression, the second target of the thesis was to find new ways of evaluating the quality and performance of view synthesis algorithms. Finally, the third objective of the thesis was to find ways of combining multi-source videos with possibly different resolutions and compression levels, in order to create a high resolution representation with increased quality. Furthermore, as part of the SURICATE project, we also investigated the problem of infrastructure surveillance, we aimed at providing new tools suited to this particular scenario.

## Summary

The following presents a short summary of the contributions and the main concepts behind them.

### View synthesis exploiting temporal correlations

As additional information is available at different time instants of a video sequence, view synthesis algorithms could take advantage of it in addition to geometric information of the scene. We investigated this problem at pixel level using both disparity and motion vector fields by imposing an epipolar constraint to link the two.

A first contribution tackles the disocclusion problem in view synthesis and proposes a Temporal Hole Filling approach (THF). As motion information cannot be reliably estimated from synthesized frames, the epipolar constraint is used to warp MVFs from the reference views in the synthesized view. Furthermore, to avoid matching disocclusions in MVFs and texture we formulate the epipolar constraint using reverse MVFs in the prediction sense (i.e. the current frame serves as a reference for a past frame in ME). As both MC, backward MC and warping operations are required we designed a robust Warping and filtering technique (Wf) that takes into account depth information for both MVF and texture warping.

Our second contribution explores the idea of replacing or combining DIBR synthesis with a blend of temporally predicted frames in the synthesized view. In this scenario we use forward MVFs and a different formulation of the epipolar constraint to generate up to four temporal predicted frames. As there is no way of determining the accuracy of each prediction we use either the average of all predictions or a blending model based on the similarity of the six predictions. It is interesting to note that even though temporal predictions are motion compensated from synthesized frames gains can still be obtained as synthesis distortions vary over time.

Finally, as view synthesis is used as a tool in MVD transmission systems we integrate these approaches with 3D-HEVC and propose a modification of the transmission system. More precisely, we send one additional intra coded frame per GOP, to be used as reference for MC. In this scenario the blend of temporal predictions alone provides a much better quality than DIBR synthesis. However these predictions are still subjected to ME errors in case of high intensity motion. To tackle this problem we formulate an adaptive fusion method that selects between inter-view blend or temporal blend at pixel level. Furthermore, we investigate a hierarchical approach for both temporal and view axes. A significant gain over direct synthesis is obtained.

## ROI based evaluation of view synthesis methods

The key observation behind this approach is that the spatial distribution of high errors within a synthesized image is highly correlated with the structure of the scene unlike compressed images where errors tend to be uniformly distributed. This is explained by the multiple sources of errors that affect synthesized frames and the different way in which they manifest. Therefore, when comparing view synthesis algorithms the evaluation should be focused on areas that are most likely to be affected by these types of artifacts.

The first contribution focuses on evaluating two synthesis methods using a common ROI. A first possibility to determine this ROI is based on the observation that the histogram of absolute errors in an image synthesized from compressed sources presents a secondary peak at higher values. This can be used to determine a threshold for the absolute errors and identify a ROI in each synthesized frames. For fairness of comparison, we merge the two ROIs and evaluate the areas using SSIM.

A second approach determines the ROI by simply selecting pixels which are predicted differently by the two methods. Our experiments show that additional information can be extracted about the behavior of each method when using this evaluation technique.

Our last contribution further extends these ideas for comparing multiple synthesis methods. We use dilation and erosion operations to extend the ROI and eliminate singleton pixels. Furthermore, we investigate other metrics in combination with multiple ROI selection methods and study their performance on a subjective evaluation database for view synthesis methods.

## Super resolution and video reconstruction

Considering the undergoing shift from H.264/AVC compression standard to HEVC, the fast adoption of high resolution displays and the adoption of cloud multimedia services, video sequences are widely available in multiple copies with different specifications.

In this context our contribution provides an efficient way of combining various descriptions of a video in order to obtain a high resolution representation with increased quality. We propose a new model based reconstruction framework that accounts for the particularities of videos compressed with hybrid video coders. A polyphase filter model is used to describe the downsampling or upsampling procedure and derive an operator composed of a simple set of matrix operations.

The framework is adapted for two encoders in our tests, first a generic VC that matches older encoders and HEVC. In order to account for the additional tools used in HEVC, several small adjustments are performed, especially for skipped blocks. Two practical applications are proposed: SR from two low resolution compressed videos and enhancing a HR compressed video from a LR one. The results indicate that our approach provides significant gains over one of the best learning based SR algorithms.

### Railroad detection for drones in SURICATE project

This work was performed in the context of SURICATE project. The aim of the project was to use enduring airborne systems in the surveillance of infrastructure. The challenge we tackled was the detection of railroads for tracking purposes. As this is a rather specific scenario, we were unable to find other methods that tackle this problem in our given parameters. More precisely, we found railroad detection algorithms for ground level pictures or small drones tracking that rely on the vanishing point in order to set a flight path direction. As our scenario involves mid range altitude the conditions are quite different. We decided to use a model based approach based on Hough transform. In order to perform the detection we create a score model for clusters of lines detected with Hough transform. The highest scored cluster is identified and its direction and position in the image are used to provide flight path information.

## Future work and perspectives

Several future work directions can be identified. From overcoming implementation issues and further improving each algorithm to tackling new challenges in each of the studied topics.

A first continuation of the work performed in this thesis is related to implementation aspects. With the exception of the railroad detection algorithm, which was integrated by Airbus Defense & Space with the existing flight control algorithms of a drone, all methods were implemented using Matlab and to a limited extent C++, as experimental testing frameworks. As such, speed optimization combined with a modular and robust implementation is required to move from experimental frameworks to easily usable software tools.

The proposed view synthesis methods have a high complexity due to the optical flow computations. As such, finding faster ways to perform ME can greatly reduce the computational time of VSTP. Another possibility to avoid this issue is to use the already available motion predictors in HEVC, a more detailed study of how motion estimation precision affects the end result of the synthesis is an interesting aspect to consider. Furthermore, the disocclusion problem is still an open matter, developing new inpainting techniques that take into account scene geometry in addition to texture patches is a subject of relevance for immersive video. Finally, these inpainting techniques can be combined with our adaptive fusion method in order to obtain a temporal and spatial consistent filling by taking advantage of the additional predictions.

The quality evaluation methods can be further extended to provide a full metric rather than a relative comparison of multiple methods, by finding a good balance between the ROI evaluation score and that of the non-ROI. Also, subjective tests can be performed in order to find a better criterion in ROI selection. Furthermore, the study of subjective evaluation methodology for video formats that enable FTV w.r.t. displaying methods, may provide future insights on establishing the impact of view synthesis distortions on perceived quality and finding which technology provides the best 3D video experience.

