# Modeling, Extracting and Description of Intrinsic Cues of High Resolution Satellite Images: Independent Component Analysis Based Approaches

Payam Birjandi

▶ **To cite this version:**

Payam Birjandi. Modeling, Extracting and Description of Intrinsic Cues of High Resolution Satellite Images: Independent Component Analysis Based Approaches. Signal and Image Processing. Télécom ParisTech, 2011. English. NNT : . pastel-00677956

## HAL Id: pastel-00677956
## https://pastel.hal.science/pastel-00677956

Submitted on 11 Mar 2012

**Doctorat ParisTech**

# T H È S E

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

## Télécom ParisTech

### Spécialité " Signal et Image "

*présentée et soutenue publiquement par*

**Payam BIRJANDI**

le 16 Septembre 2011

## Modélisation et Extraction des Descripteurs Intrinsèques des Images Satellite à Haute Résolution: Approches Fondées sur l'Analyse en Composantes Indépendantes

Directeur de thèse : **Mihai DATCU**

**Jury**

| | | |
|---|---|---|
| **Mme. Inge GAVAT**, Professeur, Politehnica, University of Bucharest | | Rapporteur |
| **M. Philippe BOLON**, Professeur, Polytech Annecy-Chambéry | | Rapporteur |
| **M. Mohammad Ali DJAFARI,** Docteur, Supéléc | | Examinateur |
| **M. Tullio TANZI**, Professeur, Telecom ParisTech | | Examinateur |
| **M. Michel ROUX**, Professeur, Telecom ParisTech | | Examinateur |
| **M. Alain GIROS**, Ingénieur de recherche, CNES | | Examinateur |
| **M. Mihai DATCU**, Professeur, German Aerospace Center (DLR) | | Directeur de thèse |

T
H
È
S
E

*To whom I love*:

*My darling wife; Sharareh, my lovely little daughter; Mojana, my dear parents; Mohammad and Mahin, my kind sisters; Parisa and Pardis and my compassionate grandmother; Ashraf.*

# Acknowledgements

# ABSTRACT

Sub-meter resolution satellite images, capture very detailed information, as for example, shape of buildings and industrial installations, detailed road and road furniture structures, vehicles, etc. Thus, their information content is incredibly rich and also complicated to be extracted. The classical image descriptors as spectral information, texture, shape, etc., are not any more sufficiently accurate to describe the image content. The main purpose of the thesis is to propose descriptors for Sub-meter resolution satellite images especially for those who contain geometrical or man-made structures. Independent Component Analysis (ICA) is a good candidate for this purpose, since previous studies demonstrated that the resulted basis vectors contain some small lines and edges, the important elements in the characterization of geometrical structures.

As a basic analysis, a study about the effects of scale size and dimensionality of ICA system on indexing of satellite images is presented and the optimum dimensionality and scale size are found.

There are two view points for feature extraction based on ICA. The usual idea is to use the ICA coefficients (ICA sources) and the other is to use the ICA basis vectors related to every image. Based on the first point of view, an ordinary ICA source based approach is proposed for feature extraction. This approach is developed and modified through a Topographic ICA system to extract middle level features which leads to a significant improvement in results.

Based on other point of view, two methods are proposed. One of them uses the Bag of words idea which considers the basis vectors as visual words. Second method uses the lines properties inside the basis vectors to extract features. Also, using the lines properties idea, another method is developed which directly detects the line segments in the images.

Finally, the capabilities of proposed descriptors are compared through a supervised classification based on Support Vector Machine (SVM).

# CONTENT

# 0  RESUME EN FRANÇAIS

## 0.1 Introduction

Les images satellites haute résolution contiennent des informations très détaillées comme la forme des bâtiments, les zones industrielles, les structures des routes, les véhicules, etc. Ainsi, leur contenu d'information est hyper riche, et aussi très compliqué à extraire. Parmi les paysages différents, les zones urbaines et des structures géométriques sont les paysages plus compliqués pour les différant domaines de recherches.

Nous allons extraire les indices intrinsèques des images satellite et proposer les descripteurs robustes. En utilisant ces descripteurs, nous serions capables de reconnaître une variété des paysages, en particulier, les structures géométriques au sein des images satellite très haute résolution. Par exemple, nous pouvons trouver des zones urbaines similaires dans une image satellite très large.

Nous insistons sur les formes géométriques ou des structures artificielles comme les sujets de caractérisation, parce que normalement il n'ya pas des difficultés majeures pour la description des paysages naturelles. Figure 0.1 (a) montre une partie d'une forêt comme un exemple des paysages naturels. Normalement, les paysages naturels ont des propriétés qui nous permettent d'utiliser un certain nombre de caractéristiques de texture comme leurs descripteurs. Par exemple, les changements dans les paysages naturels normalement se produisent d'une manière quasi périodique et continue.

De plus, généralement, ils ne contiennent pas des lignes distinctes ou des objets géométriques. D'autre part, dans les structures artificielles nous trouvons souvent des objets géométriques, contenant des lignes et des bords, qui ne sont pas nécessairement distribuées d'une manière régulière. Ainsi, ce type d'images, en comparaison avec les paysages naturels, ne peuvent pas être décrits correctement avec les caractéristiques de texture. Figure 0.1 (b) montre une zone urbaine comme un exemple de structures artificielles.

L'Analyse en Composantes Indépendantes (l'ACI) est la base théorique de cette thèse. Bell et Senjowski [2] ont utilisé l'ACI pour les images naturelles et ont trouvé que les composantes indépendantes des images contiennent des lignes et les bords courts. Ceci est une propriété importante pour la caractérisation des structures géométriques, puisque les objets géométriques contiennent normalement des lignes et des bords. Donc, l'ACI est une candidate appropriée pour définir les descripteurs des patches des images satellite contenant des structures géométriques.

   Dans la figure 0.1, il y a deux patches des images satellite, une de forêt et d'autres de la zone urbaine. Aussi, il y a des exemples des vecteurs de base de l'ACI qui sont obtenus pour chaque catégorie de données. La différence entre les deux ensembles des vecteurs de base est un signe de la capacité de d'ACI pour la caractérisation des images satellite. En particulier, les bords et les lignes dans les vecteurs de base de la région urbaine démontrent que l'ACI peut détecter les caractéristiques principales des structures géométriques.
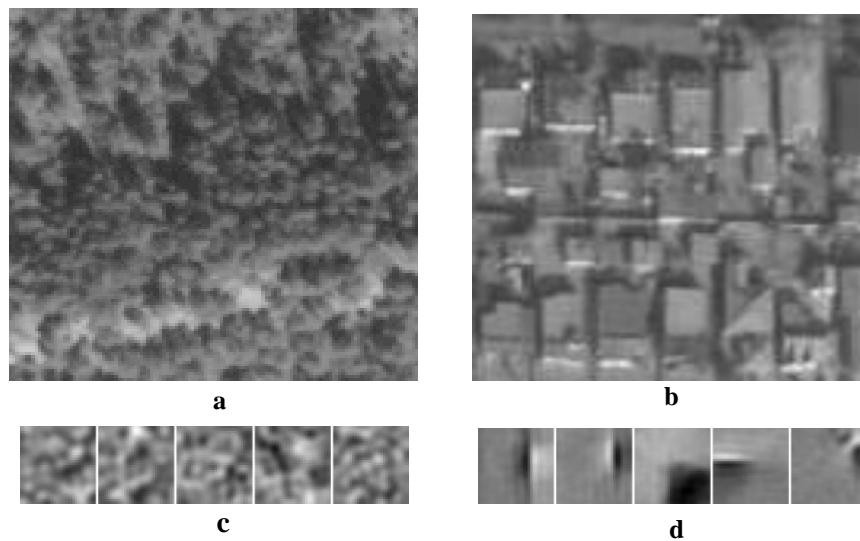


**Figure 0.1** Un exemple de deux classes des images satellites et les vecteurs de base d'ACI. (a): Forêt, typiques des paysages naturels, (b): Zone urbaine, typique des structures géométriques. (c) et (d): Vecteurs de base d'ACI obtenues pour deux classes. Les vecteurs de base de zone urbaine contiennent des lignes, des bars, des bords, etc. Vecteurs de base de forêt sont plus homogènes.

Figure 0.2 montre un schéma des contributions de thèse. La première contribution de thèse est une investigation sur l'effet de la taille de l'échelle et la dimension d'un système de l'ACI qui est utilisé pour caractérisation des images satellite. Cela nous aide à choisir le framework de notre modèle de l'ACI pour extraire des caractéristiques. On propose deux groupes des descripteurs pour les images satellites haute résolution. Le premier groupe contient deux types des descripteurs qui sont basés sur les coefficients (les sources) de l'ACI ordinaire ou l'ACI topographique et le

deuxième contient deux types des descripteurs qui sont basés sur les propriétés des vecteurs de base de l'ACI. En se basant sur notre expérience en l'ACI, nous proposons un autre descripteur qui extrait les caractéristiques des lignes dans les images satellites. Finalement, les capacités des descripteurs proposés sont comparées grâce à une classification supervisée basée sur la Machine à Vecteurs de Support.
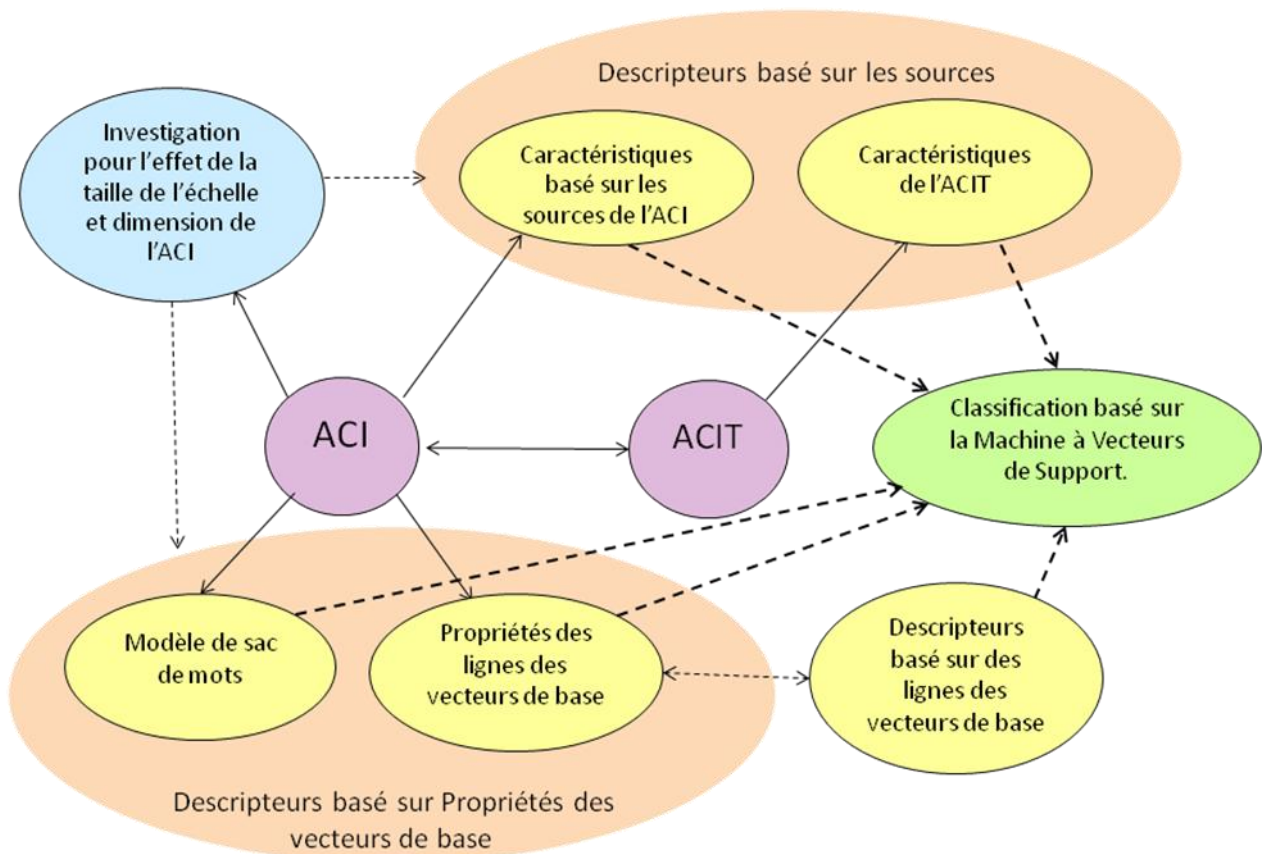


**Figure 0.2** Un schéma des contributions de thèse. Il contient une investigation sur l'effet de la taille de l'échelle et la dimension d'un système de l'ACI, proposition des descripteurs basé sur l'ACI (ou l'ACIT) et vérification des descripteurs proposés grâce à une classification supervisée basé sur la Machine à Vecteurs de Support.

## 0.2 Quels types de caractéristiques avons-nous besoin?

Les caractéristiques de texture présentent une interprétation universelle du paysage mais ne présentent pas des informations détaillées des objets dans le paysage. Au contraire, les descripteurs locaux et les opérateurs morphologiques sont des méthodes capables pour détecter les objets géométriques et caractériser la zone

urbaine, mais ils sont souvent compliqués et avec des vecteurs caractéristique très grands. En effet, nous avons besoin des caractéristiques, ni exactement au niveau de la texture et ni au niveau des descripteurs locaux. En outre, les descripteurs locaux et les opérateurs morphologiques sont généralement utilisés pour détecter les objets, mais nous n'allons pas détecter des objets géométriques dans les images satellite. Le but principal de cette thèse est de proposer des descripteurs pour les patches des images satellite contenant des paysages différents, en particulier, les structures géométriques ou artificielles.

## 0.3 Images satellite optiques d'une résolution sub-métrique

 Dans cette thèse, nous allons extraire les caractéristiques des images satellitaires optiques. Ces caractéristiques sont relatives aux propriétés spatiales des images et les caractéristiques des couleurs des images ne sont pas importantes. Ainsi, nous avons seulement besoin des images en niveaux de gris pour nos méthodes d'extraction de caractéristiques. En d'autres termes, les images satellites panchromatiques optiques sont convenables à l'objective de notre recherche. Cependant, nous pouvons utiliser les images satellites multispectrales, mais d'abord, on les transforme en images en niveaux de gris.

 La résolution spatiale est le paramètre le plus important des images satellitaires qui sont traitées dans cette thèse. Le but de thèse est de définir des descripteurs pour les images satellites contenants des structures géométriques ou artificielles. Les détails de ce genre de structure ne sont pas visibles dans les images d'une résolution spatiale plus d'un mètre par pixel. Par conséquent, nous ne considérons que les images avec une résolution spatiale d'un mètre ou sub-métrique. Par exemple, les images de QuickBird d'une résolution spatiale de 60cm ou des images Ikonos avec une résolution spatiale d'un mètre sont convenables à nos besoins.

### 0.3.1 Patch contextuel, Micro-patch

 Les images satellite ont normalement les tailles très larges et contiennent une variété de paysages artificiels ou naturels. Par conséquent, la définition des caractéristiques de ces grandes images n'est pas raisonnable. Pour l'extraction des caractéristiques, nous avons besoin des patches d'images plus petites, qui ne contiennent qu'un seul type de structure ou paysage. Bien que nous sommes capables d'extraire les caractéristiques pour les images contenant de nombreuses classes de paysages, ceci n'est pas souhaitable car chaque vecteur des caractéristiques est censé décrire une seule classe de paysage.

 Nous devons considérer nos patches suffisamment grands afin qu'ils contiennent un certain nombre d'objets et un contexte clair. Autrement dit, ils doivent présenter un paysage significatif. Par exemple, si un patch contient un seul bâtiment ou une partie d'un immeuble sans un contexte clair, il pourrait être idéal pour le but de détecter des objets mais il n'est pas approprié pour l'objectif de notre travail. En outre, si notre patch est trop grand, il peut contenir plusieurs parties, chacune d'entre elles pourrait être individuellement considérée comme un paysage interprétable.

Nous appelons les patches pour lesquelles nous allons définir les descripteurs *les patches contextuels*. En d'autres termes, nous cherchons des patches d'images qui peuvent présenter un certain nombre des formes géométriques comme les maisons, les immeubles et autres structures artificielles avec un contexte clair. Nous insistons sur le mot *contexte* pour séparer notre tâche de la détection des objets. Comme nous travaillons avec les images satellitaires avec la résolution sub-métrique, il semble que la taille des patches contextuels entre 100*100 pixels à 300*300 pixels soit raisonnable. Dans cette thèse, nous travaillons avec les patches contextuels de la taille de 200* 200 pixels.
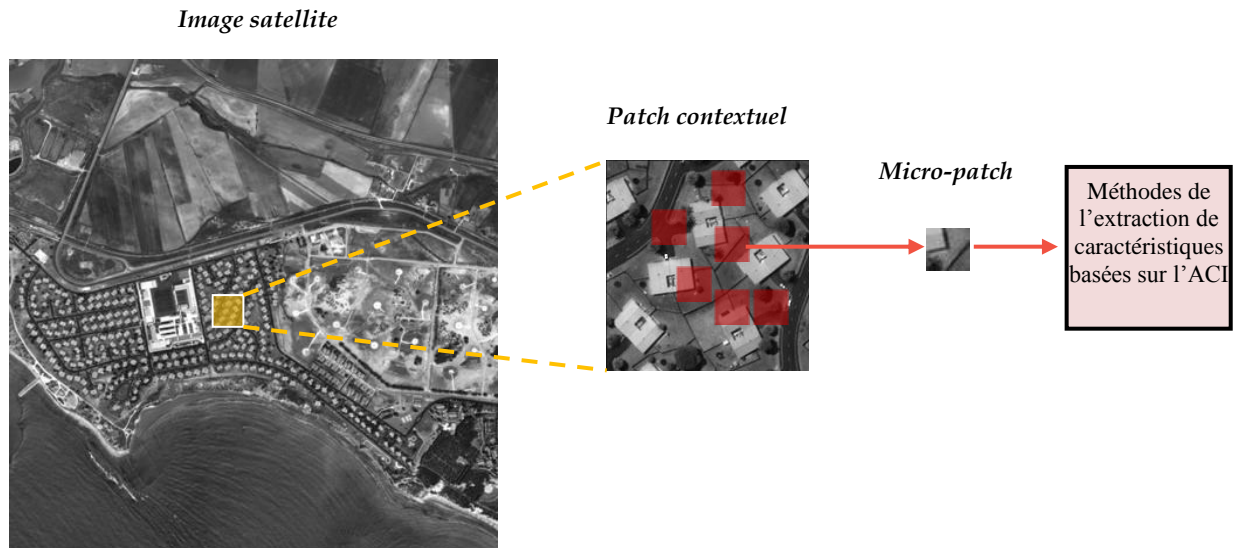


**Figure 0.3**   Trois niveaux des images qui sont utilisées dans cette thèse. Les images satellites initiales contiennent plusieurs classes de paysages. Les patches contextuels contiennent généralement une classe de paysage et conviennent à l'extraction de caractéristiques. Les Micro-patches extraits de chaque patch contextuel sont utilisés dans les procédures d'extraction de caractéristiques basées sur l'ACI car les patches contextuels sont trop grands pour être utilisé directement dans la procédure de l'ACI.

Néanmoins, les patches contextuels restent grands pour être utilisés directement dans certaines méthodes d'extraction des caractéristiques expliquées dans la thèse. Donc, nous recueillons un certain nombre de patches plus petits, *Micro-patches,* depuis chaque patch contextuel pour être traitées dans la procédure d'extraction des caractéristiques. Trois niveaux des images sont illustrés dans la Figure 0.3.

## 0.4 Etat de l'art

L'idée de l'Analyse en Composantes Indépendantes (l'ACI) est de réduire la redondance de données sans perdre les caractéristiques importantes de données.

Barlow [1] a mentionné que le cerveau humain mémorise des informations de l'environnement visible et les utilise pour diminuer la redondance de données. Ici, la redondance a un sens de la dépendance statistique. Par exemple, si nous voyons une voiture, nous attendons de voir aussi une route. Autrement dit, il existe une corrélation statistique ou la dépendance entre la voiture et la route dans notre cerveau, parce que, habituellement, nous les voyons ensembles. L'idée initiale de l'ACI est similaire mais, ici la dépendance est mesurée entre les niveaux de gris des pixels d'une image qui sont considérés comme des variables aléatoires.

*Séparation Aveugle de Sources* a été l'un des premiers problèmes pour lequel l'ACI a été élaboré. De l'autre côte, il peut être considéré comme une forme généralisée de l'*Analyse en Composantes Principales* (l'ACP).

Une étude importante a été faite par Bell et Senjowski [2] qui ont utilisé l'ACI pour des images naturelles. Ils ont trouvé que les composants indépendants des images contiennent des lignes courtes et les bords. Olshausen et Field [3] ont démontré que des propriétés similaires peuvent être trouvées dans le système visuel humain. Existence des bords et des lignes dans les composants de l'ACI est aussi intéressante pour notre recherche. Parce que, nous cherchons certains modèles pour manipuler des caractéristiques de bords des objets dans les images satellites.

Plusieurs méthodes qui appliquent l'ACI pour les images existent. La plupart de ces méthodes utilisent des modèles simples de l'ACI, mais certains d'entre eux utilisent un modèle combiné de l'ACI comme la méthode proposée par Lee, Lewicki, et TJ Sejnowsk [4]. Un exemple de l'utilisation de l'ACI pour les données de télédétection est l'étude effectuée par Zhang, X. et CH Chen [7]. En outre, Zhang et al [8] a proposé une méthode basée sur l'ACI pour la classification des images de la télédétection. Bien que l'ACI est fréquemment utilisé pour certains types d'images telles que des images naturelles, des images du texte et des images du visage, il n'a pas été très utilisé pour la caractérisation des images satellite. Ce dernier peut ouvrir un domaine de recherche, notamment, l'utilisation de l'ACI pour la caractérisation des images satellite.

## 0.5 Fondements de l'Analyse en Composantes Indépendantes

Il est supposé qu'il y a un ensemble de $n$ sources d'information ($S_1, S_2, \ldots, S_n$), chacune d'entre elles est statistiquement indépendante par rapport aux autres. Autrement dit, la valeur de chaque source n'a aucun effet sur les valeurs d'autres sources. Du point de vue statistique, nous pouvons considérer ces sources comme des variables indépendantes aléatoires. L'ensemble de ces variables peuvent être indiquées avec un vecteur aléatoire qui est appelé le vecteur des sources ($S = [S_1, S_2, \ldots, S_n]^T$). Ensuite, nous supposons que les composants indépendants sont combinés par un processus linéaire. En d'autres termes, nous avons un ensemble de variables observées, ($x_1, x_2, \ldots, x_m$) qui sont eux-mêmes des variables aléatoires parce qu'elles sont produites comme des combinaisons linéaires des variables aléatoires initiales. Nous noterons l'ensemble des variables aléatoires observées avec un vecteur aléatoire ($x_{obs} = [x_1, x_2, \ldots, x_m]^T$) et nous

appelons ce vecteur *vecteur observé* ou *signal observé*:

$$x_{obs} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \cdots & a_{1n} \\ a_{21} & a_{22} \cdots & a_{2n} \\ \vdots & & \\ a_{m1} & a_{m2} \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} = \mathbf{A}s \qquad (0.1)$$

La matrice $\mathbf{A}$ est appelée *matrice de mixture*, puisqu'elle mixe les sources indépendantes. Nous pouvons réécrire l'équation (0.1) comme suit:

$$x_{obs} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = s_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} + s_2 \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} + \cdots + s_n \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix} \qquad (0.2)$$

$$= s_1 a_1 + s_2 a_2 + \cdots + s_n a_n$$

Les vecteurs $a_1, a_2, \cdots, a_n$ ont la même dimension que le signal observé. Ces vecteurs peuvent être considérés comme les vecteurs de base d'un nouvel espace pour représenter nos données. Donc, ils sont appelés les *vecteurs de base*.

Si nous utilisons l'ACI pour les images, nos signaux observés peuvent être considérés comme des petites images (micro-patches) qui sont recueillies depuis des images initiales. Donc les vecteurs de base de l'ACI auraient la même taille que les micro-patches :



**Figure 0.4 :** Les signaux observés et vecteurs de base quant on applique l'ACI pour les images.

L'Analyse en Composantes Indépendantes est la procédure d'estimation des vecteurs de base telle que les sources de l'ACI seraient les plus indépendants que possible. En d'autres termes, un ensemble des signaux observés, $x_{obs}(k)$ , est donné et nous allons estimer la *matrice de mixture,* $\mathbf{A}$ ,qui contient les vecteurs de base, $a_1, a_2, \cdots, a_n$ , et l'ensemble des sources pour chaque signal observé, $s_1(k), s_2(k), \cdots, s_n(k)$ ,de telle sorte que ces sources seraient, statistiquement, les plus indépendantes possible.

## 0.5.1 Mesure de l'indépendance statistique

Il n'existe pas un moyen simple pour mesurer l'indépendance parmi un ensemble des variables aléatoires. *Théorème central limite* sous certaines conditions peut nous aider à évaluer le montant de l'indépendance existant entre des variables aléatoires. Ce théorème exprime que si nous créons une combinaison linéaire de $n$ variables aléatoires indépendantes, la distribution de nouvelle variable aléatoire tend vers une distribution Gaussienne si $n$ tend vers l'infini. En d'autres termes, une somme de deux variables aléatoires indépendantes généralement a une distribution qui est plus proche de Gaussienne que chacune des deux variables aléatoires initiales.

Nous avons supposé que $x_{obs}$ est une mixture des sources indépendantes. Nous définissons une nouvelle variable aléatoire, comme la combinaison linéaire des composantes de $x_{obs}$ :

$$z = \sum w_i x_i = w^T x_{obs} \tag{0.3}$$

$w$ est un vecteur qui détermine les coefficients de la combinaison linéaire. On peut remplacer $x_{obs}$ par $\mathbf{A}s$ :

$$z = w^T x_{obs} = w^T \mathbf{A}s = v^T s = \sum v_i s_i \tag{0.4}$$

$v$ est un nouveau vecteur qui est défini comme $v = \mathbf{A}^T w$. Alors, $z$ est une combinaison linéaire des sources indépendantes. Selon le *théorème central limite,* la variable aléatoire $z$ est plus Gaussienne que chacune des sources et elle devient moins Gaussienne quand il est égal à l'un des sources.

Dans ce cas, une seule composante de vecteur $v$ est non nulle. Par conséquent, l'objectif est d'estimer vecteur $w$ telle qu'elle maximise le non-Gaussianité de $z$. Ce vecteur correspond à un vecteur $v$ qui n'a qu'une seule composante non nulle. Ainsi, la procédure d'apprentissage peut commencer par la sélection d'une valeur initiale pour le vecteur $w$. Ensuit, dans les étapes d'apprentissage, nous essayons de trouver les maxima locaux de critère du non-Gaussianité de la variable $z$.

Selon notre critère pour non-Gaussianité on peut établir un algorithme d'apprentissage. Dans cette thèse, nous utilisons l'algorithme FastICA [5] pour l'estimation les vecteurs de base de l'ACI.

## 0.6 ACI pour les images satellite: effets de taille des échelles et de dimension

Il ya une relation entre la taille des vecteurs de base de l'ACI et capacité du système pour la caractérisation des images satellite. Normalement, si on augmente la taille des vecteurs de base de l'ACI, notre système sera plus capable pour caractériser les images satellite. Par contre, le volume des calculs augmentera aussi. Ainsi, nous ne pouvons pas augmenter la taille des vecteurs de base de l'ACI sans limite.
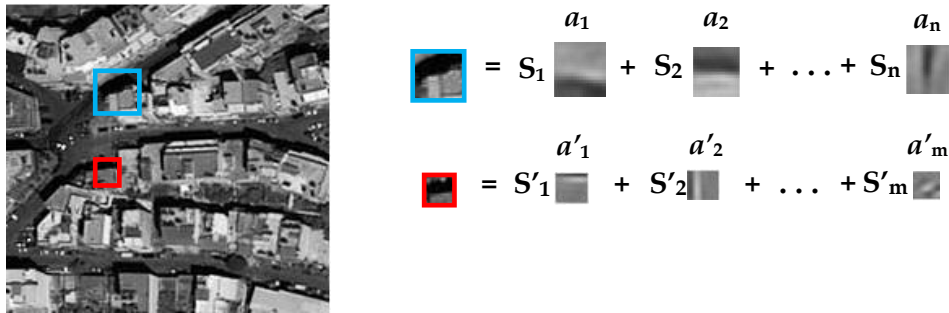


**Figure 0.5 :** L'effet de la dimension et la taille des échelles du système de l'ACI sur la caractérisation des images satellite doivent être étudiées

Une relation similaire existe entre la dimension de système l'ACI (le nombre de composants de l'ACI) et la capacité du système pour caractérisation des images satellites. Notre but est de trouver les points optimaux pour la taille des vecteurs de base de l'ACI et le nombre de composants.

### 0.6.1 Effet de dimension

La dimension du système est exprimée par le *facteur de réduction* (**r**) qui est le nombre normalisé de composants de l'ACI. Autrement dit, le nombre de composants de l'ACI divisé par $n^2$ qui est la taille des vecteurs de base de l'ACI. Ça signifie que les vecteurs de base sont des fenêtres de $n * n$ pixels.

Une façon d'évaluer l'efficacité d'un système de l'ACI est de comparer les micro-patches initiales ( $x$ ) et leurs micro-patches correspondants qui sont reconstruites en utilisant les coefficients de l'ACI ( $\hat{x}$ ). Une approche habituelle pour comparer les deux micro-patches est *l'erreur de reconstruction* qui est calculé avec l'équation (0.5):

$$e = mean(\sqrt{(x - \hat{x})^2}) / mean(\sqrt{x^2}) \qquad (0.5)$$

Nous devons considérer deux paramètres: l'erreur de reconstruction et le temps de calcul. Ceci peut être exprimé par l'optimisation d'une *fonction de coût* comme:

$$CF(r) = kt(r) + (1-k)e(r) \qquad (0.6)$$

$r$ est le facteur de réduction et $k$ est un paramètre qui représente l'importance de *temps de calcul* ($t$) par rapport à *l'erreur de reconstruction* ($e$). L'idée est d'obtenir le *temps du calcul* et *l'erreur de reconstruction* comme deux fonctions du *facteur de réduction.* La *fonction de coût* est la combinaison de ces deux fonctions et le but est de trouver le minimum de cette fonction.
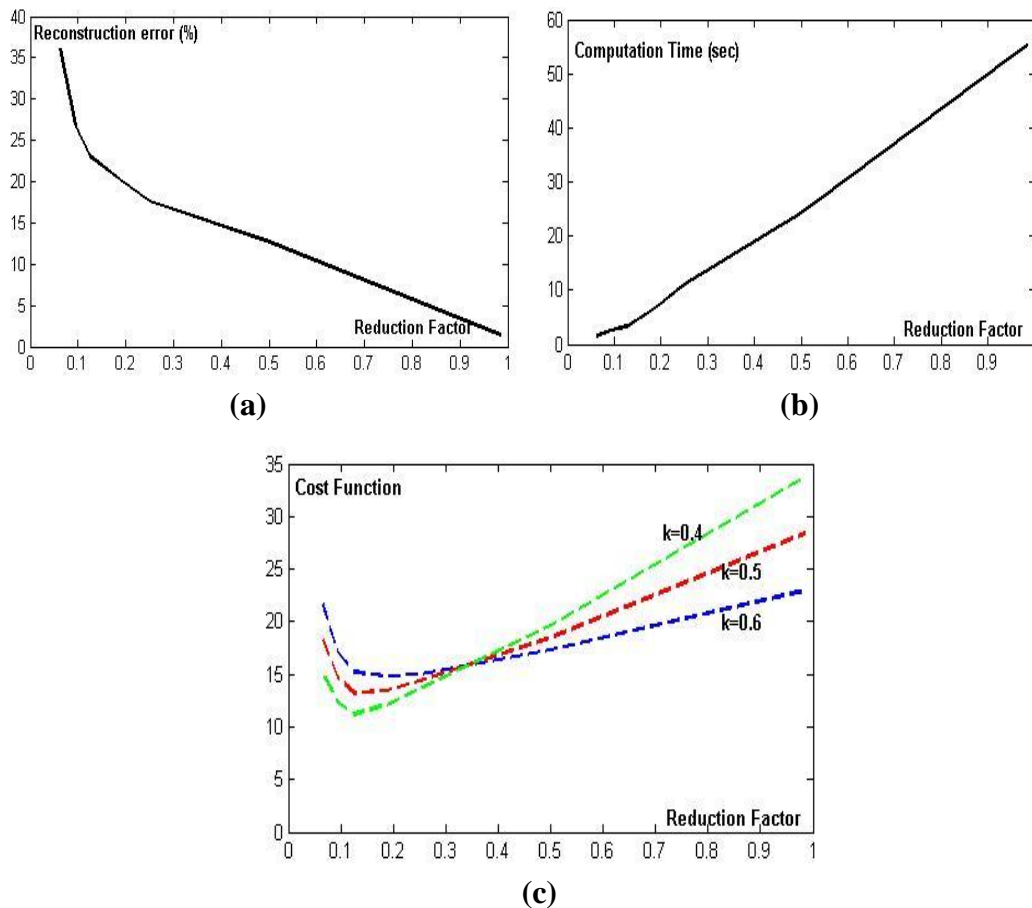


**Figure 0.6:** Détermination du *facteur de réduction* optimal (la dimension optimal) (a): Temps de calcul comme une fonction du facteur de réduction. (b): Erreur de reconstruction comme une fonction du facteur de réduction (c): Fonction de coût comme une fonction du facteur de réduction.

Nous voyons que pour les valeurs raisonnables de $k$, le facteur de réduction optimal varie entre 0,08 et 0,14.

## 0.6.2 Effet de la taille d'échelle

Nous pouvons utiliser une approche similaire pour trouver la taille optimale des échelles. Nous définissons une *fonction de coût* comme l'équation (0.6) mais, cette fois en fonction de la taille des vecteurs de base:

$$CF(m) = kt(m) + (1-k)e(m) \qquad (0.7)$$

Pour des valeurs raisonnables de $k$, la taille optimale des échelles est obtenue autour de 16*16.
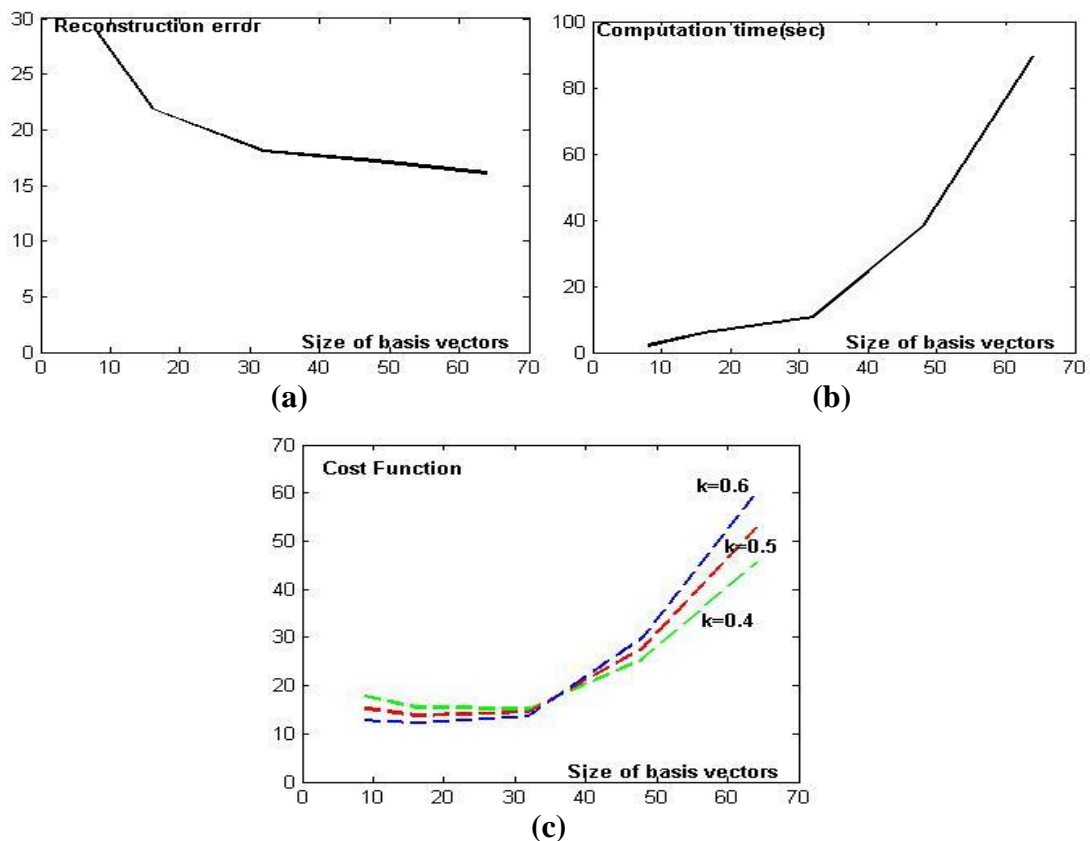


**Figure 0.7:** Détermination de la taille optimale des échelles (a): Temps de calcul comme une fonction de la taille des échelles. (b): Erreur de reconstruction comme une fonction de la taille des échelles. (c): Fonction de coût comme une fonction de la taille d'échelle.

## 0.6.3 Filtres de Gabor comme une étape de prétraitement

Nous avons découvert que dans certains vecteurs de base de l'ACI, nous ne voyons que des petits changements au niveau des coins et le reste de la surface des vecteurs de

base ne contient pas d'information importante. Nous proposons d'utiliser des filtres de Gabor-ondelettes pour mesurer la quantité des changements qui sont placés dans les parties centrales du patch d'apprentissage. Nous avons fourni un ensemble des 100 filtres de Gabor situé en 10 échelles de l'angle et 10 échelles de la fréquence. Pour chacun, le point d'origine est considéré comme le pixel central du filtre. Un patch d'apprentissage est sélectionné si son énergie correspondant à ce système est supérieure à un certain seuil.



$$G = g_1^2 + g_2^2 + \ldots + g_n^2 \overset{?}{>} \text{Threshold}$$
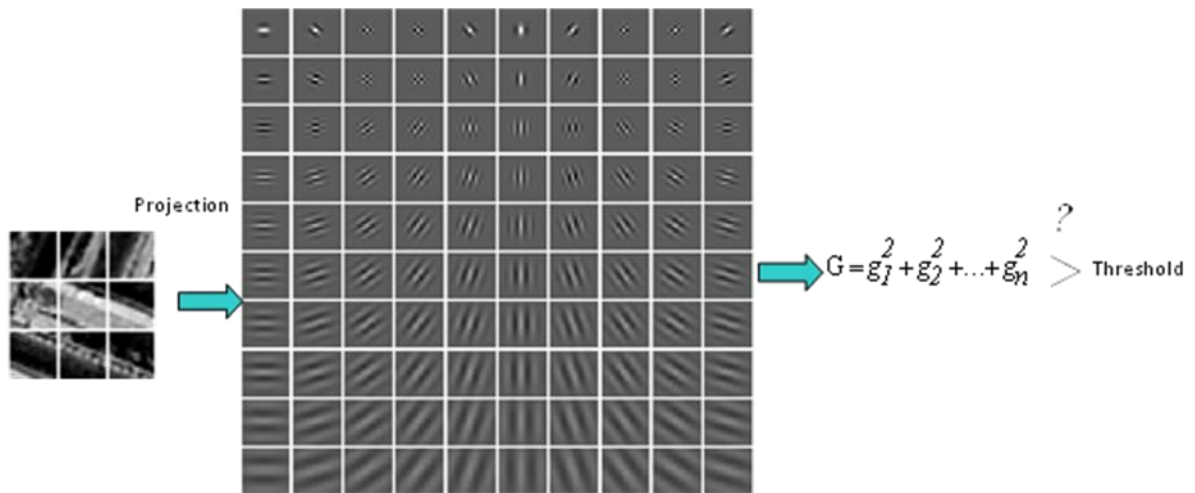
**Figure 0.8:** Filtres de Gabor comme une étape de prétraitement.

Un exemple de résultat d'utilisation de cette étape de prétraitement est montré dans le Figure 0.9.



a                    b
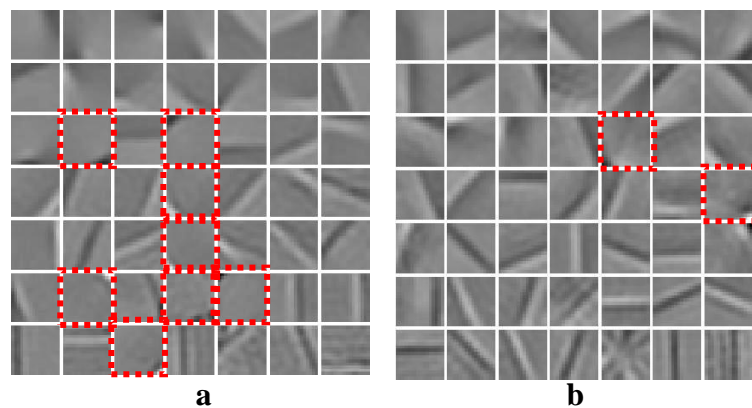
**Figure 0.9:** Résultat d'utilisation des filtres de Gabor comme une étape de prétraitement. (a) : Les vecteurs de base qui sont obtenu avec une procédure ordinaire. 8 vecteurs de base présentent seulement les petits changements à leurs coins (b) : Les vecteurs de base qui sont obtenu avec une étape de prétraitement de filtres de Gabor. Nombre de tels vecteurs de base est réduit à 2.

## 0.7 L'extraction de caractéristiques basée sur les sources de l'ACI

Il y a deux points de vue pour l'extraction de caractéristiques à l'aide de l'ACI. L'approche habituelle est basée sur les coefficients de l'ACI (les sources de l'ACI) et l'autre est basé sur les vecteurs de base de l'ACI. Dans ce chapitre, nous expliquons l'idée de l'extraction de caractéristiques depuis des coefficients de l'ACI. Cette idée est illustrée dans la Figure 0.10. Nous recueillons un nombre suffisant des micro-patches et les décomposons sur l'ensemble des vecteurs de base. Pour chaque micro-patch, nous obtenons un ensemble de $n$ sources. Nous appliquons la moyenne quadratique sur les échantillons différents d'une source et nous obtenons une caractéristique pour cette source.

$$S_1(1), S_2(1), \ldots, S_n(1)$$
$$S_1(2), S_2(2), \ldots, S_n(2)$$
$$\vdots$$
$$S_1(d), S_2(d), \ldots, S_n(d)$$

$$\sqrt{\frac{1}{d}\sum_{k=1}^{d} S_1^{2}(k)} \quad \sqrt{\frac{1}{d}\sum_{k=1}^{d} S_2^{2}(k)} \quad \sqrt{\frac{1}{d}\sum_{k=1}^{d} S_n^{2}(k)}$$

$$f_1 \qquad f_2 \qquad \bullet\bullet\bullet \qquad f_n$$
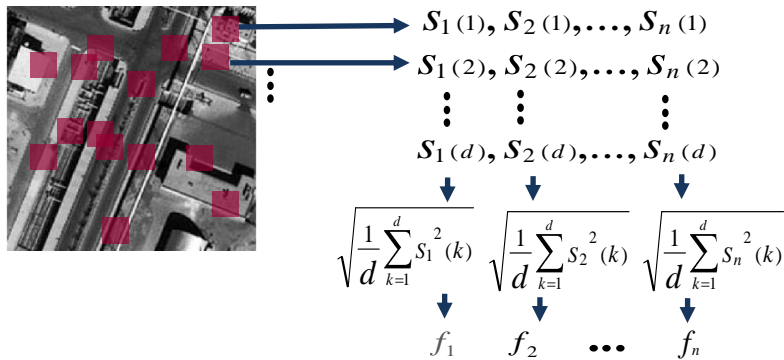
**Figure 0.10:** L'extraction de caractéristiques basée sur les sources de l'ACI

### 0.7.1 Amélioration de l'ensemble des vecteurs de base

Nous proposons une approche pour améliorer l'ensemble des vecteurs de base pour le cas où le but est de séparer la classe de zone urbaine et la classe de zone non urbaine. Nous combinons les vecteurs de base les plus importants de chaque classe pour fournir un nouvel ensemble de vecteurs de base.
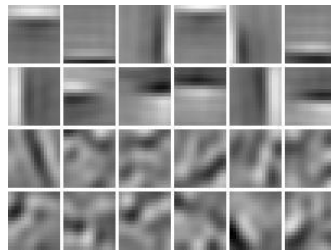
**Figure 0.11:** Nouvel ensemble de vecteurs de base. Les deux lignes supérieures sont les 12 vecteurs de base les plus significatifs de la classe de la zone urbaine et les deux lignes inférieures sont les 12 vecteurs de base les plus significatifs de la classe de la zone non urbaine.