Another interesting possibility is the application of SR and video reconstruction methods in compression. Finding solutions for GPU parallel implementation of proximal splitting based convex optimization algorithms, may enable the use of SR and video reconstruction for real-time video decoding.

Finally, the railroad detection algorithm could benefit from a temporal consistency mechanism. More precisely, we know that the change in position of a railroad is relatively limited from frame to frame, and thus incorrect detections can be eliminated by limiting the change in angle and position of the detected line cluster w.r.t. the past $N$ frames. Considering the needs of railroad companies, future research should also focus on algorithms that can identify line obstructions in order to provide a real-time warning system.

# Publications

## Journal articles

1. A. Purica, E. G. Mora, M. Cagnazzo, B. Pesquet-Popescu, and B. Ionescu, “Multiview plus depth video coding with temporal prediction view synthesis”, *IEEE Transactions on circuits and systems for video technology* , Vol. 26(2), pp. 360-374, 2016.
2. A. Purica, B. Boyadjis, B. Pesquet-Popescu, F. Dufaux and C. Bergeron “A convex optimization framework for video quality and resolution enhancement from multiple descriptions”, *IEEE Transactions on Image Processing* , *(under review)*

## International conference papers

1. **A. Purica**, B. Pesquet-Popescu, Devices and methods for video reconstruction from multiple source, Apl. Nr. EP17306724

## International conference papers

1. **A. Purica**, E. G. M., B. Pesquet-Popescu, M. Cagnazzo, and B. Ionescu, “Improved view synthesis by motion warping and temporal hole filling”, Proceeding of *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1191 - 1195, South Brisbane, Australia, April 2015.
2. **A. Purica**, M. Cagnazzo, B. Pesquet-Popescu, F. Dufaux, and B. Ionescu, “A distortion evaluation framework in 3D video view synthesis”, Proceeding of *IEEE International Conference on 3D Imaging*, pp. 1-8, Liége, Belgium, December 2015. **(Best Paper Award in the form of a Lumiére Award from the Advanced Imaging Society & VR Society)**
3. **A. Purica**, M. Cagnazzo, B. Pesquet-Popescu, F. Dufaux, and B. Ionescu, “View synthesis based on temporal prediction via warped motion vector fields”, Proceeding of *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1150-1154, Shanghai, China, March 2016.

4. **A. Purica**, G. Valenzise, B. Pesquet-Popescu, and F. Dufaux, "Using region-of-interest for quality evaluation of dibr-based view synthesis methods", Proceeding of *International Conference on Quality of Multimedia Experience (QoMEX)*, pp. 1-6, Lisbon, Portugal, June 2016.
5. **A. Purica**, B. Pesquet-Popescu and F. Dufaux, "A Railroad Detection Algorithm for Infrastructure Surveillance using Enduring Airborne Systems", Proceeding of *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, United States, March 2017.
6. **A. Purica**, B. Boyadjis, B. Pesquet-Popescu and F. Dufaux, "A study of norms in convex optimization super-resolution from compressed source", Proceeding of *IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, London, U.K., October 2017
7. B. Boyadjis, **A. Purica**, B. Pesquet-Popescu and F. Dufaux, "Video enhancement with convex optimization methods", Proceeding of *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), (under review)*

# Bibliography

[ACR97] "Itu-t study group 12", ITU-T p.910 Subjective Video Quality Assessment Methods for Multimedia Applications, 1997. *Cited in Sec.* 3.3.2

[ANR74] N. Ahmed, T. Nararajan, and K. Rao, "Discrete cosine transform". *IEEE Transactions on Computers*, pp. 90–93, 1974. *Cited in Sec.* 1.2.2

[BAS08] F. B, V. A, and S. S, "Localizing objects with smart dictionaries", in *Proceedings of the European Conference on Computer Vision*, pp. 179–192, 2008. *Cited in Sec.* 5.1

[BBMG01] C. Buehler, M. Bosse, L. McMillan, and S. Gortler, "Unstructured Lumigraph Rendering", in *Proc SIGGRAPH*, pp. 425–432, Los Angeles, California USA, August 2001. *Cited in Sec.* 2.1.1, 3.1.1

[BFB14] M. Budagavi, A. Fuldseth, and G. Bjontegaard, "HEVC Transform and Quantization", in V. Sze, M. Budagavi, and G. Sullivan, eds., *High Efficiency Video Coding (HEVC)*, Integrated Circuits and Systems, pp. 141–169, Springer International Publishing, 2014. *Cited in Sec.* 1.3.2.3, 4.4

[BHO$^+$12] B. Bross, W.-J. Han, J.-R. Ohm, G. Sullivan, and T. Wiegand, "High Efficiency Video Coding (HEVC) text specification draft 8", in *JCTVC-J1003*, Stockholm, Sweden, July 2012. *Cited in Sec.* 1.3.1, 4.4

[Bjo01] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves", in *VCEG Meeting*, Austin, USA, April 2001. *Cited in Sec.* 2.6.4.1

[BLCMP12] E. Bosc, P. Le Callet, L. Morin, and M. Pressigout, *3D-TV System with Depth-Image-Based Rendering Architectures, Techniques and Challenges*, Springer, 2012. *Cited in Sec.* 3.3.2

[BLTW16] G. Bang, G. Lafruit, M. Tanimoto, and K. Wegner, "Description of 360 3d video application Exploration Experiments on Divergent multi-view video v0.1", ISO/IEC JTC1/SC29/WG11, May 2016. *Cited in Sec.* 7, 1.5, 1.13, 1.14

[BMK11] S. Babacan, R. Molina, and A. Katsaggelos, "Variational bayesian super resolution". *IEEE Transactions in Image Process*, pp. 984–999, 2011. *Cited in Sec.* 4.1

[Bov00] A. Bovik, *Handbook of Image & Video Processing*, Elsevier Academic Press, 2000. *Cited in Sec.* 1.1

[BPc+11] E. Bosc, R. Pepion, P. L. callet, M. Koppel, P. Ndjiki-Nya, M. Pressigout, and L. Morin, "Towards a new quality metric for 3-d synthesized view assessment". *IEEE Journal of Selected Topics in Signal Processing*, vol. 5 (7), pp. 1332–1343, September 2011. *Cited in Sec.* 4, 17(b), 7, 3.1.3, 3.3.1, 3.3.3.1, 3.3.3.2, 3.6(c), 3.6, 3.7(b)

[BPPDB16] B. Boyadjis, B. Pesquet-Popescu, F. Dufaux, and C. Bergeron, "Super-resolution of hevc videos via convex optimization", in *IEEE International Conference on Image Processing (ICIP)*, September 2016. *Cited in Sec.* 4.5, 4.5.1.4

[BS00] M. Bertalmio and G. Sapiro, "Image inpainting", in *SIGGRAPH*, pp. 417–424, New Orleans, USA, Jully 2000. *Cited in Sec.* 2.1.1

[BSM+09] H. Brust, A. Smolic, K. Mueller, G. Tech, and T. Wiegand, "Mixed resolution coding of stereoscopic video for mobile devices", in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, pp. 1 –4, May 2009. *Cited in Sec.* 1.4.2

[bub] https://www.bublcam.com, accessed: 23-08-2016. *Cited in Sec.* 1.15(b)

[BYCR15] J. M. Boyce, Y. Ye, J. Chen, and A. K. Ramasubramonian, "Overview of shvc: Scalable extensions of the high efficiency video coding standard". *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, pp. 20–34, 2015. *Cited in Sec.* 4.2.1.3

[CBL72] D. Connor, R. Brainard, and J. Limb, "Interframe coding for picture transmission". *Proceedings of The IEEE*, vol. 60, pp. 779–790, 1972. *Cited in Sec.* 1.1.2

[CBYH15] J. Chen, J. Boyce, Y. Ye, and M. M. Hannuksela, "Scalable HEVC (SHVC) Test Model 10 (SHM 10)", JCT-VC of ITU-T SG16 WP 3and ISO/IEX JTC 1/SC 29/WG 11, June 2015. *Cited in Sec.* 4.2.1.3

[CCPW07] C. Chaux, P. L. Combettes, J.-C. Pesquet, and V. R. Wajs, "A variational formulation for frame based inverse problems". *Inverse Probl.*, vol. 23 (4), pp. 1495–1518, 2007. *Cited in Sec.* 4.3.3

[CFP11] "Call for Proposals on 3D video coding technology", ISO/IEC JTC1/SC29/WG11 N12036, March 2011. *Cited in Sec.* 2.4.3, 2.5.3

[CHKK07] B.-D. Choi, J.-W. Han, C.-S. Kim, and S.-J. Ko, "Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation". *CSVT*, April 2007. *Cited in Sec.* 2.1.3

[cis] http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html, accessed: 20-09-2016. *Cited in Sec.* 7

[CLLY08] C.-M. Cheng, S.-J. Lin, S.-H. Lai, and J.-C. Yang, “Improved novel view synthesis from depth image with large baseline”, in *19th International Conference on Pattern Recognition (ICPR)*, pp. 1–4, 2008. *Cited in Sec.* 2.1.1, 3.1.1

[CP04] P. L. Combettes and J.-C. Pesquet, “Image restoration subject to a total variation constraint”. *IEEE Transactions on Image Processing*, vol. 13 (9), pp. 1213–1222, 2004. *Cited in Sec.* 4.2.2.3

[CP10] ———, “Proximal splitting methods in signal processing”. *Fixed-point algorithms for inverse problems in science and engineering*, pp. 185–212, 2010. *Cited in Sec.* 4.3.3

[CP12] ———, “Primal-dual splitting algorithm for solving inclusions with mixtures of composite, lipschitzian, and parallel-sum type monotone operators”. *Set-Valued Anal*, vol. 20 (2), pp. 307–330, 2012. *Cited in Sec.* 4, 4.1, 4.3, 4.3.2, 4.3.3

[CPCP09] L. Chaari, N. Pustelnik, C. Chaux, and J.-C. Pesquet, “Solving inverse problems with overcomplete transforms and convex optimization techniques”, in *Proc. SPIE Wavelets, San Diego, CA, USA*, 2009. *Cited in Sec.* 4.3.3

[CPPPP12] G. Chierchia, N. Pustelnik, J.-C. Pesquet, and B. Pesquet-Popescu, “Epigraphical projection and proximal tools for solving constrained convex optimization problems: Part i”. *Tech. Rep. Telecom ParisTech*, 2012. *Cited in Sec.* 4.3.3

[CPT04] A. Criminisi, P. Perez, and K. Toyama, “Region filling and object removal by exemplar-based image inpainting”. *IEEE Transactions on Image Processing*, vol. 13 (9), pp. 1200–1212, 2004. *Cited in Sec.* 2.1.1, 3.1.2

[CT65] J. Cooley and J. Tukey, “An algorithm for the machine calculation of complex fourier series”. *Mathematics of Computation*, vol. 19, pp. 297–301, 1965. *Cited in Sec.* 1.2.2

[CTL$^+$10] K.-Y. Chen, P.-K. Tsung, P.-C. Lin, H.-J. Yang, and L.-G. Chen, “Hybrid motion/depth-oriented inpainting for virtual view synthesis in multiview applications”. *3DTV-CON*, pp. 1–4, 7-9 June 2010. *Cited in Sec.* 2.1.1

[CTMS03] J. Carranza, C. Theobalt, M. A. Magnor, and H.-P. Seidel, “Free-viewpoint video of human actors”. *ACM Transactions on Graphics*, vol. 22 (3), pp. 569–577, July 2003. *Cited in Sec.* 1.4.2

[CTWY14] Y. Chen, G. Tech, K. Wegner, and S. Yea, “3d-hevc test model 9 .”, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 JCT3V-N14704, 2014. *Cited in Sec.* 7, 1.11

[CVO11] G. Cheung, V. Velisavljevic, and A. Ortega, “On dependent bit allocation for multiview image coding with depth-image-based rendering”. *IEEE Transactions on Image Processing*, vol. 20, November 2011. *Cited in Sec.* 2.2.2, 2.6.3

[CWU$^+$09] Y. Chen, Y.-K. Wang, K. Ugur, M. Hannuksela, J. Lainema, and M. Gabbouj, “The emerging MVC standard for 3D video services”. *EURASIP Journal on Advances in Signal Processing*, 2009. *Cited in Sec.* 2.1.1

[Dar09] I. Daribo, *Codage et rendu de séquence vidéo 3D; et applications à la télévision tridimensionnelle (TV 3D) et à la télévision à base de rendu de vidéos (FTV)*, Ph.D. thesis, Télécom ParisTech, 2009. *Cited in Sec.* 1.4.4

[dat] “DIBR videos quality assessment (using acr-hr)”, http://ivc.univ-nantes.fr/en/databases/DIBR_Videos/. *Cited in Sec.* 3.3.2, 3.3.3.1

[DCPP14] F. Dufaux, M. Cagnazzo, and B. Pesquet-Popescu, *Motion Estimation - a Video Coding Viewpoint*, vol. 5: Image and Video Compression and Multimedia of *Academic Press Library in Signal Processing*, Academic Press, 2014. *Cited in Sec.* 2.1.1, 3.1.1