## 0.8 Caractéristiques basée sur l'ACI Topographique

L'indépendance entre les composantes de l'ACI complique l'utilisation des résultats de l'ACI, puisque nous ne connaissons pas la priorité et l'importance des composantes. L'ACI Topographiques est une version améliorée et généralisée de l'ACI. Cette dernière nous conduit à réduire le nombre de caractéristiques qui sont extraites pour les images. Selon le modèle de l'ACI, les composants sont censés être indépendants et il n'ya aucune relation entre les composants différents. Par conséquent, nous devons considérer l'ensemble des composants comme le vecteur de caractéristiques. Cependant, en l'ACI Topographique (l'ACIT) [43] la dépendance entre deux composants est une fonction de leur distance dans la topographie. Ces dépendances peuvent être utilisées pour extraire des caractéristiques de niveau intermédiaire et de réduire la dimension du vecteur de caractéristiques. La procédure de génération des vecteurs de base de l'ACIT est un peu plus compliquée mais le résultat est bien effectif pour définir les caractéristiques. Figure 0.12 montre la différence entre un ensemble des composants de l'ACI simple et un ensemble des composants de l'ACI Topographique.



| a | b |
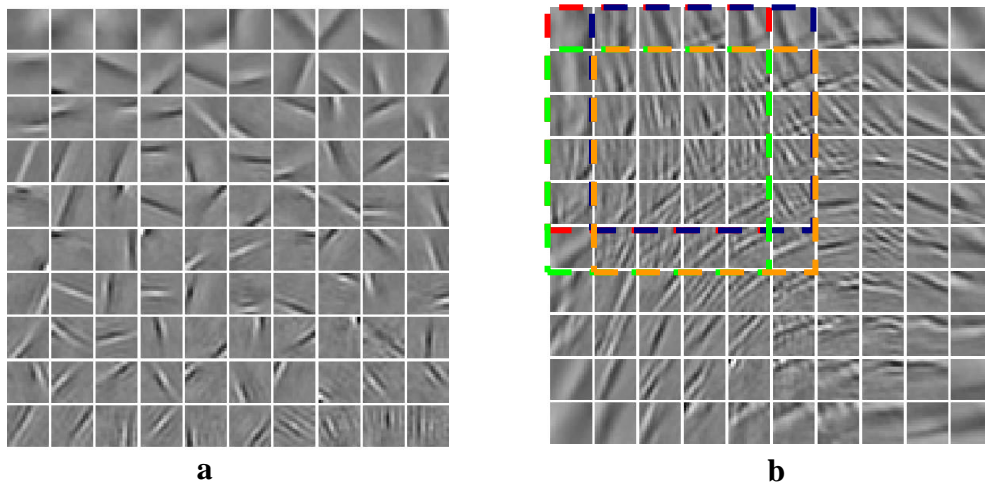
**Figure 0.12:** (a) L'ACI ordinaire. L'ordre des composants n'est pas important. Il n'y a aucune relation entre les composants (b) L'ACI Topographique. Chaque composant a une position spécifique dans la topographie. Les composant sont censées dépendants dans les régions locaux.

Nous pouvons combiner un ensemble de composants de l'ACI, qui sont censés dépendants, pour produire une caractéristique de niveau intermédiaire. Dans notre modèle, le voisinage du système de l'ACIT est 5 * 5 composants et nous prenons la moyenne des 25 caractéristiques ordinaires (composants de l'ACIT) comme une caractéristique de niveau intermédiaire. Figure 0.13 montre les régions de topographie qui génèrent les 9 caractéristiques de niveau intermédiaire.
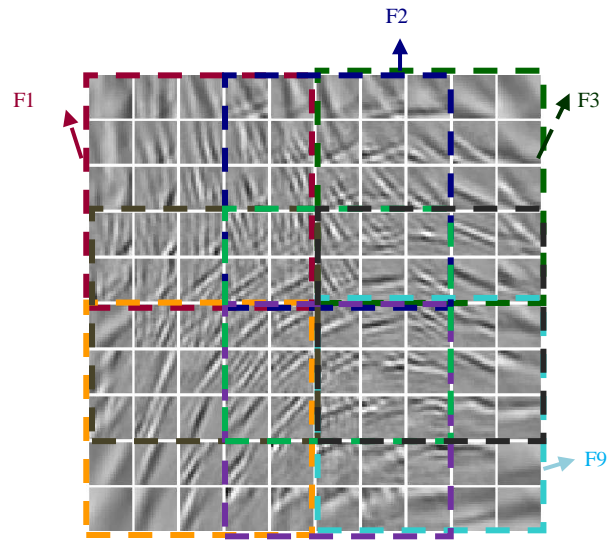
**Figure 0.13:** Génération des 9 caractéristiques de niveau intermédiaire grâce à dépendance des composants de l'ACIT.

## 0.9 Caractéristiques basée sur les vecteurs de base de l'ACI: Sac de mots

Dans la littérature de l'ACI, les caractéristiques sont normalement obtenues par un traitement de sources de l'ACI. Dans la thèse, nous proposons des méthodes qui extraient les caractéristiques depuis les vecteurs de base de l'ACI. La première méthode utilise une modèle sac de mots pour traitement des vecteurs de base de l'ACI.

Le sac de mot est une modèle pour traitement des textes. Il représente un document de texte comme un histogramme qui montre les répétitions des mots de dictionnaire dans le document. Alors, on doit déterminer une analogie entre les textes et les images. Nous définissons les patches contextuels comme les documents visuels. Après une procédure d'apprentissage de l'ACI pour chaque document (patch contextuel), les vecteurs de base correspondants sont obtenus et considérés comme les mots de document.

La prochaine étape est de définir le dictionnaire, un ensemble de tous les mots autorisés dans le modèle sac de mots. Nous proposons deux approches pour le définir. La première utilise une méthode de *clustering* parmi tous les vecteurs de base. Pour chaque cluster on prend son centre comme un mot de dictionnaire. La deuxième approche fournie un ensemble de vecteurs de base pour chaque classe des images satellites et collecte ces ensemble de vecteurs de base.

La dernier étape est l'étiquetage dans le quel pour chaque mot (vecteur de base) de document nous choisissons le mot de dictionnaire qui est le plus similaire au mot de document. Pour cela on utilise la corrélation entre les deux mots comme un critère de la similarité. Finalement, nous avons un descripteur pour chaque patch contextuel: l'histogramme qui montre les répétitions des mots de dictionnaire. Figure 0.14 est un schéma de la caractérisation des images satellite grâce à un modèle sac de mots pour les vecteurs de base des patches contextuels.

**Document *i***



**W<sub>i,1</sub>**

*Les vecteurs de base de document*

**W<sub>i,25</sub>**

**D<sub>1</sub>**   *Dictionnaire*

**D<sub>64</sub>**

*L'étiquetage*

*L' histogramme*
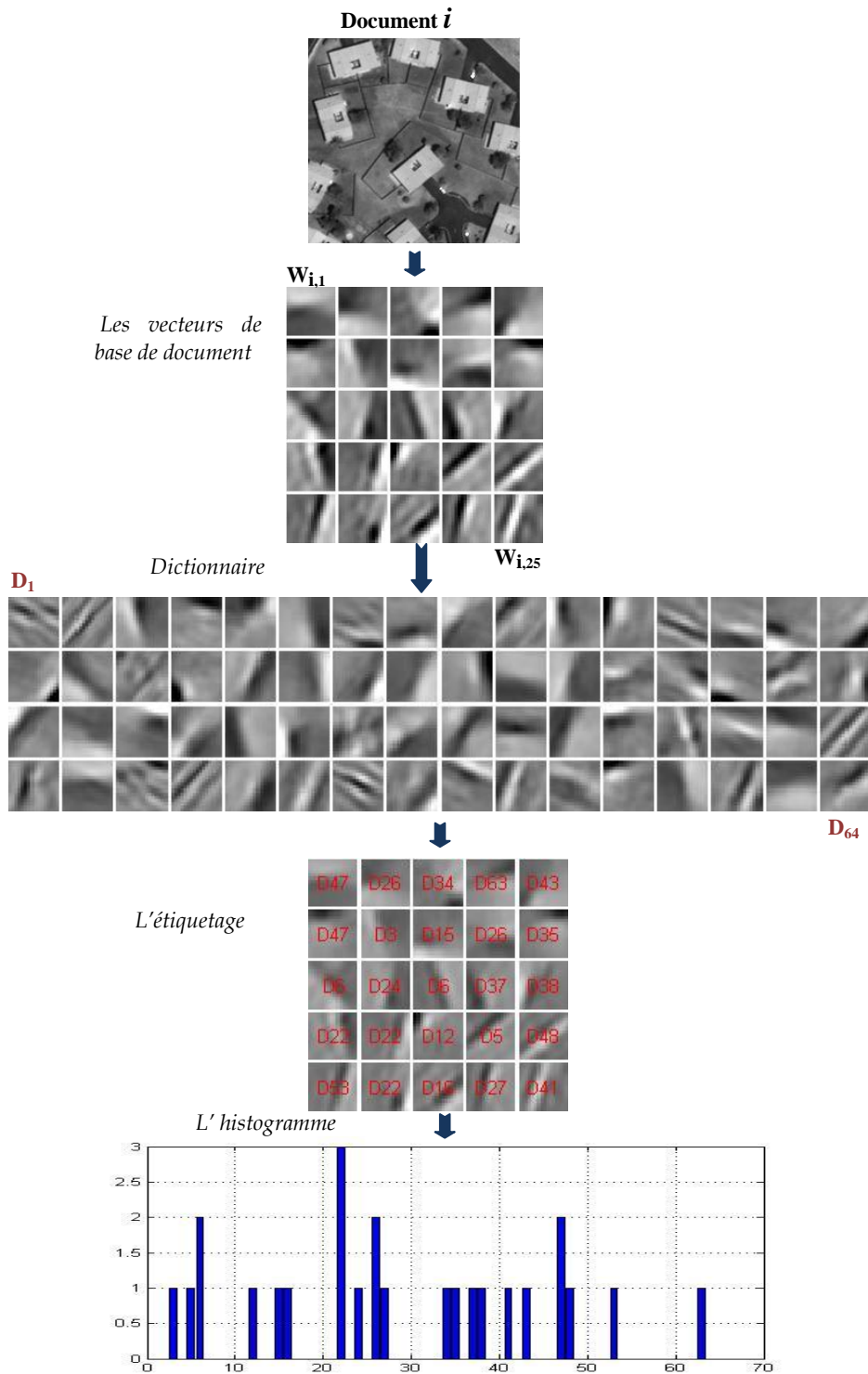
**Figure 0.14:** Caractérisation des patches contextuels grâce à un modèle sac de mots pour les vecteurs de base des patches contextuels. Chaque patch contextuel est considéré comme un document et ses vecteurs de base comme ses mots. Le descripteur est l'histogramme qui montre les répétitions des mots de dictionnaire dans le document.

## 0.10 Caractéristiques basée sur les vecteurs de base de l'ACI: les lignes des vecteurs de base

L'autre idée pour extraire les caractéristiques depuis les vecteurs de base de l'ACI est de détecter des contours dans les vecteurs de base et les modeler comme les lignes directes. Les caractéristiques sont définies comme les propriétés des lignes des vecteurs de base. Figure 0.15 montre les étapes différentes de cette idée.
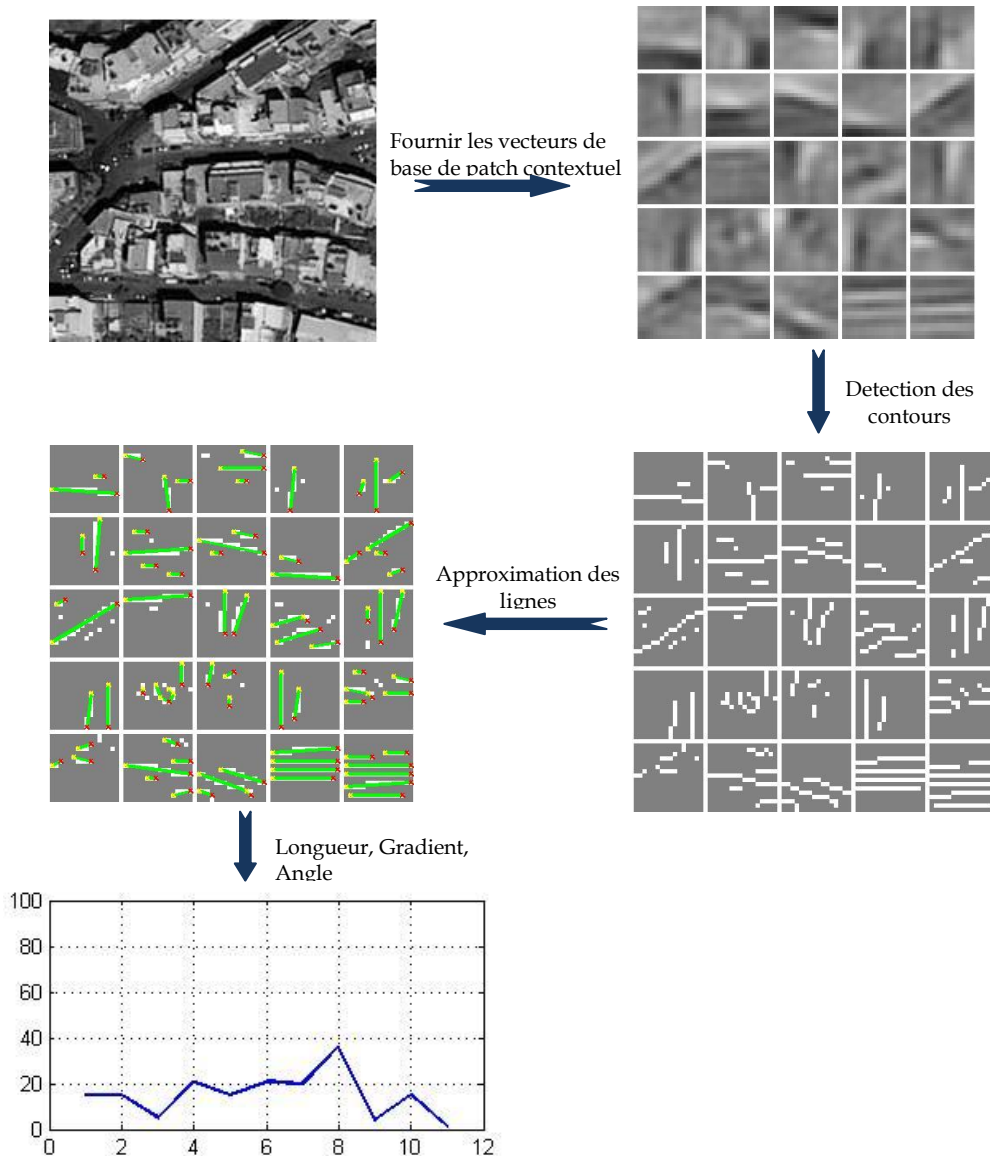


**Figure 0.15:** Caractérisation des patches contextuels grâce aux propriétés des lignes des vecteurs de base. Les lignes approximé dans les vecteurs de base d'un patch contextuel peut être considérés une signature pour patch contextuel.

Pour un patch contextuel, d'abord, on doit fournir les vecteurs de base de l'ACI. Nous détectons les contours dans les vecteurs de base de l'ACI grâce à l'operateur Sobel qui est une méthode simple de gradient du premier ordre. Après, les contours doivent être modelés par des lignes directes. Les méthodes classiques d'estimation des lignes (Hough, par exemple) sont normalement lentes et compliquées. Nous proposons, donc, notre méthode pour estimation des lignes directes. Cette méthode estime les lignes de trois pixels et ajoute des pixels à deux cotés jusqu'à ce que la direction de ligne ne change pas. La longueur, l'amplitude de gradient, et l'angle des lignes sont pris comme les propriétés plus importantes. Pour chaque propriété on détermine les boîtes, chacune représente un certain intervalle, et on dépose chaque propriété de ligne dans la boîte correspondante. Le nombre des éléments dans chaque boite est définie une caractéristique de patch contextuel.

## 0.11 Descripteur basé sur segments de lignes

D'après notre expérience sur l'ACI, on a trouvé les connexions entre l'ACI et les propriétés gradients de l'image. Par exemple, on a proposé un modèle qui utilise les lignes des vecteurs de base pour extraire des caractéristiques de l'image satellite qui sont liée aux propriétés gradients des vecteurs de base. On propose une idée similaire qui utilise directement les lignes de l'image satellite.
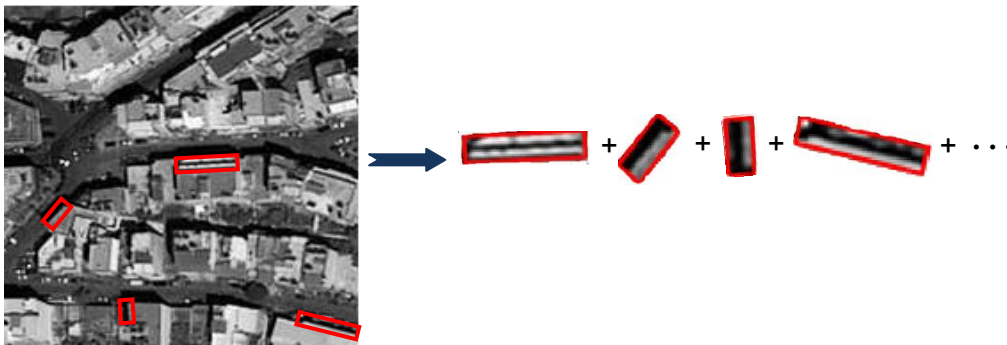


**Figure 0.16:** Les segments contenant lignes au lieu des vecteurs de base de l'ACI pour modélisation des images satellite.

Pour la plupart des vecteurs de base, une ou plusieurs lignes compose leur structure principale et les autres parties des vecteurs de base ne contiennent pas d'information importante. Les vecteurs de base, normalement, ont des formes carrées. Cela signifie que pour représenter une ligne avec la longueur de $n$ pixels nous avons besoin d'un vecteur de base de la taille de $n*n$ pixels. Cependant, cette ligne peut être représentée par un segment de $d*n$ pixels. Dans le quelle $d$ est généralement entre 3 et 5 selon la largeur de la ligne (Figure 0.16). En d'autres termes, nous pouvons utiliser directement les segments de ligne dans l'image pour caractérisation des images. Figure 0.17 montre les étapes différentes de cette idée de caractérisation.
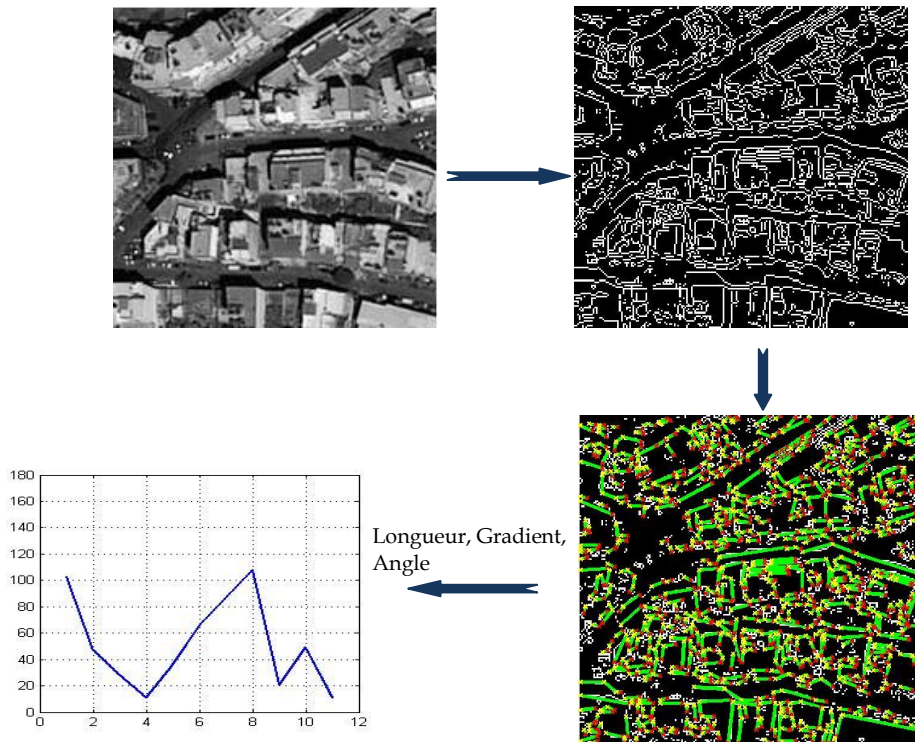
**Figure 0.17:** Caractérisation des patches contextuels grâce aux propriétés des segments de ligne

## 0.12 Evaluation des descripteurs

Pour chaque descripteur proposé, nous effectuons un clustering simple pour mesurer leur capacité. Néanmoins, nous avons besoin d'une vérification plus fiable. Ainsi, nous comparons les méthodes proposées par une classification supervisée basée sur la Machine à Vecteurs de Support (MVS).
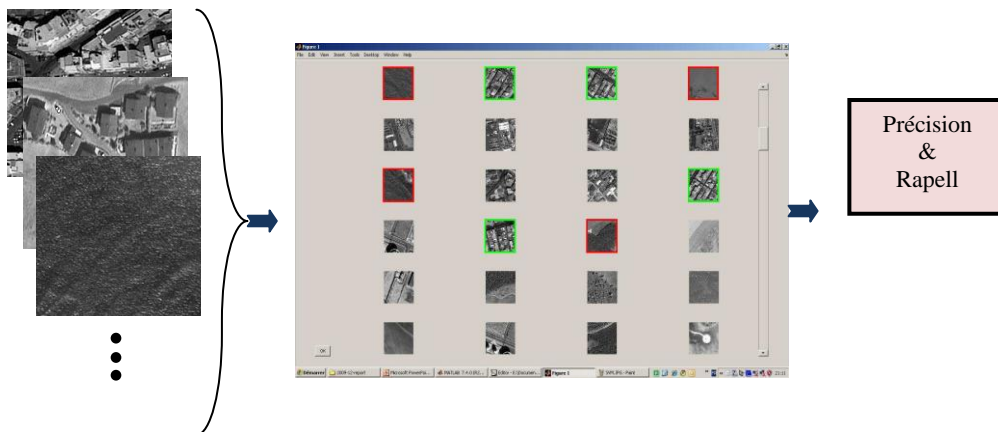


**Figure 0.18:** Détecter les classes grâce à un classificateur supervisé basé sur MVS
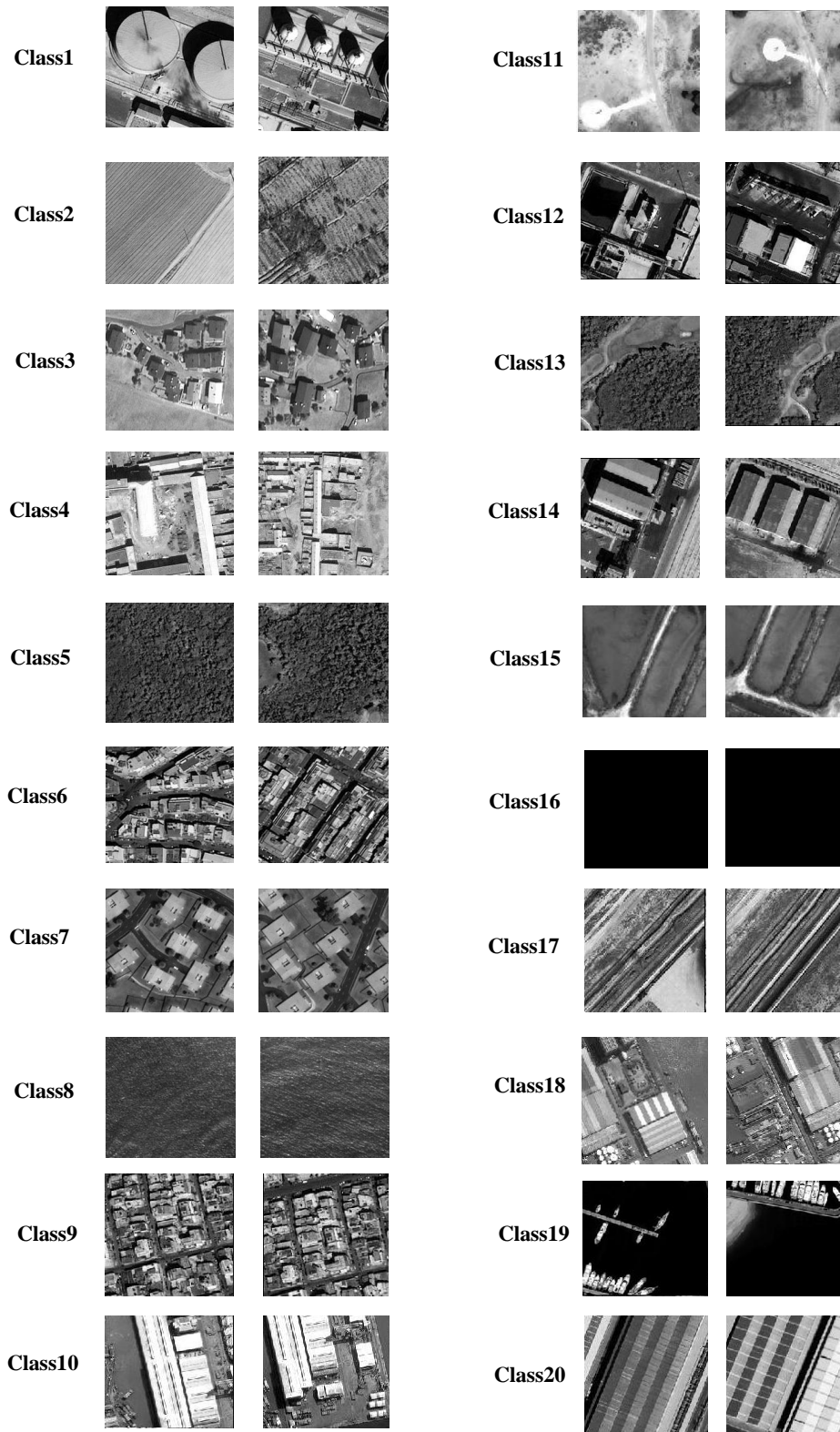
**Figure 0.19:** Les classes détectées. 2 patches contextuels sont présentés pour chaque classe.
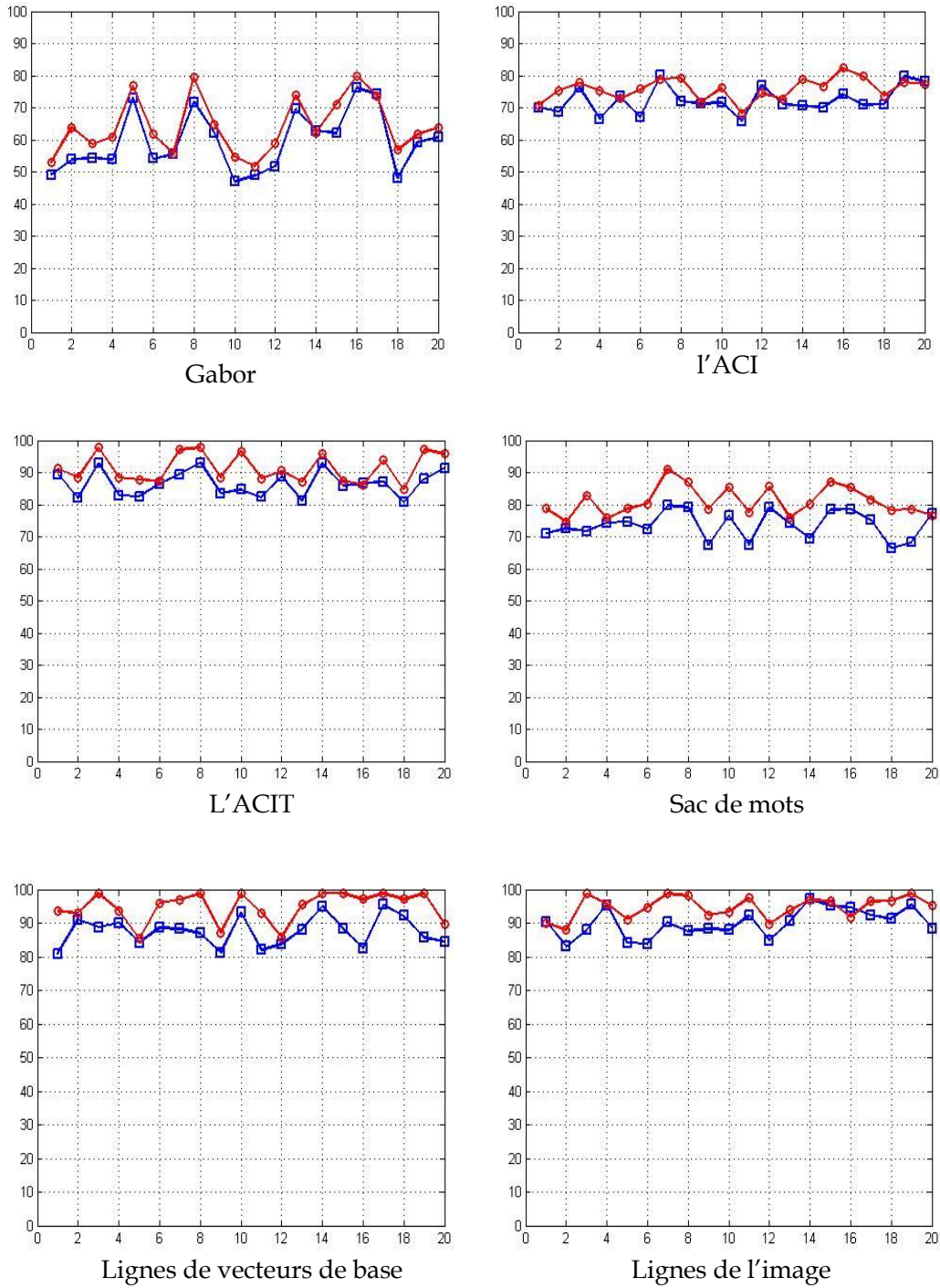
**Figure 0.20:** Les *précisions* et les *rappels* obtenus pour 20 classes détectées pour différents types de descripteurs. Les lignes rouges sont les précisions et les lignes bleues sont les rappels.

Pour cela, nous fournissons un outil visuel de MATLAB, avec un noyau de MVS, qui permet à l'utilisateur de choisir les échantillons positifs et négatifs pour la détection d'une classe des patches contextuels. Nous choisissons un descripteur et essayons de détecter 20 classes montrées dans le Figure 0.19. Après, nous répétons l'expérience avec les autres descripteurs proposés et aussi avec un descripteur de Gabor-Ondelette comme un descripteur typique de texture.

Pour 20 classes détectées nous calculons la *précision* et le *rappel* pour les différents descripteurs (Figure 0.20). Enfin, nous pouvons comparer l'efficacité des différents descripteurs selon leurs précisions -rappels, temps de calcul et longueur de descripteur. Tableau 0.1 présente un résumé de la comparaison des descripteurs. Nous voyons que les résultats de méthode Gabor n'est pas précis pour les classes contenant les objets géométriques. Mais, pour quelques classes naturelles, elle présents la *précision* et le *rappel* acceptable. Pour nos méthodes, il n'y a pas une différence entre les résultats de deux types de classes. Ça veut dire les méthodes proposées fonctionnent bien pour les classes contenant les objets géométriques.

**Tableau 0.1:** Comparaison des descripteurs

|  | Moyen de précisions et rappels | Moyen de temps de calcul | Longueur de descripteur |
|---|---|---|---|
| **Gabor** | P=64.14% R=59.41% | 0.15 sec | 27 |
| **ACI** | P=75.79% R=72.29% | 0.15 sec | 27 |
| **ACIT** | P=91.39% R=86.57% | 0.21 sec | 11 |
| **Sac de mot** | P=81.01% R=73.68% | 0.82 sec | 66 |
| **Lignes de vecteurs de base** | P=94.87% R=87.54% | 0.96 sec | 13 |
| **Lignes de l'image** | P=93.37% R=88.63% | 0.59 sec | 13 |

## 0.13 Conclusions et perspectives

Dans cette thèse, nous avons essayé de présenter une méthodologie pour étudier la nature statistique des images satellite et d'extraire leurs signatures statistiques. Les images satellite sont considérées comme des signaux aléatoires multi-variables tels que chaque pixel peut être une variable aléatoire individuelle et l'objectif est d'étudier les dépendances statistiques entre cette variable aléatoire (pixel) et les autres

variables aléatoires (pixels). L'Analyse en Composantes Indépendantes a été utilisée comme le base théorique de la thèse pour étudier les dépendances statistiques dans les images satellite.

L'objectif de la thèse est de présenter des descripteurs plus précis que les caractéristiques de texture et plus simples par rapport aux descripteurs locaux pour les images satellite haute résolution. Les descripteurs présentés sont placé quelque part entre les approches de texture et les approches locaux. D'un côté, ils donnent une interprétation globale du paysage. De l'autre côté, ils traitent des propriétés gradient qui sont importants dans les structures d'objets géométriques.

Approches présentées pour définir les descripteurs sont des approches globales. En d'autres termes, ils ne sont pas dépendants au contenu, à la résolution ou type des images. Cependant, nous les avons utilisés pour les images satellites haute résolution. Algorithmes d'extraction des caractéristiques présentées par la présente thèse peuvent être vérifiés avec les images satellitaires des autres capteurs et avec d'autres résolutions. Cela peut être une œuvre future prévue pour cette thèse. Ils peuvent également être vérifiés avec d'autres types d'images telles que des images médicales, des images naturelles, des images dans le domaine de l'astronomie, etc.

Nous avons proposé une classification supervisée pour évaluer les caractéristiques. Les classificateurs supervisés ont quelques désavantages. Par exemple, ils sont fortement dépendants au point de vue d'utilisateur. Ainsi, autre perspective de la thèse est de fournir une procédure standard pour l'évaluation des descripteurs afin de réduire les effets de point de vue d'utilisateur.

# CHAPTER 1

# INTRODUCTION

In this chapter, as an introduction, we explain the motivations and the goals of the thesis. In addition, we present a general overview of the thesis contributions.

## 1.1 Motivations and goals of the thesis

Sub-meter resolution satellite images, capture very detailed information, as for example, shape of buildings and industrial installations, detailed road and road furniture structures, vehicles, etc. Thus, their information content is incredibly rich, and also complicated to be extracted. The classical image descriptors as spectral information, texture, shape, etc., are not any more sufficiently accurate to describe the image content.

Recently, many researches are being done to study, develop, and elaborate algorithms for extraction of information from high resolution optical satellite images. Among different scenes in the satellite imagery, urban areas and geometrical structures have been the most interesting ones for many applications and studies. We are going to extract the intrinsic cues of satellite images and to propose robust descriptors so that using these descriptors we would be able to recognize a variety of the scenes, especially the geometrical structures, among the VHR (Very High Resolution) satellite imagery. For example, using these descriptors, we would be able to find the similar urban zones in different parts of a large satellite image.

Here, we insist on the geometrical shapes or the urban areas or the man-made structures inside the satellite images, as the zone of our research, because normally there is no major difficulty in the natural scenes description and recognition. Usually, images from natural landscapes have some properties which let us use a number of texture-like features as their descriptors. They correspond to a specific range of frequency and changes in their spatial domain happen in a continuant and also, usually, in a quasi periodic manner. In addition, usually they don't contain distinct

lines, edges or geometric objects. Figure 1.1(a) shows a part of forest as an example of natural landscapes. On the other side, in the man-made structures we usually find geometrical objects, containing separating lines and edges, which are not necessarily distributed in a regular manner inside the image. Thus, this kind of images, comparing with natural landscapes, cannot be described properly with the textural features. Figure 1.1(b) shows an urban area as an example of man-made structures.



**a**

**b**

**c**

**d**
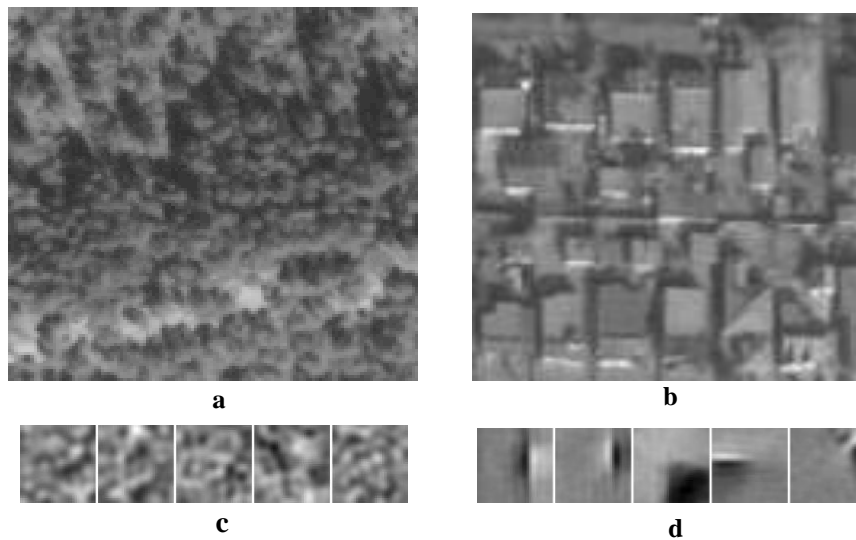
**Figure 1.1**: Examples of two classes of satellite images and ICA basis vectors obtained for them. **(a):** *Forest*, typical of natural landscapes, **(b):** *Urban area,* typical of geometrical structures. **(c)** and **(d):** ICA basis vectors obtained for two classes. Urban area basis vectors contain lines, bars, edges…but forest basis vectors are more homogeneous.

Texture-like features give a universal interpretation from the scene but don't present detailed information about the objects inside the scene. On the other side, local descriptors and morphological operators are capable methods for detecting the geometrical objects and urban area characterization, but they are usually time consuming and complicated methods with very long feature vectors. Actually, we need some features neither exactly in the level of texture and nor in the level of local descriptors. Moreover, the local descriptors and morphological operators are usually used in the object detection algorithms but in many applications we are not going to detect geometrical objects. In fact, in a lot of applications we don't need to detect particular objects or zones, but the objective is to give a semantic interpretation from the scenes containing different landscapes, particularly man-made structures. The principal purpose in this thesis is to propose patch descriptors which are capable for geometrical structures characterization with regards to the context of the satellite image patches.

Independent Component Analysis (ICA) is the theoretical basis of the thesis. Here, we just express the principal property of ICA which motivates us to use it for satellite

image characterization. Details of ICA come in chapter 5.

Bell and Senjowski [2] used ICA for natural images and found out that the independent components of images include short lines and edges. This is an important property for geometrical structure characterization, since the geometrical objects normally consist of lines and edges. Thus, ICA could be a suitable candidate to define descriptors for satellite image patches containing geometrical structures.

In Figure 1 we see two satellite image patches, one from forest and other from urban area. Also, we see examples of ICA basis vectors which are obtained for each class of data. The difference between the two sets of basis vectors is a sign of ICA capability for satellite image characterization. Particularly, the edges and lines in urban area basis vector demonstrate that ICA can detect the principal characteristics of geometrical structures.

During the thesis we try to extract features related to Independent Components Analysis from VHR optical satellite images. These features are supposed to be able to characterize this kind of images especially those who contain the man-made or geographic structures.

## 1.2 Overview of thesis contributions
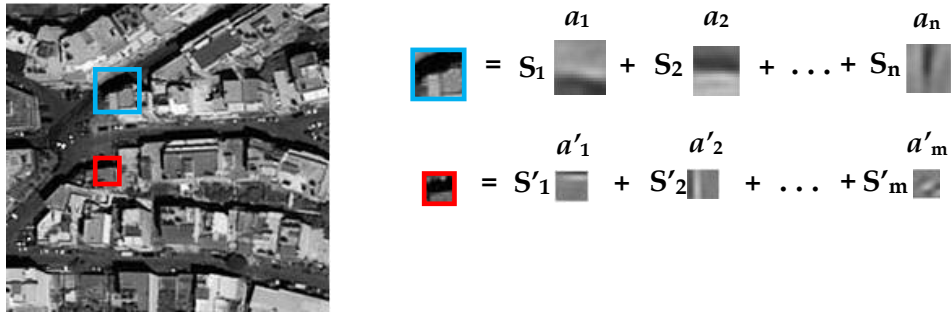
The main purpose of the thesis is to propose descriptors for optical satellite image patches. A descriptor, simply, can be defined as a vector of features and every feature is supposed to describe one characteristic of image or a pattern inside the image. Previously, many methods are presented by researchers to extract features from images. In chapter 2 we give definitions and notations for different image features, as well as the feature extraction methods. We will mostly focus on the methods which are related to our work.

On the other side, our research domain is strongly related to the Earth Observation (EO) and Remote Sensing (RS). So, in Chapter 3 we introduce the basic concepts, goals and challenges related to Earth Observation and Remote Sensing. In addition, we explain what kinds of satellite images are used in the thesis.

Then, in Chapter 4, we will review previous studies related to our work to illuminate the atmosphere of researches around the main aspects of the work. Since our objective is to characterize geometrical or man-made landscapes, it is strongly related to urban area detection and we initially investigate the related works around urban area detection and classification. We also review state of art of Independent Component Analysis (ICA) and its applications on satellite image processing.

Since Independent Component Analysis (ICA) is the theoretical framework of the thesis, it is suitable to explain its fundamentals, concepts and algorithms in a separate chapter, i.e. Chapter 5.

We started our practical work on ICA with a study about the effect of scale size and dimensionality of ICA system when it is used for satellite image indexing. There is a relation between the size of ICA basis vectors and the capability of ICA for characterization of satellite images. Normally, if we increase the size of ICA basis vectors then our ICA system will be more capable to index satellite images. But the volume of computations will grow as well. Thus, we are not able to increase the size of ICA basis vectors limitlessly.

**Figure 1.2:** Study of dimensionality and scale behavior of ICA components which are used for satellite image characterization is important for choosing the framework of ICA system.

Similar relation exists for the dimensionality of ICA system or the number of ICA components. Usually the dimensionality is expressed as the reduction factor which is the normalized number of ICA components. That is, the ratio of ICA components to $n^2$, where $n$ is the size of ICA basis vectors. The purpose of Chapter 6 is to find the optimum point for the size of ICA basis vectors and the number of components. We define the reconstruction error as a criterion of ICA system's capability for image characterization. In addition, we consider the computation time for obtaining the basis vectors as the other criterion. Using the cost functions which are combinations of these two criterions we conclude that the optimum point for the reduction factor is placed between 0.08 and 0.14 and the basis vectors with the size of 16*16 is the most suitable case for our work.

In addition, in chapter 6, an approach is proposed to reduce the redundancy in a set of basis vectors. We propose to use a set of Gabor-wavelet filters to choose the optimum learning micro patches. In other words, we choose the micro patches which have the higher energy in a set of Gabor filters.
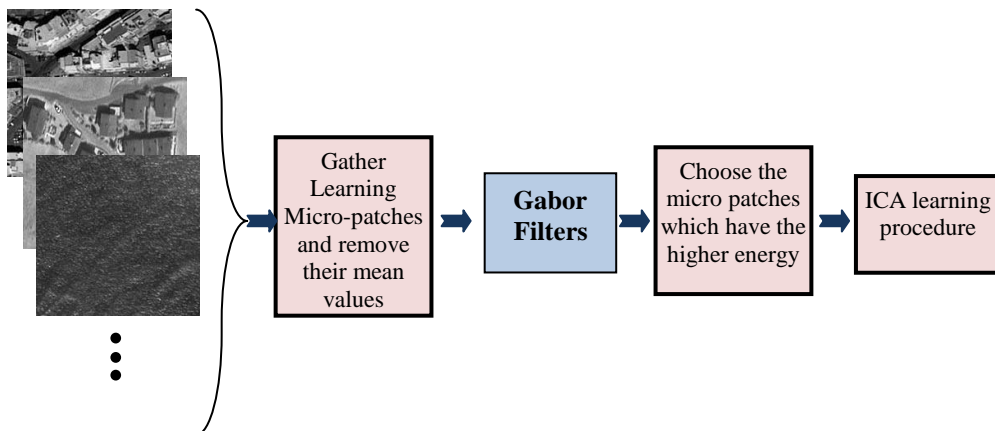


**Figure 1.3:** Using a set of Gabor wavelet filters to choose the optimum learning micro patches in order to reduce the redundancy in a set of basis vectors

There are two points of view for feature extraction based on ICA. The usual approach is to use ICA coefficients (ICA sources) and the other is based on the ICA basis vectors related to every image. In Chapter 7 we explain the idea of extracting features from the ICA coefficients. This idea is illustrated in Figure 1.4. We gather a sufficient number of micro patches and decompose them onto the set of basis vectors and for each of them we obtain a set of sources. Applying the root mean square over the same sources of different sampled micro patches we obtain the ICA features.
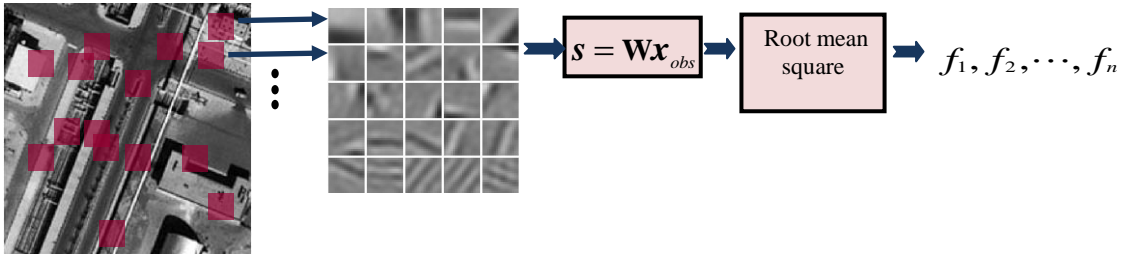


**Figure 1.4:** ICA source based feature extraction

In Chapter 7 we also propose an approach to improve the set of basis vectors when we are going to separate the urban area class from non-urban area class. We combine the most important basis vectors of each class to make a new set of basis vectors:
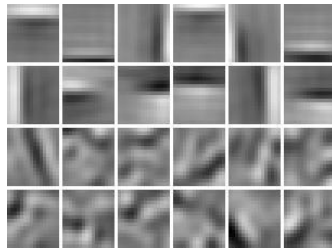


**Figure 1.5:** Improved set of basis vectors. The two upper rows are the 12 most significant basis vectors of urban area and the lower two rows are the 12 most significant basis functions of non-urban area.