[DH72] R. Duda and P. Hart, “Use of the hough transformation to detect lines and curves in pictures”. *Commun. ACM*, vol. 15, pp. 11–15, 1972. *Cited in Sec.* 5.2

[DMPP10] I. Daribo, W. Milded, and B. Pesquet-Popescu, “Joint Depth-Motion Dense Estimation for Multiview Video Coding”. *Journal of Visual Communication and Image Representation*, vol. 21, pp. 487–497, 2010. *Cited in Sec.* 2.3

[DPP10] I. Daribo and B. Pesquet-Popescu, “Depth-aided image inpainting for novel view synthesis”, in *IEEE MMSP*, Saint Malo, France, 4-6, October 2010. *Cited in Sec.* 2.1.1, 3.1.2

[DPPC13] F. Dufaux, B. Pesquet-Popescu, and M. Cagnazzo, eds., *Emerging technologies for 3D video: content creation, coding, transmission and rendering*, Wiley, May 2013. *Cited in Sec.* 7, 2.1.1, 2.2.1.1, 2.4.1, 3.1.1

[EXP10] “Report on experimental framework for 3D video coding”, ISO/IEC JTC1/SC29/WG11 MPEG2010/N11631, October 2010. *Cited in Sec.* 7, 2.1.1, 2.2.1.1, 2.1

[fac] https://www.facebook.com/help/851697264925946/, accessed: 23-11-2016. *Cited in Sec.* 7, 1.18

[FCSK02] C. Fehn, E. Cooke, O. Schreer, and P. Kauff, “3D analysis and image-based rendering for immersive TV applications”. *Signal Processing: Image Communication*, vol. 17 (9), pp. 705 – 715, 2002. *Cited in Sec.* 1.4.2

[Fec60] G. T. Fechner, *Elements of psychophysics [Elemente der Psychophysik]*, vol. 1, D. H. Howes and E. G. Boring, 1966 [First published 1860]. *Cited in Sec.* 1.1.2

[Feh03] C. Fehn, "A 3D-TV approach using depth-image-based rendering", in *3rd IASTED Conference on Visualization, Imaging, and Image Processing*, pp. 482–487, Benalmadena, Spain, 8-10 September 2003. *Cited in Sec.* 7, 2.1.1, 3.5, 3.3.1

[Feh04] ———, "Depth-image-based-rendering (dibr) , compression and transmission for a new approach on 3D-TV", in *Proc. of SPIE Stereoscopic Displays and Virtual Reality Sistems*, vol. 5291, pp. 93–104, 2004. *Cited in Sec.* 3.3.2

[FLG13] M. S. Farid, M. Lucenteforte, and M. Grangetto, "Depth image based rendering with inverse mapping", in *IEEE MMSP*, pp. 135–140, Pula (Sardinia), Italy, September 30-October 2, 2013. *Cited in Sec.* 2.2.2

[FO08] L. A. Fernandes and M. M. Oliveira, "Real-time line detection through an improved hough transform voting scheme". *Pattern Recognition*, vol. 41, pp. 299–314, September 2008. *Cited in Sec.* 5.2

[GAM04] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Super-resolution reconstruction of compressed video using transform-domain statistics". *IEEE TRANSACTIONS ON IMAGE PROCESSING*, vol. 13 (1), pp. 33–43, January 2004. *Cited in Sec.* 4.1

[GCPP12] R. Gaetano, G. Chierchia, and B. Pesquet-Popescu, "Parallel implementations of a disparity estimation algorithm based on a proximal splitting method", in *IEEE Visual Communications and Image Processing (VCIP)*, pp. 1–6, 2012. *Cited in Sec.* 4.5.3

[gea] http://www.samsung.com/global/galaxy/gear-360/, accessed: 23-08-2016. *Cited in Sec.* 1.15(d)

[gir] https://www.giroptic.com/, accessed: 23-08-2016. *Cited in Sec.* 1.15(c)

[GM14] C. Guillemot and O. L. Meur, "Image inpainting: Overview and recent advances". *IEEE Signal Processing Magazine*, vol. 31, pp. 127–144, 2014. *Cited in Sec.* 2.1.1, 3.1.2

[GMM$^+$13] D. Grois, D. Marpe, A. Mulayoff, B. Itzhaky, and O. Hadar, "Performance comparison of h.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders", in *2013 Picture Coding Symposium (PCS)*, Institute of Electrical and Electronics Engineers (IEEE), dec 2013. *Cited in Sec.* 1.3.1

[gop] https://shop.gopro.com/EMEA/virtualreality/, accessed: 23-08-2016. *Cited in Sec.* 7, 1.17

[GPPC13] R. Gaetano, B. Pesquet-Popescu, and C. Chaux, “A convex optimization approach for image resolution enhancement from compressed representations”, in *18th International Conference on Digital Signal Processing (DSP)*, pp. 1–8, July 2013. *Cited in Sec.* 4.1

[H2605] “Advanced Video Coding for generic audiovisual services”, ITU-T Recommendation H.264 and ISO/IEC 14496-10 AVC, 2005. *Cited in Sec.* 1.3.1

[Had93] J. Hadamard, “Resolution d'une question relative aux determinants”. *Bulletin des Sciences Mathematiques*, vol. 17, pp. 240–246, 1893. *Cited in Sec.* 1.2.2

[HEV13] “High Efficiency Video Coding”, ITU-T Recommendation H.265 and ISO/IEC 23008-2 HEVC, April 2013. *Cited in Sec.* 7, 1.3.1, 1.3.2.1, 2.1.1

[HKA13] S. Huq, A. Koschan, and M. Abidi, “Occlusion filling in stereo: theory and experiments”. *Computer Vision and Image Understanding*, vol. 117, pp. 688–704, June 2013. *Cited in Sec.* 2.4.1, 3.1.2

[Hou62] P. Hough, “Methods and means for recognizing complex patterns”, 1962. *Cited in Sec.* 5.2

[HS81] B. Horn and B. Schunck, “Determining optical flow”. *Artificial Intelligence*, vol. 16, pp. 185–203, August 1981. *Cited in Sec.* 4.1

[Huf52] D. Huffman, “A method for the construction of minimum-redundancy codes”. *PROCEEDINGS OF THE I.R.E.*, vol. 40, pp. 1098–1101, 1952. *Cited in Sec.* 1.2.3

[ISO00] ISO/IEC 15444-1, “Jpeg 2000 image coding system”, Tech. Rep., JPEG, 2000. *Cited in Sec.* 4.2.1.3

[Kar47] H. Karhunen, “Uber lineare methoden in der wahrscheinlichkeitsrechnung”. *Ann. Acad. Sci. Fennicae*, vol. 37, pp. 1–79, 1947. *Cited in Sec.* 1.2.2

[KEB91] N. Kiryati, Y. Eldar, and A. M. Bruckstein, “A probabilistic hough transform”. *Pattern recognition*, vol. 24 (4), pp. 303–316, 1991. *Cited in Sec.* 5.1, 5.2

[Key81] R. Keys, “Cubic convolution interpolation for digital image processing”. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-29 (6), December 1981. *Cited in Sec.* 4.2.1.3

[KGV13] K. P. Kumar, S. Gupta, and K. S. Venkatesh, “Spatio-temporal multi-view synthesis for free viewpoint television”, in *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, pp. 1 – 4, Aberdeen, 7-8 October 2013. *Cited in Sec.* 2.1.1

[KH96] D. Kundur and D. Hatzinakos, “Blind image deconvolution”. *IEEE Signal Processing Magazine*, vol. 13 (3), pp. 43–64, May 1996. *Cited in Sec.* 4.1

[KKS$^+$00] J. Kim, Y.-G. Kim, H. Song, T.-Y. Kuo, and Y. J. Chung, “TCP-friendly internet video streaming employing variable frame-rate encoding and interpolation”. *CSVT*, October 2000. *Cited in Sec.* 2.1.3

[KMN$^+$02] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, “An efficient k-means clustering algorithm: Analysis and implementation”. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, pp. 881–892, 2002. *Cited in Sec.* 5.2

[KMW95] R. Krishnamurthy, P. Moulin, and J. Woods, “Optical flow techniques applied to video coding”, in *IEEE International Conference on Image Processing (ICIP)*, vol. 1, pp. 570–573 vol.1, 1995. *Cited in Sec.* 2.1.1, 3.1.1

[KNND$^+$10] M. Koppel, P. Ndjiki-Nya, D. Doshkov, H. Lakshman, P. Merkle, K. Muller, and T. Wiegand, “Temporally consistent handling of disocclusions with texture synthesis for depth-image-based rendering”, in *17th IEEE International Conference on Image Processing (ICIP)*, pp. 1809–1812, 2010. *Cited in Sec.* 3.3.2

[KPS$^+$11] Y. J. Kim, J. H. Park, G. S. Shin, H.-S. Lee, D.-H. Kim, S. H. Park, and J. Kim, “Evaluating super resolution algorithms”, in *SPIE-IS&T Electronic Imaging, Image Quality and System Performance VIII*, vol. 7867, 2011. *Cited in Sec.* 4.1

[KYDK16] A. Kappeler, S. Yoo, Q. Dai, and A. K. Katsaggelos, “Super-resolution of compressed videos using convolutional neural networks”, in *IEEE International Conference on Image Processing (ICIP)*, September 2016. *Cited in Sec.* 4.1

[LCCJ12] Z. Liu, G. Cheung, J. Chakareski, and Y. Ji, “Multiple description coding of free viewpoint video for multi-path network streaming”, in *IEEE Globecom*, December 2012. *Cited in Sec.* 2.1.1

[LE10] P.-J. Lee and Effendi, “Adaptive edge-oriented depth image smoothing approach for depth image based rendering”, in *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pp. 1–5, Shanghai, 24-26 March, 2010. *Cited in Sec.* 2.2.2, 3.1.2

[Lea93] V. Leavers, “Survey: which hough transform?” *Graphical Model Image Process*, vol. 58, pp. 250–264, 1993. *Cited in Sec.* 5.2

[LF80] G. Legge and J. Foley, “Contrast masking in human vision”. *Journal of Optical Society of America*, vol. 70, pp. 1458–1471, 1980. *Cited in Sec.* 1.1.2

[LH96] M. Levoy and P. Hanrahan, “Light field rendering”, in *Proceedings of SIGGRAPH*, SIGGRAPH ’96, pp. 31–42, ACM, New York, NY, USA, 1996. *Cited in Sec.* 2.1.1, 3.1.1

[Liu] C. Liu, "Optical flow Matlab/C++ code", http://people.csail.mit.edu/celiu/OpticalFlow/. *Cited in Sec.* 2.4.1, 2.4.3, 2.5.3, 2.6.4.1, 3.2.3

[Liu09] ———, *Beyond pixels: exploring new representations and applications for motion analysis*, Ph.D. thesis, Massachusetts Institute of Technology, May 2009. *Cited in Sec.* 2.4.3, 2.5.3, 2.6.4.1

[Llo82] S. Lloyd, "Least squares quantization in pcm". *IEEE Transactions on Information Theory*, vol. 28, pp. 129–137, 1982. *Cited in Sec.* 1.2.1

[LLZ$^+$14] S. Li, J. Lei, C. Zhu, L. Yu, and C. Hou, "Pixel-based inter prediction in coded texture assisted depth coding." *IEEE Signal Processing Letters*, vol. 21, pp. 74–78, 2014. *Cited in Sec.* 2.2.2, 2.6.3

[Loe48] M. Loeve, "Fonctions aleatoires de seconde ordre". *Processus Stochastiques et Mouvement Brownien*, 1948. *Cited in Sec.* 1.2.2

[Low99] D. G. Lowe, "Object recognition from local scale-invariant features", in *Proceedings of the International Conference on Computer Vision*, pp. 1150–1157, 1999. *Cited in Sec.* 5.1

[LS14] C. Liu and D. Sun, "On bayesian adaptive video super resolution". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36 (2), pp. 346–360, 2014. *Cited in Sec.* 4.1

[LSK$^+$08] C. Liu, R. Szeliski, S. Kang, C. Zitnick, and W. Freeman, "Automatic estimation and removal of noise from a single image". *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30 (2), pp. 299–314, February 2008. *Cited in Sec.* 4.1

[MBXV06] E. Martinian, A. Behrens, J. Xin, and A. Vetro, "View synthesis for multiview video compression". *Picture Coding Symposium*, 2006. *Cited in Sec.* 2.1.1

[MFdW07] Y. Morvan, D. Farin, and P. H. N. de With, "Multiview depth-image compression using an extended h.264 encoder". *LNCS*, 2007. *Cited in Sec.* 2.1.1

[Mil10] P. Milanfar, *Super-Resolution Imaging, Digital Imaging and Computer Vision*, Taylor&Francis/CRC Press, 2010. *Cited in Sec.* 4.1

[Mit96] J. Mitchell, *MPEG Video: Compression Standard*, Chapman and Hall, New York, 1996. *Cited in Sec.* 1.1.2

[MJW15] Z. Ma, J. Jia, and E. Wu, "Handling motion blur in multi-frame super-resolution", in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. *Cited in Sec.* 4.1

[MNCM01] R. Molina, J. Nunez, F. Cortijo, and J. Mateos, "Image restoration in astronomy bayesian perspective". *IEEE Signal Process. Mag.*, vol. 18, pp. 11–29, 2001. *Cited in Sec.* 4.1