The independence of ICA components can make it difficult to use the results of ICA, since we don't know the priority and importance of the components. Topographic ICA is a generalized and also improved version of ICA. It leads us to reduce the number of features which are extracted from the image. In ICA model, the components are supposed to be independent and there is no relation among different components. Therefore, we have to consider the set of all components as the feature vector. However, in TICA the dependency between two components is a function of their distance in the topography. These dependencies can be used to extract some middle level features and reduce the dimension of feature vector. This is shown in Figure 1.6. In other words, TICA has the capability of combining a number of low level features to provide some middle level features. The principals of Topographic ICA and the

method that uses Topographic ICA for feature extraction is explained in Chapter 8.
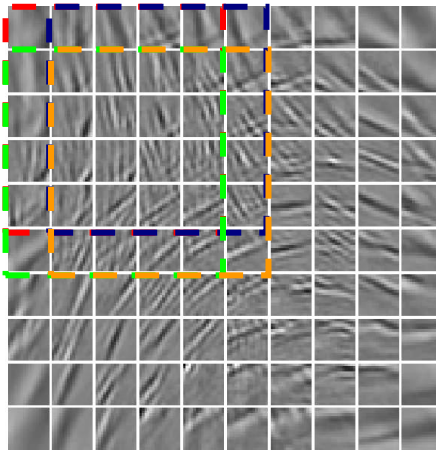


**Figure 1.6:** In a set of TICA basis vectors we are able to combine a number of low level features because of existing dependencies among the components.

The second viewpoint for feature extraction is to consider the ICA basis vectors which are obtained for each image. Actually, we are going to work with the characteristics of basis vectors related to each image. This idea is explained in Chapter 9. Moreover we use the *Bag of words* model and a Bayesian approach to make it easier to extract features. In this model we consider each satellite image as a visual document and its related basis vectors as its visual words.

Figure 1.7 illustrates the principles of ICA words idea for satellite image characterization. Initially, we have to apply the ICA learning procedure for one image to extract the related basis vectors as its visual words. We also need a dictionary in which all possible visual words are placed. Two approaches are proposed in chapter 9 for defining the dictionary. At last, a histogram is made which shows the number of each dictionary's word repeats in the document (satellite image). In addition we propose a Bayesian approach which helps us to classify different visual documents.
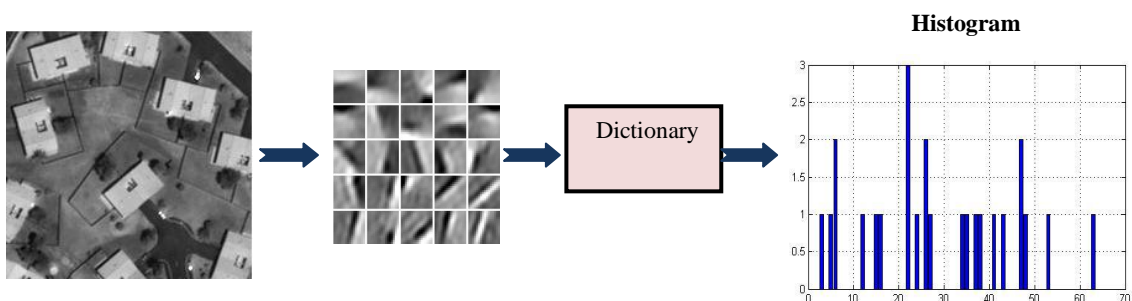


**Figure 1.7:** Bag of words model for satellite image characterization. We consider each satellite image as a visual document and its related basis vectors as its visual words.

In Chapter 10, we try to extract features from a set of basis vectors, using the characteristics of lines which are detected in each basis vector. The idea is shown in Figure 1.8:
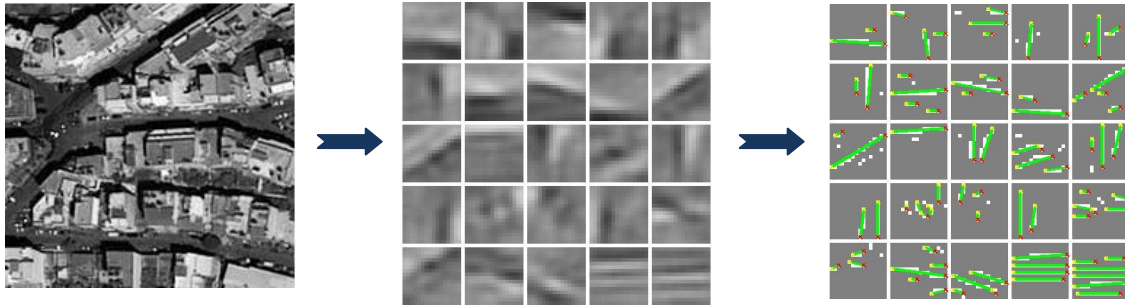


**Figure 1.8:** Feature extraction from basis vectors of a satellite image, using their lines properties.

There are several steps, such as obtaining basis vectors, edge detections and line approximation. For line approximation a new approach is proposed in chapter 10. Finally, for each line, we put each of its characteristics (length, average of gradient magnitude and angle) into the corresponding bin in order to make a feature vector.

In Chapter 11, we introduce a method whose idea is taken, during the thesis, from ICA basis vectors. Using a line detection method and considering the variation around each line, we define new components (segments) to describe the image. In fact, we apply the edge detections and line approximation directly for a satellite image:
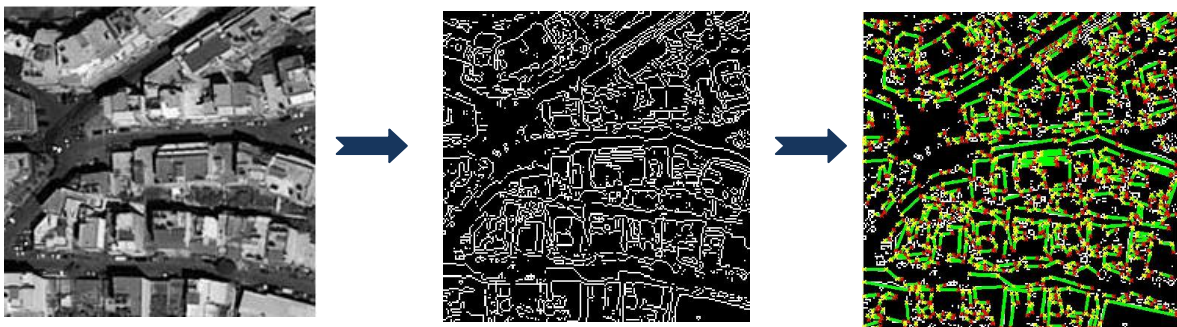


**Figure 1.9:** Feature extraction from a satellite image patch, using its lines properties.

For every proposed descriptor in chapters 7 to 11 we perform a simple clustering to demonstrate their capability for VHR satellite image characterization. However, in chapter 12, to have a more reliable verification, we compare the proposed methods through a supervised classification. This supervised classification is based on the Super Vector Machine (SVM). For that, we provide a visual relevance feedback tool which allows the user to choose the positive and negative samples for detecting a class.
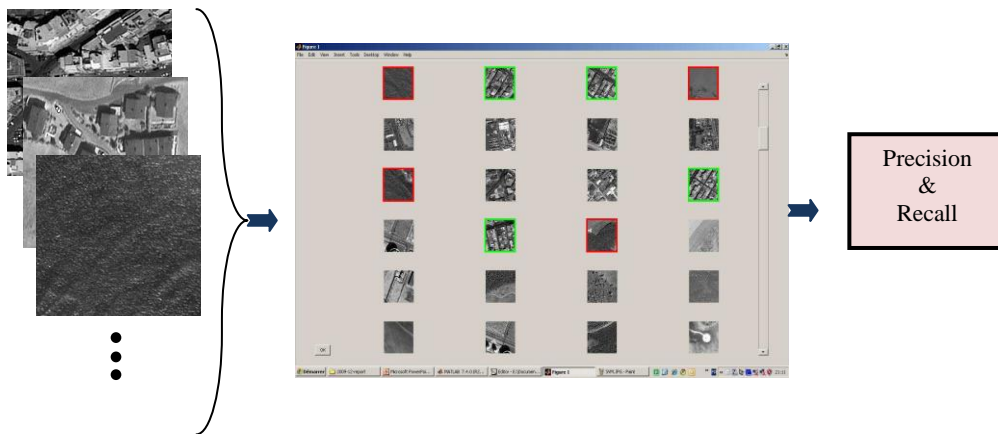
**Figure 1.10:** Detecting classes through a supervised classification based on SVM

For 20 detected classes we calculate the precision and recall for different descriptors. Finally, we are able to compare efficiency of different descriptors regarding to their calculated precisions and recalls.

# CHAPTER 2

# FEATURE EXTRACTION METHODS

According to the previous chapter, we aim to define some reliable descriptors for satellite images. A descriptor is usually a vector of features and every feature is supposed to describe one characteristic of an image or a pattern inside the image. In other words, a descriptor could be considered as a signature of an image. In the literature, there are many methods to extract features from images. In this chapter we study some of related textural and local descriptors and explain the required properties of a VHR satellite image descriptor which is supposed to characterize the geometrical structures.
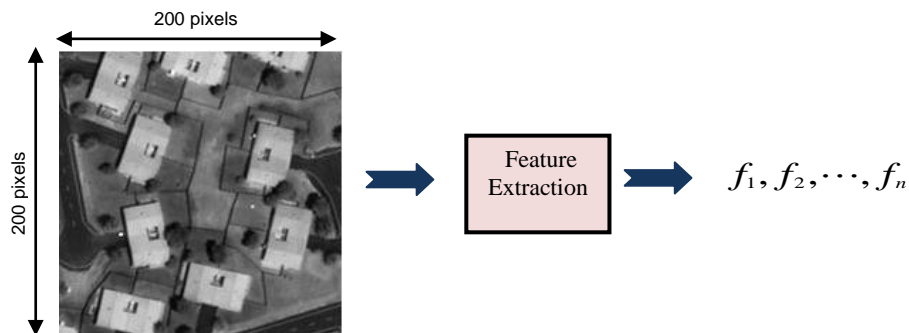


**Figure 2.1**: Feature extraction transforms the data with a large dimensionality into a feature vector with a reduced dimensionality. In this example, the initial space of data representation is a matrix of 200*200 pixels which is transformed to a shorter vector of features

The goal of *Feature extraction* is to transform our data with an initial dimensionality into another space with a reduced dimensionality in order to simplify its process and analysis. The new representation of data is usually called a feature vector. In fact, the initial data is so large to be processed and also is usually redundant. Here, redundancy means that we will not lose important information if we carefully reduce the dimensionality of data. For different tasks, we may extract different features from a set

of data. Regarding to a desired task, we have to choose some features such that the feature vector perfectly works with the specific task. When the initial data are images, the features are used to represent an image instead of using the original pixel values. In many cases, as in this thesis, image features have strong links with the image semantic properties.

Many feature extraction methods are presented by researcher for different types of images. We can cluster different features from many points of view. When we are working with satellite images, a simple way is to cluster the feature types into 3 main groups: image intensity features, texture features and local features.

## 2.1 Image intensity features

Image intensity features are the basic features which are widely used to describe any types of images. They give a very general description of an image and are computed based on statistical moments such as central moments of the first (mean value) and second order (standard deviation). Here we suppose that image $I$ is a square matrix which has $N_r$ pixels of rows from top to bottom and $N_c$ of columns from left to right: $I = \left\{ i_{rc} : r = 1,...N_r, c = 1,...,N_c \right\}$.

Where $i_{rc} \in \left\{ 0,...,L \right\}$ is the gray level of a pixel which is placed in row $r$ and column $c$ and $L$ is the number of gray levels. A *mean value* of gray level intensity of image $I$ is:

$$f_1 = \frac{1}{N_r N_c} \sum_{r=1}^{N_r} \sum_{c=1}^{N_c} i_{rc} \qquad (2.1)$$

This feature helps us to have a common sense about the brightness or darkness of a satellite image patch.

The other feature, *Standard deviation* of intensity level, is obtained as:

$$f_2 = \sqrt{\frac{1}{N_r N_c} \left( \sum_{r=1}^{N_r} \sum_{c=1}^{N_c} i_{rc} - f_1 \right)^2} \qquad (2.2)$$

This feature is generally known as an indicator for the amount of gray levels variation in an image patch. The standard variation of a satellite image patch is low if the image seems to be homogenous. Contrary, it is high for images whose gray level values of pixels obviously differ from one pixel to another one. For example if we have two satellite image patches, one from an urban area and other from a sea, which are taken with the same sensor and same resolution, we logically expect that the urban area has a higher standard variation.

Some higher order statistical features may be extracted from the image. For example the *Skewness* is obtained as:

$$f_3 = \frac{1}{N_r N_c f_2^{\,3}} \left( \sum_{r=1}^{N_r} \sum_{c=1}^{N_c} i_{rc} - f_1 \right)^3 \qquad (2.3)$$

And the *Kurtosis* is defined as:

$$f_4 = \frac{1}{N_r N_c f_2^{\,4}} \left( \sum_{r=1}^{N_r} \sum_{c=1}^{N_c} i_{rc} - f_1 \right)^4 \qquad (2.4)$$

If we consider each image pixel as an observed sample of a random variable, then the image patch could be the set of our variable's observations. So we can define another useful feature, the *entropy,* as:

$$f_5 = -\sum_{l=0}^{L-1} \frac{N_l}{N_r N_c} \log(\frac{N_l}{N_r N_c}) \qquad (2.5)$$

Where $N_l$ is the number of pixels with the gray level $l\,(0 \leq l \leq L-1)$. Entropy is a measurement of the unpredictability of a random variable. The entropy for an image shows how much its pixels gray levels are distributed in a random manner.

## 2.2 Texture features

In the field of image processing there is no a clear-cut definition for texture. Available texture definitions are based on texture analysis methods and the features which are extracted from the image. However, texture can be considered as the repeated patterns of pixels over a spatial domain. But the textures usually appear to be random and unstructured because in their model it is supposed that an amount of noise is added to the patterns and also the repetition frequencies changes from an area to another one.

Regularity, directionality, smoothness and coarseness are different examples of texture properties which are perceived by the human eye. Texture analysis has been extensively used to characterize and classify the remotely sensed images. In this sub chapter we study Haralick and Gabor-wavelet features, as the typical texture features to investigate the properties of textural analysis.

### 2.2.1 Haralick features

Haralick features [44] are one of the well known features for describing the textures presented by an image. These features are calculated based on the second-order histogram of a matrix which contains the joint probability distribution of pairs of pixels in the image. This matrix is called as *co-occurrence* matrix and could be

computed for several directions which are indicated by their angles with respect to the horizontal axis ($\theta$). If, for example, $\theta = 0$, the matrix computes the number of occurrences for the pairs of pixels that are separated by $\rho$ pixels in horizontal direction, that is, the pixels with coordinates $(m,n)$ and $(m, n \pm \rho)$, that have the specific gray levels:

$$\mathbf{P}_{\rho, \theta=0}(l_a, l_b) = \sum_m \sum_n (i_{mn} = l_a, i_{m,n\pm\rho} = l_b) \tag{2.6}$$

In which, $l_a, l_b$ are two arbitrary gray levels ($0 \le l_a, l_b \le L-1$) and $\mathbf{P}_\rho(l_a, l_b)$ is the co-occurrence matrix for the arguments $l_a, l_b$ which shows the number of occurrences $i_{mn} = l_a$ and $i_{m,n \pm \rho} = l_b$. Usually, $\rho$ can take some units and $\theta$ takes four angles 0°, 45°, 90° and 135°. For example, if we are going to obtain the co-occurrence matrix in vertical direction, i.e. $\theta = 90°$, then co-occurrence matrix is defined as:

$$\mathbf{P}_{\rho, \theta=90°}(l_a, l_b) = \sum_m \sum_n (i_{mn} = l_a, i_{m\pm\rho, n} = l_b) \tag{2.7}$$

The co-occurrence for other directions is obtained to have more robustness with respect to the rotation of image. If $P_{ij}$ is an element of this matrix then the corresponding element in normalized co-occurrence matrix, $p_{ij}$, is defined as

$$p_{ij} = P_{ij} / \sum_{i,j} P_{ij} \tag{2.8}$$

Several Haralick features could be extracted from the normalized co-occurrence matrix. For example, the *Angular second moment* could be obtained as:

$$f_6 = \sum_{i,j} p_{ij}^2 \tag{2.9}$$

And the *Contrast* is calculated as:

$$f_7 = \sum_{n=0}^{L-1} n^2 \left( \sum_{|i-j|=n} p_{ij} \right) \tag{2.10}$$

## 2.2.3 Gabor wavelet features

Gabor wavelet [46] filters represent models of visual perception of a texture and are widely studied and applied for texture modeling and classification. Moreover, there is another motivation for us behind the Gabor-wavelet filters. As we will see, some of them are visually similar to some of ICA basis vectors.

Gabor filters are usually considered as two dimensional functions of $x$ and $y$ and can be achieved from a general relation as:

$$G(x, y) = \exp(-\frac{x'^2 + y'^2}{2\sigma^2})\cos(2\pi\frac{x'}{\lambda}) \qquad (2.11)$$

In which:

$$x' = x\cos\theta + y\sin\theta$$
$$y' = -x\sin\theta + y\cos\theta \qquad (2.12)$$

In equation 2.11, $\lambda$ represents the wavelength of the cosine factor, $\theta$ represents the orientation of the Gabor function and $\sigma$ is the sigma of the Gaussian envelope. Usually, in a set of Gabor filters, $\sigma$ is constant but $\theta$ and $\lambda$ take some scaled levels. Number of Gabor filters is a function of the number of scales which are considered for $\theta$ and $\lambda$.

Another point in Gabor-wavelet filters is to choose the size of each filter. This is strongly depended on the application. Generally, smaller filters could detect the detailed characteristics and bigger filters give more general features in an image. Moreover, using smaller filters makes the feature extraction faster. Figure 2.2, as an example, shows a set of 16*16 Gabor filters.
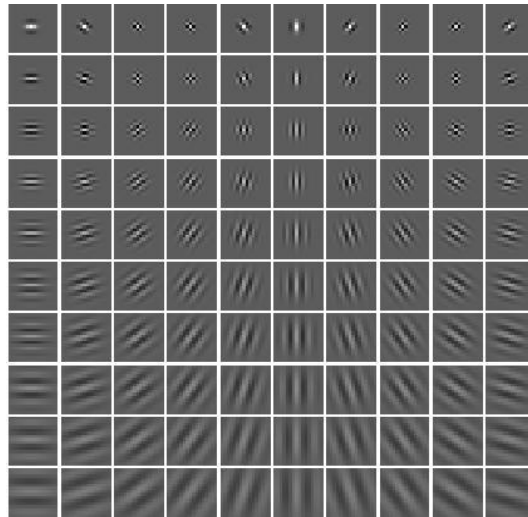


**Figure 2.2**: A set of 16*16 Gabor filters. We used 10 scales for frequency (wavelength) and 10 scales for the orientation (angle).

We took $\sigma$ as 5 and change the $\theta$ in 10 scales and also the $\lambda$ in 10 scales in order to produce a set of 100 filters. $\theta$ varies from 0 to 180 by a step of 18 degrees. $\lambda$ is obtained from an exponential equation as $1.2^n$, in which $n$ varies from 1 to 10. So, $\lambda$ varies from

1.2 to 6.2. Figure 2.2 shows the obtained set of 100 Gabor filers in which $\theta$ increases from left to right and $\lambda$ increases from up to down.

Extracting features from an image using a set of Gabor filters is similar to the feature extraction using a set of ICA basis vectors which is explained in chapter 7. Briefly, we gather a sufficient number of samples from initial image and decompose them into the set of filters. Using a specific function (for example root square average) upon all coefficients (obtained from decomposition of all samples) corresponding to one filter we obtain the feature which is related to that filter.

## 2.3 Local Features

The local approaches are mostly used for the purpose of object detection in image processing. They try to detect some features which are able to characterize existing objects in the image. According to this goal, local descriptors are usually robust to the rotation, scale and illumination. But they contain some complexities comparing with textural methods.

Scale Invariant Feature Transform (SIFT) [47] and Speeded Up Robust Features (SURF) [48] are two well known local approaches which are used by researches to model the images. Here, we are going to study the principles of local features in order to compare it with our requirements for satellite image descriptors. In the following, we study the SIFT model to show the principles and the complexities of local descriptors.

### 2.3.1 Scale Invariant Feature Transform

SIFT features are supposed to be *scale* invariant. Here, *scale* means the level of clearance of image. In other words, we want some features that are able to detect the objects regardless to their details. These details appear differently in different levels of clearance. So we have to provide some scale invariant features. The first step is to provide a scale-space of the image. Given an image, $I(x, y)$, we initially provide its corresponding *octaves*. An *octave* is obtained by dividing the dimensions of images by 1, 2, 4, etc. Then for each octave we obtain some blurred versions of image using a convolution between the image and Gaussian filters:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \qquad (2.13)$$

Where

$$G(x, y, \sigma) = 1/(2\pi\sigma^2)e^{-(x^2+y^2)/\sigma^2} \qquad (2.14)$$

If we use the Gaussian filters with different *scale*s of $\sigma$, we will have some blurred images which make the *scale space* of original image. An example is shown in Figure 2.3.
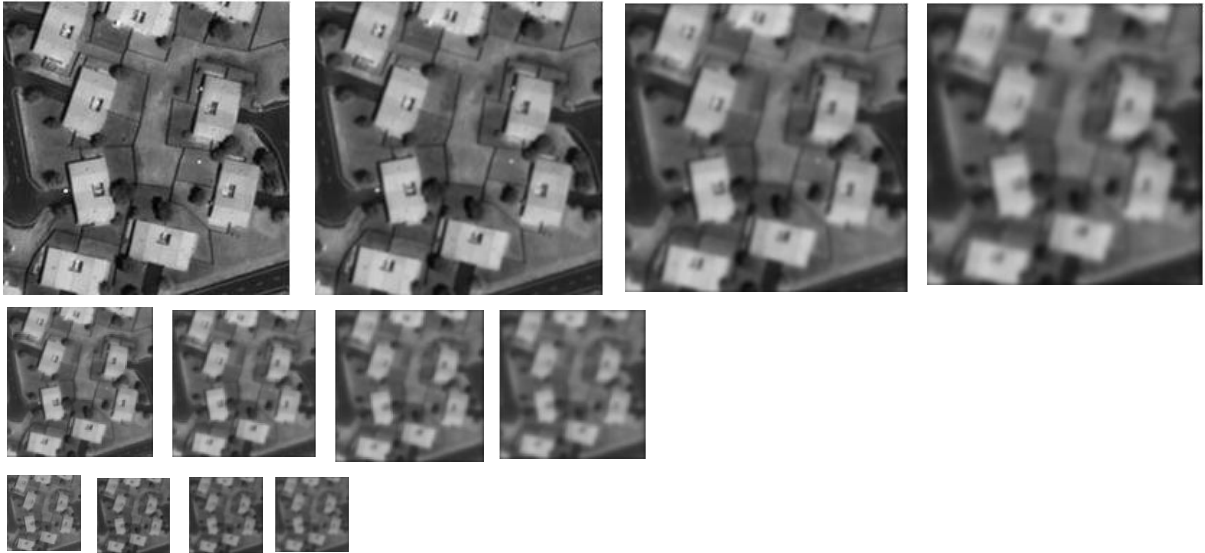
**Figure 2.3**: Obtaining scale space of original image using the Gaussian filters with different scales for 3 octaves.

Then we apply the *Difference of Gaussian* (*DoG*) operator for different scales of blurred image in order to obtain some new images, usually called *DoG* images.



**Figure 2.4**: Applying the *DoG* operator on the blurred images of scale space

*DoG* is defined as an operator which calculates the difference of two consecutive scales of blurred image :

$$DoG = G(x, y, k\sigma) - G(x, y, \sigma) \tag{2.15}$$

The results of applying this operator is shown in figure 2.4. In fact, we only use the differences of blurred images in order to remove the details of objects in the image so that the resulted features would be invariant with respect to different scales of blurring. *keypoints* in the SIFT framework are known as local maxima or minima of the images which are obtained after applying the *DoG* operators on the scales of blurred images. The pixels in such images will be compared with their 8 neighbours at the same scale, in addition with the 9 corresponding neighbours in two consequence scales. If the pixel is a local maximum or minimum, it is selected as a candidate *keypoint*. If some of these candidates are placed on one of the edges, or don't have enough contrast, they are not useful and will be removed.

We need some features which are invariant with respect to the details of objects (scale invariant) and also to the rotation. Detected keypoints in different scales of blurred image guaranties the scale invariance of our features. The next step is to determine an orientation for each keypoint to provide the rotation invariance. The idea is to obtain the most significant orientation(s) around each keypoint. We calculate the gradient angles and gradient magnitudes in a neighbourhood of each keypoint. Then, a histogram is created for the orientations according to their magnitudes. Based on this histogram we choose the most orientation(s) around the keypoint.
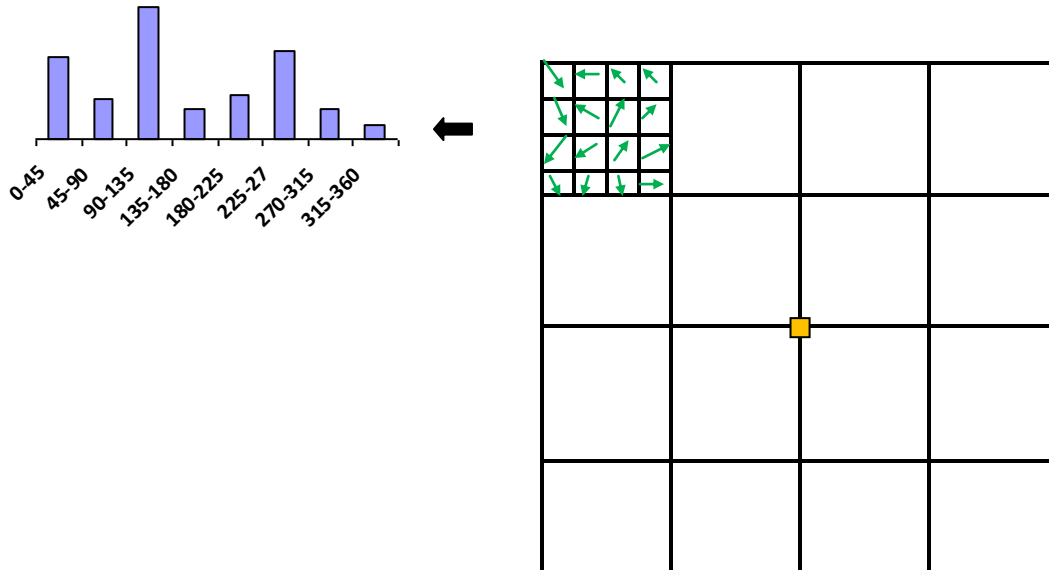


**Figure 2.5**: Generating the keypoint descriptors. A window of 16*16 pixels is considered around the keypoint and is divided to sixteen 4*4 region. In each of 16 regions, the gradient orientations are placed in a histogram of 8 bins to provide a vector of 128 elements. Then the keypoint's orientation will be subtracted from each 128 number to achieve rotation independence.

Next step is to define a descriptor for each keypoint. To do this, a 16*16 window around the keypoint is considered. This 16*16 window is divided into sixteen 4*4 windows. In every 4*4 window, we calculate the gradient magnitudes and gradient angles (orientations) for each of sixteen pixels. Then, we again create a histogram of gradient angles containing 8 bins of 45 degrees and for each gradient angel we add a number to the corresponding bin. This number depends on the magnitude of the gradient and also depends on the distance of pixel from the keypoint. If we do the same procedure for all sixteen 4*4 regions we will have a set of 16*8 = 128 numbers corresponding to the gradient orientations for each keypoint. This set of 128 numbers will be normalized at the end of this procedure.

Then, to obtain the rotation independence, the keypoint's orientation(s) is removed from each of these 128 gradient orientations. Thus, each gradient orientation will be relative to the keypoint's orientation. In addition, If we threshold the numbers that are big, we can achieve illumination independence. So, if for example one of the 128 values is greater than 0.15, it will be changed to 0.15. Finally, each keypoint is uniquely identified by its feature vector of 128 elements.

## 2.4 What kind of features do we need?

During this chapter we studied the principles of some texture and local descriptors. Each of texture or local features has its own properties, advantages or disadvantages. Table 2.1 present a comparison between the texture and local features. It also explains that what we expect from the features which are supposed to be defined.

**Table 2.1**: Comparison of texture and local features with the needed features

|  | **Texture Features** | **Local Features** | **Needed Feature** |
|---|---|---|---|
| **Objective of features in remote sensing** | Characterization of natural landscapes or scenes of man-made structures that are homogenous. | Geometrical objects or urban area detection, etc | Characterization of scenes specially those who contain geometrical structures |
| **General or detailed description** | Give a general description of the image | Give a description of geometrical details of the image | Give a general description of the image but is able to distinguish some of important geometrical characteristics of the image |
| **Length of feature vector** | Usually not long | Usually very long | Preferably, not long |
| **Complexity** | Usually simple | complex | As simple as possible |
| **Computation Time** | Usually fast | slow | As fast as possible |

We could say that our needed features are placed somewhere between textural and local features. Local features are mostly used for the goal of object detection which is different with respect to our objective in this thesis. We are not going to detect objects from the satellite images but we want to define features which are able to describe all scenes, especially those who contain geometrical objects. Although, the needed features seem to be closer to the textural features in the sense of presenting a general description of the image instead of presenting the geometrical details of the image, they are supposed to be able to distinguish some of important geometrical characteristics of the image.

In other words, we need some features which are more precise for geometrical characteristics in comparison with the textural features. In this sense, we move from the origin of textural features toward the local features. Simultaneously, we are supposed to keep the advantages of texture features like simplicity, short feature vector and being fast in computation.

In chapter 12 we compare all features which are defined in the thesis. Also we compare them with Gabor features as a typical texture features.

# CHAPTER 3

# SATELLITE IMAGES PROPERTIES

Our research domain is strongly related to the Remote Sensing. So, in this chapter we introduce the basic concepts related to the Earth Observation and Remote Sensing. In addition, we explain what sorts of satellite images are being processed during the thesis.

Remote Sensing is referred to the set of methods and techniques by which we are able to gather and process the data from the Earth surface for many applications. The data used for the remote sensing is obtained from the scenes which are sensed by instruments of measurements at remote distance, such as cameras or sensors which are installed at planes or satellites.

## 3.1 Active and passive sensors

Remote sensing sensors receive the electromagnetic waves that are reflected from the Earth's surface and convert them into a data form, usually an image, which can be processed by a machine or a specialist to obtain a variety of information from the Earth's surface.

There are active and passive remote sensing sensors. Active sensors send an electromagnetic pulse and process the reflected pulse. However, Passive sensors collect the electromagnetic waves emitted from the Sun and reflected by the Earth's surface.

Every sensor is supposed to work in a specific range of electromagnetic wavelengths. Particularly, optical sensors that belong to the passive group are sensitive principally to the electromagnetic wavelengths in the range of 0.4 μm to 0.75 μm . This is the same range of wavelengths to which the human eye is sensitive.

Thermal and Hyperspectral sensors are two other examples for passive systems. Thermal sensors measure the electromagnetic radiations reflecting from the Earth's surface in the thermal region. Hyperspectral sensors work with the electromagnetic radiations in the infrared, visible and ultraviolet regions. They collect electromagnetic waves using very narrow bands (e.g. 10 nm) as opposed to the wide bands (e.g. 250 nm) optical sensors.

RADAR (RAdio Detecting And Ranging) and LIDAR (LIght Detection And Ranging) are two examples for active sensors. These sensors send different types of pulses and process the reflected pulse also the time for each pulse to be received back at the sensor.

## 3.2 Optical satellite sensors

In this thesis we work with the optical satellite images, so we focus on this type of satellite images and study some of their important properties.

### 3.2.1 Resolution

Three types of resolution are defined in the field of remote sensing: *Spatial resolution*, *radiometric resolution* and *temporal resolution*.

*Spatial resolution* determines the actual dimensions of a pixel in an image on the Earth's surface. For example, for a satellite image with 1m resolution, each pixel represents an area of 1m*1m on the Earth's surface.

*Temporal resolution* refers to the time it takes for the satellite to return to the same orbit. The time it takes for the satellites to return to exactly the same orbit (usually called the revisit time) depends on the altitude of the satellite, which can vary from 400km to 800km above the Earth. In other words, this term is used to describe the time interval in which the same area on the Earth can be seen by the sensor using a different view angle.

*Radiometric resolution* refers to the range of numbers that can be stored in a single pixel which is described using *bits*. For example, an 8 bit image stores numbers between 0 and 255 ($2^8$) and an 11 bit image stores numbers between 0 and 2047 ($2^{11}$) for each pixel.

### 3.2.2 Panchromatic or Multispectral

Depending on the number of bands by which a sensor takes images from the Earth's surface, we have *Panchromatic* or *Multispectral* satellite images. The panchromatic sensors collect the electromagnetic radiations for the full range of the visible spectrum and give a grayscale image. Multispectral images are given by sensors working with several (normally four) bands. Each band covers a slice of the visible spectrum, normally the blue, green, red and infrared portions. Usually the features which are related to the spatial or shape characteristics of satellite images are extracted from the panchromatic images. Multispectral images are often used for objectives such as land

cover classification. This is possible due to the fact that objects on the Earth's surface react differently to various wavelengths of electromagnetic radiations.

## 3.3 Sub-meter optical satellite images

In this thesis we are going to extract the features from the optical satellite images. These features are related to the spatial characteristics of the images, so the color characteristics of the images are not important.
Thus, we only need *gray scale* images for our methods. In other words, the *panchromatic optical* satellite images are suitable for the purposes of our works. However, we can use multispectral or colored satellite images, but we transform them to gray scale images at the beginning.
Temporal resolution of satellite images is not our concern, because every received satellite image can be considered as the object of feature extraction, regardless to the previous or next received images.
Radiometric resolution is not a critical point for us. It is possible to work with both of 11 bit and 8 bit images. 11 bit images have more accuracy but take more memory and time to be processed. It is important especially when we have to apply some learning or training methods such as the ICA learning procedure which is used during the thesis. On the other side, the accuracy of 8 bit images is usually enough for us and our proposed methods work accurately with this kind of images. So we prefer to use the 8 bit satellite images to avoid more computational problems. Although, the proposed methods are able to work with the 11 bit images, we initially transform them to 256 grayscale level images to keep the same format of images during the thesis.
Spatial resolution is the most important characteristic of satellite images which are processed in this thesis. As it is mentioned before, the purpose of the thesis is to define some descriptors for satellite images containing various landscapes especially the geometrical or man-made structures. Details of this kind of structures are not visible in images which are taken with a spatial resolution over 1m. Therefore, during the thesis we consider only images with spatial resolution about one meter or under meter. For example, the QuickBird images with 60cm spatial resolution or Ikonos images with 1m spatial resolution are suitable for our purposes.
In figure 3.1 we show some samples of satellite images with the resolution of 60 cm which are used in the thesis containing a variety of natural and man-made landscapes. These images are transformed to 256 gray scale level images.

## 3.4 Contextual image patches for feature extraction

Satellite images which are shown in Figure 3.1 have sizes around 3000*3000 pixels and each of them contain a variety of man-made and natural landscapes. Consequently, defining features for such big images doesn't seem to be logical. For the purpose of feature extraction we need some smaller image patches which contain only one type of structure or landscape. Although we are able to extract features from images containing many classes of landscapes, this is not desirable because every vector of features is supposed to describe only one class of landscape.
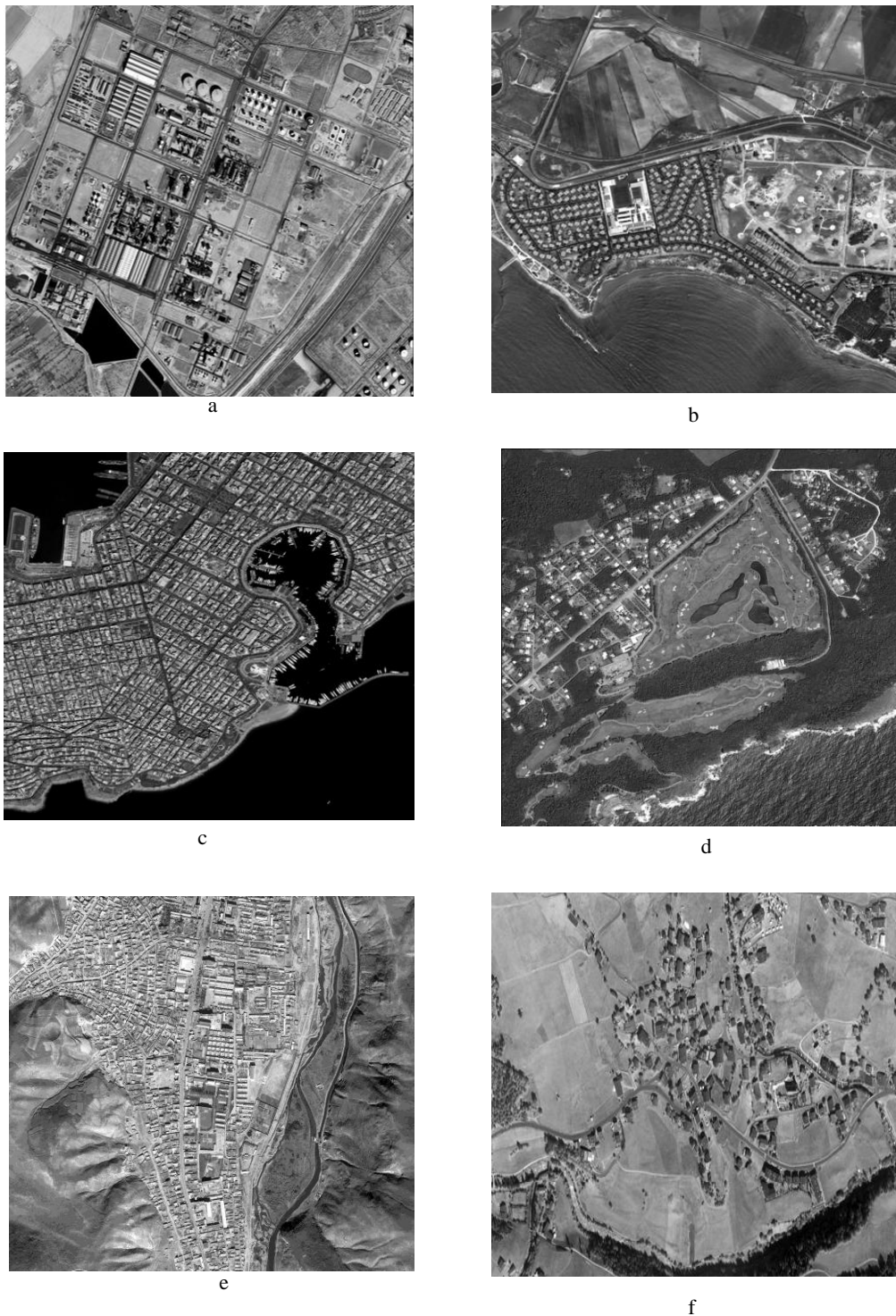
**Figure 3.1:** Samples of satellite images with the resolution of 60$^{cm.}$. Images have sizes around 3000*3000 pixels, so they cover a surface about 2$^{km}$*2$^{km}$. (a) Factory in Arak (Iran) , (b) Houses, farms, sea in Spain  (c) Port of Piraeus (Greece) , (d) Small town , Guam  , (e) City in china,  (f) Village in Austria. Images contain a variety of man-made and natural landscapes.

**Class1:** Factory

**Class2:** Farm

**Class3:** Village

**Class4:** City-1

**Class5:** Forest

**Class6:** City-2

**Class7:** Town

**Class8:** See
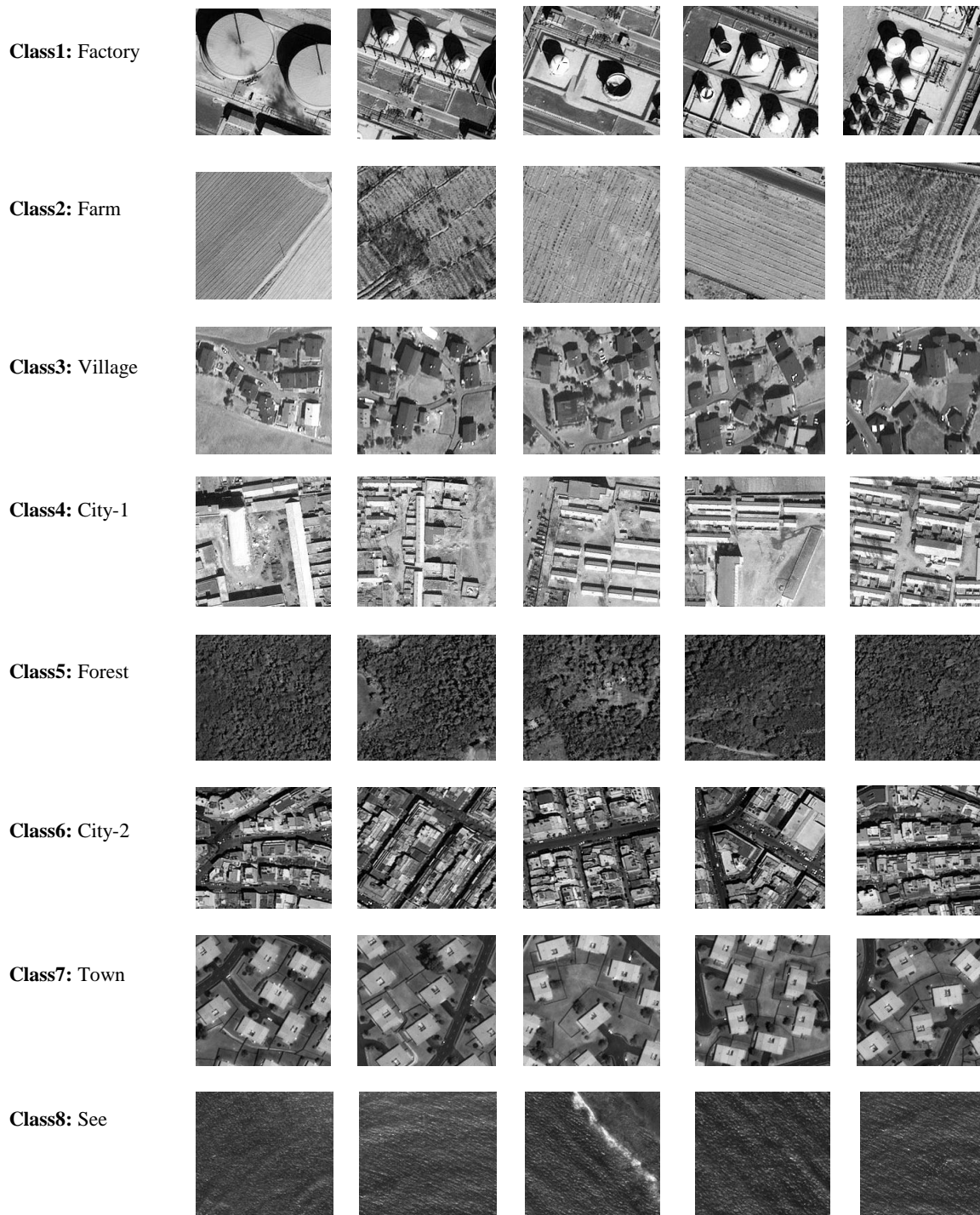
**Figure 3.2**: Test set, Samples of contextual image patches with the size of 200*200 pixels which are clustered in 8 classes of natural and man-made landscapes. For each class we prepared 100 samples.