[Mor65] J.-J. Moreau, “ProximitÂ´e et dualitÂ´e dans un espace hilbertien”. *Bull. Soc. Math. France*, vol. 93, pp. 273–299, 1965. *Cited in Sec.* 4.3.1

[Mor14a] E. G. Mora, *Codage multi-vues multi-profondeur pour de nouveaux services multimedia*, Ph.D. thesis, chapter 6, pg. 120, EDITE, Telecom Paristech, 2014. *Cited in Sec.* 2.1.3

[Mor14b] ———, *Codage multi-vues multi-profondeur pour de nouveaux services multimedia*, Ph.D. thesis, chapter 1, pg. 4, EDITE, Telecom Paristech, 2014. *Cited in Sec.* 2.6.3

[MSD+08] K. Muller, A. Smolic, K. Dix, P. Merkle, and P. Kauff, “View synthesis for advanced 3d video systems”. *EURASIP Journal on Image and Video Processing*, 2008. *Cited in Sec.* 3.3.2

[MSMW07] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, “Multi-view video plus depth representation and coding”. *IEEE International Conference on Image Processing*, vol. 1, pp. 201–204, 2007. *Cited in Sec.* 2.1.1

[NB05] D. N and T. B, “Histograms of oriented gradients for human detection”. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 886–893, 2005. *Cited in Sec.* 5.1

[NB12] S. K. Nelson and A. Bhatti, “Performance evaluation of multi-frame super-resolution algorithms”, in *International Conference on Digital Image Comput-ing: Techniques and Applications (DICTA)*, Centre for Intelligent Systems Research, Deakin University, Geelong,Victoria,Australia, 2012. *Cited in Sec.* 4.1

[NNKD+11] P. Ndjiki-Nya, M. Koppel, D. Doshkov, H. Lakshman, P. Merkle, K. Muller, and T. Wiegand, “Depth image based rendering with advanced texture synthesis for 3-d video”. *IEEE Transactions on Multimedia*, vol. 13 (3), pp. 453–465, June 2011. *Cited in Sec.* 3.3.2

[nok] https://ozo.nokia.com, accessed: 23-08-2016. *Cited in Sec.* 7, 1.16

[ope] “OpenHEVC Open source HEVC decoder”, https:/github.com/openhevc/openhevc/. *Cited in Sec.* 4.4.1.1

[PCPP+15] A. Purica, M. Cagnazzo, B. Pesquet-Popescu, F. Dufaux, and B. Ionescu, “A distortion evaluation framework in 3D video view synthesis”, in *Proceedings of International Conference on 3D Imaging*, Liege, Belgium, December 2015. *Cited in Sec.* 3.4

[PCPP+16] ———, “View synthesis based on temporal prediction via warped motion vector fields”, in *Proc. of IEEE ICASSP*, IEEE, March 2016. *Cited in Sec.* 2.7, 3.2.1, 3.2.3

[PMC+16] A. Purica, E. G. Mora, M. Cagnazzo, B. Pesquet-Popescu, and B. Ionescu, “Multiview plus depth video coding with temporal prediction view synthesis”. *IEEE Transactions on circuits and systems for video technology*, vol. 26 (2), pp. 360–374, 2016. *Cited in Sec.* 2.7

[PMPP$^+$15] A. Purica, E. G. M., B. Pesquet-Popescu, M. Cagnazzo, and B. Ionescu, “Improved view synthesis by motion warping and temporal hole filling”, in *Proc. of ICASSP*, pp. 1191–1195, IEEE, South Brisbane, 19-24 April 2015. *Cited in Sec.* 2.7, 3.2.1, 3.2.3

[PMTB14] E. Pali, K. Mathe, L. Tamas, and L. Busoniu, “Railway track following with the ar.drone using vanishing point detection”, in *IEEE International Conference on Automation, Quality and Testing, Robotics*, pp. 1–6, May 2014. *Cited in Sec.* 5.1

[PRDD10] F. PF, G. RB, M. D, and R. D, “Object detection with discriminatively trained part-based models”. *IEEE Trans Pattern Anal Machine Intell*, vol. 32, pp. 1627–1645, 2010. *Cited in Sec.* 5.1

[PVPPD16] A. Purica, G. Valenzise, B. Pesquet-Popescu, and F. Dufaux, “Using region-of-interest for quality evaluation of dibr-based view synthesis methods”, in *2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX)*, pp. 1–6, Lisbon, Portugal, 2016. *Cited in Sec.* 3.4

[RHFL10] S. Reichelt, R. Haussler, G. Fatterer, and N. Leister, “Depth cues in human visual perception and their realization in 3D displays”. *Proceedings of the SPIE Three-Dimensional Imaging, Visualization, and Display*, pp. 76900B–76900B–12, 2010. *Cited in Sec.* 1.4.1

[RMV13] D. Rusanovsky, K. Muller, and A. Vetro, “Common Test Conditions of 3DV Core Experiments”, ITU-T SG16 WP3 & ISO/IEC JTC1/SC29/WG11 JCT3V-D1100, April 2013. *Cited in Sec.* 2.4.3, 2.5.3, 2.6.4.1, 3.2.3

[RS01] S. Rane and G. Sapiro, “Evaluation of jpeg-ls, the new lossless and controlledlossy still image compression standard, for compression of high-resolution elevation data”. *IEEE Transactions on Geoscience and Remote sensing*, pp. 2298–2306, 2001. *Cited in Sec.* 1.1

[Sal07] D. Salomon, *Data Compression: The Complete Reference*, Springer, fourth edition edn., 2007. *Cited in Sec.* 1.1

[SAX$^+$12] W. Sun, O. C. Au, L. Xu, Y. Li, and W. Hu, “Novel temporal domain hole filling based on background modeling for view synthesis”, in *IEEE International on Image Processing (ICIP)*, pp. 2721 – 2724, Orlando, FL, 30 Sept. - 3 Oct. 2012. *Cited in Sec.* 2.1.1

[SDZ04] R. Stoica, X. Descombes, and J. Zerubia, “A gibbs point process for road extraction from remotely sensed images”. *International Journal of Computer Vision*, vol. 57, pp. 121–136, 2004. *Cited in Sec.* 5.1

[SH99] H.-Y. Shum and L.-W. He, “Rendering with concentric mosaics”, in *Proceedings SIGGRAPH*, pp. 299–306, Los Angeles, California USA, 1999. *Cited in Sec.* 2.1.1, 3.1.1

[Sha48] C. E. Shannon, "A mathematical theory of communication". *The Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948. *Cited in Sec.* 1.2.3