Here, a class of landscape is the set of landscapes that present similar characteristics to user so that he or she can identify them as the same type of landscape. Defining the classes of landscapes and structures, itself, is a challenge because it is depended on the users' point of view. Everyone may determine a definition for a class of image which differ from other definitions. In this thesis, the class definition is not our concern because classification is not our main task; however we use classification to evaluate the capability of extracted features.

To illustrate what is called a class of landscapes in this thesis, we selected 8 classes of various landscapes from initial satellite images shown in Figure 3.1. For each class we prepared 100 image patches with the size of 200*200 pixels. We show 5 samples of every class in Figure 3.2. This set of 800 samples of 8 classes is used during the thesis to examine the proposed methods and we call it as *test set*.

The principal purpose of our work is to propose the patch descriptors which are capable for characterization of different landscapes specially the geometrical structures with regards to the context of the image patch. We have to consider our image patches to be enough large so that they contain a number of objects and a clear context. In other words, they must present meaningful scenery. For example, if an image patch contains only one building or a part of a building without any context, it may be ideal for the purpose of object detection, but it is not suitable for the goal of our work. In addition, if our image patch is too large then it may contain several parts that each of them could be individually considered as interpretable scenery.

*Initial satellite image*



*Contextual image patch*

*Micro image patch*

Feature extraction methods based on ICA
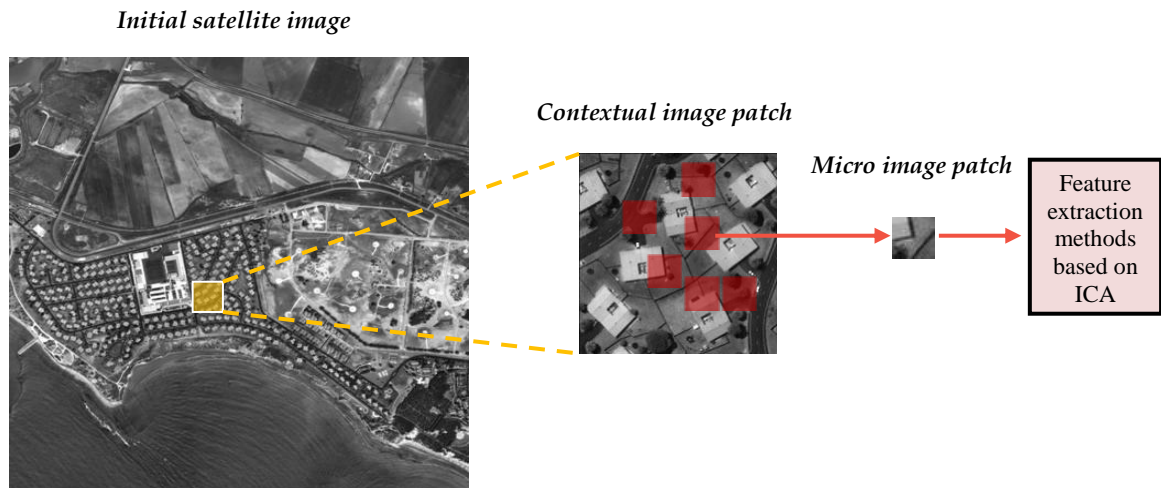
**Figure 3.3**: Three levels of images which are used in the thesis: *Initial satellite images* contain many classes of landscapes. *Contextual patches* usually contain one class of landscape and are suitable for feature extraction. *Micro patches* extracted from each contextual patch are used in the feature extraction procedures based on ICA because the contextual patches are too large to be used directly in ICA procedure.

During the thesis, we call the image patches for which we are going to define the descriptors as *contextual patches*. In other words, we are looking for image patches that

may present a number of geometrical shapes such as houses, buildings and other man-made structures with a clear context. We emphasise on the word *context* to separate our task from the object detection. Since we are working with the satellite images with the sub-meter resolution, it seems that a size of patches between 100*100 pixels to 300*300 pixels is reasonable. In this thesis we work with the contextual patch with the size of 200*200 pixels regarding to our considerations.

The 200*200 contextual patches are too large to be used directly for many feature extraction methods explained in the thesis. So we have to gather a number of smaller patches, called *micro patches*, from each contextual patch to be processed in feature extraction procedure. Details of this issue are explained in chapter 6. Here, we just mention that we have three levels of images in the thesis: *Initial satellite images* that are big images and contain a lot of landscapes. These images are not suitable for feature extraction. *Contextual patches* with the size of 200*200 which are extracted from the initial satellite images are those for whom the descriptors are defined. *Micro patches* are small image patches which are extracted from each contextual patch and are used in the procedure of extracting features from the contextual patch. These three levels of images are illustrated in Figure 3.3.

# CHAPTER 4

# STATE OF THE ART

In this chapter, we will review previous studies related to our work to illuminate the atmosphere of researches around the main aspects of the work. First, we have a look at the related works around urban area detection and classification, because one of the goals of the thesis is to extract features and to define descriptors which are able to characterize geometrical structures in satellite images.

On the other hand, methods presented during the thesis for extracting features from satellite images are strongly related to Independent Component Analysis (ICA). So, we will also review the history of Independent Component Analysis and its applications on image processing.

## 4.1 Urban area characterization state of the art

Nowadays a lot of satellite images are being received every day from several satellite sensors and with different resolutions. These images, especially Very High Resolution (VHR) satellite imagery (such as Ikonosand and Quickbird, etc) provide valuable information for researchers for different applications.

However, during last years the volume of these remotely sensed data has been rapidly grown and makes it impossible for a researcher or an expert to extract manually the valuable information from these remotely sensed data. So we have to apply some automated techniques and methods to extract this information. Unfortunately, developing such automated techniques is not straightforward since classical image processing and pattern recognition methods are not sufficient for this purpose. We need some new and more complicated techniques which benefits of many sources of data and a combination of feature extraction methods and classification approaches to extract important information from high resolution satellite imagery.

Among different landscapes in the satellite imagery, urban areas have attracted especial attention of researchers from different fields of science and engineering. This is because urban zones are rich of information and important from different points of view and also because of the complexity and difficulty of the modeling and characterization methods which are used for them.

On the other side, obvious improvement in sensors' resolution caused the content of satellite imagery to present the details of objects that makes it more difficult to characterize different landscapes, especially urban areas, by traditional methods. In fact, in the satellite images with low level of resolution, different scenes, even from the urban areas, could be seen as a kind of texture and could be modeled using textural approach. The images in Figure 4.1 show the differences between the properties of an urban area at 10m resolution and details of the same area at 1m resolution. We can see that at 10m resolution the image can be modeled as some kinds of textures but when the resolution is improved to 1m, objects (buildings, roads, trees, etc.) clearly appear in the image and make it difficult to consider the scene as a texture.
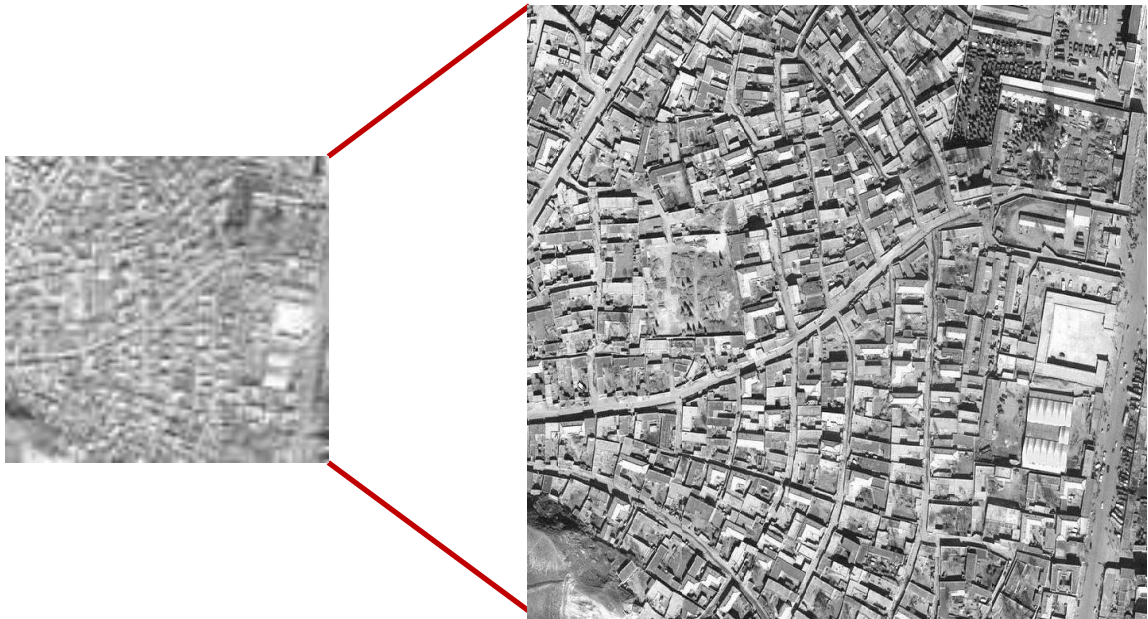


**Figure 4.1**: An example of urban area which may be considered as a texture model at 10m resolution and appears as a set of objects and geometrical shapes at 1m resolution.

Even if a scene in such high resolution images could be estimated by a texture, the majority of landscapes and objects would be much more complex, and difficult to be analyzed with texture based methods because of the non-homogeneity. Actually, it is supposed to apply some more complicated methods which are able to take into account such details of objects and structures inside urban area scenes taken from high resolution satellite images. Recently, a wide variety of methods and techniques are used by researchers for characterization and indexing of urban areas in high resolution satellite images.

The schema in Figure 4.2 illustrates a general procedure of what is usually done for urban area characterization.
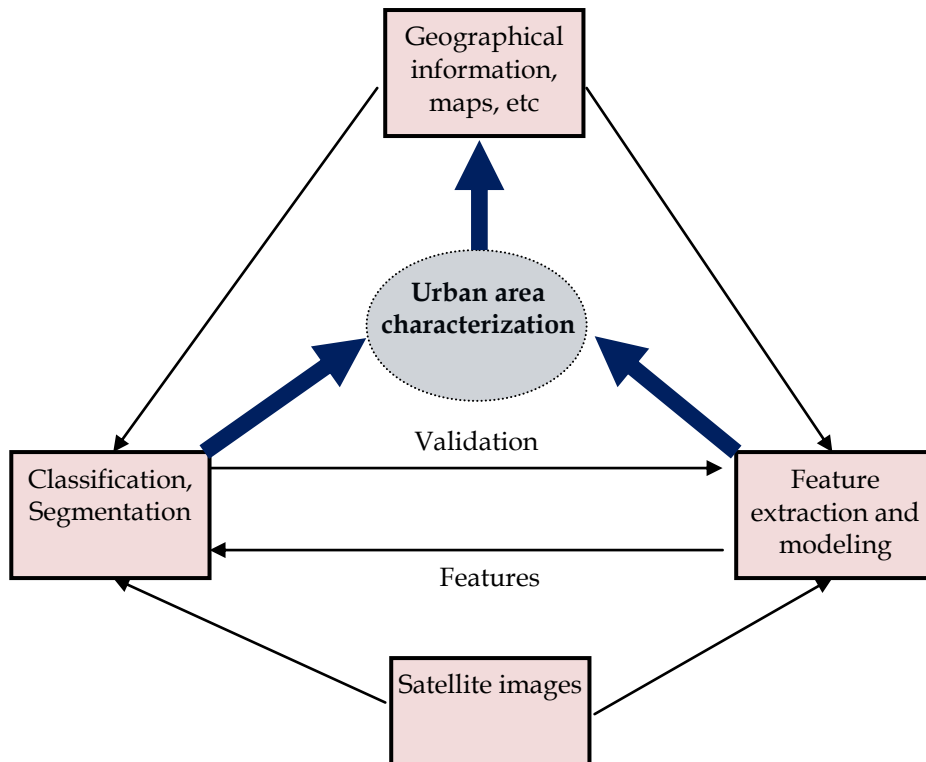


**Figure 4.2**: A general illustration of procedure which is performed for urban area characterization.

Characterization of satellite imagery, including urban areas, is usually done based on two main levels of activities. On one hand, a variety of researches have been done to extract features from the scenes and to model them using several approaches. Different types of features and descriptors may be extracted from satellite images. However, two types of features can be seen more or less in the literature: the texture-like features and the local features. As it was mentioned, textural approaches are not anymore efficient to perfectly characterize VHR satellite images who present the details of objects. Particularly, the urban areas contain some geometrical structures that could be hardly modeled by texture-like features. Although local descriptors are capable for such a purpose but they have some disadvantage such as complexity and being time consuming. Extracted features and descriptors may be used by different approaches like bag of words, graph theory,... to provide a model giving a semantic interpretation of the scene.

On the other hand, the extracted features can be used for classification and segmentation of satellite images. The goal of such activities is to distinguish different classes of landscapes and structures on the Earth's surface. Results of classification is not only used for urban area characterization but also could be considered as a

feedback for feature extraction which shows the efficiency of the feature extraction method. Researchers usually benefit two important sources of data and information for urban area characterization. The main source is satellite images but some methods use also the geographical information like maps as the extra source of data and information which themselves benefit of the results of urban area characterization.

Here we review some of previous studied which have been done for characterization of satellite images specially the urban areas.

Scale-Invariant Feature Transform (SIFT), explained in chapter 2, provides a set of local features which is widely used for urban area detection. Sırmac and Unsalan, [12] proposed a method based on SIFT key-points for urban area and building detection on very high resolution satellite images. Their approach also benefits multiple sub graph matching, and graph cut methods. They picked two template building images, one representing dark buildings and the other representing bright buildings and obtained their SIFT keypoints. Also, SIFT keypoints are obtained for the test image. Then, by applying multiple sub graph matching between template and test image SIFT keypoints, urban area are detected in the test image. Finally, from the detected urban area, separate buildings are detected using a graph cut method. They obtained a 89.62% correct urban area detection performance with a 8.03% false alarm rate and 88.4% correct building detection performance with a 14.4% false alarm rate. But their building detection method may not detect buildings if the contrast between their rooftop and the background is low.

Among texture-like methods, Gabor features are known as effective descriptors for urban areas. Yang and Newsam [11] compared two classification methods and two feature extraction methods for labeling and indexing different classes and structures inside the satellite images such as residence area, forest, roads, etc. As the feature extraction, they applied Gabor texture features and SIFT descriptors and as the classification method, they used Super Vector Machine (SVM) and Maximum A Posteriori (MAP) classifier to evaluate the results of feature extractions. Their work is interesting because they compared a texture-like descriptor with a local descriptor. They presented their results as the following table:

**Table 4.1:** Results of Yang and Newsam. Comparison of SIFT and Gabor, using MAP and SVM

|  | SIFT | Gabor |
|---|---|---|
| **MAP** | 84.5% | 73.9% |
| **SVM** | 76.9% | 89.9% |

Principal Components Analysis (PCA), sometimes called Karhunen-Loève Transform (KLT), is another method which has been used by researchers to characterize the satellite images. KLT or PCA, in some pattern recognition methods, is used to reduce the images dimensionality without losing the important information. In a KLT procedure, eigenvectors of covariance matrix is calculated as new orthogonal basis

vectors for describing data. Quintiliano and Santa-Roza [9] proposed a target detection approach, based on KLT, in order to detect streets, using very high resolution multispectral or hyperspectral satellite images. They transformed the data from multispectral images to some large vectors and computed eigenvectors of their covariance matrix as new basis vectors. They applied positive and negative training, for detecting asphalt street (what is asphalt street and what is not asphalt street), and reported very better results when they used only positive training.

 Since different types of features may be used in urban area characterization, in many cases, feature selection methods are applied to find the optimum features. Tuia et al [17] used a feature selection method based on SVM called Recursive Features Elimination (RFE) to search among the morphological features to find an optimal set of features based on the importance of the features in the classifier. An important challenge in their work is that the input space could become rapidly untreatable since it is possible to extract many morphological features by simply using different filters or by changing size and shape of the structuring elements.

 Nowadays, some text retrieval approaches are used by researchers to model different types of images including remotely sensed images. *Bag of words* approach tries to model each image patch as a document containing some visual words. Then some topics are extracted using these words and a semantic interpretation of the documents is presented. In this model, a document is viewed as an un-ordered set of words and is statistically modeled as a frequency of occurrence histogram along the dictionary. Weizman and Goldberger [14] proposed an approach which is based on the idea of bag of words for modeling of satellite images. Here we explain some aspects of their works because we also use the bag of words model in chapter 9. They needed a visual analogy of word and a visual analogy of dictionary that contains a list of all possible words. They used image patches with the size of 10*10 pixels and applied PCA transformation to the patches to reduce the dimensionality from 100 to 7 and used a clustering algorithm to the data vectors to group the patches into clusters. The mean of every cluster was defined as a dictionary's visual word. Next step was to provide a histogram of words repetitions for urban and non-urban areas. They extracted all the patches from the manually labeled training images and normalized the patches, applied PCA and found the nearest word from the dictionary. Then, they built a histogram which displays the frequency of every visual word in urban areas. A similar histogram was built for the non-urban areas. Then the words that best differentiate urban areas from the non urban areas were found. As a result, a set of "urban detection words" was defined. For a new unlabeled test image, they selected the patches that correspond to urban detection words as a first detection step for urban areas. Then, as a post-processing step they applied spatial consistency constraints on the detected urban patches to obtain a global decision on urban regions.

 Methods based on graph theory are also important for urban area characterization. Dogrusoz and Aksoy [10] introduced a graph-theoretic method for analyzing land development in high-resolution satellite imagery in terms of spatial arrangements of buildings. Buildings are detected using spectral classification and morphological post-processing. These buildings form the nodes of a graph where the edges are constructed using the Voronoi tessellation of the scene. Building groups are formed by thresholding the minimum spanning tree of this graph. These groups are classified as organized or unorganized by examining the distributions of the angles between neighboring nodes of the clusters. Their experiments for detecting the urban area show

an accuracy of about 80%. They proposed to incorporate new properties of building groups into the graph to improve the clustering stage.

As an example of using the geographical information and maps in urban area characterization, we can mention the work of Newsam and Yi Yang [13]. They proposed a method which uses gazetteers and remote-sensed imagery, simultaneously, to improve the level of characterization.

Beside the urban area detection and characterization, Building extraction has been one of the interesting zones for researchers in the field of satellite image processing. Mayunga et al. [31] proposed a semi-automatic method based on the edge characteristics of estimated polygons to extract buildings from high resolution panchromatic imagery. Their method benefits of a snake algorithm to approximate buildings with some polygons. They reported a result of 91% for their experiments of building extraction from a variety of tested satellite images. An improved version of snake algorithm is also used by Peng et al. [32] to detect buildings. However they used their method for colored satellite images instead of panchromatic images. They also reported good detection results with their method.

There are many other interesting and valuable works and studies which have done by researchers in the field of urban area detection and characterizations. Some of them are mentioned in the bibliography. See for example references from [18] to [30].


## 4.2 ICA State of the art

Visual data that can be obtained from all around us has certain characteristics like other measured signals. They contain for example different textures, specific bands of frequencies and also edges, lines which are related to objects and their properties. An efficient processing system which is supposed to operate in such environment should utilize these characteristics.

The idea of Independent Component Analysis (ICA) is to reduce the redundancy of data without losing the important characteristics of data. Barlow [1] proposed that human brain memorizes some information about all visible environments and use it when we are looking to an environment to decrease the redundancy of the data. Here, the redundancy has a meaning of statistical dependency. For example, if we see a car we expect to see also a street or a road. That is, there is a statistical correlation or dependency between the car and the street in our brain because we usually see them together. The initial idea of ICA is the same but here the dependency is measured between the gray levels of different pixels of an image which are considered as some random variables.

Fundamentals of ICA are explained in chapter 5. Blind Source Separation was one of the first problem for which the ICA was developed. In addition, it may be considered as a generalized form of Principle Component Analysis (PCA). The difference is that in PCA the components are supposed to be statistically uncorrelated, however, in ICA, the components are statistically independent. We know that statistical independence is a generalized form of being uncorrelated. That is, two independent random variables are certainly uncorrelated. But if two random variables are uncorrelated they may be not independent.

An important study was done by Bell and Senjowski [2] who used ICA for natural images and found out that the independent components of images contain short lines

and edges. Olshausen and Field [3] showed that similar properties can be found in human visual system. Existence of edges and lines in ICA components is interesting for us as well, because we are looking for some models to deal with the edge characteristics of objects in the satellite images.

Recently, several methods have been proposed which apply ICA for image data. Some of these methods use simple ICA models, but some of them use an ICA mixture model like the method proposed by Lee, Lewicki, and T. J. Sejnowsk [4].

An example of using ICA for remotely sensed data is the study which is done by Zhang, X. and C. H. Chen [7]. Also, Zhang et al [8] proposed a method based on ICA for classification of remotely sensing images.

Although ICA is frequently used for some types of images such as natural images, text images and face images, but it has not been widely used for satellite image characterization and there are many aspects to be investigated by researchers. This is another reason which motivates us to work on the ICA for satellite image characterization in this thesis.

# CHAPTER 5

# PRINCIPLES OF INDEPENDENT COMPONENT ANALYSIS

Independent Component Analysis is the principal framework upon which several methods are proposed during the thesis to extract features from high resolution satellite images. In this chapter we illustrate fundamentals, concepts and algorithms of Independent Component Analysis.

## 5.1 Why ICA for satellite images?

The initial question in this chapter could be around the motivation which leads us to use Independent Component Analysis for VHR satellite image characterization. Although reader could find the motivations behind the idea of using ICA for atellite image characterization during the thesis, for more illustrations, here we give some explanations.

We know that Bell and Senjowski [2] used ICA for natural images and found out that the most of independent basis vectors of images include short lines and edges. This is an important property for satellite image characterization especially for those who contain geometrical structures, because the geometrical objects normally contain lines and edges. We show in Figure 5.1 some of ICA basis vectors which are obtained for four of classes of contextual satellite image patches shown in Figure 3.2. Here, the size of ICA basis vectors is 32*32 pixels. Later, we explain about the size of ICA basis vectors.

It is clear that the ICA basis vectors which are obtained for one class differs from the other classes. For example the basis vectors which are obtained for class of City contain both of long and short lines, but the basis vectors of class of Village contain only the short lines in several directions. We also see some types of smooth edges inside the basis vectors which are obtained for class of Sea which are probably because of the existing waves in the scenery. But for class of Forest we hardly could find the lines or

edges in its basis vectors. They present some special structures which are completely different from other classes. However, the ICA basis vectors of classes Forest and Sea which can be called as natural landscapes are, in general, more homogenous comparing with the basis vectors of classes City and Village which are man-made landscapes and include the geometrical structures. ICA basis vectors of such classes usually contain obvious lines and edges.

**Class6:** City-2

**Class3:** Village

**Class8:** See

**Class5:** Forest
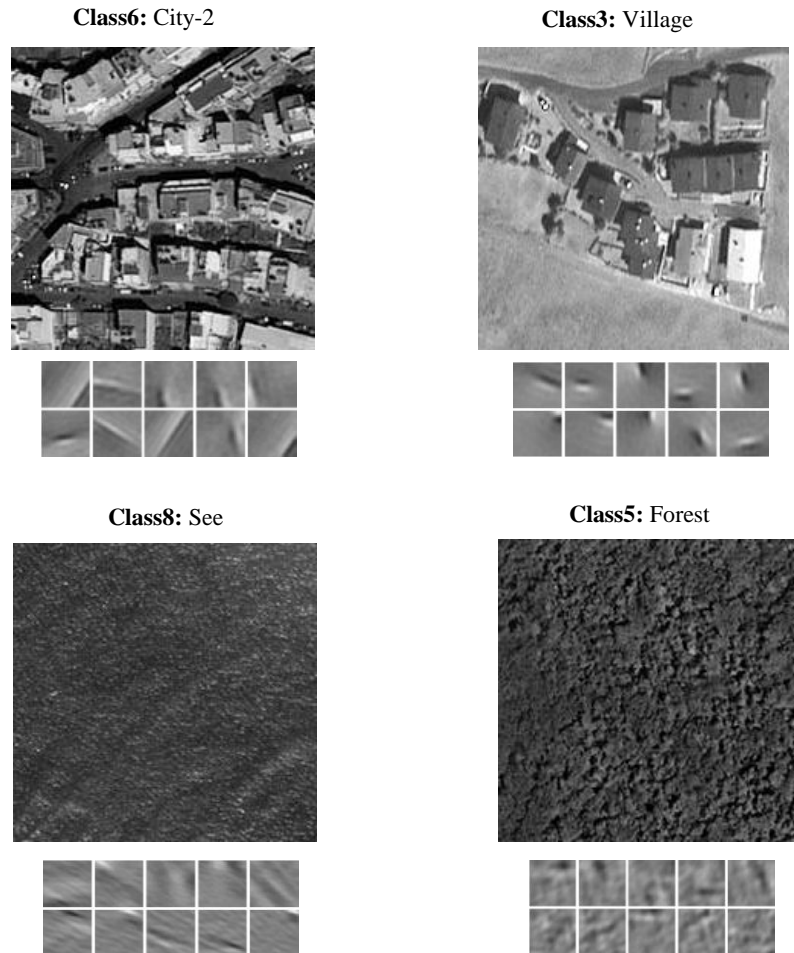


**Figure 5.1**: ICA basis vectors differ from one class to other class. For every class we show a sample of its contextual patches and also 10 samples of its obtained basis vectors. The size of contextual patches is 200*200 and the size of basis vectors is 32*32 pixels and their scales are proportional in this figure. The basis vectors of classes Forest and Sea, as natural landscapes, are more homogenous comparing with those of classes City and Village, as man-made landscapes. ICA basis vectors of classes City and Village (Man-made landscapes), contain obvious lines and edges. The difference among the ICA basis components which are obtained for different classes is a sign of ICA capability for satellite image characterization.

Actually, it seems that the ICA basis vectors carry the important characteristics of every class of images and the difference among the obtained basis vectors for different classes of satellite images can be interpreted as ICA capability for satellite image characterization. In this chapter we are going to present the theoretical fundamentals and concepts of Independent Component Analysis including some necessary assumptions and some pre-processing steps.

## 5.2 Fundamentals of Independent Component Analysis

Independent Component Analysis (ICA) is a method that initially developed to deal with problems close to source separation or cocktail-party problem. In such problems, we have some observed signals which are combined from a number of signal sources that are simultaneously generating its own signals. For example, in a cocktail party, where many people are talking simultaneously, we are going to separate the different speeches. In this example, each person in cocktail party is a source which is generating its own information independently. Human brain, naturally, is able to perform a kind of source separation. That is why we can follow somebody's speech among other speeches in a cocktail party.

In ICA, it is assumed that there are a set of $n$ sources of information ($S_1, S_2, \ldots, S_n$) which are statistically independent with respect to each other. That is, the value of each source does not have any effect on other sources values. From the statistical point of view we could consider these sources as the independent random variables. The set of these variables could be denoted with a random vector $S = [S_1, S_2, \ldots, S_n]^T$ which is called the *source vector*. Then we suppose that the original independent source components are combined via a linear process. In other words, we have a set of observed variables, ($X_1, X_2, \ldots, X_m$) which are themselves random variables because they are produced as linear combinations of the initial random variables:

$$
\begin{aligned}
X_1 &= a_{11}S_1 + a_{12}S_2 + \ldots + a_{1n}S_n \\
X_2 &= a_{21}S_1 + a_{22}S_2 + \ldots + a_{2n}S_n \\
&\vdots \\
X_m &= a_{m1}S_1 + a_{m2}S_2 + \ldots + a_{mn}S_n
\end{aligned}
\tag{5.1}
$$

We denote the set of observed random variables with a random vector $X_{obs} = [X_1, X_2, \ldots, X_m]^T$ and call this vector as *observed vector* or *observed signal*. As an example, when we apply ICA to images, each image patch is considered as our observed signal, $X_{obs}$, in which $X_1, X_2, \ldots, X_m$ are the pixels of image patch. More details about ICA for image data are explained in Chapter 6. Notice that the dimension of the observed signal, $X_{obs}$, (that here is $m$) is not necessarily equal to the dimension of the source vector, $S$, (that here is $n$).

Since the process is assumed to be linear, the relation between $x_{obs}$ and $s$ can be modeled as a matrix form:

$$x_{obs} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \cdots & a_{1n} \\ a_{21} & a_{22} \cdots & a_{2n} \\ \vdots & & \\ a_{m1} & a_{m2} \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} = \mathbf{A}s \tag{5.2}$$

Here, the matrix $\mathbf{A}$ is called as *mixing matrix*, since it mixes the independent sources. We can also rewrite equation (5.2) as:

$$x_{obs} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = s_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} + s_2 \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} + \cdots + s_n \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix} \tag{5.3}$$

$$= s_1 a_1 + s_2 a_2 + \cdots + s_n a_n$$

In which the vectors $a_1, a_2, \cdots, a_n$ are with the same dimension as observed signal (which here is $m$). These vectors can be considered as the basis vectors of a new space for representing our data. So, they are usually called the *basis functions* or the *basis vectors*. When we use ICA for image data, our observed signals could be considered as small image patches (micro patches) which are gathered from the initial images. So the ICA basis vectors would be with the same size as the micro patches.



**Figure 5.2**: when we use ICA for image data, observed signals are small image patches which are gathered from the initial images and the basis vectors obtained with the same size.

In fact, the initial space for representing data are some orthogonal basis vectors that have just one non-zero element. This is expressed with the equation (5.4):

$$x_{obs} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = x_1 \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} + \cdots + x_m \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \quad (5.4)$$

Through ICA we transfer our data into a new space whose basis vectors are $a_1, a_2, \cdots, a_n$. It is important to notice that the dimension of initial space (*m*) could be different from the dimension of ICA space that is the number of sources (*n*). This can be considered as dimension reduction step, a pre-processing step, which is explained in sub chapter 5.4.1.

*Independent Component Analysis* or ICA is defined as the procedure of basis vectors estimation such that the ICA sources will be as independent as possible. In other words, having a set of *d* observed signals, $x_{obs}(k)$, $k = 1, \cdots, d$, we are going to estimate mixing matrix, $\mathbf{A}$, which includes the basis vectors, $a_1, a_2, \cdots, a_n$ and the set of sources for each observed signal, $s_1(k), s_2(k), \cdots, s_n(k)$, $k = 1, \cdots, d$ such that these sources would be, statistically, as independent as possible.

The main goal of ICA is to estimate the mixing matrix. Once we estimate the mixing matrix, we are able to obtain the corresponding source vector for each observed signal according to the equation (5.2). Actually, since the columns of mixing matrix $\mathbf{A}$, are the *n* basis vectors which have *n* different directions, the rank of $\mathbf{A}$ is equal to *n.* Thus the matrix $\mathbf{A}$ would be left invertible. It means that we are able to find matrix $\mathbf{W}$ such that:

$$\mathbf{WA} = \mathbf{I_n} \quad (5.5)$$

So the source vector could be easily obtained by multiplying two sides of equation (5.2) into $\mathbf{W}$:

$$s = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} \cdots & w_{1m} \\ w_{21} & w_{22} \cdots & w_{2m} \\ \vdots & & \\ w_{n1} & w_{n2} \cdots & w_{wm} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = \mathbf{W}x_{obs} \quad (5.6)$$

Matrix $\mathbf{W}$ is called *separating matrix* since it helps us to separate the sources which are mixed through the observed signals.

## 5.3 Assumptions for the mean and variance of sources

When we are working with ICA, it would be very useful if we assume that the mean values of sources are zero:

$$E\left\{S_i\right\} = 0, \qquad i = 1, \cdots, n \tag{5.7}$$

This assumption makes the learning procedure easier to be performed. If the mean values of sources are equal to zero then according to equation (5.1) the mean value of every component of observed signal would be zero because they are the linear combinations of independent sources whose mean values are zero. So, at the beginning of learning procedure we remove the mean value from each element of observed signal.

We have to consider another assumption to solve the problem of estimation of ICA components. The reason is that we cannot determine the variances of the independent sources. In fact, according to equation (5.2) both $S$ and $\mathbf{A}$ are unknown and must be estimated. So, if one of the sources $S_i$ is multiplied by a scalar, and the corresponding column $a_i$ of $\mathbf{A}$ is divided by the same scalar the equation is still satisfied. Consequently, it is reasonable that we assume all sources have a constant variance which is usually considered equal to one:

$$\mathrm{var}\left\{S_i\right\} = 1, \qquad i = 1, \cdots, n \tag{5.8}$$

Then the matrix $\mathbf{A}$ will be obtained in the ICA learning procedure regarding to this assumption.

## 5.4 Pre-processing steps

The theoretical fundamentals of Independent Component Analysis are explained during last sub-chapters. Before an ICA learning algorithm, we usually apply some pre-processing steps on our data which make the learning procedure easer and cause better results. In this sub-chapter, we explain some general pre-processing techniques that usually are used for all types of data which must be processed by ICA. Then in next chapter, we introduce another pre-processing step which could be used exclusively for the image data.

## 5.4.1 Dimension reduction

When we apply ICA for multi-dimensional data, it is very useful if we eliminate the dimensions in which the variance of data is very low. It is illustrated in Figure 5.3 with an example. In this example our data is defined in three-dimensional space. However, the most important variations happen in a two-dimensional plate and the variation of data out of this plate is not significant. So, if we consider the variation of data just in two dimensions corresponding to this plate we will not loss important information but on the other side the data processing will be really simpler.
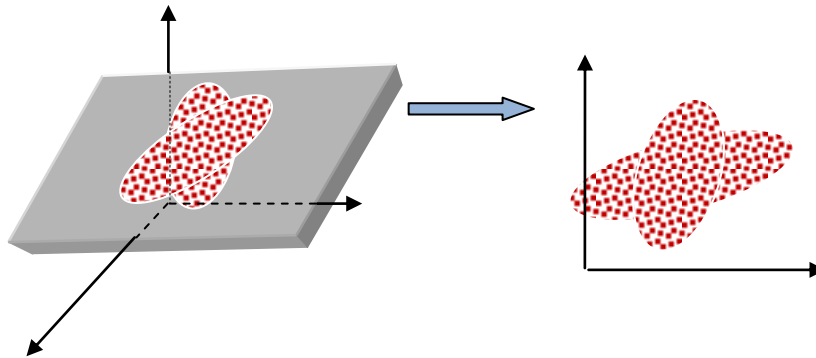


**Figure 5.3:** Example of dimension reduction. It is reasonable to reduce the data dimension from 3D to 2D, since the variance of data out of a two-dimensional plate is not important.

This step is usually done simultaneously when we are performing the *Principal Component Analysis* pre-processing step which is explained later in this chapter. In other words, this step usually is not performed independently.

## 5.4.2 Removing mean vector

Data which is processed by ICA are considered as the several samples of a random vector and in ICA we deal with the statistical properties of such data. So, every pre-processing step that reduces the statistical complication of data would be useful. As a pre-processing step to simplify the problem we can *center* our data. This means removing the mean vector of $x_{obs}$ ($\bar{x} = E\left\{x_{obs}\right\}$), in order to have a zero mean random vector. Figure 5.4 shows this pre-processing step. Consequently, the mean value of source vector ($S$) obtained through ICA learning procedure would be zero as well. This can be verified according to both sides of Equation. (5.2). Thus, the first assumption in sub-chapter 5.3 is satisfied. If it is needed, using equation (5.6), we are able to estimate the mean value of sources as well. When we obtain the separating matrix, $\mathbf{W}$, (and the mixing matrix, $\mathbf{A}$) through a learning procedure with centered data, we can estimate the mean vector of $S$ using the following equation:

$$\bar{s} = \mathbf{W}\bar{x} \qquad\qquad (5.9)$$

In which $\bar{s}$ is the mean of source vector: $\bar{s} = E\left\{s\right\}$
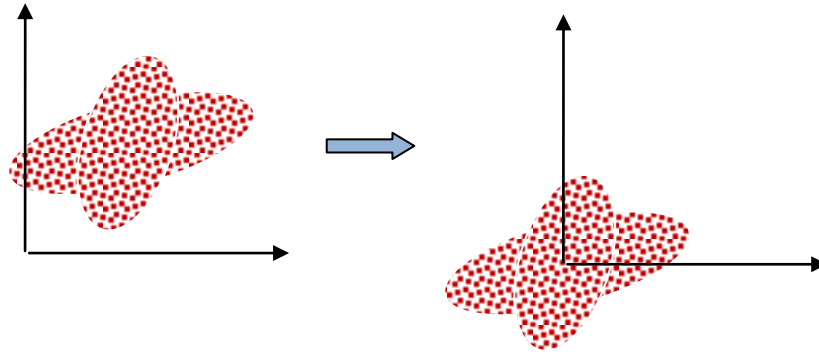


**Figure 5.4:** Centering as a preprocessing of data means that we subtract the mean vector of observed signal.

## 5.4.3 Principal Component Analysis step

In ICA we aim to transfer data into a new space in which the component would be mutually independent. If we initially transfer our data into a space in which the components are *uncorrelated*, it reduces many statistical complications. This is the goal of Principal Component Analysis (PCA) which could be considered as one of the basic ideas for Independent Component Analysis. In fact, the statistical independence is a generalized form of being uncorrelated. When two random variables are statistically independent, it implies that they are uncorrelated as well. But if these random variables are uncorrelated, we cannot conclude that they are necessarily independent.
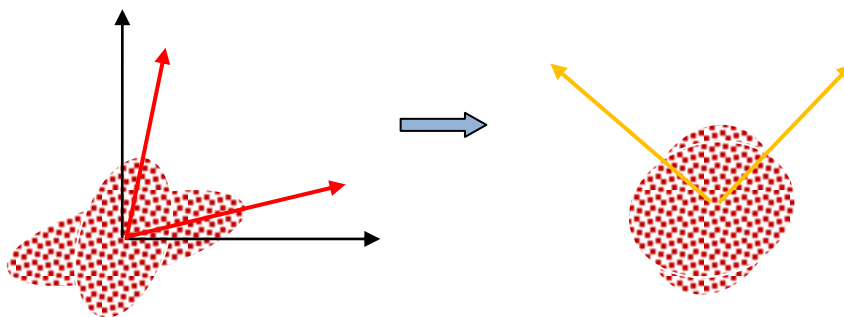


**Figure 5.5:** Whitening is transferring the data into a space such that the components are uncorrelated and with the same variance. In this pre-processing step, by transferring the original basis vectors, we obtain the PCA basis vectors (orange) which are orthogonal. The ICA basis vectors (red) are not necessarily orthogonal and make a space in which the components are independent

This preprocessing step, sometimes, is called whitening. In whitening, as a pre-processing step, we aim to transform the observed vector $x_{obs}$ linearly so that we obtain a new vector $\tilde{x}_{obs}$ which is white. That is, the components of $\tilde{x}_{obs}$ are supposed to be uncorrelated. In addition, we expect that their variances would be the same (usually equal to one).

In other words, we are looking for a transformation of original data such that the covariance matrix of $\tilde{x}_{obs}$ will be equal to the identity matrix:

$$E\left\{\tilde{x}_{obs}\,\tilde{x}_{obs}^{T}\right\} = I \tag{5.10}$$

To transform data to such space we could perform the eigenvalue decomposition of the covariance matrix of original data. This decomposition is always possible because the covariance matrix ($cov$) is a Hermitian matrix ($cov = cov^{T}$). So we are able to find real matrix $E$ and $\Lambda$ such that:

$$E\left\{x_{obs}\,x_{obs}^{T}\right\} = E\Lambda E^{T} \tag{5.11}$$

Where $E$ is the orthogonal matrix of eigenvectors ($e_1, e_2, \ldots, e_n$) of covariance matrix and $\Lambda$ is the matrix of its eigenvalues ($\lambda_1, \lambda_2, \ldots, \lambda_n$) which is the diagonal matrix:

$$E = [e_1, e_2, \ldots, e_n] \tag{5.12}$$
$$\Lambda = diag[\lambda_1, \lambda_2, \ldots, \lambda_n] \tag{5.13}$$

Then we define the transformation matrix ($Q$) as:

$$Q = E\Lambda^{-1/2}E^{T} \tag{5.14}$$

And the whitened data is obtained as:

$$\tilde{x}_{obs} = Qx_{obs} = E\Lambda^{-1/2}E^{T}x_{obs} \tag{5.15}$$

Now, if we multiply two sides of equation (5.11) by $Q$ from left and by $Q^{T}$ from right, we will have:

$$E\left\{Qx_{obs}\,x_{obs}^{T}Q^{T}\right\} = (E\Lambda^{-1/2}E^{T})E\Lambda E^{T}(E\Lambda^{-1/2}E^{T}) = I \tag{5.16}$$

And if in this equation we replace $\mathrm{Q}x_{obs}$ with $\widetilde{x}_{obs}$, we exactly obtain equation (5.10). As it is illustrated in Figure 5.5, by performing whitening step we make a new space whose basis vectors are PCA basis vectors that are orthogonal. However, ICA basis vectors are not necessarily orthogonal.

When we transfer our data into the whitened data, the *mixing matrix* will be transferred into a new matrix ($\widetilde{\mathrm{A}}$) which could be obtained by replacing equation (5.2) into equation (5.15) :

$$\widetilde{x}_{obs} = \mathrm{Q}\mathrm{A}s = \widetilde{\mathrm{A}}s \qquad (5.17)$$

In other words:

$$\widetilde{\mathrm{A}} = \mathrm{Q}\mathrm{A} = \mathrm{E}\Lambda^{-1/2}\mathrm{E}^{T}\mathrm{A} \qquad (5.18)$$

Multiplying two sides of this equation by $\mathrm{E}\Lambda^{1/2}\mathrm{E}^{\mathrm{T}}$ we obtain:

$$\mathrm{A} = \mathrm{E}\Lambda^{1/2}\mathrm{E}^{\mathrm{T}}\widetilde{\mathrm{A}} \qquad (5.19)$$

Based on assumption which expressed by equation (5.8), we can show that:

$$\widetilde{\mathrm{A}}\widetilde{\mathrm{A}}^{\mathrm{T}} = \widetilde{\mathrm{A}}E\left\{ss^{T}\right\}\widetilde{\mathrm{A}}^{\mathrm{T}} = E\left\{\widetilde{x}_{obs}\widetilde{x}_{obs}^{T}\right\} = I \qquad (5.20)$$

This means that $\widetilde{\mathrm{A}}$ is orthogonal. It is important because estimation of an orthogonal matrix is absolutely simpler. After obtaining $\widetilde{\mathrm{A}}$ in an ICA learning procedure, we can easily obtain $\mathrm{A}$, using equation (5.19) which is not necessarily orthogonal.

As it was mentioned in sub-chapter 5.4.1 during whitening we could eliminate the dimensions of data that don't contain important information. We can easily do this by eliminating very small eigenvalues and their corresponding eigenvectors from equations (5.12) and (5.13)

## 5.5 Measurement of statistical independence

In ICA we aim to estimate basis vectors such that sources would be statistically independent. So we have to create a criterion to measure the independence among the random variables which are called sources. Usually, there is no straightforward way to measure the independence among a set of random variables. Central Limit Theorem under certain conditions could help us to see how much a set of random variables are mutually independent. This theorem expresses that if we create a linear cobination of *n* independent random variables, the distribution of new random variable tends toward a Gaussian distribution if *n* tends to infinity. In other words, a sum of two

independent random variables usually has a distribution that is closer to Gaussian than any of the two initial random variables.

We assumed that the data vector $x_{obs}$ is a mixture of independent sources. Now, We define a new random variable, $z$, as the linear combination of components of $x_{obs}$ :

$$z = \sum w_i x_i = w^T x_{obs} \tag{5.21}$$

Where $w$ is a vector which determines the coefficients of the linear combination. According to equation (5.2) we can replace $x_{obs}$ with $\mathbf{A}s$ :

$$z = w^T x_{obs} = w^T \mathbf{A}s = v^T s = \sum v_i s_i \tag{5.22}$$

Here $v$ is a new vector which is defined as $v = \mathbf{A}^T w$. In other words, $z$ is a linear combination of independent sources ($s_i$) as well, with weights which are given by components of vector $v$ .

Based on Central Limit Theorem, the sum of independent random variables is more Gaussian than the original variables. So we can conclude that the random variable $z = v^T s$ is more Gaussian than each of the $s_i$ . We also can conclude that this variable becomes least Gaussian when it equals one of the $s_i$ . In this case, only one of the components $v_i$ of $v$ is nonzero. Therefore, the goal is to estimate vector $w$ such that it maximizes the *non-Gaussianity* of $w^T x_{obs}$. This vector corresponds to a vector $v$ which has only one nonzero component.

So, the learning procedure could start with selecting an initial value for vector $w$. Then through the learning steps we try to find the local maxima of non-Gaussianity criterion for the variable $w^T x_{obs}$.

It is clear that we can use non-Gaussianity to measure the level of mutual independence among a set of random variables. Now, the question is how to measure the non-Gaussianity. It is supposed to find a quantitative criterion which shows how much a probability distribution is close to (or far from) the Gaussian distribution. Kurtosis or the fourth-order cumulant is the most usual way to measure the non-Gaussianity. The kurtosis of a Gaussian distribution is equal to zero and when a probability distribution tends toward a Gaussian form; its kurtosis tends toward zero. So, it could be used as a criterion to measure the non-Gaussianity of a distribution. However, in practice, estimation of Kurtosis is not easy and encounters some problems. Thus, we have to search for another quantitative criterion which is more practicable.

We know from information theory that a random variable has the largest *entropy* if its distribution is Gaussian comparing with all other possible distributions with the same variance.

This means that entropy could be used as a measurement of non-Gaussianity. The entropy of a random variable expresses the level of its unpredictability. When we have a random variable with high level of entropy it means that we hardly could predict the value of random variable for next instant.

Since the entropy for the Gaussian variable is maximum, it is reasonable to define a new function called *negentropy* as the difference between the entropy of our random variable and the entropy of a Gaussian random variable with the same variance [6]:

$$Neg(z) = H(u) - H(z) \qquad (5.23)$$

Here $H$ stands for entropy and $u$ is a Gaussian random variable with the same variance of $z$ .thus, the negentropy is always non-negative, and it is zero if and only if $z$ has a Gaussian distribution.

Based on our criterion for negentropy, we are able to design an algorithm for learning which estimate the $w$ that minimize the non-Gaussianity of $z = w^T x_{obs}$ . In this thesis we use the Fast-ICA algorithm [5] for estimating the ICA basis vectors.

# CHAPTER 6

# ICA FOR SATELLITE IMAGES: SCALE AND DIMENSIONALITY BEHAVIOR

In this chapter we are going to perform an investigation about the advantages and challenges of applying Independent Component Analysis for VHR satellite image characterization. We explain the details of applying ICA for image data. Then, we study the scale and the dimensionality behavior of an ICA system. The goal is to find the optimum points for the scale and the dimensionality of an ICA system when it is applied for VHR satellite images.

We also propose an extra pre-processing step based on using Gabor-wavelet filters which leads to remove some unwanted redundancies in the set of ICA basis vectors.
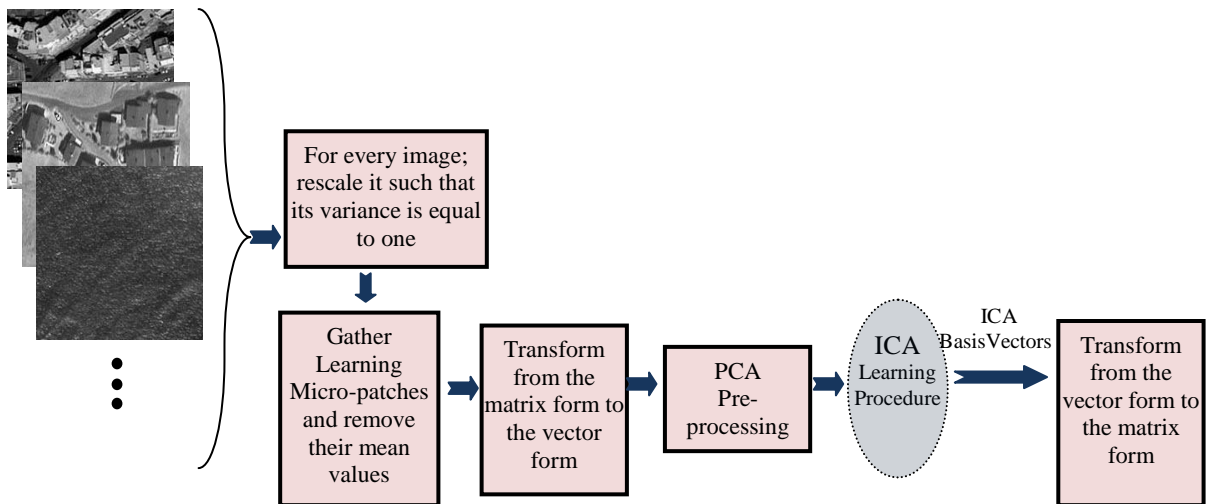


**Figure 6.1:** Different steps of applying ICA for image data

## 6.1 ICA for image data

In chapter 5 we explained the basics of Independent Component Analysis. In an ICA learning procedure, the inputs are the observed signals which are considered as vectors of random variables. But, when we are working with the images, we are dealing with matrices of data instead of vectors. In other words, we have two-dimensional observed signals instead of one-dimensional ones. So, the question is that how we can adapt the ICA methods to the image data.

Another problem is the large volume of data belongs to every image which must be processed during the ICA procedure. Actually, if we have an image with the size of $n*n$ pixels it means that we have $n^2$ random variables to be processed by ICA in the same time. So if $n$ gets larger the amount of computations increase with the rate of $n^2$. Thus, the ICA procedure gets difficult when the size of image patches increases and it gets impossible for the size of contextual satellite image patches.

In Figure 6.1, different steps for applying ICA to the image data are illustrated. At the beginning, for each image we rescale it such that its variance will be equal to one. Then we gather some smaller patches (micro patches) from initial images which are not enough small to be processed by ICA. Next step is to transform the image patch matrices to the vectors. Then we perform the PCA pre-processing step before starting the ICA learning procedure. At the end we have to transform the resulted basis vectors which are obtained in the form of vector to the form of matrix. In the following, we explain the details of these steps.

### 6.1.1 Image rescaling

In chapter 5, it is explained that in ICA system the input observed signals are supposed to be transferred such that the variance of each element of observed signals will be equal to one (See Equation 5.10). If we have a set of initial images to be sampled for ICA learning procedure and divide every image to its norm, it helps us to have sampled learning micro patches whose pixels variances are approximately equal to one.

### 6.1.2 Micro patches

It was mentioned in sub-chapter 3.4 that the objective of thesis is to define the descriptors for contextual patches with the size of 200*200. When we are going to apply ICA for these contextual image patches, ICA has to process 40000 random variables in the same time. This is impossible with the existing normal computer systems.

The solution is to gather some smaller patches from the contextual image patches to make it possible to be processed by ICA. Figure 6.2 demonstrates an example of gathering smaller image patches from the larger contextual patches. During the thesis we call these smaller image patches on which the ICA learning procedure is performed as *micro patches*. It is important not to mix up the *contextual patches* and the *micro*

*patches*. The *contextual patches* are the image patches for which we are going to define the descriptors and during the thesis we get their size as 200*200 pixels (see chapter 3.4). However, the *micro patches* are the smaller image patches which are sampled from *contextual patches* and are used by ICA learning procedure.

The size of micro patches is an important issue which is explained in sub-chapter 6.3. If we get it very small then the results don't preset necessary characteristics of the contextual patches. And if we get it too large, we will have the same problem with the contextual patches, i.e. the computational problem. In the literature, researchers usually get the size of such micro patches between 8*8 and 32*32. In the example of Figure 6.2, the size of micro patches is considered as 8*8 pixels and their scales are proportional to the initial contextual image patches in the figure.

When we are gathering the micro patches from the initial contextual patches, another important issue is the number of sampled micro patches. The number of micro patches must be enough large to cover all important events and variations inside the images. For example, if we have 200*200 contextual patches and we want to gather 8*8 micro patches, it is reasonable to gather at least 100 to 200 samples from every contextual patch.

If we increase the number of sampled micro patches, the results will be more reliable but we have to take care about the total number of samples. If there are many contextual patches to be sampled, there may be some computational problems because the total number of learning micro patches may be very high for ICA learning procedure. After gathering the micro patches, we have to remove their mean values according to the ICA supposition.
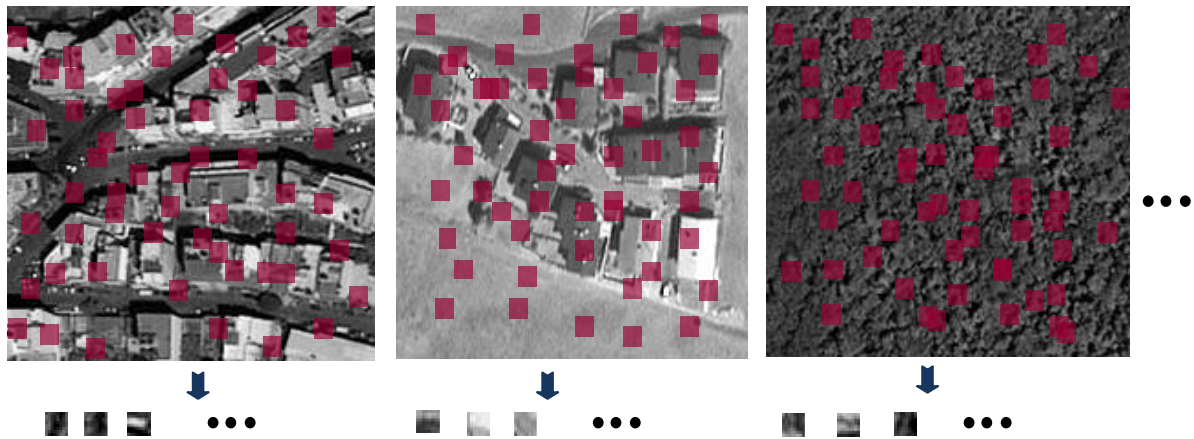


**Figure 6.2:** Gathering the micro patches from contextual patches. Here, the size of contextual patches is 200*200 and the size of micro patches is 8*8 pixels and their scales are proportional in the figure.

## 6.1.3 Micro patch conversion to the vector form

From chapter 5 we know that ICA works with the vectors of random variables. But here, the micro patches are in the form of matrix. Thus, we have to transform the matrices related to the micro patches to the vectors. Actually, images present their

information in two-dimensional format. In other words, in the images the position of every pixel is important. However in ICA the order of pixels is not important. ICA considers each pixel as a random variable and tries to find dependencies between this pixel (random variable) with other pixels (random variables).
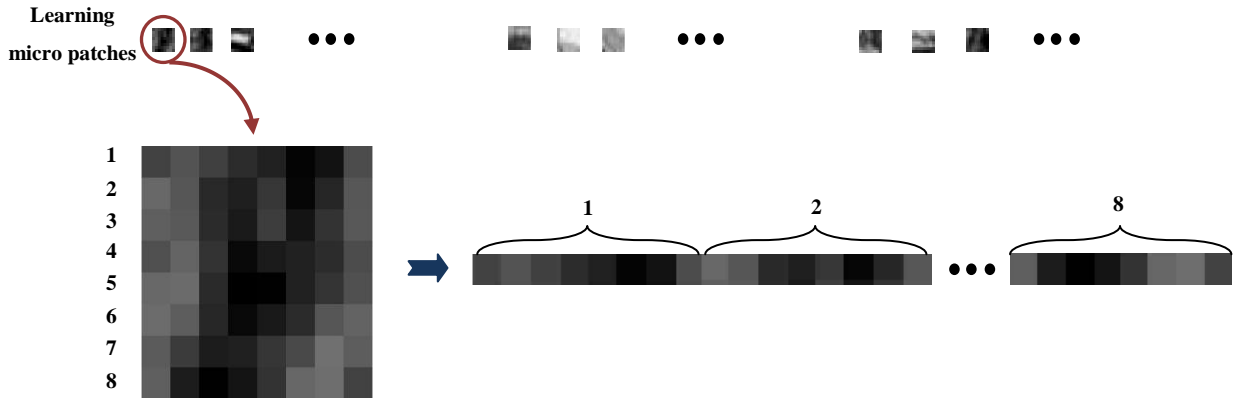


**Figure 6.3:** Converting of every micro patch matrix to the vector. Here, the size of micro patches is 8*8. The usual approach is to put each row of the matrix beside the previous one to make a vector from the matrix.

Therefore, the arrangement of the random variables in the input vectors is not significant for ICA. Consequently, we can convert the matrix of an image to the vector just by placing its pixels beside each other with an arbitrary arrangement. Of course, we have to respect the same arrangement for all micro patches which contribute in the ICA procedure. Usually, we put each row of the matrix beside the previous one to make a vector from the matrix. This is shown in Figure 6.3.

## 6.1.4 Principal Components

The ICA learning procedure usually begins with whitening step (sub-chapter 5.4.3) that includes a PCA procedure to obtain the principal basis vectors. In PCA we are going to find the new basis so that the components of data in new basis will be uncorrelated. This can help us in ICA which aims to find new basis such that the components of data in new basis are independent. In addition, through this step, we are able to reduce the dimensionality of the data. This issue is explained in sub-chapter 6.2. Here, as an example we obtained the principal basis vectors without any reduction in dimensionality for the set of test contextual patches shown in Figure 3.2. We use the micro patches with the size of 8*8 pixels and initially remove the mean value from micro patches. The normal dimensionality i.e. without any reduction for such a system is 64. If we perform PCA or ICA without any reduction in dimensionality, one of resulted basis vectors is the DC component which presents the mean value of micro patches. But we already removed the mean values of micro patches, so the maximum possible dimensionality is 63. The result is seen in Figure 6.4. The basis vectors are

converted from the vector form to the matrix form respecting the same arrangement which is used for converting the matrix form to the vector form.
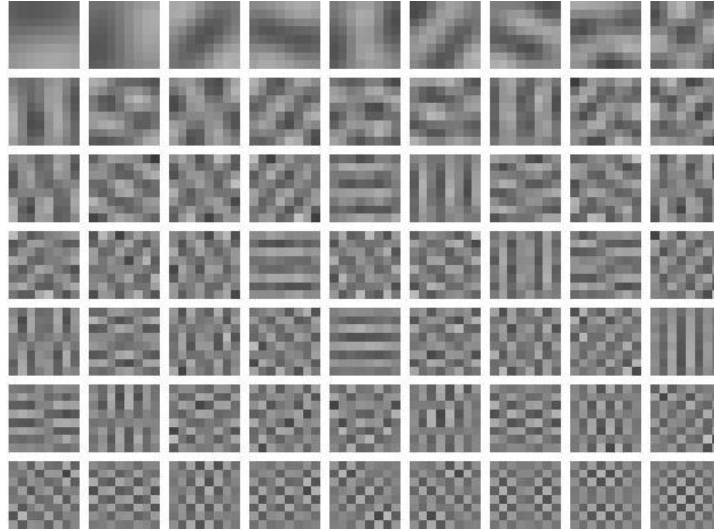


**Figure 6.4:** The set of 63 principal basis vectors with the size of 8*8 pixels which are obtained for the set of test contextual patches shown in Figure 3.2. The basis vectors are converted from the vector form to the matrix form.
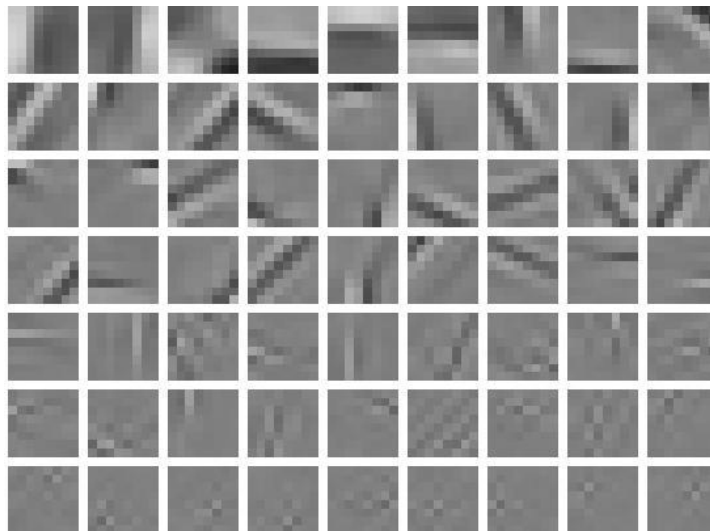


**Figure 6.5:** The set of 63 ICA basis vectors with the size of 8*8 pixels which are obtained for the set of test contextual patches shown in figure 3.2. The ICA basis vectors are converted from the vector form to the matrix form. These are the basis vectors for new space of data. These basis vectors obviously contain some types of lines or edges.

### 6.1.5 ICA basis vectors

Finally, we obtain the ICA basis vectors from the whitened learning micro patches through a learning ICA procedure which is explained in chapter 5. At the end we convert them to the matrix form respecting the same arrangement which was used for converting the matrix form to the vector form. Here, we obtained the 63 ICA basis vectors for the set of test contextual patches explained in Figure 3.2, for which we already obtained the PCA basis vectors demonstrated in Figure 6.4. The result is shown in Figure 6.5.

It is interesting to compare structures of ICA basis vectors (Figure 6.5) with PCA basis vectors (Figure 6.4). The most important difference between the two sets of components is that in the set of ICA basis vectors there are some components which obviously contain some types of lines or edges.

## 6.2 Dimensionality behavior of ICA components

If the size of micro patches is $m \times m$ pixel, it means that in the initial space, our data are represented using $n^2$ basis vectors. In other words, each pixel represents a basis vector. This is shown in Figure 6.6. In fact, in every basis vector of initial space of micro patches, only one pixel is equal to one and other pixels are zeros.



**Figure 6.6:** Initial space for micro patches. In each basis vector only one pixel is equal to one and other pixels are zeros

As it was mentioned in chapter 5, through the whitening pre-processing step, we are able to reduce the dimensionality of the data. We find the eigenvectors and eigenvalues of covariance matrix and then we choose the $n$ eigenvectors that correspond to the highest eigenvalues and eliminate the others. So we define the PCA basis vectors for the data as remained eigenvectors (see Figure 6.4 for example). Using the ICA system we transfer our data into a newer space of basis vectors, i.e. ICA basis vectors. (See Figure 6.5 for example). If we are working with an ICA system with $m$ components we define the *Reduction Factor* as:

$$ r = \frac{n}{m^2} \tag{6.1} $$

The *Reduction factor* is greater than zero and less than one because the number of components, $m$, cannot be selected over the dimensionality of initial space of data, i.e. $n^2$. Moreover, as it is mentioned before, if we have an ICA system without any reduction in dimensionality, one of resulted components is the DC component which presents the mean value of micro patches. But we already removed the mean values of

---

micro patches, so the maximum possible dimensionality for an ICA system that works with $n \times n$ micro patches is $n^2\text{-}1$.
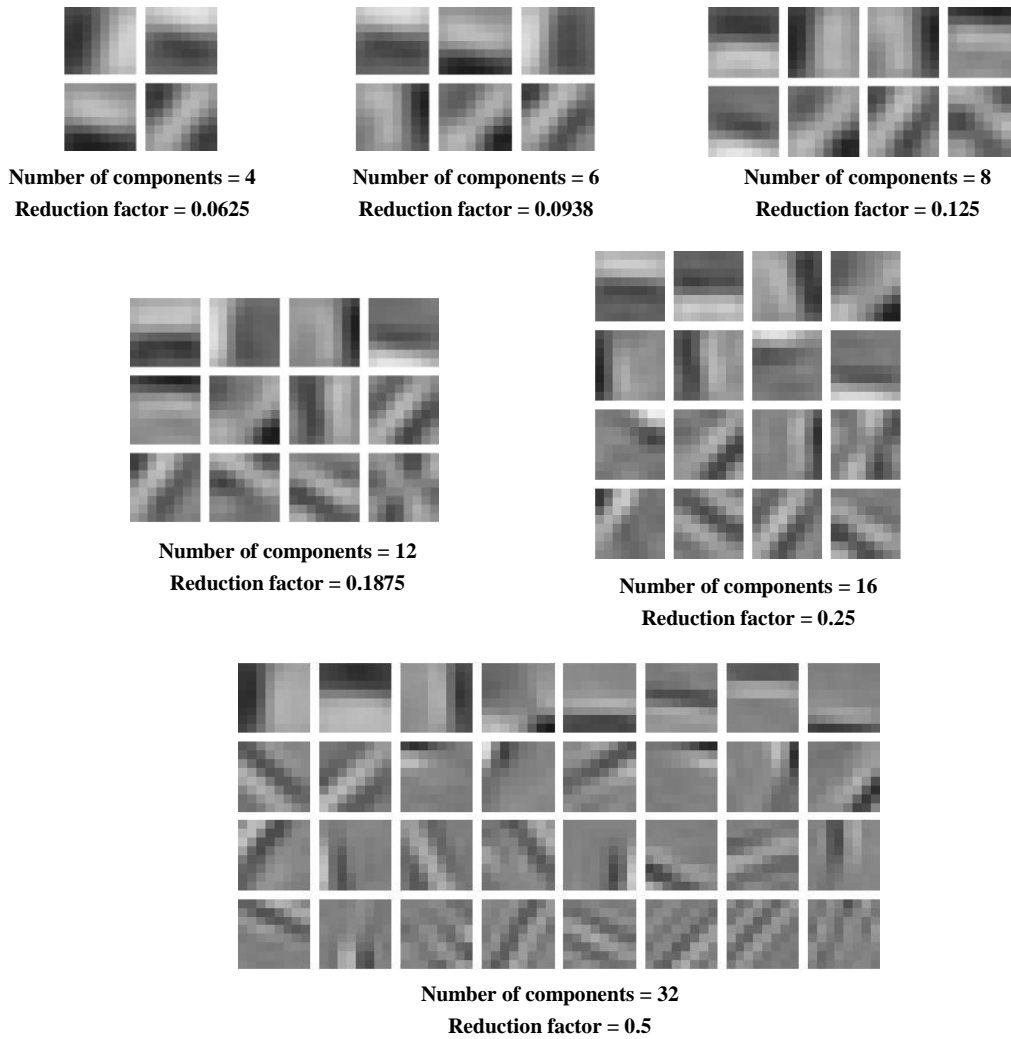


**Figure 6.7:** Six sets of ICA basis vectors with six different reduction factors. These basis vectors are obtained for the set of test contextual patches in figure 3.2. The size of basis vectors is 8*8 pixels. It is interesting to compare the structures of basis vectors in different sets. When the dimensionality increases, more various types of lines or edges appear in the set of basis vectors.

We already obtained the set of 63 ICA components with the size of 8*8 pixels for our set of test contextual patches (see Figure 6.5). This is an example of ICA system without any reduction in dimensionality. To make a comparison, we obtained the ICA components with other reduction factors in Figure 6.7. We see that when the dimensionality increases, more various types of lines or edges appear in the set of

basis vectors. It is a good sign for capability of ICA basis vectors for characterization of geometrical structures. But on the other side, when the dimensionality increases, the time of learning procedure increases as well. In the Table 6.1 we see the times which are computed for the learning procedures which are used for obtaining different sets of ICA basis vectors.

**Table 6.1:** Computation times of learning procedures for obtaining 8*8 ICA basis vector. Number of learning micro patches is 15000

| *Number of components* | 4 | 6 | 8 | 12 | 16 | 32 | 63 |
|---|---|---|---|---|---|---|---|
| *Reduction factor* | 0.0625 | 0.0938 | 0.125 | 0.1875 | 0.25 | 0.5 | 0.9844 |
| *Time (sec)* | 1.8 | 2.7 | 3.7 | 6.2 | 11.8 | 24.2 | 56.3 |

The question here is that how much we are able to reduce the dimensionality of the data without losing the important information of images. In other words, we are going to know how many basis vectors are necessary to be produced by ICA learning procedure. Answering to this question is dependent on the application for which we produce the ICA basis vectors. Here, we aim to use the ICA basis vectors and ICA sources to extract features from the contextual image patches. So our criterion for the dimensionality may differ from other applications (for example data compression) which may use ICA basis vectors and sources. However, generally we expect that the important information existing inside the image must be held when we decompose our image into the ICA basis vectors and reconstruct it back. This leads us to use the reconstruction error as the criterion.

## 6.2.1 Reconstruction

When we finish ICA learning procedure we have the *mixing matrix* and the *separating matrix*. The mixing matrix, $\mathbf{A}$, includes the ICA basis vectors as its columns and separating matrix, $\mathbf{W}$, includes the inverse filters as its rows. Now, we are going to decompose the micro patches into the set of ICA basis vectors. Here, decomposition means obtaining the ICA coefficients (ICA sources) in the space of ICA basis vectors for every micro patch. According to chapter 5, ICA sources can be computed using the separating matrix, $\mathbf{W}$, as it is stated in the equation (6.2):

$$s = \mathbf{W}x_0 \qquad\qquad (6.2)$$

In which $x_0$ is the micro image patch from which the mean value is removed and $s$ is the vector of ICA coefficients i.e. ICA sources. In other words, given a micro

image patch and also the separating matrix, $\mathbf{W}$, (and the set of basis vectors, $\mathbf{A}$ ), it is possible to obtain the set of ICA coefficients (ICA sources). In fact, the vector of ICA coefficients, $S$, is the projection of the micro patch into the space of ICA basis vectors. Reconstruction is the inverse operation of decomposition. That is, given the set of ICA basis vectors and the set of ICA coefficients related to one micro image patch, we are supposed to reconstruct the micro image patch. This can be expressed by equation (6.3):

$$x_0 = \sum_{i=1}^{n} s_i a_i \qquad (6.3)$$

In which $a_i$ is one of the ICA basis vectors that corresponds to the source $s_i$. So we are able to reconstruct a micro patch if we are given its corresponding ICA coefficients and also the set of ICA basis vectors. Dividing a contextual patch to several micro patches we are able to produce the reconstructed contextual patch. Figure 6.8 is an illustration of reconstruction procedure.
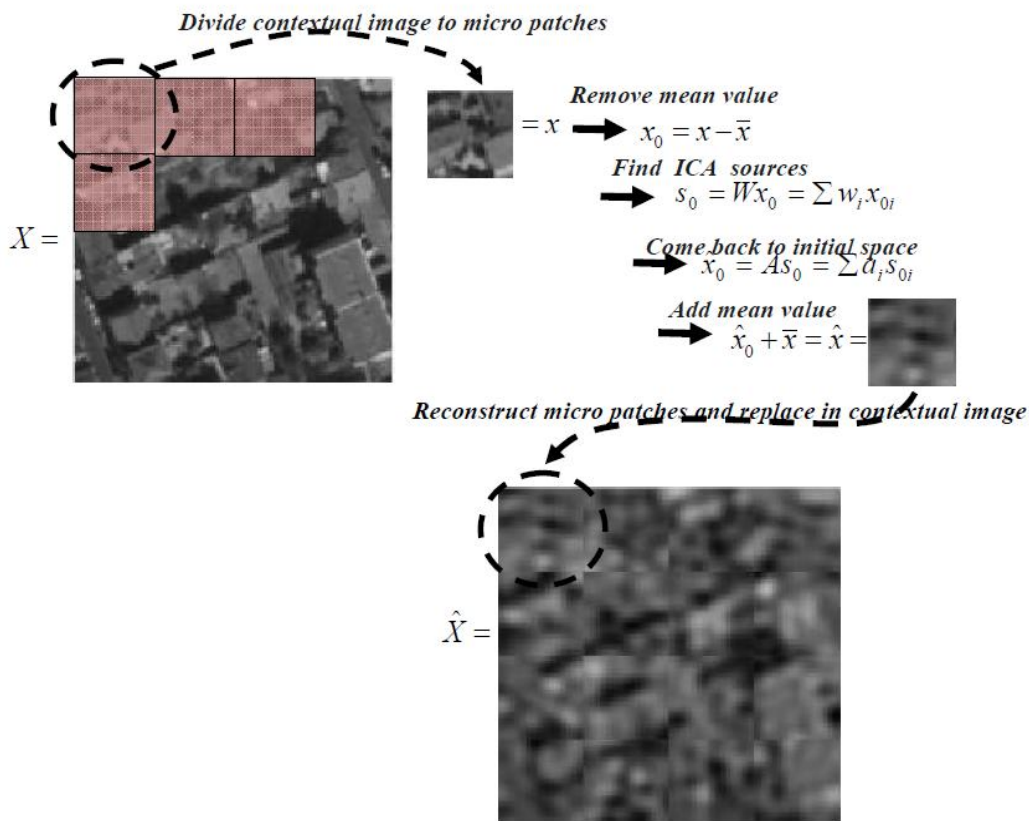


**Figure 6.8:** Reconstruction of a contextual patch by dividing it to the micro patches. Then each micro patch could be reconstructed based on its ICA coefficients (ICA sources). Finally, the reconstructed image patch will be replaced in the contextual image patch.

## 6.2.2 Reconstruction error

When we are working in the space of a set of ICA basis vectors, we expect that it holds the important information of input signals (here, the micro image patches), in its corresponding ICA coefficients (ICA sources). A way to evaluate the efficiency of an ICA system is to compare the initial micro patch and its corresponding micro patch which is reconstructed given the ICA coefficients, to see how much the reconstructed micro patch is similar to the initial one. A usual approach to compare the two micro patches is the error of reconstruction which is computed with the equation (6.4):

$$e = mean(\sqrt{(x - \hat{x})^2}) / mean(\sqrt{x^2}) \qquad (6.4)$$

Where $x$ is the initial micro patch, $\hat{x}$ is the reconstructed micro patch and $e$ is the reconstruction error. The idea is to calculate the reconstruction error as a function of dimensionality of ICA system, i.e. the number of ICA basis vectors. Table 6.2 contains the average of reconstruction error for different reduction factors when we use 8*8 basis vectors. These results are obtained for a set of 10000 micro patches which are randomly gathered from the set of test contextual patches.

**Table 6.2:** Average of reconstruction error for different reduction factors which are obtained for 10000 of 8*8 gathered micro patches.

| Number of components | 4 | 6 | 8 | 12 | 16 | 32 | 63 |
|---|---|---|---|---|---|---|---|
| Reduction factor | 0.0625 | 0.0938 | 0.125 | 0.1875 | 0.25 | 0.5 | 0.9844 |
| Reconstruction error (%) | 36.1 | 26.9 | 23.1 | 20.2 | 17.6 | 12.7 | 1.1 |

## 6.2.3 Optimum reduction factor

The objective is to select the optimum number of components for an ICA system. For that, we have two criteria to be considered: the computation time and the reconstruction error. It could be seen as an optimization problem with a *Cost Function (CF)* like the following equation:

$$CF(r) = kt(r) + (1 - k)e(r) \qquad (6.5)$$

Where $r$ is the reduction factor and $0 \leq k \leq 1$ is a parameter which represents the importance of computation time ($t$) with respect to the reconstruction error ($e$). If $k$ is zero, it means that $t$ is not important at all; so the cost function is exactly equal to $e$.

When $k$ is increasing to one the importance of $t$ will be increasing with respect to $e$ and for $k = 1$ the cost function is exactly equal to $t$. It is possible to define other forms for the cost function; however, it must be an increasing function with respect to $e$ and $t$, because we expect that both of $e$ and $t$ are in their minimum levels. Equation (6.5) is a simple form for such a cost function. Parameter $k$ is adjusted regarding to the variation ranges of $e$ and $t$ and, of course, the importance of $t$ comparing with $e$ in the application. In our work both of $e$ and $t$ are important so $k$ must not be very close to zero or one.
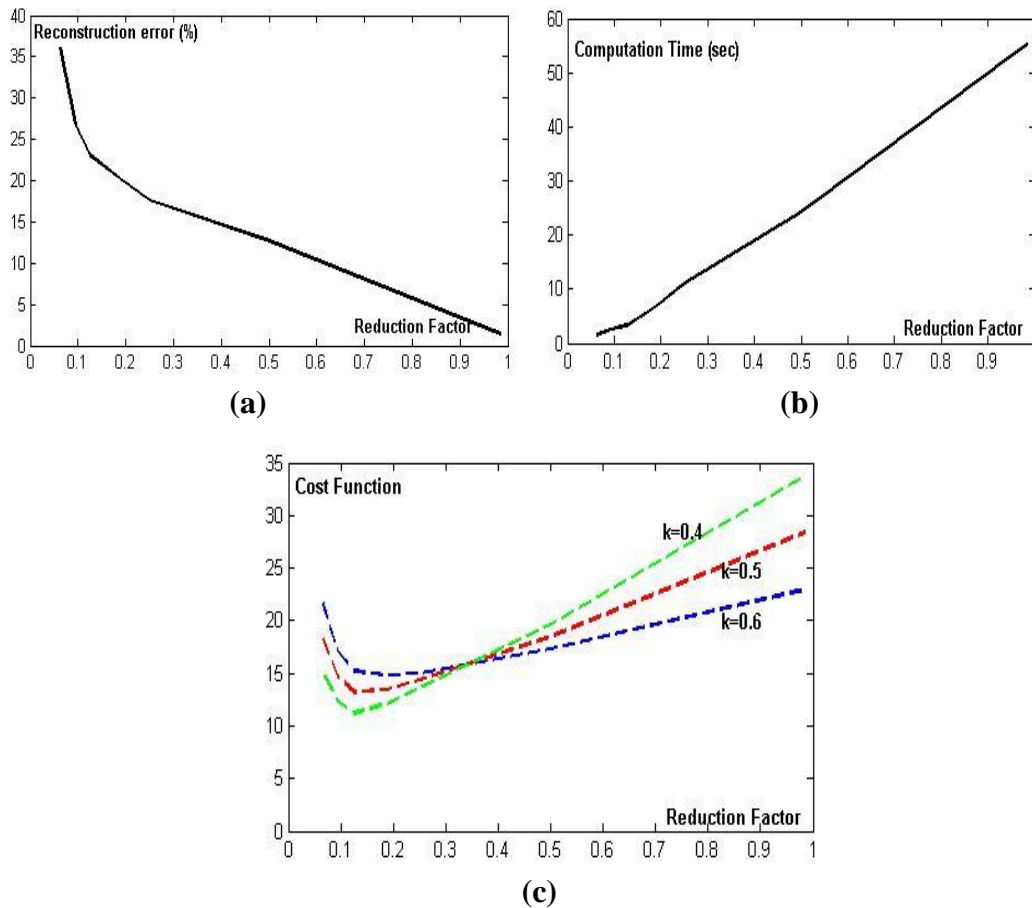


**Figure 6.9:** (a): Reconstruction error as a function of reduction factor. (b): Computation time as a function of reduction factor. (c): Cost function as a function of reduction factor. Here, the size of basis vectors is 8*8 pixels.

Figure 6.9 (c) shows the cost function for 3 values of $k$. The optimum point is the local minimum of cost function. We see that for 3 values of $k$, the optimum point differs between 0.1 and 0.12. In general, for an interval of 0.3 around 0.5, the optimum

reduction factor could be found somewhere between 0.08 and 0.14. This result is obtained for an ICA system with the 8*8 basis vectors. However, we performed similar experiments for the cases 16*16, 32*32, 48*48, 64*64 and we found very similar results for all cases.

## 6.3 Scale behavior of ICA components

Choosing the size of basis vectors for an ICA representation which here is called as *scale size* is very important in our work. The ideal case is when the size of contextual patches, for which we are going to extract features, is equal to the size of basis vectors. But usually it is not possible because of the computational problems. For example, we have 200*200 contextual patches but an ICA learning procedure with such a big size is not possible. Therefore, we have to use some smaller learning micro patches which lead to a set of basis vectors with the same size.

When the size of basis vectors is getting larger, we could see more various forms of edges, lines and other structures. This is shown in Figure 6.10.



**Size of basis vectors=16*16**     **Size of basis vectors =32*32**     **Size of basis vectors =64*64**
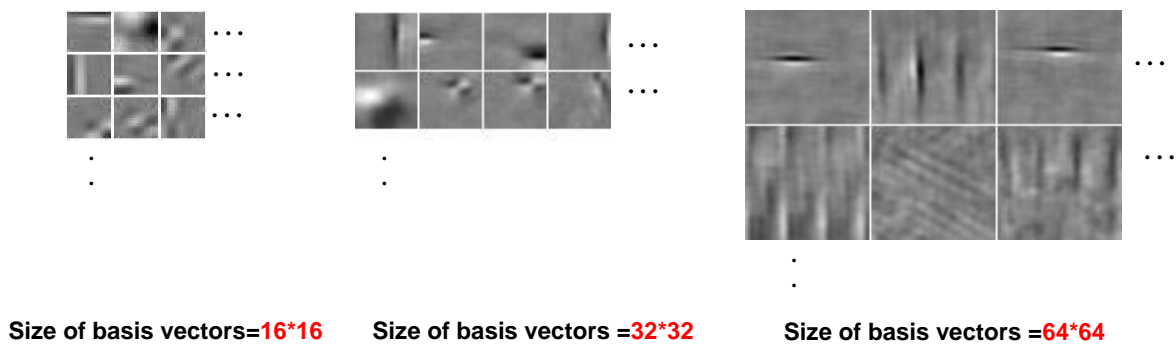
**Figure 6.10:** Samples of basis vectors for different scale sizes

Thus, we expect better results when we use larger basis vectors but in the same time we have more computational problem. We could use a similar approach to find the optimum scale size. We define a cost function like the equation (6.5) but as a function of the size of basis vectors:

$$CF(m) = kt(m) + (1 - k)e(m) \qquad (6.6)$$

Where $m$ is the size of basis vectors. We calculated the cost function for 5 different scale sizes: 8*8, 16*16, 32*32, 48*48 and 64*64. For all cases we selected the number of basis vectors such that the reduction factor is (exactly or very close to) 0.1 and the number of learning samples is 8500. In Figure 6.11(c) we could see this cost function for 3 different values of $k$. For small $k$ the optimum point converges to $m = 32$ (That is equivalent to size of 32*32) because the importance of time is little with respect to

the reconstruction error. When $k$ increases, the optimum point moves to $m = 16$.

In our work, both of time and the efficiency of ICA system (which here is represented by reconstruction error) are important so it seems that the ICA basis vectors with the size of 16*16 could be the optimum point.
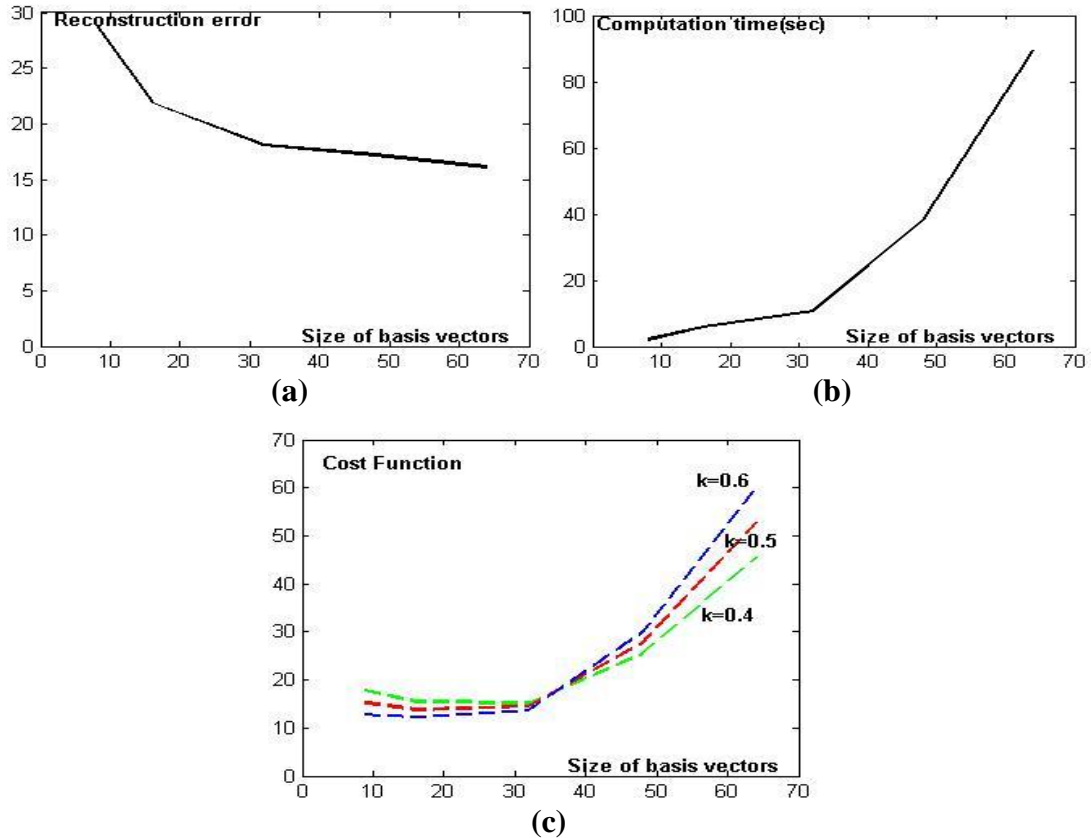


**Figure 6.11:** (a): Reconstruction error as a function of size of basis vectors. (b): Computation time as a function of size of basis vectors. (c): Cost function. We calculated the cost function for 5 different scale size: 8*8, 16*16, 32*32, 48*48 and 64*64. For all cases, the reduction factor is 0.1 and the number of learning samples is 8500

## 6.4 Gabor filters pre-processing step

During our experiments with ICA we noticed that in some of ICA basis vectors we see only small changes at their corners and the rest of the basis vector's surface does not contain important information (see Figure 6.12(a)). Our experiments show that such of these basis vectors do not play important role in image characterization, because they cannot present significant character of the image. The reason of appearing such basis vectors is that many micro patches that are gathered randomly for learning procedure, present only very small parts of a structure at their corners.
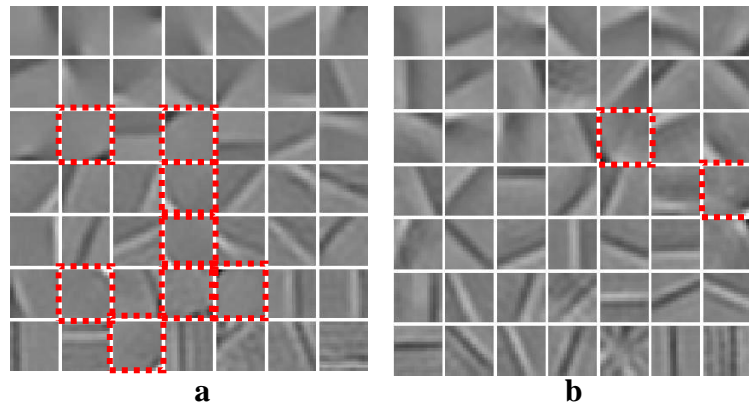
**Figure 6.12:** Example of effect of Gabor pre-processing filters onto a set of 49 ICA basis components. (a) is obtained with an ordinary procedure. 8 of basis vectors present only small changes at their corners (b) is obtained with a Gabor filtering pre-processing. Number of such basis vectors reduces to 2.

Therefore, the large parts of these micro patches do not contain important information. This causes our model to present a kind of redundancy. On the contrary, the learning image patches that all parts of their surfaces, in particular the central parts, contain structures, shapes, edges, etc, contain more information from the scene and make the model more reliable. The idea is to increase the number of these learning image patches with respect to the first group. In other words we expect that our learning micro patches present their information in their central parts as much as possible.

We propose to use Gabor-wavelet filters to measure how much of information are placed in the central parts of learning image patch. We provide a set of 100 Gabor filters with 10 scales in angle and 10 scales in frequencies, but for all of them the origin point is considered as the central pixel of filter (see Figure 6.13). Details of this Gabor system is explained in sub chapter 2.2.2.

To verify if a learning image patch present enough information, we project it into the set of Gabor filters and obtain coefficients $g_1$, $g_2$,... $g_n$ and compute the sum of their squares as the energy of image patch in the Gabor system:

$$G = g_1^2 + g_2^2 + \cdots g_n^2 \tag{6.7}$$

We call this value as $G$ parameter of the micro patch. So, the micro patches with higher $G$ parameters present more information in their central parts.

In Figure 6.12 we show an example of two sets of ICA basis vectors. One of them is obtained with a Gabor filtering pre-processing filters and the other is obtained with an ordinary procedure. Both of these sets contain 49 components. When we don't apply the Gabor pre-processing step, 8 of these basis vectors present only very small changes in their corners. However, when we use Gabor-wavelet filters as a processing step, the number of such basis vectors reduces to two. Although this is an example of effect of

Gabor filtering step on the ICA basis vectors, the same effect happens for the TICA basis vectors (explained in chapter 8) if we apply Gabor filtering step. In our work, after randomly gathering $2L$ micro patches, we selected $L$ of them which have higher $G$ parameters as the learning micro patches.



Projection

$$G = g_1^2 + g_2^2 + ... + g_n^2 \quad \overset{?}{>} \quad \text{Threshold}$$
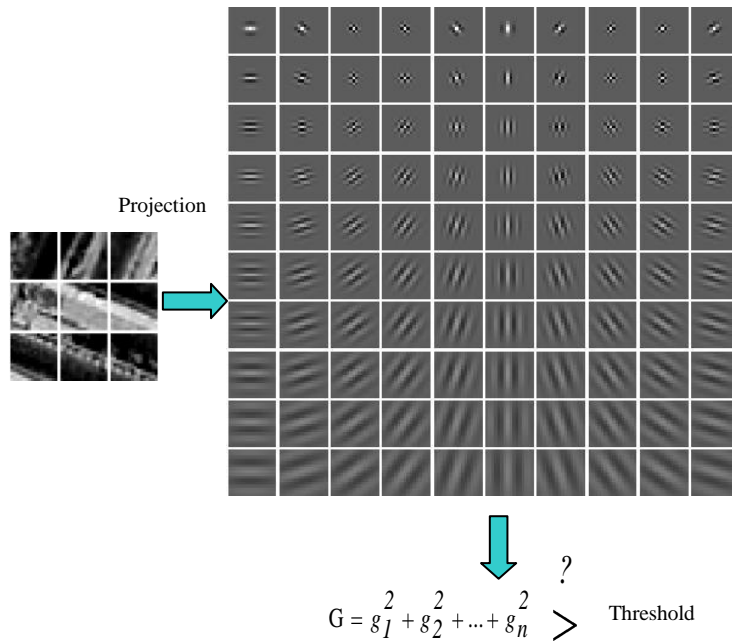
**Figure 6.13:** Gabor filtering as a pre-processing step for selecting optimum leaning micro patches. We project the learning micro patch into the set of Gabor filters and obtain the coefficients. Then we compute the energy of micro patch in the Gabor system (G parameter) as the criterion which shows how much of changes appear in the central part of learning micro patch.