[Siu12] K.-W. H. W.-C. Siu, "Depth-assisted nonlocal means hole filling for novel view synthesis", in *ICIP*, September 2012. *Cited in Sec.* 2.1.1

[SK00] H. Shum and S. B. Kang, "Review of image-based rendering techniques". *SPIE Visual Communications and Image Processing*, vol. 4067, pp. 2–13, 2000. *Cited in Sec.* 2.1.1, 3.1.1

[SK10] S. Shimizu and H. Kimata, "Improved view synthesis prediction using decoder-side motion derivation for multiview video coding", in *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, pp. 1–4, 2010. *Cited in Sec.* 2.1.1

[SKMM02] C. A. Segall, A. K. Katsaggelos, R. Molina, and J. Mateos, *Super-Resolution from Compressed Video*, Chap. 11, pp. 211–242, Springer US, Boston, MA, 2002. *Cited in Sec.* 4.1

[SMRA14] S.Villenaa, M.Vegaa, R.Molinab, and A.K.Katsaggelosc, "A non-stationary image prior combination in super-resolution". *Digital Signal Processing*, vol. 32, 2014. *Cited in Sec.* 4.1

[SS08] Y. Q. Shi and H. Sun, *Image and Video Compression for Multimedia Engineering*, CRC Press, 2008. *Cited in Sec.* 1.1.1, 1.1.2

[SSA13] S. Saluja, A. K. Singh, and S. Agrawal, "A study of edge-detection methods". *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, pp. 994–999, 2013. *Cited in Sec.* 5.2

[TCM+15] G. Tech, Y. Chen, K. Muller, J.-R. Ohm, A. Vetro, and Y.-K. Wang, "Overview of the multiview and 3d extensions of high efficiency video coding". *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, pp. 35–49, 2015. *Cited in Sec.* 7, 1.4.3

[Tec] G. Tech, "HTM-7.0 software", Available: https://hevc.hhi.fraunhofer.de/. *Cited in Sec.* 2.2

[Tel04] A. Telea, "An image inpainting technique based on the fast marching method". *Journal of Graphics, GPU and Game Tools*, vol. 9, pp. 25–36, 2004. *Cited in Sec.* 3.3.2

[TFK+] M. Tanimoto, T. Fujii, K.Suzuki, N. Fukushima, and Y. Mori, "Reference software for depth estimation and view synthesis", ISO/IEC JTC1/SC29/WG11 MPEG 2008/M15377. *Cited in Sec.* 3.3.2

[TG90] K. Turkowski and S. Gabriel, "Filters for common resampling tasks". *Andrew S. Glassner Graphics Gems I, Academic Press*, pp. 147–165, 1990. *Cited in Sec.* 4.2.1.3

[TGM08] A. Tikanamaki, A. Gotchev, and A. S. S. Miller, “Quality assessment of 3-d video in rate allocation experiments”, in *IEEE International Symposium on Consumer Electronics, 2008. ISCE 2008*, pp. 1–4, IEEE, Vilamoura, 14-16 April 2008. *Cited in Sec.* 3.1.3

[the] https://theta360.com/en/about/theta/, accessed: 23-08-2016. *Cited in Sec.* 1.15(a)

[TLZ16] Z. Teng, F. Liu, and B. Zhang, “Visual railway detection by superpixel based intracellular decisions”. *Multimedia Tools and Applications*, vol. 75 (5), pp. 2473–2486, March 2016. *Cited in Sec.* 5.1

[TSG15] R. Timofte, V. D. Smet, and L. V. Gool, “A+: Adjusted anchored neighborhood regression for fast superresolution”, in D. Cremers, I. Reid, H. Saito, and M. Yang, eds., *Computer Vision ACCV 2014*, vol. 9006 of *Lecture Notes in Computer Science*, pp. 111–126, Springer International Publishing, 2015. *Cited in Sec.* 3, 7, 4.1, 4.5, 4.5.1.1, 4.5.1.3, 4.5.2.1, 4.4, 4.6

[TSK10] L. Tian, A. Suzuki, and H. Koike, “Task-oriented evaluation of superresolution techniques”, in *nt. Conference on Pattern Recognition (ICPR)*, pp. 493–498, 2010. *Cited in Sec.* 4.1

[TTFY11] M. Tanimoto, M. P. Tehrani, T. Fujii, and T. Yendo, “Free-Viewpoint TV”. *IEEE Signal Processing Magazine*, vol. 28e, pp. 67–76, 2011. *Cited in Sec.* 7, 2.1.1, 3.1.1

[vdBF08] E. van den Berg and M. P. Friedlander, “Probing the pareto frontier for basis pursuit solutions”. *SIAM J. Sci. Comput.*, vol. 31 (2), pp. 890–912, 2008. *Cited in Sec.* 4.3.3

[Vid00] Video Quality Experts Group, “Final report from the video quality experts group on the validation of objective models of video quality assessment”, VQEG, March 2000. *Cited in Sec.* 3.3.3.1

[VQE] “VQEG 3DTV Group”, http://www.its.bldrdoc.gov/vqeg/projects/3dtv/. *Cited in Sec.* 3.1.3

[vtJoIMITV13] J. video team (JVT) of ISO/IEC MPEG & ITU-T VCEG, “H.265/HEVC HM reference software”, http://hevc.fraunhofer.de//, May 2013. *Cited in Sec.* 4.4.1.2, 4.5.1.1

[VVB$^+$13] S. Villena, M. Vega, D. Babacan, R. Molina, and A. Katsaggelos, “Bayesian combination of sparse and non sparse priors in image super resolution”. *Digital Signal Processing*, vol. 23 (2), pp. 530–541, 2013. *Cited in Sec.* 4.1

[Wal23] J. Walsh, “A closed set of normal orthogonal functions”. *American Journal of Mathematics*, vol. 45, pp. 5–24, 1923. *Cited in Sec.* 1.2.2

[Wat87] A. Watson, “Effciency of a model human image code”. *Journal of Optical Society of America*, vol. 4, pp. 2401–2417, 1987. *Cited in Sec.* 1.1.2

[WBSS04] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity". *IEEE Transactions on Image Processing*, vol. 13, pp. 600–612, 2004. *Cited in Sec.* 3.1.3, 3.2.2.1, 3.3.3.1

[WLS+11] L. Wang, J. Liu, J. Sun, Y. Ren, W. Liu, and Y. Gao, "Virtual view synthesis without preprocessing depth image for depth image based rendering", in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, pp. 1–4, Antalya, Turkey, 16-18 May 2011. *Cited in Sec.* 2.2.2

[WSB04] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment", in *Proceedings of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*, vol. 2, pp. 1398–1402, November 2004. *Cited in Sec.* 3.3.3.1