This step could be considered as a pre-processing step for the ICA learning procedure. It is not mandatory and is not used in the usual ICA approach in the literature. So we didn't bring it as the other pre-processing steps in sub-chapter6.4 and preferred to explain it as an individual sub-chapter to introduce it as new approach for reducing the redundancy in the ICA basis vectors. However, it helps us for characterization of satellite images especially when we are going to extract features from the basis vectors (chapters 9 and 10).

## 6.5 Conclusions

In this chapter we explained the technical details for applying ICA for satellite images. Moreover, we studied the scale and dimensionality behavior of an ICA system when we use it for VHR satellite images. We found that the optimum reduction factor could be detected somewhere between 0.08 and 0.14. In addition, the basis vectors

with the size of 16*16 and 32*32 pixels are more efficient according to their results and their computation times. Between the two choices, the size of 16*16 seems to be more suitable for our work. These optimum points were obtained regarding to two criteria: the time of computation and the reconstruction error.

In addition we introduced an approach to reduce the redundancy in a set of basis vectors. We proposed to use Gabor-wavelet filters to choose the optimum learning micro patches.

# CHAPER 7

# FEATURE EXTRACTION FROM ICA SOURCES

There are two points of view for feature extraction using ICA from image data. The usual approach is to use the ICA coefficients (ICA sources) and the other is to use the ICA basis vectors which are corresponding to every image. In this chapter we concentrate on extracting features from the ICA coefficients for every contextual image patch.We initially explain that how we can define features for micro image patches with the size of given basis vectors, then we generalize our approach for larger image patches, particularly for contextual image patches. In addition we introduce a simple clustering approach to evaluate the efficiency of the features.

## 7.1 Features for a micro patch

From previous chapters we know how to obtain the basis vectors through an ICA learning procedure. Here, we suppose that a set of ICA basis vectors is given. It means that we have the mixing matrix $\mathbf{A}$, and the separating matrix $\mathbf{W}$. The objective is to define features for a test image patch using the set of ICA basis vectors. The size of image patch is important for feature extraction. Usually our image patches for which we are going to extract features are larger than given basis vectors. This case is explained in next sub-chapter. However, if our image patch is with the same size as given basis vectors, we can simply decompose the image patch onto the given basis vectors using the equation (6.2). Then we define the features as the ICA sources which are obtained for our image patch.

There are two technical points when we are obtaining the corresponding ICA sources given a set of ICA basis vectors. The first point is about the mean value of our test micro patch. Referring to the Chapters 5 and 6, we remove the mean value from each of learning micro patches at the beginning of learning procedure. Thus, here it is reasonable to remove the mean value of our test micro patch before decomposing it onto the ICA basis vectors.

Test Micro Patches ($x_{obs}$)
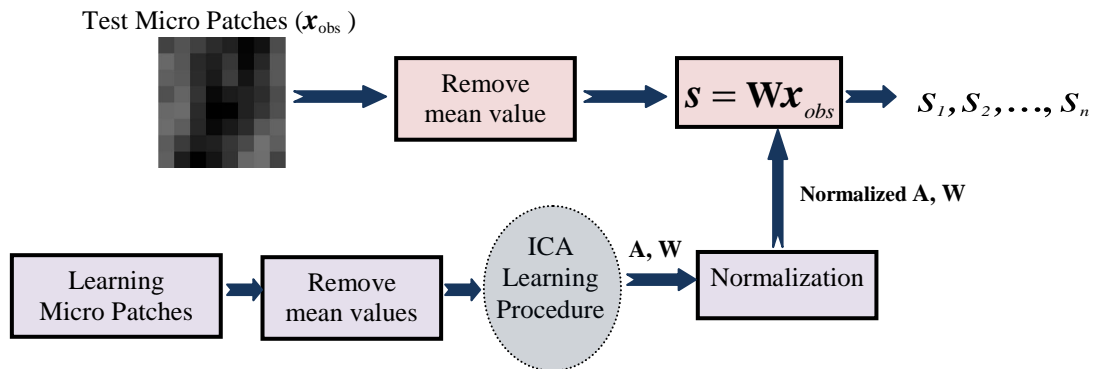


**Figure 7.1:** Obtaining the source based features for a test micro patch when mixing matrix and separating matrix are given. We decompose the micro patch onto the set of basis vectors. These features are called micro features.

The other point is related to the normalizing of basis vectors. In sub-Chapter 5.3 we assumed that each of ICA sources has unit variance. See equation (5.8). So, the matrix $\mathbf{A}$ is adapted in the ICA learning procedure such that this assumption will be satisfied. In fact, the basis vectors, $a_i$, are obtained with different *norm*s so that the variances of ICA sources would be the same. In other words, the norm of every basis vector, $a_i$, gives some information about the variation of data around the direction of the basis vector, $a_i$ $_i$. Normally, a basis vector with a larger norm is a sign of larger variation of data around its direction.

However, when we use ICA sources as features, they carry the information from the image patches and we don't expect that different sources have the same variance. Moreover, when we are going to represent our data in an arbitrary space of basis vectors it is reasonable that the basis vectors have the same norms. Consequently, for obtaining the ICA sources as the features of a test micro patch, we initially normalize the basis vectors (the columns of matrix $\mathbf{A}$ ) and their corresponding filters in separating matrix, i.e. the rows of matrix $\mathbf{W}$. The procedure of obtaining the source based features for a test micro patch is illustrated in Figure 7.1. We call these features as *micro features*.

## 7.2 Features for contextual patches

If our image patch for which we are going to extract features is larger than given basis vectors, as in the case of contextual patches, we have to sample sufficient number of micro patches with the same size as basis vectors, then we remove their mean values and decompose them onto the set of given basis vectors.

Thus for every sampled micro patch we obtain a set of $n$ sources. $n$ is the number of basis vectors. If, for example, we gather $d$ micro patches for our contextual patch, then

for each $S_i$ , $i = 1, 2, \cdots, n$ we have $d$ samples ( $S_i(1), S_i(2), \cdots, S_i(d)$ ) and we have to find a way to define only one feature for every $S_i$ .

We can conclude from sub-chapter 5.3 that the sign of ICA sources is not important because both of sources and basis vectors are supposed to be estimated and the sign of a source can be canceled by the sign of corresponding basis vector . So, a simple way is to apply a root mean square over the $d$ samples of the same $S_i$ to define the feature $f_i$ :

$$f_i = \sqrt{\frac{1}{d} \sum_{k=1}^{d} S_i^2(k)} \qquad (7.1)$$

This procedure is shown in Figure 7.2.
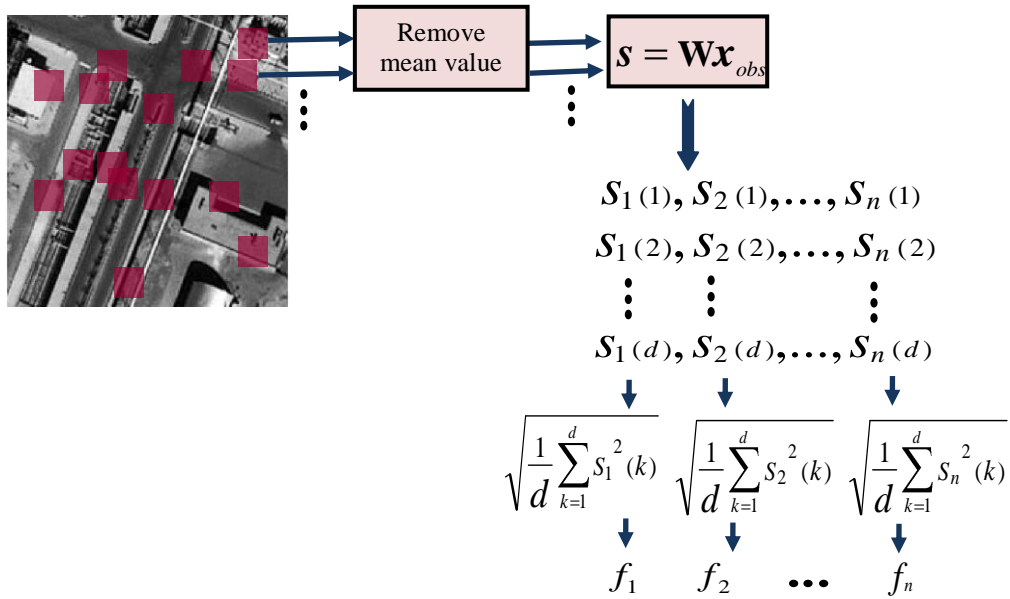


**Figure 7.2:** Defining contextual features for a contextual image patch when mixing matrix and separating matrix are given. We gather a sufficient number of micro patches and decompose them onto the set of basis vectors and for each we obtain a set of $n$ sources. Applying the root mean square over samples of a specific source we obtain the corresponding feature.

The basis vectors which are used for feature extraction were already obtained through a learning procedure which takes 10000 micro patches with the size of 16*16 from all types of classes as learning samples. We used a reduction factor of 0.098 for the dimensionality of ICA system that is equal to obtaining 25 basis vectors. Therefore, the number of features for each contextual patch is 25. The set of these basis vectors is shown in Figure 7.3.
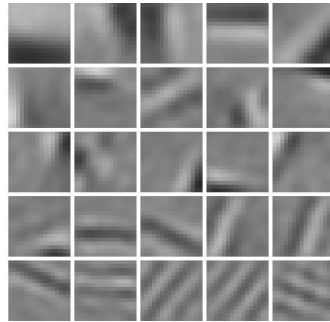
**Figure 7.3:** Set of 25 ICA basis vectors with the size of 16*16 pixels which are obtained from a set of 10000 learning micro patches gathered from all types of classes. The basis vectors are sorted ascending by their mean frequencies from left to right and up to down.

There are other ways to define feature $f_i$ from the samples of $s_i$. For example, we may count the number of sampled micro patches for which the value of $|s_i|$ is the maximum comparing with the other sources. We experimentally found that the two ways works with a similar level of efficiency but the idea of using root mean square over the samples of a specific source, sometimes, works a little better. So, we decided to use this idea to define the corresponding ICA feature. We call these features as *contextual features*.

Here is also another technical point. As it was mentioned in sub-Chapter 6.1.1, In an ICA learning procedure, before gathering micro patches; we rescale the contextual patches such that their norm is equal to one. So our basis vectors are obtained with such condition from contextual patches. Therefore, it is reasonable to do the same when we are going to extract features from contextual patches.

In Figure 7.4 we see the obtained ICA features for the test contextual patches which are classified in 8 classes (See Figure 3.2). From each class we selected the first 10 contextual patches for feature extraction. Then we gathered 1000 micro patches from each contextual patch to be decomposed onto the set of basis vectors. In sub-Chapter 7.4 we explain about the number of micro patches that must be sampled from each contextual patch.

Looking to the produced features for different classes in Figure 7.4, we could see similarities among the features which are obtained for 10 images of one class and also differences among the features which are obtained for images from different classes. In addition, we could see that when two classes are close to each other, their features are more similar. For example look at the features which are obtained for class 7 (Town) and the class 6 (City-2).

All of these observations from Figure 7.4 could be good news for us because we expect the same roles for a descriptors or a set of features. However, we may have concerns about confusing between similar classes and also about classes whose contextual features have larger variances. For example look at contextual features which are obtained for class 2 (Farm).

**Figure 7.4:** ICA features for 8 classes of contextual patches. From each class, 10 contextual patches are selected for feature extraction.

### 7.2.1 Number of sampled micro patches

The number of sampled micro patches is selected experimentally. In fact, in similar approaches in the literature, for each pixel of image, they gather one micro patch to be decomposed. For example for this case we would have 200*200=40000 samples. Here, we found that if we randomly sample a number of micro patches the resulted feature vector does not change significantly. We performed an experiment: For some contextual patches, we tried to extract features. We started from the 40000 samples and reduced the number and for each step we verified the changes of feature vectors. We found that if we use 600-1000 samples the changes in the feature vector is not significant.

## 7.3 Simple clustering for evaluation

To evaluate the efficiency of such features for separating different classes, we perform a simple clustering that is based on Euclidean distance measurement. This is not a perfect evaluation of features but is useful to observe how they work. In chapter 12 we will perform a perfect evaluation for each kind of descriptors. The idea is to put every test contextual image patch in a cluster whose members are more similar to that contextual patch. One way to measure this similarity is to compute the distance of new sample from the *center* of each cluster. Then we put new sample in a cluster whose distance from the sample, i.e. the distance of its center from the new sample, is minimum. The question here is to define center for a cluster. Usually, we consider the mean of existing members as the center of a cluster. In a clustering problem, sometimes we do know the initial positions of cluster centers and sometimes we don't.

**Table 7.1**: Results of simple clustering. We used the ICA source based feature vectors

| Clusters | Class1 Factory | Class2 Farm | Class3 Village | Class4 City-1 | Class5 Forest | Class6 City-2 | Class7 Town | Class8 Sea | Total |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 61 | 8 | 6 | 3 | 4 | 8 | 0 | 1 | 91 |
| 2 | 4 | 44 | 4 | 0 | 4 | 4 | 7 | 13 | 80 |
| 3 | 5 | 6 | 66 | 2 | 2 | 2 | 7 | 0 | 91 |
| 4 | 6 | 1 | 0 | 67 | 4 | 7 | 4 | 8 | 97 |
| 5 | 0 | 10 | 1 | 4 | 67 | 0 | 2 | 8 | 92 |
| 6 | 8 | 6 | 6 | 2 | 1 | 64 | 5 | 0 | 92 |
| 7 | 6 | 4 | 7 | 5 | 2 | 0 | 65 | 0 | 89 |
| 8 | 0 | 11 | 0 | 7 | 6 | 5 | 0 | 60 | 89 |
| Total | 90 | 90 | 90 | 90 | 90 | 90 | 90 | 90 | 720 |

Here, we are going to perform a clustering with 8 clusters and we supposed that initial values of clusters' centers are the mean value of 10 contextual patches for which we obtained the features in Figure 7.4. Consequently, for each class of test set 90 contextual patches remain to be clustered, that is, totally 720 samples. For each of these samples we compute the Euclidean distance from every cluster centers and put it into the cluster which has the minimum distance from the sample. Then we could change the center of cluster regarding to the new member of the cluster.

Table 7.1 shows that from which classes samples of each cluster come, however, Table 7.2 presents the same results in the percent format. In fact, the bold diagonal numbers in these tables show the amount of samples which are classified correctly. This is usually known as the *Precision* value. According to Table 7.2 we have an average of 68.6 % for this case. We present the results for other feature extraction approaches in next chapters in the same format.

**Table 7.2**: Results of simple clustering in the percent format.

| **Clusters** | **Class1** Factory | **Class2** Farm | **Class3** Village | **Class4** City-1 | **Class5** Forest | **Class6** City-2 | **Class7** Town | **Class8** Sea |
|---|---|---|---|---|---|---|---|---|
| **1** | **67.7** | 8.9 | 6.7 | 3.3 | 4.4 | 8.9 | 0 | 1.1 |
| **2** | 4.4 | **48.9** | 4.4 | 0 | 4.4 | 4.4 | 7.8 | 14.4 |
| **3** | 5.5 | 6.7 | **73.3** | 2.2 | 2.2 | 2.2 | 7.8 | 0 |
| **4** | 6.7 | 1.1 | 0 | **74.4** | 4.4 | 7.8 | 4.4 | 8.9 |
| **5** | 0 | 11.1 | 1.1 | 4.4 | **74.4** | 0 | 2.2 | 8.9 |
| **6** | 8.9 | 6.7 | 6.7 | 2.2 | 1.1 | **71.1** | 5.5 | 0 |
| **7** | 6.7 | 4.4 | 7.8 | 5.5 | 2.2 | 0 | **72.2** | 0 |
| **8** | 0 | 12.2 | 0 | 7.8 | 6.7 | 5.5 | 0 | **66.7** |

## 7.4 Dimensionality and Scale size effects

In chapter 6 we studied the effects of dimensionality and scale size of basis vectors in an ICA system which is used for satellite image characterization. We defined the reconstruction error as the criterion for capability of ICA system. Here, we are going to verify the previous results with a new criterion: average of precision.

For that, we have to repeat the same clustering experiments with different reduction factor and scale sizes. We also need a new definition for our cost function:

$$CF = kt + (1-k)(100-P) \tag{7.2}$$

Where $P$ is the average of precision. If we compare equation (7.2) with the equations (6.5) and (6.6), we find that the reconstruction error is replaced by the term $100 - P$. Initially, we take the size of basis vector as 16*16 and perform the clustering experiments using different reduction factors. The result is shown in Figure 7.5(a). Then we take the reduction factor as 0.098 (equivalent to 25 basis vectors) and repeat the experiments for different sizes of basis vectors. The result is shown in Figure 7.5(b).
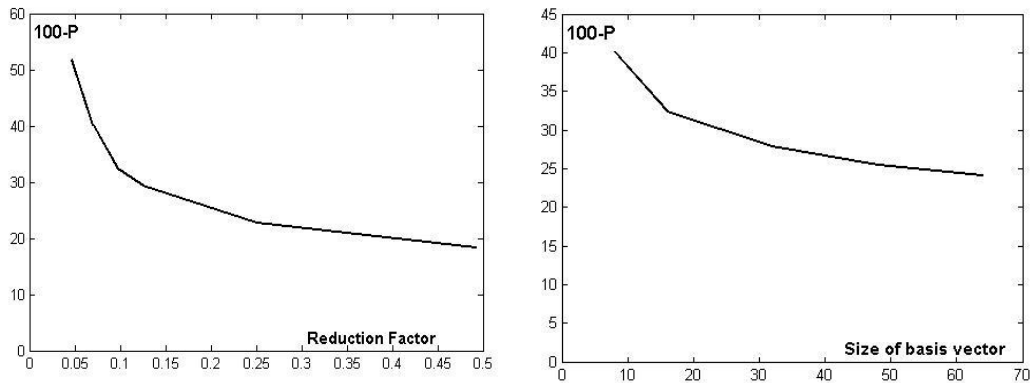


**Figure 7.5:** Dimensionality and scale size effects on results of clustering. (a):The size of basis vectors is taken constant (16*16) and *100-precision* is obtained for different reduction factors. (b):The number of basis vectors is constant (25) and *100-precision* is obtained for different scale sizes.

Comparing the results in Figure 7.5 with the Figures 6.9(a) and 6.11(a), we could conclude that the variation of term 100-P is very similar to the variation of reconstruction error. Thus the same results for the optimum dimensionality and scale size in chapter 6 could be validated.

## 7.5 Basis vectors improvement

As it was mentioned in chapter 4, the main goal in many applications is to detect the urban zones. In other words we are going to separate the class of urban area from other classes. Here we propose an approach to improve the basis vectors for such a purpose. Suppose that we have two classes: the urban area and none-urban area. Here, to simplify the problem we take *class2* (City-1) as the urban area and *class5* (Forest) as the none-urban area.

The idea is to choose the most important basis vectors from each class and bring them together to make a new set of basis vectors. In the ICA learning procedure, the basis vectors of each class are obtained such that the ICA sources can be considered as independent random variables with the same mean (usually equal to zero) and the same variance (usually equal to one). Thus, if we project some training micro patches from a specific class onto its basis vectors and obtain ICA sources ($s_i$) then we

compute the probabilities of the squared ICA sources ($P(s_i^2)$), we expect a flat curve, i.e. $P(s_1^2) \approx P(s_2^2) \approx \cdots$. However, if we project micro patches from a specific class onto the basis vectors of another class, we don't expect a curve being as flat as the previous one. We see this in Figures. 7.6(c) and 7.6(d).
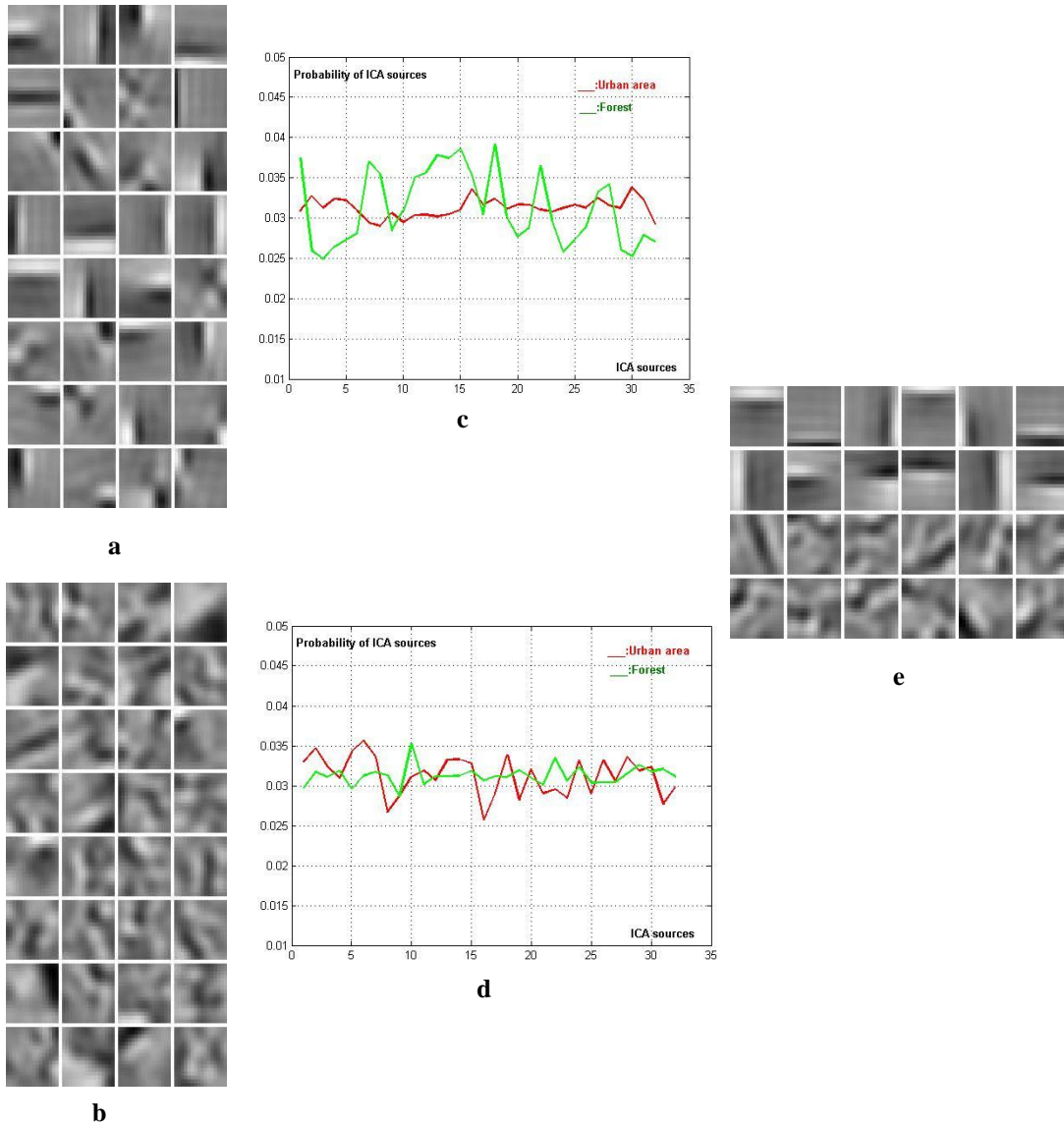


**Figure 7.6:** Providing a new set of basis vectors by collecting the most important ones. **(a)** and **(b):** Initial basis vectors for forest and urban area **(c):** Probability of ICA sources for patches projected onto urban area basis vectors. **(d):** Similar curves for the projection onto basis vectors of forest. **(e):** New set of basis vectors. The two upper rows are the 12 most significant basis vectors of urban area and the lower two rows are the 12 most significant basis functions of forest.

In this example, initially we obtain 32 basis vectors for each class. Using these curves, we can choose the most important basis vectors for each class. For example, for the urban area class, we select a basis vector if $P(s_i^2)$ for the urban area is obviously higher than $P(s_i^2)$ for other class. As an example, see the 25th basis component of an urban area in Figure 7.6(c). In other words, we can say that these basis vectors faithfully represent the urban area class in comparison with the class of forest. The new set of basis vectors is made by combining the most important basis functions from two classes. We have selected the 12 most important basis functions for each class in the example of Figure7.6.

The new combined set of basis vectors is shown in Figure 7.6(e). Our clustering experiments show that this set of basis vectors has a better result in comparison with the ordinary basis vectors in Figure 7.3 for the goal of separating the urban area from forest images. The results are summarized in Tables 7.3 (a) and 7.3(b).

**Table 7.3**: Results of clustering in the percent format for separating the urban area from the forest. (a) Result of ordinary set of basis vectors. (b) Results of the new set of basis vectors combined from the most important basis vectors of two classes.

(a)

| Clusters | Class4 City-1 | Class5 Forest |
|----------|---------------|---------------|
| 1 | **84.4** | 18.9 |
| 2 | 15.6 | **81.1** |

(b)

| Clusters | Class4 City-1 | Class5 Forest |
|----------|---------------|---------------|
| 1 | **90** | 15.6 |
| 2 | 10 | **84.4** |

## 7.5 Conclusions

In this chapter we explained the approach for extracting features from contextual patches, based on ICA sources. In this approach we initially need to generate a set of basis vectors using learning micro patches which are gathered from some initial contextual patches. Then, we generated 25 features for each contextual patch. We verified the feature vector through a simple clustering method. The results show the significant capability of features for separating the 8 classes of VHR satellite images. In terms of computation time, if we have already produced the set of ICA basis vectors, it averagely takes just 0.15 sec to generate a feature vector for a contextual patch.

We also proposed an approach to improve the set of basis vectors for the purpose of separating urban area from other classes. This approach is based on choosing the most important basis vectors from each class and bringing them together to make a new set of basis vectors.

# CHAPTER 8

# MIDDLE LEVEL TOPOGRAPHIC ICA FEATURES

In ordinary ICA, the components are assumed to be completely independent, and they do not necessarily have any meaningful order relationships. Notice that we usually talk about the dependency of two basis components, but we mean the dependency of their corresponding sources. The independence of ICA components can make it difficult to use the results of ICA, since we don't know the priority and importance of the components. On the other hand in practice, however, the estimated "independent" components are often not perfectly independent. Topographic Independent Component Analysis (TICA) is a method which uses these residual dependences to define a topographic order for the components [43].

Moreover, an important empirical motivation for this kind of dependency can be found in feature extraction. In ICA model, the components are supposed to be independent and there is no relation among different components. Therefore, we have to consider the set of all components as the descriptor. However, in TICA the dependency between components can be used to extract some middle level features and to reduce the dimension of feature vector. In this chapter we initially illustrate the principles of Topographic ICA, then we explain the procedure of generating the TICA basis vectors and finally we explain how to extract mid-level TICA features.

## 8.1 Principles of Topographic ICA

In TICA model, the observed variables $\mathcal{X}_{obs}$ are generated as a linear transformation of the sources $s_i$ just as in the basic ICA model. The point is that the sources are not assumed to be completely independent. We assume that there are dependencies between every two sources. These dependencies define the topography. Topography is known as the arrangement of the basis components in which the distance of two components is proportional to the dependencies of their corresponding sources (as an example see Figure 8.4(b)). Actually, the topography is defined by simultaneous

activations of two sources. In ICA it is supposed that variances $\sigma_i$ of the $s_i$ are the same. However, in TICA they are not constant and are assumed to be random variables which are generated using a specific model. After generating the variances, the variables $s_i$ are generated independently from each other. In other words, the $s_i$ are independent given their variances:

$$s_i = y_i \sigma_i \qquad (8.1)$$

In which $y_i$ is a zero-mean independent variable. Dependence among the $s_i$ is implied by the dependence of their variances. According to the principle of topography, the variances corresponding to near-by components are supposed to be correlated, and the variances of components that are not close should be almost independent.

To specify the model for generating the variances $\sigma_i$, we initially need to define the topography. The first step to do that is to determine a neighborhood function, $nb(i, j)$, like the following relation, which expresses the proximity between the i-th and j-th components:

$$nb(i, j) = \begin{cases} 1 & \text{if } d(i,j) \leq L \\ 0 & \text{if } d(i,j) \succ L \end{cases} \qquad (8.2)$$

Here, $d(i, j)$ is the distance of i-th and j-th components in the topography and $L$ defines the width of the neighborhood. That is, the neighborhood of the component with index $i$ consists of those components whose distances are less than $L$ components. Using the topographic relation $nb(i, j)$, many different models for generating the variances $\sigma_i$ could be used. A simple way is to define them in such a nonlinear function:

$$\sigma_i = \varphi(\sum_k nb(i, k) u_k) = (\sum_k nb(i, k) u_k)^{-1/2} \qquad (8.3)$$

Where $u_i$ are the "higher-order" independent components used to generate the variances $\sigma_i$, and $\varphi$ is the nonlinearity function.

The resulting topographic ICA model is summarized in Figure 8.1. Note that the two stages of the generative model can be expressed as a single equation, as follows:

$$s_i = \varphi(\sum_k nb(i, k) u_k) y_i \qquad (8.4)$$

Where $y_i$ is a random variable that has the same distribution as $s_i$ if $\sigma_i$ is fixed to

unity. The $u_i$ and the $y_i$ are all mutually independent.
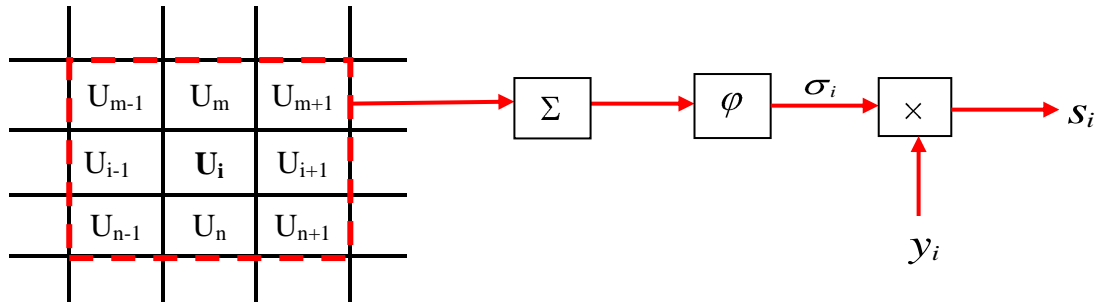


**Figure 8.1:** Generating the sources in a Topographic ICA model. The first step is to generate the higher order variables $u_i$. These variables are generated randomly. They are then mixed linearly inside their topographic neighborhoods. The mixtures are then given to the function $\varphi$ to produce the local variances $\sigma_i^2$. Components $s_i$ are then generated by multiplying $\sigma_i$ and a random variable $y_i$.

An important different between TICA and ICA is that TICA components are represented in a specified topography. In other words, the result of a TICA learning procedure is not only the basis components but also their positions in a specified topography. This means that we are not allowed to change the positions of the components because the topography of components specifies which components are statistically depended to each other. However, the result of an ICA learning procedure is just the basis components and their positions are not important at all. Usually, the TICA components are represented in a square topography. That is, if we have $n^2$ TICA components, they are represented in a *n\*n* topography. More details of the TICA model which is used in this paper are explained in [43].

When we apply ICA or TICA to the images, our observed data ($x_{obs}$) are considered as small patches gathered from big original images. Basis vectors resulted in the learning procedure will be also some small patches with the same size of observed patches. Actually, at the beginning of the learning procedure, we transform the matrix related to each image patch to a vector by placing the rows of matrix besides each other and after obtaining the basis components we transform them from a vector form to a matrix form, using the inverse logic. Before gathering small patches from one original big image, we usually remove the mean value of the original image and rescale it so that the variance of image pixels will be normalized to one. This can balance the roles and contributions of different original big images in obtaining basis components. In addition, as it is explained in chapter 6.2 we remove the mean value from each learning patch. This is done because of a theoretical assumption in a TICA (or ICA) algorithm. Later we perform the whitening pre processing step before starting the learning procedure.

Notice that just like the ICA learning procedure, small patches (micro patches) which are used in a TICA learning procedure differ from 200*200 contextual patches for which we are going to define the descriptors. In fact, we are not able to apply TICA directly for the 200*200 contextual patches because of the huge dimension of the data. So we need other small patches which can be used by a TICA algorithm.

## 8.2 TICA basis vector production

In this sub-chapter the technical details of obtaining the TICA components are outlined. Initially we have to choose the scale size and dimensionality of TICA system, in addition, the dimensionality and the neighborhood for corresponding topography:

### 8.2.1 Scale size of TICA system

For choosing the size of micro patches, we have to respect two considerations. First consideration is related to computational aspects. In chapter 6 we performed a study for the effect of the size of ICA components on the satellite image indexing. We demonstrated that the results would be improved when the size of components increases, however the computational problems would be rising exponentially as well. During our experiments for TICA we have found similar results. We found out that a TICA learning procedure with the 16*16 micro patches is about 8 times faster than a learning procedure with the 32*32 micro patches and 100 times faster than a learning procedure with the 64*64 micro patches. Over this size (64*64 pixels) the computation is almost impossible because of the huge dimension of the data. Regarding to all of these considerations we chose the size of basis vectors as 16*16 pixels.

### 8.2.2 Dimensionality of TICA components

In chapter 6, we studied the effect of the dimensionality of ICA components on the satellite image indexing. We demonstrated that the optimum *reduction factor* is usually between 0.08 and 0.14. In chapter 7 we used a set of 16*16 basis vectors with a reduction factor about 0.1 (25 the basis vectors).

For TICA we chose the size of basis vectors equal to 16*16 pixels. Therefore, initially, it seems to be reasonable if we choose the same number of components for TICA system. But here we decide to increase the number of components to 100. The reason is that in an ICA system, as it was mentioned, we don't have any relation or priority between the components. So we have to consider all of them as the features. Thus, if we use an ICA system with 100 basis vectors, the feature vector will be too long. However, for TICA system, as it will be explained in sub-chapter 8.3, it is possible to mix the components to make a very shorter feature vector. This is an important property for TICA because it permits us to increase the number of components which leads to a very better accuracy and in the same time we have a short feature vector.

### 8.2.3 Topography dimensions

Topography in a TICA system determines arrangement of the components. If two dimensions of topography are equal, dependencies among the components can be used more effectively for extracting middle level features. So, here we simply choose our topography as 10 components by 10 components.

### 8.2.4 Neighborhood dimensions

Neighborhood in a specific topography of TICA components specifies depth of dependence. In other words, it determines the size of region in which the components are supposed to be statistically depended. For example, if we have a TICA system with a 5*5 neighborhood, it means that the TICA components placed in every arbitrary region of 5*5 components in the topography are statistically depended (see Figure 8.2(b)).We cannot conclude that two components whose distance is out of the region 5*5 components are not depended but the dependency rapidly decreases when their distance gets larger. The TICA system with a neighborhood of 1*1 is equal to ICA system because it implies independence for every component (see Figure 8.2(a)).

We examined different neighborhoods from 3*3 to 7*7. Our experiments show that for neighborhood of 5*5 components, comparing with other cases, the extracted features are more capable to separate different classes of landscapes. Thus, we choose it as the neighborhood for our TICA system.

### 8.2.5 Pre-processing steps

The Pre-processing steps for TICA learning procedure is the same as ICA. The first step is to collect enough number of micro patches for learning procedure. We gather randomly 20000 micro patches from initial images. Then in next step we selected 10000 of them which have the higher **G** parameter as learning micro patches. As it was mentioned in chapter 6, before gathering learning micro patches from one original big image, we remove the mean value of the original image and rescale it so that the variance of image pixels will be normalized to one. Then, we also remove the mean value from each learning micro patch. The last pre-processing step is the centering and whitening which are explained in chapter 5.

### 8.2.6 TICA learning procedure

Now, we are ready to start the TICA learning procedure. The TICA learning rules are different from the ICA learning rules from some points of view because here we have a specific topography for the components. The details of TICA learning procedure are explained in [43]. Input of the learning procedure is the set of 10000 learning micro

patches selected by Gabor filters as the optimum samples and the output is the set of 100 TICA components represented in a specified 10*10 topography with the neighbourhood of 5*5 which are shown in Figure 8.2(b). Closer components are supposed to be statistically more depended. We see the similarity between close basis vectors in Figure 8.2(b).



**a**



**b**
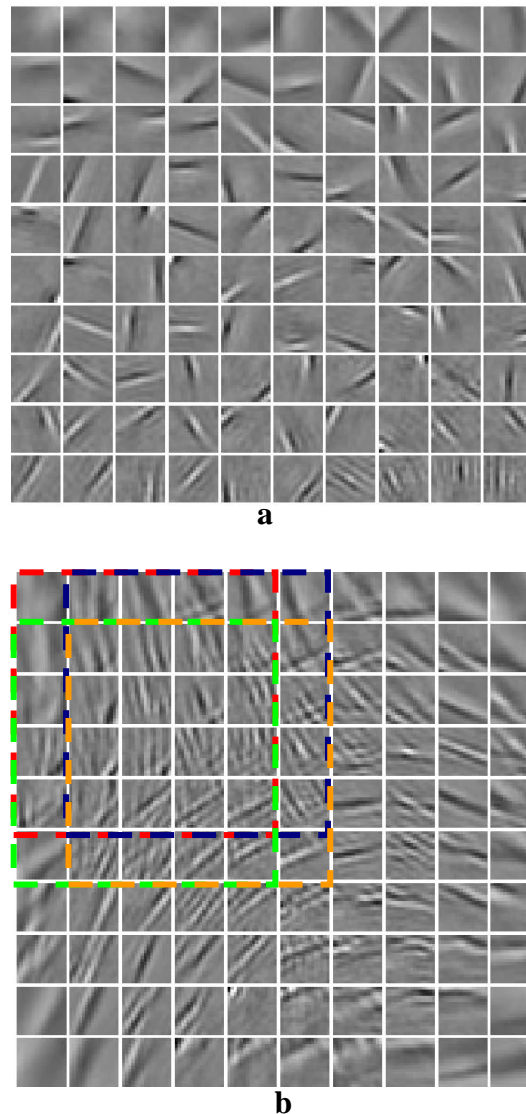
**Figure 8.2**:  **(a)** TICA system with the 1*1 neighborhood and 100 basis vectors. It is equal to an ICA system. Order and positions of basis vectors are not important. However, we sort them based on their mean frequencies from left to right and up to down. **(b)** TICA system with 100 basis vectors and a topography of 10 components by 10 components and also a neighborhood of 5 components by 5 components. Closer components are supposed to be statistically more depended. In particular, the components placed in a region of 5*5 components are statistically depended (For example, the region surrounded by red or green or orange lines).

To have a better comparison, we also preformed the learning procedure with 1*1 neighbourhood. The output is a set of 100 components which can be considered as ICA components. They are shown in Figure 8.2(a). For this case, orders and positions of components are not important. However, we sort them based on their mean frequencies from left to right and up to down. TICA basis vectors which are obtained in this sub-chapter will be used in next sub-chapter for extracting features from contextual patches.

## 8.3 Middle-level TICA features

In previous sub-chapter, the procedure of generating the TICA basis vectors is explained. In this sub-chapter we explain how to define the features for contextual patches using these TICA basis vectors. Figure 8.3 presents a general illustration of the TICA features generation procedure.
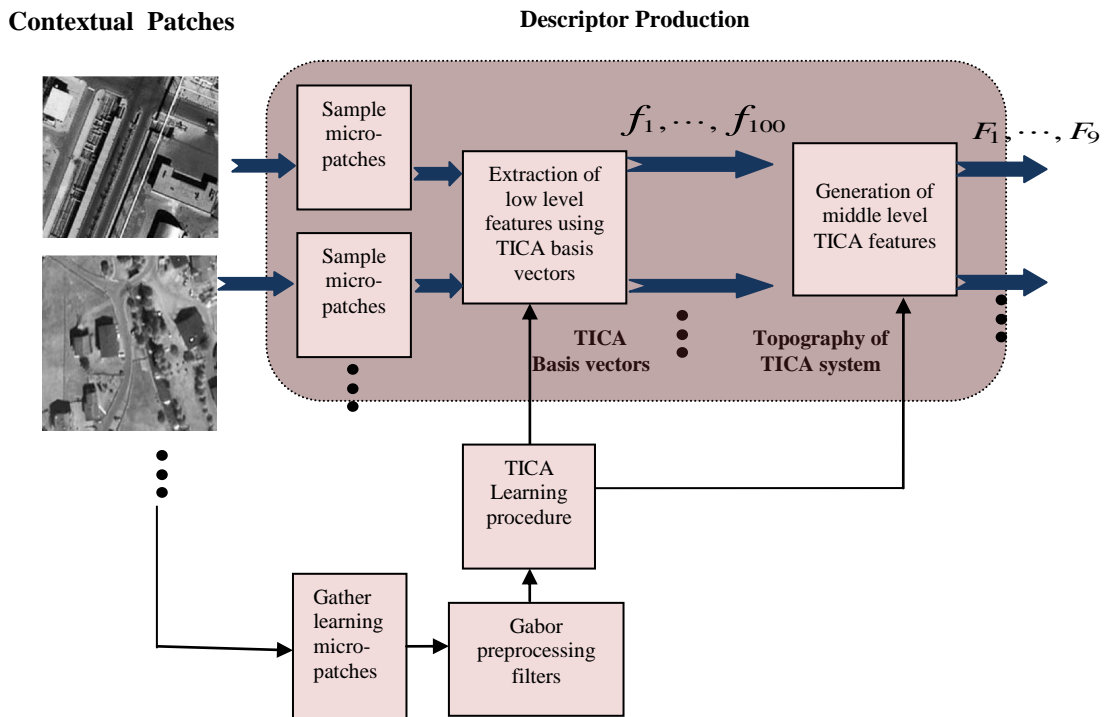


**Figure 8.3:** Illustration of different steps for generating the middle level TICA features for contextual patches. We gather a set of micro patches (with the size of 16*16) and apply a TICA learning procedure including a Gabor pre-processing step to produce the TICA components. To define a feature vector for each 200*200 contextual patch, we project its related micro patches into TICA components to obtain 100 low level features. Then using the dependencies among the TICA sources, we produce 9 middle level TICA features.

## 8.3.1 Low level TICA features generation

The procedure of the low level TICA features generation is very similar to the generating of ICA source based features which is explained in sub-chapter 7.2, except here we use the set of 100 TICA basis vectors instead of 25 ICA basis vectors. So each low level feature is obtained as :

$$f_n = \sqrt{\frac{1}{1000} \sum_{k=1}^{1000} s_n^2(k)} \tag{8.5}$$

In Which $n$ is a number between 1 and 100. Thus, for each contextual patch we obtain 100 low level TICA features.

## 8.3.2 Middle-level features definition

Now, using the low level features, we aim to define the middle level features for each contextual patch.
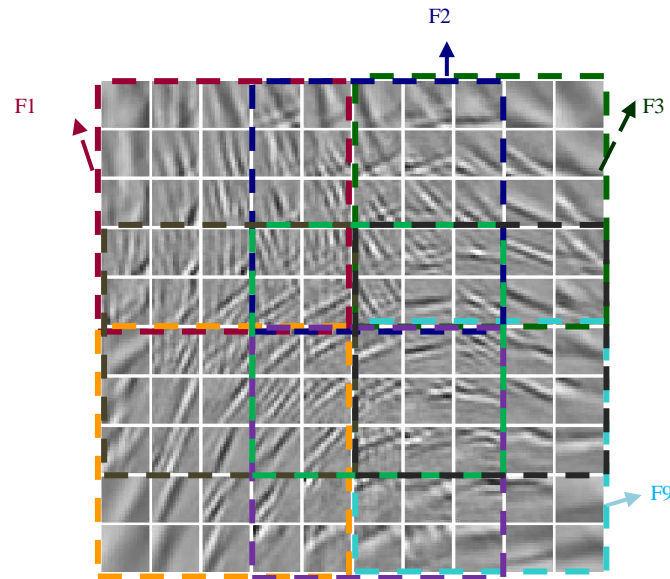


**Figure 8.4**: Defining middle level TICA features. We are able to combine a number of components that are supposed to be mutually dependent in a group, in order to provide a set of middle level features. In our TICA system, the neighborhood is 5*5. So, we average low level features related to a set of 5*5 components to produce one middle level feature. In our work we define 9 middle level features from 5 regions of topography which are specified in this figure.
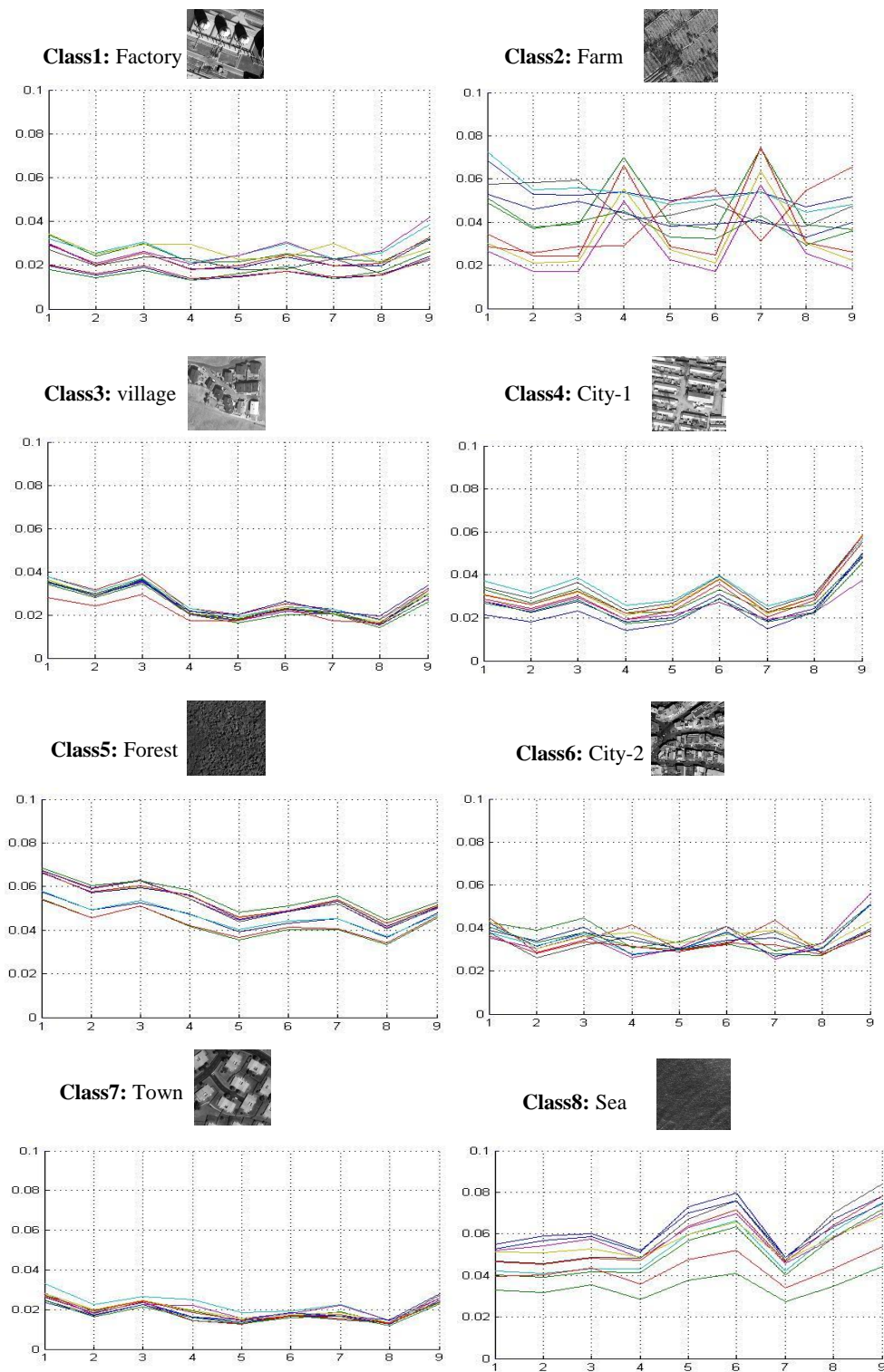
**Figure 8.5:** TICA mid level features for 8 classes of contextual patches. From each class, 10 contextual patches are selected for feature extraction.

If we use an ICA system (that is equal to TICA with the 1*1 neighbourhood), then we have to consider all low level features, individually, as our contextual features. The reason is that the components are supposed to be independent and there is not any kind of dependency or other relation between different components. Thus, for the case that we have 100 ICA basis components we have to consider 100 low level ICA features, as the descriptor of contextual patch. However, for the TICA decomposition of an image patch, we are able to combine a number of components that are supposed to be mutually dependent in a group, in order to make a set of middle level features. So, we are able to reduce the number of contextual features.

Depending on dimensionality of neighbourhood, we may choose number of components which must be grouped together to produce one middle level feature. In our TICA system, the neighbourhood is 5*5. So it is reasonable to bring together every set of 5*5 components. We simply define one middle level feature ($F_m$) by applying an average upon the 25 low level features related to the grouped component:

$$F_m = \frac{1}{25}\sum_{}^{25} f_n \qquad (8.7)$$

Here, the question is that the grouped components that produce one middle level feature must be selected from which part of topography. Other question is that how many middle level features are necessary to be extracted. The important point is that each low level feature must be contributed for producing at least one middle level feature. In addition, it is desired to define the minimum number of features for one contextual patch. We performed some experiments to estimate the optimum number of middle level features. Initially, we defined the minimum possible number of middle level features. We chose 4 regions at 4 corners of the topography for defining 4 middle level features. These regions cover entire surface of the topography. Then, step by step, we increased the number of features and evaluated the results of clustering based on these features. This clustering is explained in next sub-chapter. Our experiments showed that if we extract 9 middle level features as it is illustrated in Figure 8.4 then the capability of the middle level features set obviously grows up. So, our feature vector will be defined with 9 middle level features.

Figure 8.5 demonstrates sets of 9 middle level TICA features produced for 10 first contextual patches of each class of test set (see Figure 3.2). Although we reduced the number of features comparing with the ICA source based features explained in chapter 7, we visually may detect even more capabilities to separate different features. This can be concluded from the obvious differences between the feature vectors of different classes. This is validated also by results of clustering which is explained in next sub-chapter.

## 8.4 Simple clustering for evaluation

To evaluate our features we performed the same clustering which is explained in sub-chapter 7.3 but with the new middle level TICA features. The results of such clustering are summarized in Table 8.1.

**Table 8.1**: Results of simple clustering. We used the mid level TICA feature vectors .

| Clusters | Class1 Factory | Class2 Farm | Class3 Village | Class4 City-1 | Class5 Forest | Class6 City-2 | Class7 Town | Class8 Sea |
|---|---|---|---|---|---|---|---|---|
| **1** | **81.1** | 3.3 | 3.3 | 2.2 | 1.1 | 1.1 | 3.3 | 1.1 |
| **2** | 1.1 | **74.4** | 2.2 | 1.1 | 8.9 | 2.2 | 0 | 7.8 |
| **3** | 3.3 | 0 | **83.3** | 3.3 | 1.1 | 3.3 | 1.1 | 1.1 |
| **4** | 5.5 | 0 | 3.3 | **86.7** | 0 | 3.3 | 3.3 | 1.1 |
| **5** | 0 | 11.1 | 1.1 | 1.1 | **81.1** | 1.1 | 1.1 | 5.5 |
| **6** | 4.4 | 1.1 | 1.1 | 3.3 | 0 | **86.7** | 2.2 | 1.1 |
| **7** | 3.3 | 1.1 | 4.4 | 2.2 | 1.1 | 1.1 | **88.9** | 0 |
| **8** | 1.1 | 8.9 | 1.1 | 0 | 6.7 | 1.1 | 0 | **82.2** |

## 8.5 Conclusions

In this chapter we explained the approach for extracting TICA middle level features from contextual patches. We initially generated a set of TICA basis vectors then we obtained low level features similarly to the approach explained in sub-chapter 7.2. Finally we combined groups of low level features into 9 middle level features. We also verified the feature vector through a simple clustering method. The results show the significant improvement in comparison with the ordinary ICA features. In fact, here we benefit a set of 100 basis vectors (instead of 25 basis vectors for the case of ordinary ICA) and in the same time we generate only 9 features (instead of 25 features for the case of ordinary ICA).

In terms of computation time, if we have already produced the set of TICA basis vectors, it averagely takes 0.21 sec to generate a feature vector for a contextual patch which is a little more than the case of ICA but it is still enough fast.

# CHAPTER 9

# FEATURE EXTRACTION FROM ICA BASIS VECTORS: BAG OF WORDS MODEL

In chapters 7 and 8 we explained methods for extracting features from images using (Topographic) ICA sources. The other viewpoint for feature extraction is to consider the ICA basis vectors which are obtained for each image. Actually, we are going to work with the characteristics of basis vectors extracted from each image. This idea is explained in this chapter.

Firstly we illustrate the idea of obtaining basis vectors for one image. Then we propose an approach for extracting features from one image's basis vectors. This approach is based on the idea of *Bag of Words* model. In this approach we consider every contextual patch as a visual document and each basis vector as its word. Then we explain how to define features using the words of document.

## 9.1 Basis vectors of a contextual patch carry its signature

In the literature, the usual approach is to apply ICA learning procedure for all images or a set of images which are in one class. This helps us to find a set of new basis vectors for our data. Using this new basis vectors we are able to represents images with a fewer number of coefficient. Although this is usually considered as the main goal of ICA, it is possible to find other applications for the basis vectors which are obtained through an ICA learning procedure.

Actually, there is some information extracted from initial images in the obtained basis vectors. They tell us that in which directions the independent sources are distributed. That means if the set of basis vectors which are obtained for a set of images is given, then we could guess something about the structures of initial images. This is clearly observed in Figure 5.1 where the sets of obtained basis vectors for different classes are shown. Each set of basis vectors has its own properties. We could express this fact as: the signature of a class of images appears on its basis vectors.

**Class6:** City-2

**Class3:** Village



**Class8:** Sea

**Class5:** Forest
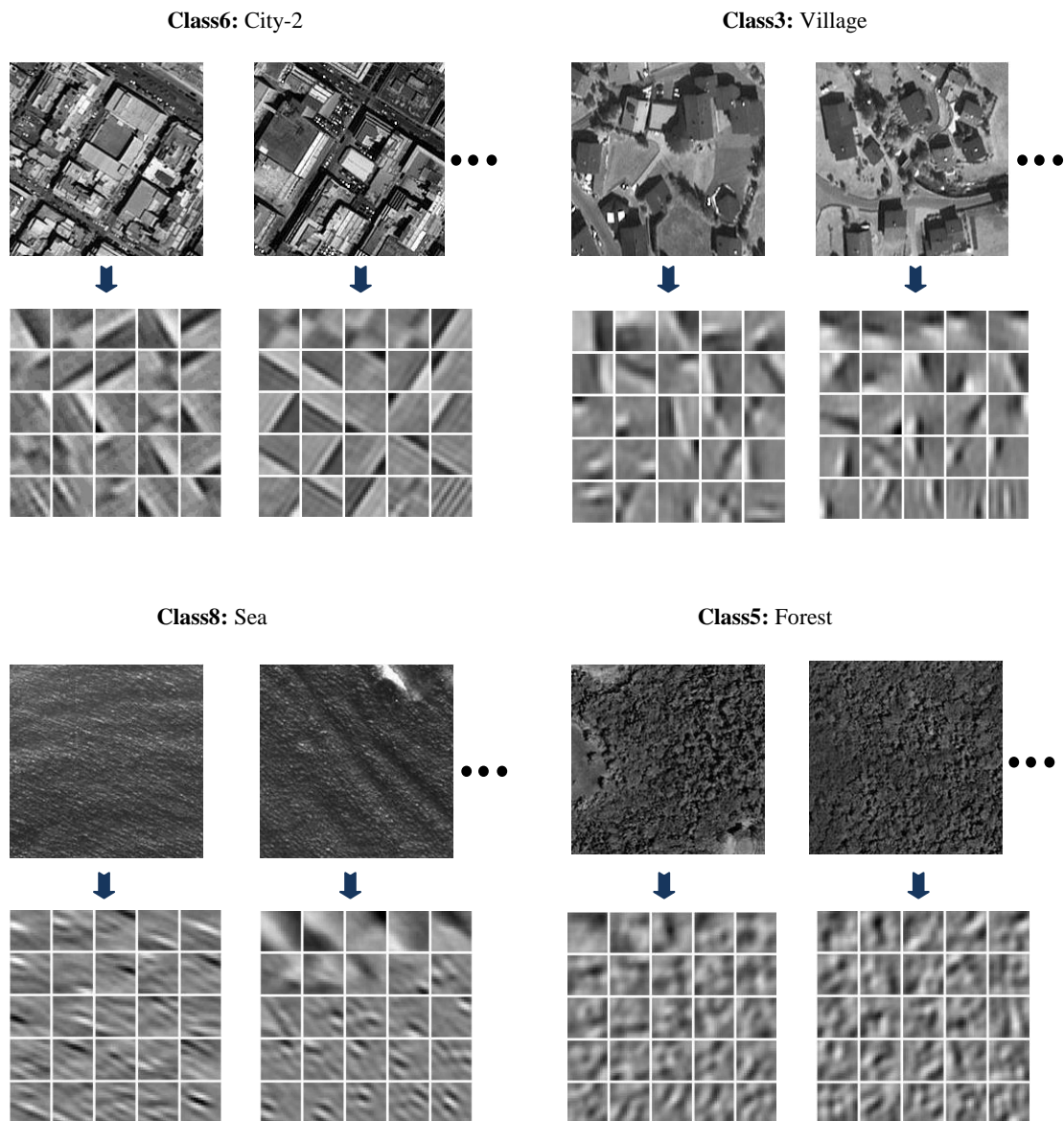


**Figure 9.1:** Obtaining the basis vectors for each contextual image patch, individually, instead of obtaining basis vectors for a set or a class of images. The basis vectors which are obtained from each image carry signatures of that image. Here, we selected 4 classes of landscapes and for each class we chose two contextual patches. Then we obtained 25 basis vectors with the size of 16*16 pixels for each 200*200 contextual patch. Notice that the basis vectors and the contextual patches are not presented with the same scale in the figure. As it is clear, the basis vectors of two contextual patches which are in one class have more similar characteristics comparing with the case that the two contextual patches come from different classes.

If this class includes just one image instead of several images, then the basis vectors carry the signature of that image. This is the basic idea of this chapter. Figure 9.1 shows the sets of basis vectors which are obtained for several contextual patches from different classes. We could see that when two contextual patches are in the same class, their basis vectors are more similar, comparing with the case that the two contextual patches are not in the same class.

## 9.1.1 Learning procedure for one contextual patch

The procedure for obtaining basis vectors for one contextual patch is the same as the procedure which is performed to obtain basis vectors for a set of contextual patches. Figure 6.1 shows the necessary steps for obtaining the basis vectors for a set of images. Here, the only difference is that we gather our learning micro patches from one image. It would be very useful if we apply the pre-processing step of Gabor filters to select the best micro patches for learning. This step was explained in sub-chapter 6.4

## 9.1.2 Choosing the dimensionality and the size of basis vectors

In Chapter 6 we studied the dimensionality and multi-scale behavior of an ICA system. We concluded that the basis vectors with the size of 16*16 and 32*32 pixels are more efficient according to their results and their computation times. In addition we mentioned that a reduction factor between 0.08 and 0.14 is suitable to determine the dimensionality of system. These conclusions are general facts when we apply ICA to VHR satellite images and can be used in this chapter because we are using an ICA system to extract features from a contextual patch.

Between the basis vectors with the size of 16*16 and 32*32, we expect that the second one gives more details about the characteristics of the contextual patch. However, here the time of learning is extremely important because we have to perform learning procedure for each contextual image patch. Table 9.1 shows the average computation times for obtaining a set of basis vectors for one contextual patch.

**Table 9.1**: Average times for obtaining a set of basis vectors for one contextual patch. For all cases, number of learning micro patches is 1000 and the number of iterations to learn is 100.

| Size of basis vectors | 8*8 | 16*16 | 32*32 |
|---|---|---|---|
| Number of basis vectors | 8 | 25 | 100 |
| Reduction factor | 0.125 | 0.098 | 0.098 |
| Time (sec) | 0. 39 | 0.69 | 5.21 |

The reduction factors for all cases are close to each other and in the normal range

(0.085-0.14). Also, for all cases, the number of learning micro patches is equal to 1000 and we perform the same number of learning iterations that is 100 learning iterations.

Actually, the computation time consists of two parts. The first part is related to the gathering micro patches, computing the covariance matrix and the PCA preprocessing. This part of time is a function of two factors, the number of learning micro patches and the size of micro patches. But, the second factor is more effective upon this part of the time, comparing with the second factor. The second part of computation time is related to the ICA learning procedure which is affected almost equally by the size and the number of learning micro patches.

Nevertheless, it is clear that between the basis vectors of the size of 16*16 and the size of 32*32, there is a big difference in their computation speeds. In fact, the first one is about 8 times faster than second one. But obtaining the basis vectors with the size of 8*8 is only 2 times faster than the size of 16*16.

The last point for this part is that we have to take care about the ratio of the size of contextual image patches and the size of basis vectors. If the size of basis vectors is very large then we hardly could gather proper learning samples from the contextual patch. For example, consider the case that we have 200*200 contextual patches and 64*64 basis vectors. We have to gather a large number of 64*64 learning micro patches from the 200*200 contextual patch. It seems that in this case many learning micro patches are the same or extremely similar to each other.

According to table 9.1, in this chapter we use a set of 25 basis vectors with the size of 16*16 for each 200*200 contextual image patch.

### 9.1.3 Number of learning micro patches

An important technical point for obtaining a contextual patch's basis vectors is the number of learning micro patches which must be gathered. The number of learning micro patches should be enough large so that the basis vectors will be obtained correctly. Actually the task of basis vectors is to represent the directions in which the independent sources' variations happen. If the number of micro patches which are gathered for learning procedure is not enough then the resulted basis vectors don't represent the correct directions for independent components' variations.

From the other side we have to take care about the time of obtaining the basis vectors, because we know that the learning procedure should be performed for every contextual patches and this can make the procedure of feature extraction very long.

Figure 9.2 gives an example of obtaining basis vectors for a contextual patch with different number of learning samples. For 100 or 250 learning micro patches, the basis vectors don't seem to be normal. For the case of 600 learning samples, the basis vectors are on the boundary of correct situation but still not reliable. Differences between the cases of 1000, 2000 and 4000 learning samples are very small but the time for the case of 1000 learning samples is about one third of the time for the case of 4000 learning samples. All of these lead us to choose the number of learning micro patches for obtaining basis vectors of one contextual patch as 1000 samples.

Of course, choosing the size and dimensionality of basis vectors and also the number of learning micro patches, is dependent on the user and the application, but here we chose the parameters such that they are suitable for a general purpose.

**Contextual patch**

**Number of learning samples = 100**
**Time (sec) = 0.21**

**Number of learning samples = 250**
**Time (sec) = 0.34**

**Number of learning samples = 600**
**Time (sec) = 0.52**

**Number of learning samples = 1000**
**Time (sec) = 0.71**

**Number of learning samples = 2000**
**Time (sec) = 1.23**

**Number of learning samples = 4000**
**Time (sec) = 2.12**
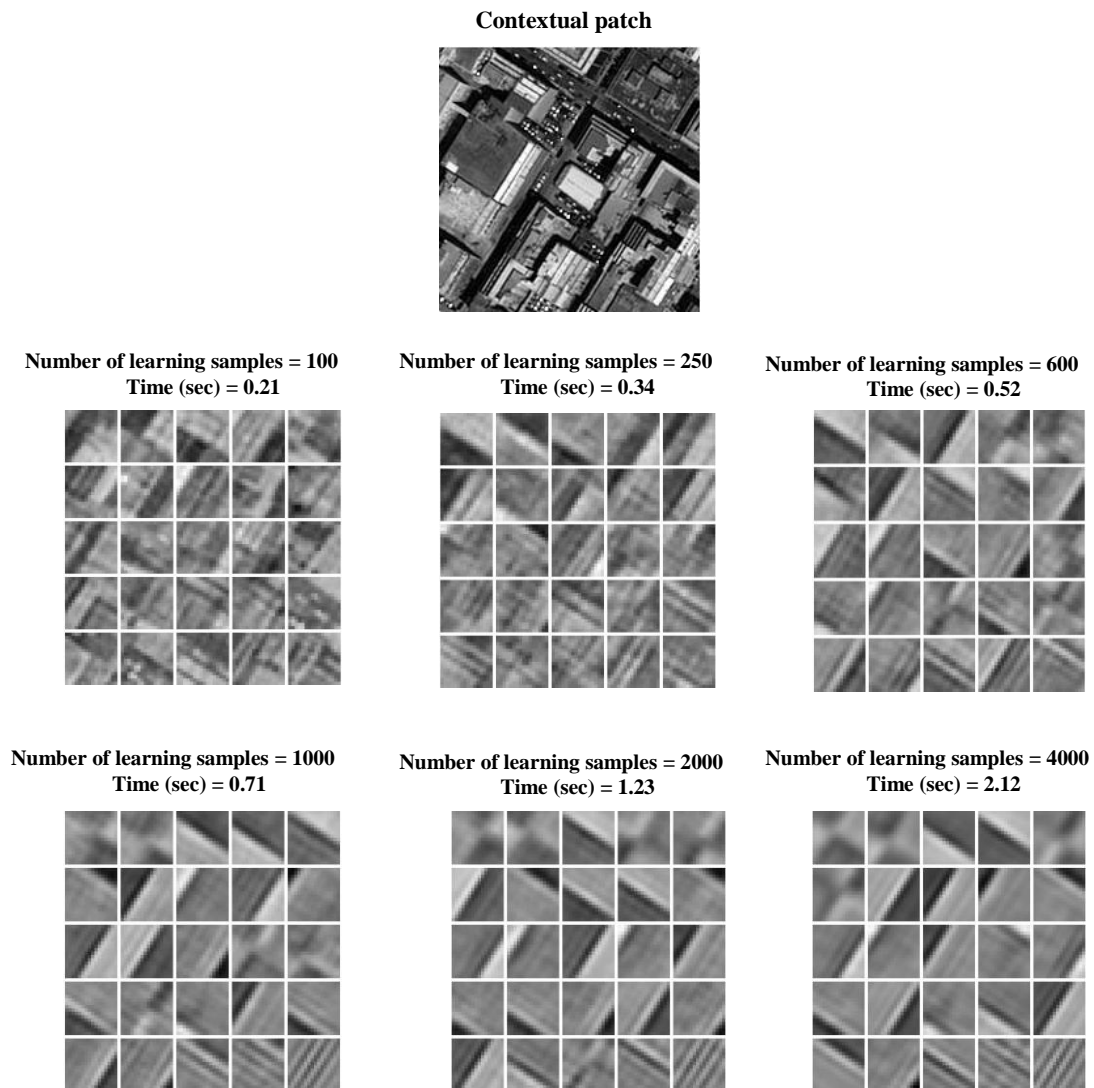
**Figure 9.2:** Obtaining the basis vectors for one contextual image with different number of learning micro patches. For all cases the number of obtained basis vectors is 25 (reduction factor = 0.098). The size of contextual patch is 200*200 and the size of basis vectors is 16*16 .

In this chapter, up to know we have shown that we are able to obtain the basis vectors for one contextual image patch instead of for a set of images. Also we have explained that these basis vectors carry the signatures of that contextual image patch. However, we may not able to take the set of basis vectors as the vector of features. The vector of features is a vector of numbers that each of them describes one characteristic of image. So we are supposed to extract features from the set of basis vectors which, themselves, carry the signatures of the contextual image patch.

In this chapter and next chapter we propose two approaches to extract features from

the basis vectors which are obtained from one contextual patch. One of these approaches is based on the Bag of Words model which is explained in this chapter and the other is based on detecting lines in basis vectors and characteristics of their lines which is outlined in next chapter.

## 9.2 Bag of words model

In text retrieval approaches, the Bag of Words (BoW) is a model for representing the documents as a set of words in which the arrangement of words is not important. Actually, in a BoW model a dictionary of all possible words is defined and each document is considered as a *bag* which contains a set of dictionary words. From a statistical point of view; each document is modeled as a vector which represents the occurrence histogram along the dictionary words. Here, we give an example to illustrate the BoW model. Suppose that we have a dictionary with 10 words: $\{W_1, W_2, W_3, \ldots, W_{10}\}$. Also suppose that we have a document (or text) as $W_5\ W_2\ W_3\ W_2\ W_7\ W_2\ W_3\ W_8$. This document can be represented by the vector: [0 3 2 0 1 0 1 1 0 0]. In other words, we count the number of repeats of every dictionary word in our document.

Recently the idea of BoW has been used for satellite images. See for example [14]. For using the BoW idea to characterize the images, the basic problem is to define an analogy between texts and images, i.e. defining visual words, documents and vocabulary. Our first goal in this part is to introduce an analogy between texts and images using ICA. This Bag of Words model for images is supposed to explain about:

- Definition of a visual document
- Definition of visual words for each document
- Definition of visual vocabulary
- Labeling each word of document by dictionary words
- Features definition

In the following we explain each part separately

### 9.2.1 Visual documents

Defining visual documents is not a critical step of our model. We can simply define visual documents as $l*l$ images. However, it is important to choose the size of visual documents ($l$) regarding to the size of visual words. In sub-chapter 9.2.2 it is explained how to extract visual words from each visual document using an ICA procedure. We have to choose a suitable size for visual documents so that visual words can be properly obtained for each visual document. If we take visual words with the size of $n*n$ pixels, we experimentally found that a minimum value for $l$ for providing sufficient number of proper learning micro patches for ICA procedure is about $5n$. Here, our contextual image patches with the size of 200*200 are considered as the visual documents.

## 9.2.2 Visual words for each document

Defining visual words for each document is one of the most important steps for introducing an analogy between text and image. Many methods propose that we simply gather some small micro patches from the visual document. See for example [14]. Here, we propose to obtain the basis vectors for a visual document (Here, a contextual patch) through an ICA learning procedure and consider them as the visual words of the document. We produce 25 basis vectors with the size of 16*16 for each contextual patch. Thus each visual document has 25 words.

## 9.2.3 Dictionary

Dictionary or vocabulary is known as the set of all possible visual words. Usually we have a lot of visual documents to be processed and the number of visual words extracted from the documents is enormous. Avoiding the computational problems, we have to limit the number of all possible words. For example, if we consider our test set of contextual patches, (see Figure 3.2) which contains 800 images, there will be totally 800*25=20000 visual words (basis vectors) which are extracted from all contextual images.

Number of dictionary words is dependent on the application. Dictionary is supposed to cover all possible forms of words, so we should take care about the number of dictionary words. Usually, number of dictionary words is determined experimentally. In our work we experimentally found out that if the number of words in the dictionary is about 60 ($\pm 20\%$ ), it is enough to describe all possible forms of visual words.

Here, we propose two approaches for defining dictionary. The first approach is based on performing a clustering among the visual words and the other is based on obtaining basis vectors for each class.

## 9.2.3.1 Obtaining the Dictionary from clustering

In the literature, the usual way for defining the dictionary is to perform a clustering on all words which are extracted from training documents. We collect all words from all training documents, and then a clustering method is applied to put a group of similar words in a cluster. For each cluster of words, one word which represents all words of this cluster (usually the center of the cluster), is introduced as a word of dictionary.

Here, from the *test set* of contextual patches which is explained in sub-chapter 3.4, we selected the first 10 contextual patches from each class as the training documents. That is, totally, 80 contextual patches are selected for defining dictionary. Then, a simple K-means clustering is applied on all the visual words which are extracted from 80 training contextual patches to place the 2000 visual words (25 words from each document) into 64 clusters. Finally, The 64 centers of clusters are considered as the words of our dictionary. The procedure of obtaining dictionary through a clustering on the visual words and the resulted dictionary, here called *Dictionary1*, are shown in Figure 9.3.
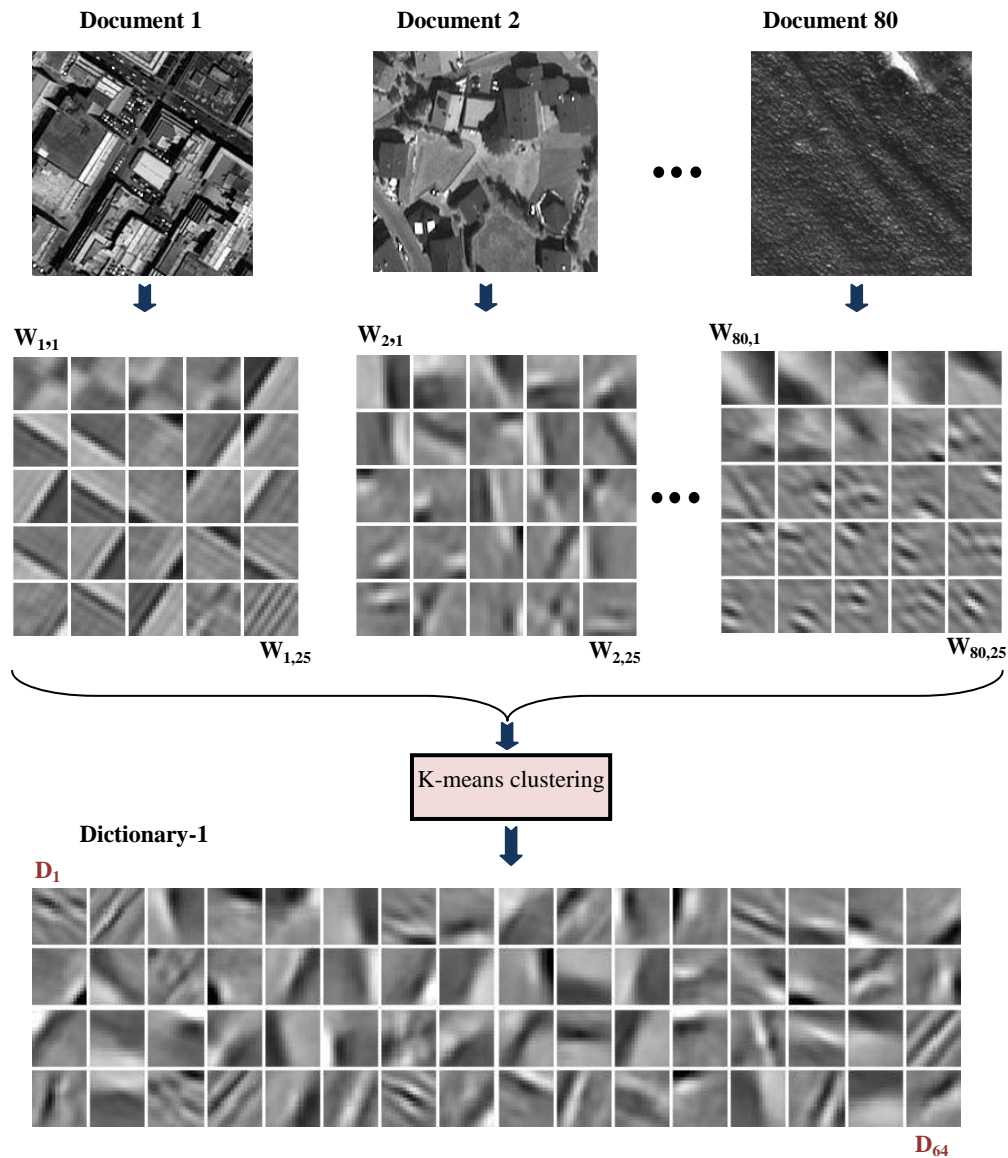
**Figure 9.3:** Producing the *Dictionary1* through a clustering on all the basis vectors which are obtained for the set of 80 contextual patches. From each class we selected 10 contextual patches for producing the dictionary. The 64 centers of clusters are considered as the words of our dictionary.

## 9.2.3.2 Obtaining the Dictionary from the basis vectors of classes

The other idea for defining the dictionary is explained in Figure 9.4. That is, for all of documents in each class we apply an ICA procedure and obtain basis vectors related to this class.

**Class 1:** Factory  **Class 2:** Farm  **Class 8:** Sea
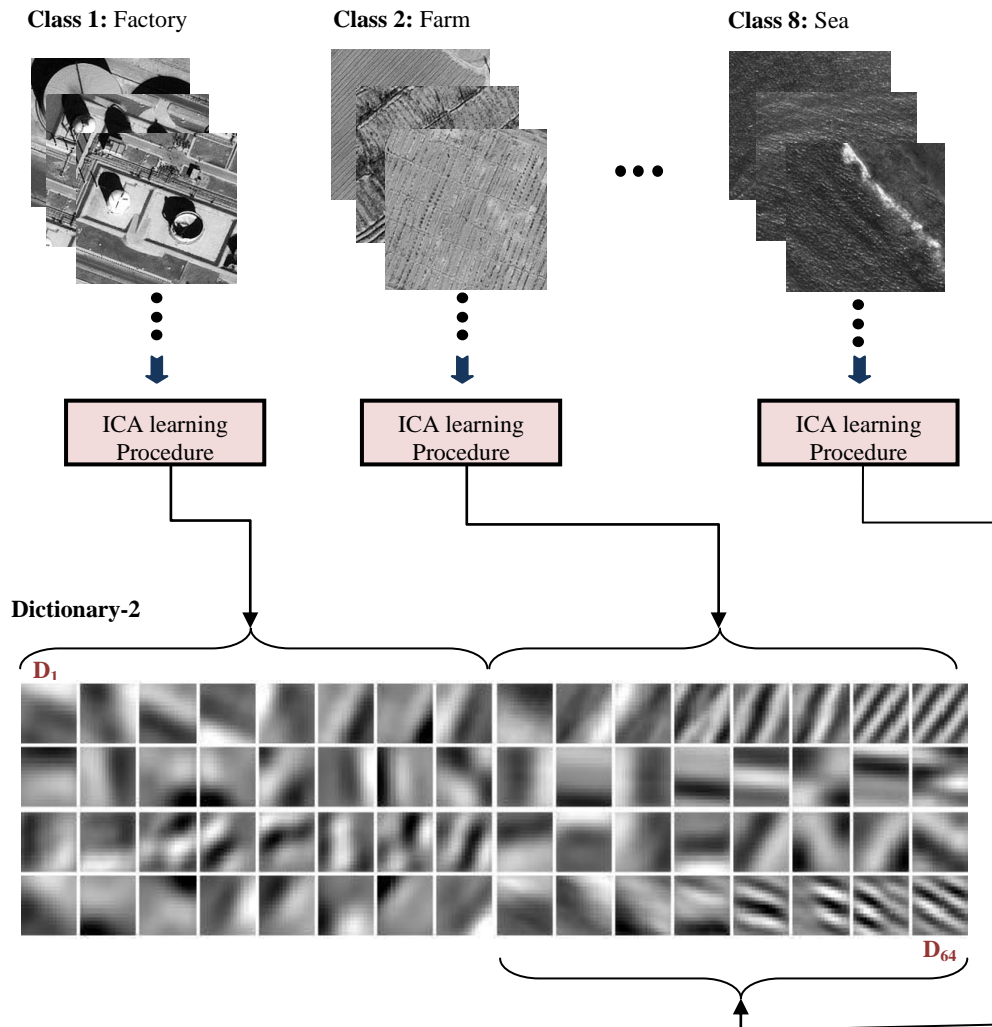


**Figure 9.4:** Producing the *Dictionary2* through obtaining basis vectors for all training documents in each class. For every class we selected 10 training visual documents and obtain 8 basis vectors using an ICA learning procedure. Dictionary is defined by bringing all of these basis vectors together.

As it is shown above, from each class we select the first 10 contextual patches as the training documents and through an ICA learning procedure we obtain 8 basis vectors for these 10 contextual patches. Therefore, totally we will have 64=8*8 basis vectors for all classes.

*Dictionary2* is defined by bringing all of these basis vectors together. This method for defining the dictionary is dependent on our primary knowledge about the of classes of images and also we are supposed to have some training samples from each class. This may limit us for using this kind of dictionary but on the other side could improve the results as we will see later.

## 9.2.4 Labeling each word of document by dictionary words

The normal approach in a Bag of Words model is to label each word of document by one dictionary word which is the most similar to the document word. The question here is how to find the most similar dictionary word. In other word, what is our criterion to measure the similarity between two words?

A variety of methods could be considered to measure this similarity. To avoid complexity in computation, we choose correlation as a simple criterion for similarity between two words:

$$C(w_1, w_2) = \frac{|w_1 w_2|}{|w_1||w_2|} \qquad (9.1)$$

In which, $W_1$ and $W_2$ are two arbitrary words and $C$, as their normalized correlation, measure the similarity between two words. For each word of document, we compute correlation between this word and all of dictionary words. Then we select the dictionary word which maximizes this correlation. When we label all words of a document using vocabulary words we can obtain histogram of a documents which shows the number of repeats for each dictionary word in the document. An example of labeling is shown in figure 9.5.

The histogram which is obtained for each visual document (contextual patch) can be considered as its vector of features.

## 9.2.5 Bayesian approach for classification

When we use the Bag of Words model, the approaches for classifying the documents is usually related to the Bayesian methods which are based on the probability of existence of a word in a document. Here we explain a simple Bayesian approach for classification of documents.

In this approach we calculate the posterior probability for existing of a document in one class, based on its primary probability and the probabilities of the words of documents.

Initially, for each word of dictionary, $D_k$, using the histograms of training documents, we calculate the probability of occurrence of $D_k$ in class $m$ :

$$P(D_k | class\_m) = \frac{num_{class\_m}(D_k)}{\sum_k num_{class\_m}(D_k)} \qquad (9.2)$$

In which, $num_{class\_m}(D_k)$ is the number of repeats of dictionary word $D_k$ in the histograms of all training documents of class $m$ and $\sum_k num_{class\_m}(D_k)$ is the number of repeats of all dictionary words in the same documents.

**Document *i***



**W$_{i,1}$**



**W$_{i,25}$**

Labeling by *Dictionary-1*

**D$_1$**



**D$_{64}$**
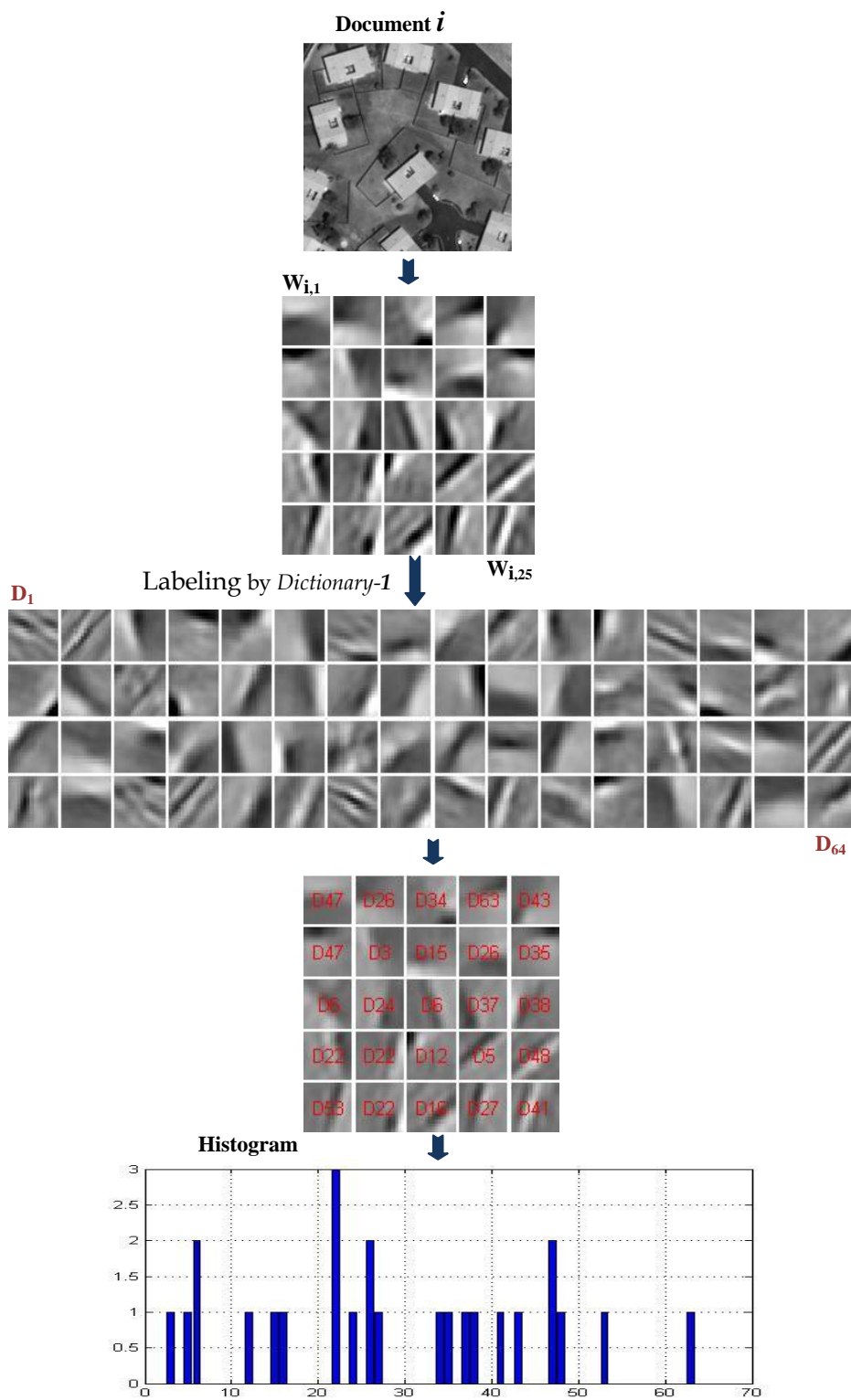


**Histogram**



**Figure 9.5:** Normal labeling. For each word of document, we compute correlation between this word and all of dictionary words and select the dictionary word which maximizes this correlation. Then we can obtain the histogram

Then we use the Bayesian theorem to calculate the posterior probability of existence of an arbitrary document **i** in class **m** if we know that the dictionary word $D_k$ exists in the document **i**:

$$P_i(class\_m | D_k) = \frac{P(D_k | class\_m)P_{i,1}(class\_m)}{\sum_m P(D_k | class\_m)P_{i,1}(class\_m)} \qquad (9.3)$$

In which $P_{i,1}(class\_m)$ is the primary probability of existence of the document **i** in class **m**. Since here we don't have any knowledge about the primary probabilities, we suppose that for all classes (for all **m)**, $P_{i,1}(class\_m)$ are the same. So the equation (9.3) can be changed to:

$$P_i(class\_m | D_k) = \frac{P(D_k | class\_m)}{\sum_m P(D_k | class\_m)} \qquad (9.4)$$

Finally we calculate the probability of existence of the document **i** in class **m** by adding all of these partial probabilities:

$$P_i(class\_m) = \sum_k P_i(class\_m | D_k) p_i(D_k) \qquad (9.5)$$

In which $p_i(D_k)$ is the probability of existence of dictionary word $D_k$ in the document **i** and is obtained from the histogram of the document **i** with the following equation:

$$p_i(D_k) = \frac{num_{doc\_i}(D_k)}{\sum_k num_{doc\_i}(D_k)} \qquad (9.6)$$

In which $num_{doc\_i}(D_k)$ is the number of repeats of dictionary word $D_k$ in the histogram of document **i** and $\sum_k num_{doc\_i}(D_k)$ is the total number of words in the document that here is equal to 25.

Using this approach we are able to put an arbitrary document in a class that gives the maximum probability in equation (9.5). We performed such a classification for the *test set* of contextual patches . The first 10 images from each class are considered as the training samples to obtain the dictionaries and also to calculate the probabilities $P(D_k | class\_m)$. Thus 720 images remain to be classified.