[WSBL03] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard". *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13 (7), pp. 560–576, July 2003. *Cited in Sec.* 1.3.1

[WSR+07] T. Wiegand, G. Sullivan, J. Reichel, H. Schwarz, and M. Wien, "Joint draft itu-t rec. h. 264 iso/iec 14496-10/amd. 3 scalable video coding". *Joint Video Team (JVT) JVT-X201*, vol. 108, p. 1990, 2007. *Cited in Sec.* 4.2.1.3

[WYT09] C. Wang, G. Yang, and Y.-P. Tan, "Reconstructing videos from multiple compressed copies". *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, vol. 19 (9), pp. 1342–1352, September 2009. *Cited in Sec.* 4.1

[WZ11] Z. Wang and J. Zhou, "A novel approach for depth image based rendering, based on non-linear transformation of depth values", in *International Conference on Image Analysis and Signal Processing (IASP)*, pp. 138–142, Hubei, People's Republic of China, 21-23 October 2011. *Cited in Sec.* 2.2.2, 3.1.2

[WZHT10] C. Wang, L. Zhang, Y. He, and Y.-P. Tan, "Frame rate up-conversion using trilateral filtering". *CSVT*, June 2010. *Cited in Sec.* 2.1.3

[XHT+14] J. Xiao, M. Hannuksela, T. Tillo, M. Gabbouj, C. Zhu, and Y. Zhao, "Scalable bit allocation between texture and depth views for 3d video streaming over heterogeneous networks". *IEEE Trans. Circuits and Systems for Video Technology*, p. 1, Juin 2014. *Cited in Sec.* 2.2.2

[YHFK08] S. L. P. Yasakethu, C. Hewage, W. Fernando, and A. Kondoz, "Quality analysis for 3-d video using 2-d video quality models". *IEEE Trans. Consumer Electron.*, vol. 54 (4), pp. 1969–1976, November 2008. *Cited in Sec.* 3.3.1

[YLLL12] H. Yuan, J. Liu, Z. Li, and W. Liu, “Virtual view synthesis for 3d video system: Theoretical analyses and implementation”. *IEEE Transactions on Broadcasting*, vol. 58, pp. 558–568, Mars 2012. *Cited in Sec.* 2.1.1

[YV09] S. Yea and A. Vetro, “View synthesis prediction for multiview video codings”. *Elsevier, Signal Processing: Image Communication*, January 2009. *Cited in Sec.* 2.1.1

[ZTWY13] L. Zhang, G. Tech, K. Wegner, and S. Yea, “3D-HEVC test model 5”, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 JCT3V-E1005, July 2013. *Cited in Sec.* 7, 2, 2.1.1, 2.2.1.2, 2.2, 2.4.3, 2.5.3, 2.6.3, 3.2.1

[ZWPSxZy07] L. Zhan-Wei, A. Ping, L. Su-xing, and Z. Zhao-yang, “Arbitrary view generation based on DIBR”, in *International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pp. 168–171, Xiamen, People’s Republic of China, 2007. *Cited in Sec.* 2.1.1, 3.1.1

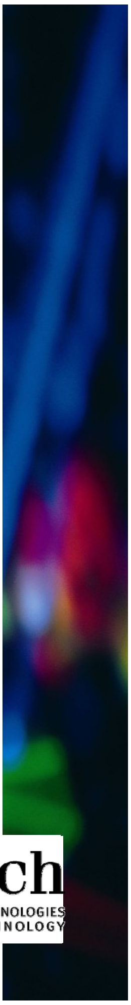
[ZZC+11] Y. Zhao, C. Zhu, Z. Chen, D. Dian, and L. Yu, “Boundary artifact reduction in view synthesis of 3d video: from perspective of texture-depth alignment”. *IEEE Transactions on Broadcasting*, vol. 57, pp. 510–522, 2011. *Cited in Sec.* 2.2.2, 2.6.3

# Synthèse et reconstruction de vues à partir de vidéos compressées multi-vues et multi-sources

**Andrei Iacob PURICA**

**RESUME :** De nos jours, les vidéos sont la forme de multimédia la plus demandée. Ce grand intérêt a alimenté une évolution continue des technologies d'affichage, de transmission et de compression vidéo. Également, il y a aussi beaucoup d'intérêt à trouver le meilleur moyen d'offrir une expérience multimédia dite immersive. Plusieurs solutions ont été étudiées au cours des dernières années et le format vidéo multi-vues plus profondeur a été trouvé pour fournir une solution prometteuse en combinaison avec des algorithmes de synthèse visuelle. Dans cette thèse, nous explorons plusieurs sujets liés à la synthèse et à la reconstruction des vues. Tout d'abord, nous explorons l'utilisation des corrélations temporelles en combinaison avec les techniques traditionnelles de rendu basé sur la profondeur d'image et proposons plusieurs approches pour aborder les problèmes communs des algorithmes de ce type qui sont démontrés pour améliorer la qualité de la synthèse. Comme les algorithmes de synthèse de vues produisent des distorsions localisées élevées, nous évaluons également l'efficacité des mesures d'évaluation de qualité courantes et proposons une évaluation ciblée sur la région d'intérêt. Enfin, nous étudions le problème de la reconstruction vidéo multisource et proposons un modèle de reconstruction qui utilise des algorithmes proximaux primal-dual d'optimisation convexes pour améliorer la qualité et la résolution des vidéos provenant de sources multiples avec des résolutions et des niveaux de compression éventuellement différents.

**MOTS-CLEFS :** synthèse de vues, 3D-HEVC, super-resolution, compression vidéo, évaluation de qualité



**ABSTRACT :** Nowadays, videos are the most demanded form of multimedia. This high interest fueled a continuous evolution of display, transmission and compression technologies. Furthermore, there is also a lot of interest in finding the best way to provide a so-called immersive multimedia experience. Several solutions were investigated over the past years and the Multi-View video plus Depth format was found to provide a promising solution in combination with view synthesis algorithms. In this thesis we explore several topics related to view synthesis and view reconstruction. First, we explore the use of temporal correlations in combination with the traditional Depth-Image-Based-Rendering techniques and propose several approaches to tackle common problems in DIBR type algorithms which are shown to improve the quality of the synthesis. As view synthesis algorithms produce localized high distortions, we also evaluate the effectiveness of common quality evaluation metrics and propose a targeted Region-Of-Interest evaluation. Finally, we investigate the problem of multi-source video reconstruction and propose a model based framework that uses primal-dual splitting proximal convex optimization algorithms to enhance the quality and resolution of videos from multiple sources with possibly different resolutions and compression levels.

**KEY-WORDS :** view synthesis, 3D-HEVC, super-resolution, video compression, quality evaluation