**Table 9.2**: Results of classification. We used the BoW model with the *Dictionary-1* and a normal labeling. Then we applied our Bayesian approach for classification.

| Clusters | Class1 Factory | Class2 Farm | Class3 Village | Class4 City-1 | Class5 Forest | Class6 City-2 | Class7 Town | Class8 Sea |
|---|---|---|---|---|---|---|---|---|
| 1 | **68.9** | 5.5 | 4.4 | 3.3 | 4.4 | 7.8 | 0 | 1.1 |
| 2 | 3.3 | **53.3** | 8.9 | 2.2 | 5.5 | 6.7 | 7.8 | 12.2 |
| 3 | 5.5 | 5.5 | **70** | 3.3 | 3.3 | 1.1 | 6.7 | 1.1 |
| 4 | 4.4 | 2.2 | 0 | **76.7** | 0 | 7.8 | 3.3 | 6.7 |
| 5 | 2.2 | 13.3 | 2.2 | 0 | **77.8** | 1.1 | 2.2 | 4.4 |
| 6 | 8.8 | 4.4 | 6.7 | 3.3 | 2.2 | **68.9** | 4.4 | 0 |
| 7 | 6.6 | 6.7 | 7.8 | 5.5 | 1.1 | 1.1 | **75.6** | 1.1 |
| 8 | 0 | 8.8 | 0 | 5.5 | 5.5 | 5.5 | 0 | **73.3** |

**Table 9.3**: Results of classification. We used the BoW model with the *Dictionary-2* and a normal labeling. Then we applied our Bayesian approach for classification.

| Clusters | Class1 Factory | Class2 Farm | Class3 Village | Class4 City-1 | Class5 Forest | Class6 City-2 | Class7 Town | Class8 Sea |
|---|---|---|---|---|---|---|---|---|
| 1 | **71.1** | 6.6 | 5.5 | 4.4 | 5.5 | 6.7 | 3.3 | 0 |
| 2 | 3.3 | **56.7** | 10 | 1.1 | 4.4 | 3.3 | 8.9 | 6.6 |
| 3 | 4.4 | 5.5 | **70** | 1.1 | 3.3 | 3.3 | 4.4 | 5.5 |
| 4 | 4.4 | 1.1 | 1.1 | **78.9** | 2.2 | 7.8 | 2.2 | 4.4 |
| 5 | 2.2 | 11.1 | 1.1 | 1.1 | **75.6** | 2.2 | 3.3 | 2.2 |
| 6 | 5.5 | 5.5 | 4.4 | 2.2 | 3.3 | **72.2** | 3.3 | 2.2 |
| 7 | 7.8 | 6.7 | 5.5 | 4.4 | 0 | 1.1 | **73.3** | 1.1 |
| 8 | 1.1 | 6.7 | 2.2 | 6.7 | 5.5 | 3.3 | 1.1 | **77.8** |

According to the results of classifications in tables 9.2 and 9.3 we could say that there is a little improvement when we use *Dictionary-2* instead of *Dictionary-1*. We experimentally found that when we are defining our dictionary by obtaining a set of basis vectors for each class (the case of *Dictionary-2*), if the number of basis vectors for each class is close to the number of visual words for each document (here 25), then we will have better results. In recent experiment, we chose the number of dictionary words as 64 and we had 8 classes, so the number of basis vectors for each class was selected as eight. If for example we had 4 classes instead of 8 classes then the number of basis vectors for each class would be 16 which is closer to the number of visual words for each document (25). Nevertheless, we chose the number of dictionary words as 64 in order to keep the similar conditions for the two dictionaries.

## 9.2.6 Improved labeling and features

As it was mentioned in sub chapter 9.2.4, the normal approach in a Bag of Words model is to label each word of document by only one dictionary word which is the most similar to the document word. Then, obtained histogram for each document could be considered as the feature vector of the document. An example of such feature vector is shown in Figure 9.5. As we see, the length of feature vector is the same as the number of dictionary words (here 64) and many of the features are zero because usually the number of words for each document (here 25) is less than the number of words in the dictionary. This fact that many of features are zero may leads us to make a conclusion, that is, the number of features is more than what is needed. On the other hand, we are not able to reduce the number of features because for each dictionary word we need one feature. However, maybe we can change the labeling and feature definition so that the vector of features would be more reach and efficient.

The idea is to measure the total similarity between one dictionary word with all of words which are extracted from an arbitrary document. In other words, if we have an arbitrary document $\mathbf{i}$, then for each dictionary word $\mathbf{D_k}$, we can define a feature $f_{\mathbf{i,k}}$ with the equation (9.7):

$$f_{i,k} = \underset{j}{mean}(\mathrm{C}(\mathrm{D_k}, \mathrm{W_{i,j}}))$$
(9.7)

In which $\mathbf{W_{i,j}}$ is the visual word number $\mathbf{j}$ of document $\mathbf{i}$ and $\mathrm{C}(\mathrm{D_k}, \mathrm{W_{i,j}})$ that measure the similarity between dictionary word $\mathbf{D_k}$ and document word $\mathbf{W_{i,j}}$ is calculated from equation (9.1). In fact, $f_{\mathbf{i,k}}$ expresses how much the dictionary word $\mathbf{D_k}$ is similar to the words of document . This is a kind of labeling because it is a criterion for the level of existence of dictionary word $\mathbf{D_k}$ in document $\mathbf{i}$ .

Figure 9.6 shows the improved features which are obtained for 10 first contextual patches of each class of the *test set* (see Figure 3.2). In this case we used *Dictionary-1* in our BoW model. Figure 9.7 shows the extracted features when we use *Dictionary-2* in our BoW model. As we see the feature vectors is completely different with the case that we use normal labeling (see Figure 9.5 for comparison) and we rarely find zeros in the feature vectors.
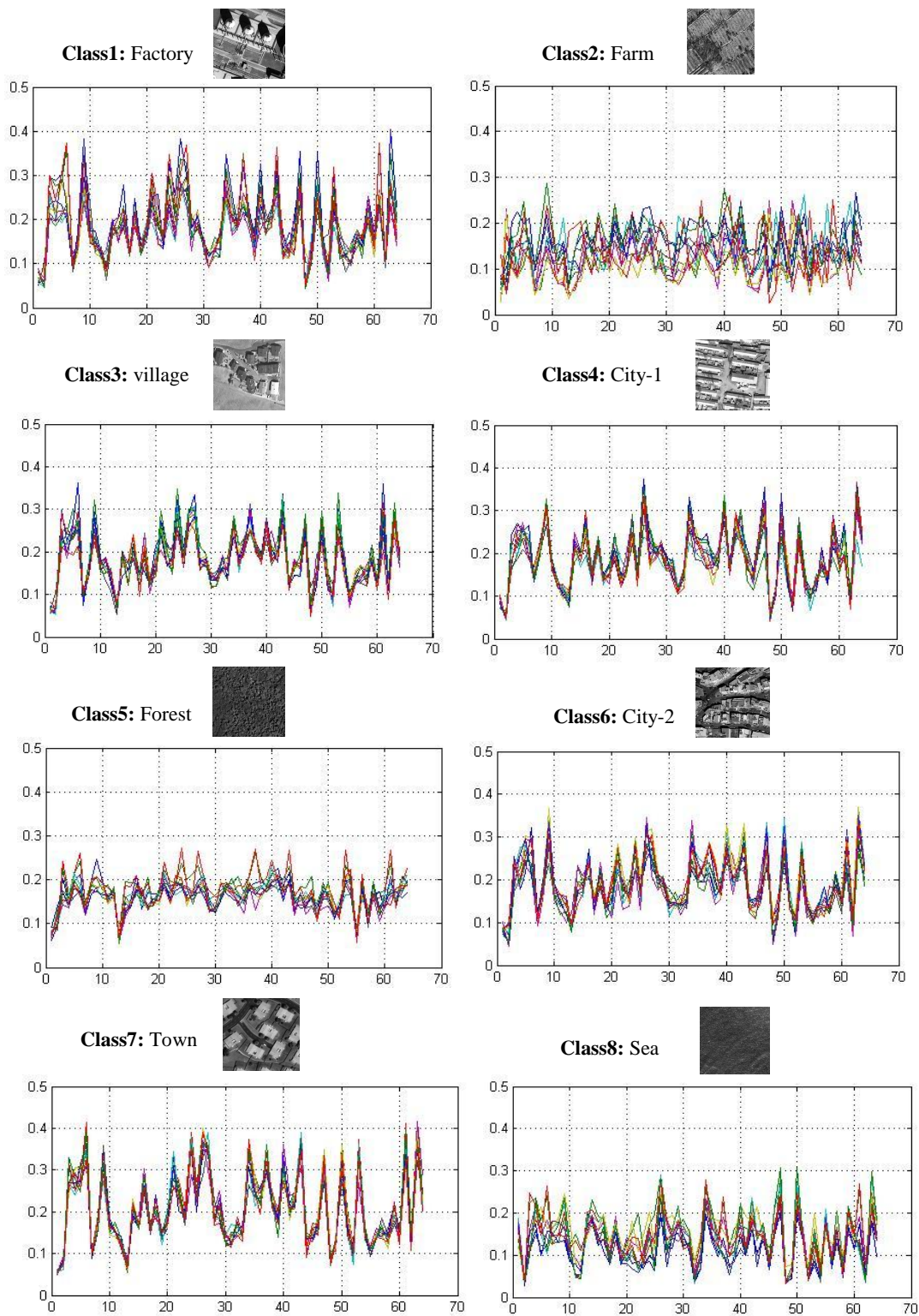
**Figure 9.6:** Improved BoW features. These features are obtained for 10 first contextual patches from each class of the *test set*. In this case we used *Dictionary-1* in our BoW model.
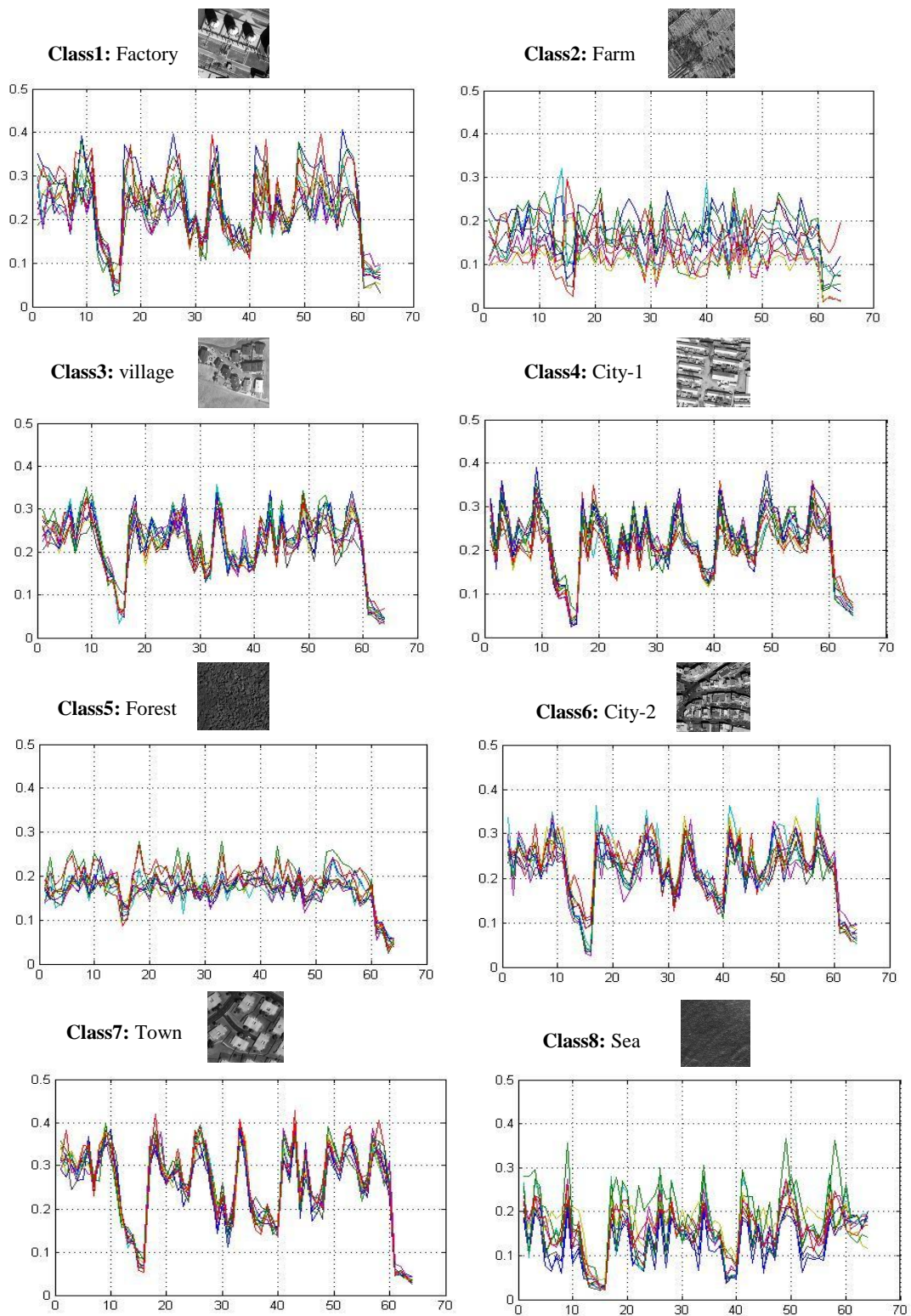
**Figure 9.7:** Improved BoW features. These features are obtained for 10 first contextual patches from each class of the *test set*. In this case we used *Dictionary-2* in our BoW model.

**Table 9.4**: Results of clustering. We used the BoW model with the *Dictionary-1* and improved labelling and features. Then we applied a simple clustering on the features

| Clusters | Class1 Factory | Class2 Farm | Class3 Village | Class4 City-1 | Class5 Forest | Class6 City-2 | Class7 Town | Class8 Sea |
|---|---|---|---|---|---|---|---|---|
| 1 | **71.1** | 6.7 | 2.2 | 3.3 | 4.4 | 7.7 | 3.3 | 1.1 |
| 2 | 4.4 | **55.6** | 10 | 1.1 | 5.5 | 3.3 | 10 | 6.7 |
| 3 | 5.5 | 6.7 | **71.1** | 2.2 | 4.4 | 4.4 | 4.4 | 5.5 |
| 4 | 3.3 | 1.1 | 2.2 | **77.8** | 1.1 | 6.6 | 2.2 | 3.3 |
| 5 | 5.5 | 10 | 3.3 | 2.2 | **76.7** | 1.1 | 2.2 | 2.2 |
| 6 | 5.5 | 7.8 | 4.4 | 3.3 | 3.3 | **72.2** | 4.4 | 2.2 |
| 7 | 4.4 | 6.7 | 5.5 | 4.4 | 1.1 | 2.2 | **72.2** | 0 |
| 8 | 0 | 5.5 | 1.1 | 5.5 | 3.3 | 2.2 | 1.1 | **78.9** |

**Table 9.5**: Results of clustering. We used the BoW model with the *Dictionary-2* and improved labelling and features. Then we applied a simple clustering on the features

| Clusters | Class1 Factory | Class2 Farm | Class3 Village | Class4 City-1 | Class5 Forest | Class6 City-2 | Class7 Town | Class8 Sea |
|---|---|---|---|---|---|---|---|---|
| 1 | **74.4** | 5.5 | 2.2 | 3.3 | 2.2 | 5.5 | 3.3 | 0 |
| 2 | 2.2 | **58.9** | 7.8 | 1.1 | 4.4 | 4.4 | 5.5 | 8.9 |
| 3 | 3.3 | 7.8 | **75.6** | 2.2 | 5.5 | 3.3 | 3.3 | 3.3 |
| 4 | 5.5 | 1.1 | 1.1 | **81.1** | 1.1 | 4.4 | 4.4 | 5.5 |
| 5 | 3.3 | 5.5 | 3.3 | 0 | **78.9** | 0 | 2.2 | 2.2 |
| 6 | 5.5 | 3.3 | 4.4 | 3.3 | 2.2 | **77.8** | 4.2 | 2.2 |
| 7 | 4.4 | 7.8 | 5.5 | 4.4 | 1.1 | 2.2 | **75.6** | 1.1 |
| 8 | 1.1 | 10 | 0 | 4.4 | 4.4 | 2.2 | 1.1 | **76.7** |

### 9.2.7 Simple clustering for evaluation

  To evaluate the improved BoW features we performed the same clustering which is explained in sub-chapter 7.3 but with the improved BoW features. The results are summarized in table 9.4 for the case that we use *Dictionary-1* in our BoW model and also in table 9.5 for the case that we use *Dictionary-2* in our BoW model. We can see a little improvement when we use *Dictionary-2* instead of *Dictionary-1*. Moreover, the results show that the new features are more effective comparing with the case that we use normal labeling and Bayesian approach for classification.


## 9.3 Conclusions

  In this chapter we proposed a Bag of words model to extract features from the basis vectors which are obtained for each contextual patch.
  The comparison of the tables 9.4 and 9.5 with the tables 9.3 and 9.2 shows that the improved BoW features are more effective comparing with the case that we use normal labeling and Bayesian approach for classification.
  For comparison with the features which are obtained from ICA sources we have to compare tables 9.4 and 9.5 with the table 7.2 . This comparison leads us to conclude that the BoW model is more efficient comparing the models which generate the features on the base of ICA sources. But we have to take care about the time of obtaining the features for each contextual image. The average time for obtaining the BoW features for a contextual patch is about 0.82sec and for the features based on ICA sources this time is about 0.15 seconds. Thus the BoW model is about 8 times slower. If we compare tables 9.4 and 9.5 with the results of mid-level TICA features in table 8.1 we can conclude that the mid-level TICA model is generally more effective in comparison with the BoW model. Moreover, we know that mid-level TICA model is absolutely faster than BoW model.
  Another point is the length of feature vector that can represent the complexity of model. The BoW model generates the feature vectors which are longer than feature vectors based on ICA sources and the feature vectors based on TICA model. So, we could say that the BoW model is generally more complex comparing with two previous models.

# CHAPTER 10

# FEATURE EXTRACTION FROM ICA BASIS VECTORS: LINE AND GRADIENT FEATURES

In chapter 9 we explained that one viewpoint for feature extraction is to consider the ICA basis vectors which are obtained for each image. In addition, we proposed the Bag of Words model to define features from the basis vectors of one contextual patch.

In this chapter we are going to propose another approach for extracting features from the basis vectors of one contextual image patch. This approach is based on detecting lines in the basis vectors and extracting features from the characteristics of these lines.

## 10.1 Lines and gradient as basic characteristics of basis vectors

As it was mentioned in chapter 9, the basis vectors which are obtained for one contextual image patch carry its signatures. In other words, the basis vectors of a contextual patch have some characteristics which may differ from the other contextual patch. In this chapter we aim to investigate about the most important characteristics in the basis vectors which can lead us to identify the original image's properties.

We are going to demonstrate that it is possible to estimate the structures on the basis vectors by some lines and extract features from these lines. We emphasize on the lines because as it was mentioned in previous chapters, our goal in the thesis is to define some descriptors for VHR satellite images especially for those who contain the geometrical objects. Naturally, lines play important roles for modeling the geometrical objects so the characteristics of lines (such as their lengths, their gradients and their angles) in the basis vectors could help us to characterize the initial image.

Figure 10.1 shows two examples of contextual patch and their basis vectors. As we see, in each basis vector we could find some structures which can be modeled by different lines. These lines are different from many points of view. Particularly, in the

basis vectors of one image (right) we see some lines which are mostly long, and intense from gradient point of view. Moreover, their angles are distributed in two narrow intervals around 45° and 135°. However, the other image has some basis vectors that present lines which are not very long and also not very strong from gradient point of view. In addition, the line angles are distributed differently. Most of them are distributed around 0° and 90°. This could make an idea for extracting features from the basis vectors of one image.
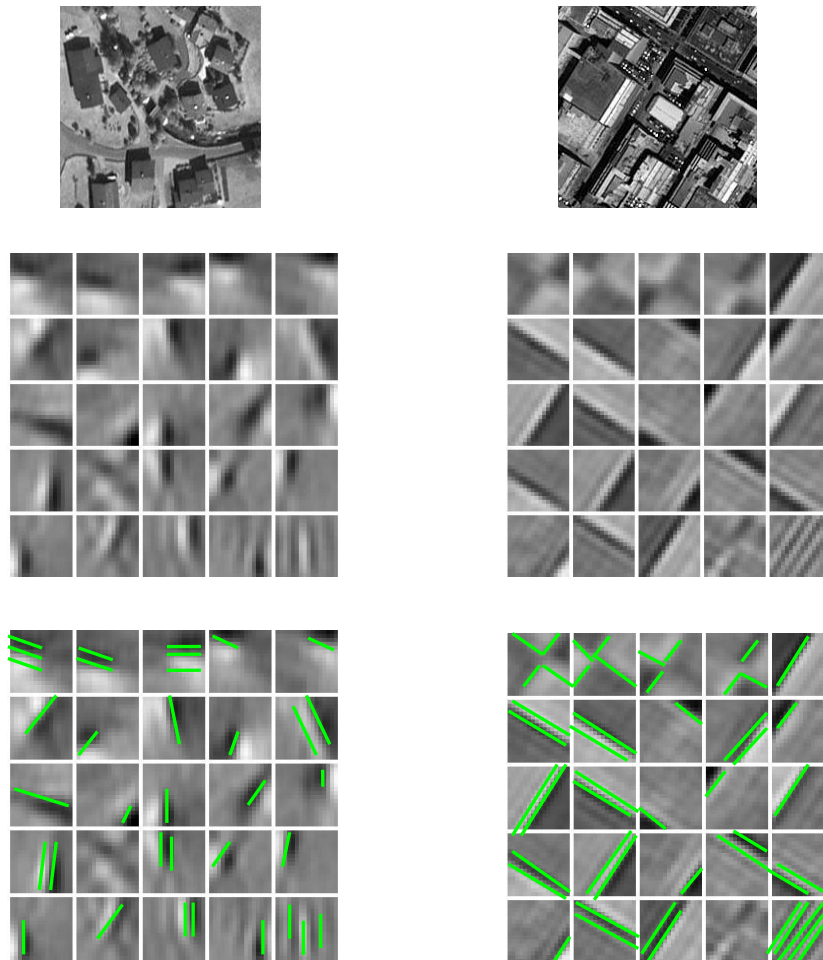


**Figure 10.1:** Basis vectors could be modelled by lines. The basis vectors of right image contain some lines which are mostly long and intense from gradient point of view. However, the basis vectors of left image have some lines which are shorter and weaker from gradient point of view. Moreover, the distributions of lines angles in two sets of basis vectors are different. This can be an idea for extracting features from the basis vectors of one image.

Our goal in this chapter is to extract features from the characteristics of lines which are found on the basis vectors of one contextual image. Thus, the first step is to

estimate lines on the basis vectors. In the literature, this is usually performed in two essential steps:

- Edge detection from the original images (here, basis vectors)
- Line estimation from the produced edges

In the following we explain the basic concepts of edge detection and line detection and introduce our methods for each of them.

## 10.2 Edge detection

The goal of edge detection is to find the boundaries of objects or segments in the images. Edge detection methods usually use a first order or second order derivation to measure the strength of an edge, then they compare it with a threshold to decide if it can be detected as an edge (thresholding), finally they edit the resulted edges to obtain one pixel thick edge elements (edge thinning). Different methods of edge detection exist. D. Ziou and S. Tabbone [51] presented a study about a number of different edge detection techniques.

The Canny edge detector [52] is known as one of the most effective methods for edge detection in the literature. The only disadvantage of Canny method is that it is not as fast as some other edge detectors such as Sobel edge detector. Since in our work the time of feature extraction is an important parameter, we preferred to use Sobel edge detector instead of Canny method. Sobel can be categorized as the first-order gradient method which is explained below.

### 10.2.1 Edge strength estimation based on first-order gradient

Gradient amplitude could be a criterion for the edge strength in the image. Actually, it can show the level of change between the neighbor pixels. For using the first-order gradient as a criterion for edge detection, we have to apply an operator to estimate the gradient function for a digital image. The easiest way is to use central differences to estimate the gradient:

$$G_x(x, y) = -\frac{1}{2}I(x-1, y) + 0 \times I(x, y) + \frac{1}{2}I(x+1, y) \qquad (10.1)$$

$$G_y(x, y) = -\frac{1}{2}I(x, y-1) + 0 \times I(x, y) + \frac{1}{2}I(x, y+1) \qquad (10.2)$$

In which, $I(x,y)$ is the gray scale level of original image at the point of $(x,y)$ and $G_x$ and $G_y$ are the gradient in the $x$ and $y$ directions. The equations (10.1) and (10.2) can be written as :

$$G_x(x, y) = [-\frac{1}{2} \quad 0 \quad \frac{1}{2}]I(x, y) = L_x I(x, y) \tag{10.3}$$

$$G_y(x, y) = \begin{bmatrix} \frac{1}{2} \\ 0 \\ -\frac{1}{2} \end{bmatrix} I(x, y) = L_y I(x, y) \tag{10.4}$$

The two matrix $L_x$ and $L_y$ (here vectors) are called the *gradient operators*. Here, each of gradient operators considers only two pixels in the neighborhood of corresponding pixel (left and right or up and down). If we consider other pixels which are in oriented neighborhoods of corresponding pixel, we expect that the resulted gradients would be more reliable. For example, the well-known Sobel operator is based on the following filters:

$$L_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad L_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{10.5}$$

Given such estimates of first-order derivatives, we are able to obtain the gradient magnitude and the angle of gradient for each point. The magnitude of gradient can be computed as:

$$|\nabla G| = \sqrt{G_x^2 + G_y^2} \tag{10.6}$$

While the gradient orientation can be estimated as

$$\theta = \arctan(\frac{G_y}{G_x}) \tag{10.7}$$

## 10.2.2 Thresholding and edge thinning

After computing the gradient estimation, we have to apply a threshold, to decide if it is enough strong to be considered as an edge. If we take a low value for threshold, more edges will be detected. Simultaneously, the result will be sensitive to the noise. On the other side a high threshold may lose some edges. The usual approach is to consider the hysteresis thresholding. That is, using multiple thresholds to find edges.

Next step after thresholding is edge thinning which is a technique used to remove the unwanted points on the edges and, if it is necessary, add some points so that we will have one pixel thick edges.

The rules of a thinning procedure are dependent to the needed accuracy and the forms of edges which are expected. However, if the thinning is applied carefully, it removes all the unwanted points and normally results in one pixel thick edge elements. So, we will have sharp and thin edges that lead to greater efficiency in line detection algorithms.

Usually a thinning method is performed based on the number of neighborhoods of each pixel in the edge. It also verifies if the pixel can be considered as the connection of two or more different edges or not. In the thinning procedure sometimes we add one pixel and sometimes we replace one pixel to its neighborhoods to fill the gaps inside the edges. The complicated cases are when we have a pixel in the edge with more than two pixels in its neighborhood. We have to determine if this pixel is a linking point of two or more different edges or it should be eliminated from the set of edge pixels.

## 10.3 Line estimation

After detecting edges in one image, we are going to estimate them with lines. The result of edge detection is a Binary Matrix (BM) that contains only ones for the pixels which are detected as the edge pixels and zeros for the pixels which are not detected as the edge pixels. In the literature there are methods for detecting lines from the binary matrix. For example, Hough is a very well known approach which detects all the possible lines inside the image [50]. However, most of these methods, including Hough, are not suitable for our purpose. Their results usually contain many unwanted lines because they are designed to detect all the possible lines. So, some pixels may belong to several lines.

Here, we propose our own approach for line detection which is more efficient than Hough method. This approach starts with detecting the 3-pexel lines as the primary kernel of the lines. Then we enlarge each 3-pixel lines by adding the pixels from two sides until the direction of line does not change. Below, we explain this approach of line detection.



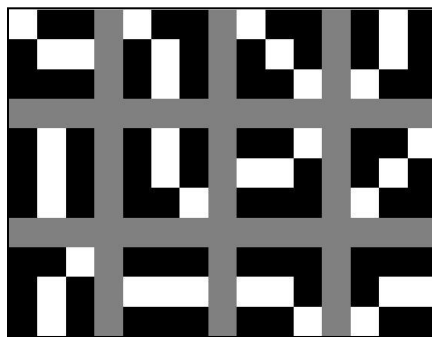**Figure 10.2:** Twelve possible forms for a 3-pixel line. These are possible forms if we suppose that in edge thinning step, we removed all unwanted pixels and we have one pixel thick edge pieces.
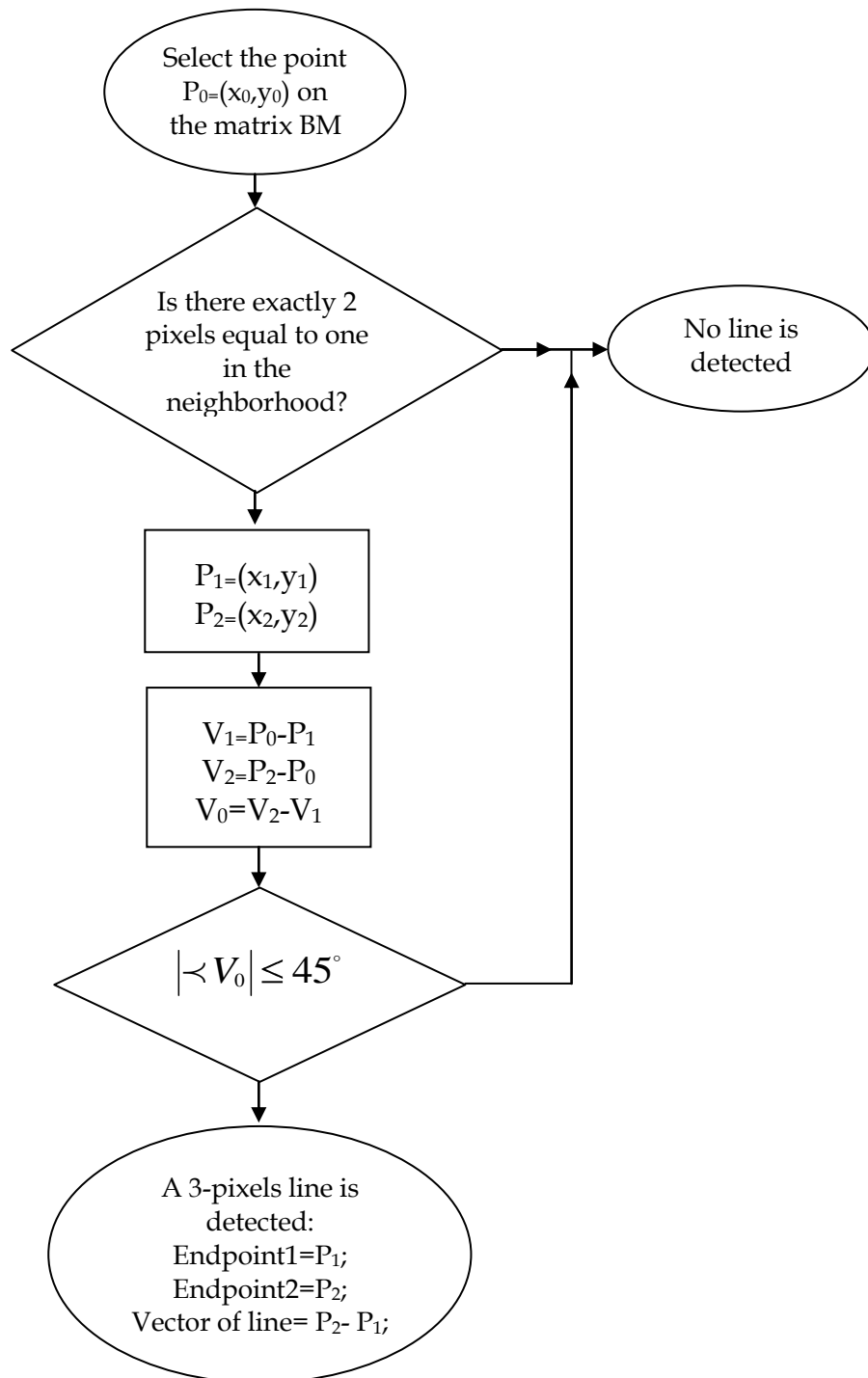
**Figure 10.3:** Algorithm of 3-pixel lines detection. The input is one pixel on the binary matrix which specifies the edges. If a 3-pixel line is detected, the output is the 2 endpoints and the vector which shows the direction of the line.

### 10.3.1 Three-pixel line detection

The first step of our line detection approach is to detect 3-pixel lines. We suppose that in edge thinning step, we removed all unwanted pixels and we have one pixel thick edge pieces. So, all the possible forms for 3-pixel lines are 12 forms which are illustrated in Figure 10.2.

Given a pixel on the binary matrix, resulted from edge detection, the objective of a 3-pixel line detector is to verify if this pixel with two other pixels in its neighborhood could from a 3-pixel line as the 12 possible forms shown in Figure 10.2. This can be performed with the algorithm shown in Figure 10.3.

As we see the 3-pixel lines detector as a function, gets a pixel on the BM, the binary matrix, which is resulted from the edge detection. This matrix contains only ones for the pixels which are detected as the points of edges and zeros for the pixels which are not detected as the points of edges. Then, the function verifies if there are exactly two pixels equal to one in the neighbourhood of selected pixel. If so, it obtains the two vectors corresponding to the selected pixel and its 2 neighbour pixels and verifies if the difference between 2 vectors is not greater than 45°. If so, a 3-pixel line is detected and the output of detector is the two endpoints of the line and the line's vector which is obtained from the difference of two endpoints. This vector shows the direction of detected line. Using this algorithm we are able to detect all the 12 possible forms for 3-pixel line which are shown in Figure 10.2

### 10.3.2 Enlarging the three-pixel lines

Once a 3-pixel line is detected, we can enlarge it from its two sides until the direction of line does not change significantly. We designed an algorithm for enlarging a 3-pixel line. The idea is to add one pixel to the end of the line if it does not change the direction of the line. Actually, for each added pixel to one end of the line we could detect a 3-pixel line that terminates with the added pixel. We accept the added pixel as new point of the line if the vector of this 3-pixel line does not differ significantly from the initial 3-pixel line, which is the kernel of corresponding line. Otherwise the line is terminated from the corresponding side. Our criterion for difference between the directions of two lines is defined based on the interior product of their vectors:

$$d(\mathbf{v}_i, \mathbf{v}_j) = \left| \frac{\mathbf{v}_i \mathbf{v}_j}{\|\mathbf{v}_i\| \|\mathbf{v}_j\|} \right| \tag{10.8}$$

In which, $\mathbf{V_i}$ and $\mathbf{V_j}$ are the vectors of two lines which are obtained by subtracting the initial point of the line from the its end point. $d$ is zero if the two lines are orthogonal and increases to one if the directions of two lines are exactly the same. In the following we explain the details of our algorithm for enlarging the 3-pixel lines:

1- If there are pixels that don't belong to one line, select one of them: $\mathbf{P_0}$

2- Apply *3-pixels line detection* to verify if $\mathbf{P_0}$ is the central point of a 3-pixel line. If the answer is *no* go to the step 1 and if the answer is *yes*, for this line ($\mathbf{L_0}$) find the two endpoints ($\mathbf{Endpoint_1}=\mathbf{P_1}$, $\mathbf{Endpoint_2}=\mathbf{P_2}$) and the line-vector ($\mathbf{V_0}$)

3- If $\mathbf{Endpoint_1}$ is already selected for another line, from the side of $\mathbf{Endpoint_1}$ the line cannot be enlarged. So go to the step 6. If $\mathbf{Endpoint_1}$ is not yet selected for another line, go to the step 4.

4- Apply *3-pixels line detection* to verify if $\mathbf{Endpoint_1}$ is the central point of a 3-pixels line. If the answer is *no* the line cannot be enlarged from the side of $\mathbf{Endpoint_1}$. So, go to the step 6. If *yes,* for this line ($\mathbf{L_1}$) find the two endpoints and the line-vector ($\mathbf{V_1}$).

5- One of the endpoints of $\mathbf{L_1}$ must be already a pixel of the line. We take the other endpoint as $\mathbf{P_3}$. Verify If $\left|\dfrac{\mathbf{V_1}\mathbf{V_0}}{\|\mathbf{V_1}\|\|\mathbf{V_0}\|}\right| \geq 0.8$ and $\mathbf{P_3}$ is not yet selected as a point of another line. If the two conditions are satisfied, take $\mathbf{P_3}$ as the new pixel of the line and change the endpoint of the line to $\mathbf{P_3}$ ($\mathbf{Endpoint_1}=\mathbf{P_3}$) and go back to the step 4. If one of two conditions is not satisfied, the line cannot be enlarged from the side of $\mathbf{Endpoint_1}$. So, go directly to the step 6.

6- If $\mathbf{Endpoint_2}$ is already selected for another line, from the side of $\mathbf{Endpoint_2}$ the line cannot be enlarged. So go to the step 1. If $\mathbf{Endpoint_2}$ is not yet selected for another line, go to the step 7.

7- Apply *3-pixels line detection* to verify if $\mathbf{Endpoint_2}$ is the central point of a 3-pixels line. If the answer is *no* the line cannot be enlarged from the side of $\mathbf{Endpoint_2}$. So, go to the step 1. If *yes,* for this line ($\mathbf{L_2}$) find the two endpoints and the line-vector ($\mathbf{V_2}$).

8- One of the endpoints of $\mathbf{L_2}$ must be already a pixel of the line. We take the other endpoint as $\mathbf{P_4}$. Verify If $\left|\dfrac{\mathbf{V_2}\mathbf{V_0}}{\|\mathbf{V_2}\|\|\mathbf{V_0}\|}\right| \geq 0.8$ and $\mathbf{P_4}$ is not yet selected as a point of another line. If the two conditions are satisfied, take $\mathbf{P_4}$ as the new pixel of the line and change the endpoint of the line to $\mathbf{P_4}$ ($\mathbf{Endpoint_2}=\mathbf{P_4}$) and go back to the step 7. If one of two conditions is not satisfied, the line cannot be enlarged from the side of $\mathbf{Endpoint_2}$. So, go to the step 1.

As we see, before accepting a pixel as the new point of a line, we always verify if it is not yet selected for another line. Therefore, each pixel can belong only to one line. This approach has strong links with the Piecewise Linear Approximation (PLA) methods. The purpose of PLA is to approximate digitized curves (edges) using consecutive line segments. The advantage of our approach is that we could have more control for detecting each line individually. For example, it is possible to add an option to change the criterion for the direction of line when the line is getting larger. However, in PLA usually the goal is to minimize a total error for all parts of curves (edges) and we do not have access to each line separately.

## 10.4 Feature extraction from basis vectors using lines properties

We are going to extract features from the characteristics of lines which are found on the basis vectors of one contextual image. The procedure of feature extraction starts with finding lines inside the basis vectors. Then, we compute three characteristics of each line: the length, the angle and the average of gradient of the line. For each characteristic we divide its entire possible interval to some smaller intervals which are called bins and put every characteristic of a line in its corresponding bin. The number of elements which are placed in a bin can be considered as a feature. Below we explain the details of feature extraction.

### 10.4.1 Finding lines inside the basis vectors

After obtaining the basis vectors for a contextual patch, next step of feature extraction is to find the possible lines inside the basis vectors. This is performed using the edge detection and line estimation algorithms which are explained in sub chapters 10.2 and 10.3. We use the Sobel method for computing the gradient in our edge detector. There is a parameter that adjusts the threshold of gradient magnitude for the edges to be detected. We have to adjust this parameter such that we detect all necessary edges and avoid unwanted edges.

In addition in our line estimator we are able to determine the minimum length of lines which must be detected. We decided to detect the lines which are longer than 4 pixels. Figure 10.4 shows the results of edge detector (Binary Matrix) and line estimators which are applied for the basis vectors of a contextual patch.

### 10.4.2 Length, Gradient and angle as the important line properties

The length, the average of gradient magnitudes and the angle of a line is considered as its most important characteristics. The length of the line is the total number of its pixels. The angle of line is computed based on the vector which connects the first pixel and last pixel of the line. The average of gradient magnitudes of all pixels on detected line indicates the intensity of change which is represented by detected line.
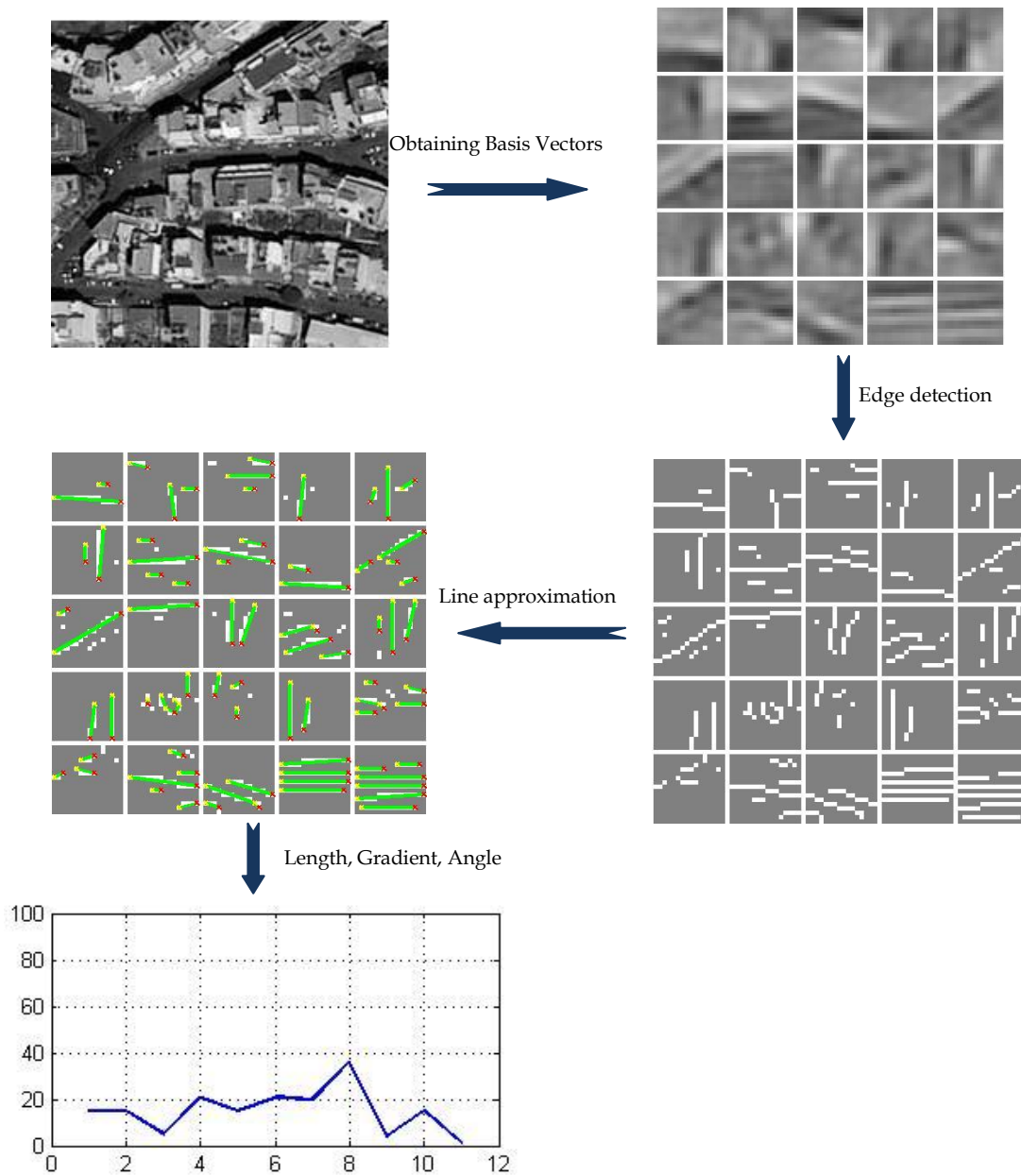
**Figure 10.4:** Feature extraction from basis vectors of a contextual image patch, using their lines properties. There are several steps, such as obtaining basis vectors, edge detections and line approximation. Finally, for each line, we put each of its characteristics (length, average of gradient magnitude and angle) into the corresponding bin.

### 10.4.3 Number of elements in a bin as feature

For each characteristic we divide its entire possible interval to some smaller intervals which are called bins. Consider the length of a line for example. We suppose that this line may be *short* or *medium* or *long* or *very long*. These are labels of bins for the length of the line whose intervals must be defined. For this case, we define *short* line as the line whose length is between 4 to 6 pixels, *medium* line as the line whose length is between 7 to 10 pixels, *long* line as the line whose length is between 11 to 13 pixels and *very long* line as the line whose length is greater than 14 pixels. For the average of gradient magnitude we define 3 intervals for *week*, *strong* and *very strong* lines.

Also, for the angle of lines we define 4 intervals for the lines whose angles are around 0°, 45°, 90° or 135°. In fact, our 4 intervals for the angle are defined as $0^{\circ} \pm 22.5^{\circ}$, $45^{\circ} \pm 22.5^{\circ}$, $90^{\circ} \pm 22.5^{\circ}$ and $135^{\circ} \pm 22.5^{\circ}$.

Consequently, here we have a total number of 11 bins for the length (4 bins), the average of gradient magnitude (3 bins) and the angle (4 bins). For each line, we put each of its characteristics (length, average of gradient magnitude and angle) into the corresponding bin.

The number of elements in each bin could be considered as a feature of our contextual patch. So we have a feature vector with 11 features whose first 4 features correspond to the length, the next 3 features correspond to the average of gradient magnitude and the last 4 features correspond to the angel. For example, the first feature of a contextual patch indicates the number of *short* lines in its basis vectors, the sixth feature indicates the number of lines in the basis vectors that are *strong* from the gradient point of view and the eleventh feature indicates the number of lines in the basis vectors that are around 135°. In the following, Table 10.1 contains the descriptions and the intervals for each feature.

**Table 10.1**: Features, their description and their intervals

| Feature | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---------|---|---|---|---|---|---|---|
| Description | *Short,* | *Medium,* | *Long,* | *Very Long* | *Weak* | *Strong* | *Very Strong* |
| Interval | $4 \le L \le 6$, | $7 \le L \le 10$, | $11 \le L \le 13$ | $14 \le L$, | $\lvert G \rvert < 0.11$ | $0.11 \le \lvert G \rvert < 0.14$ | $\lvert G \rvert \ge 0.14$ |

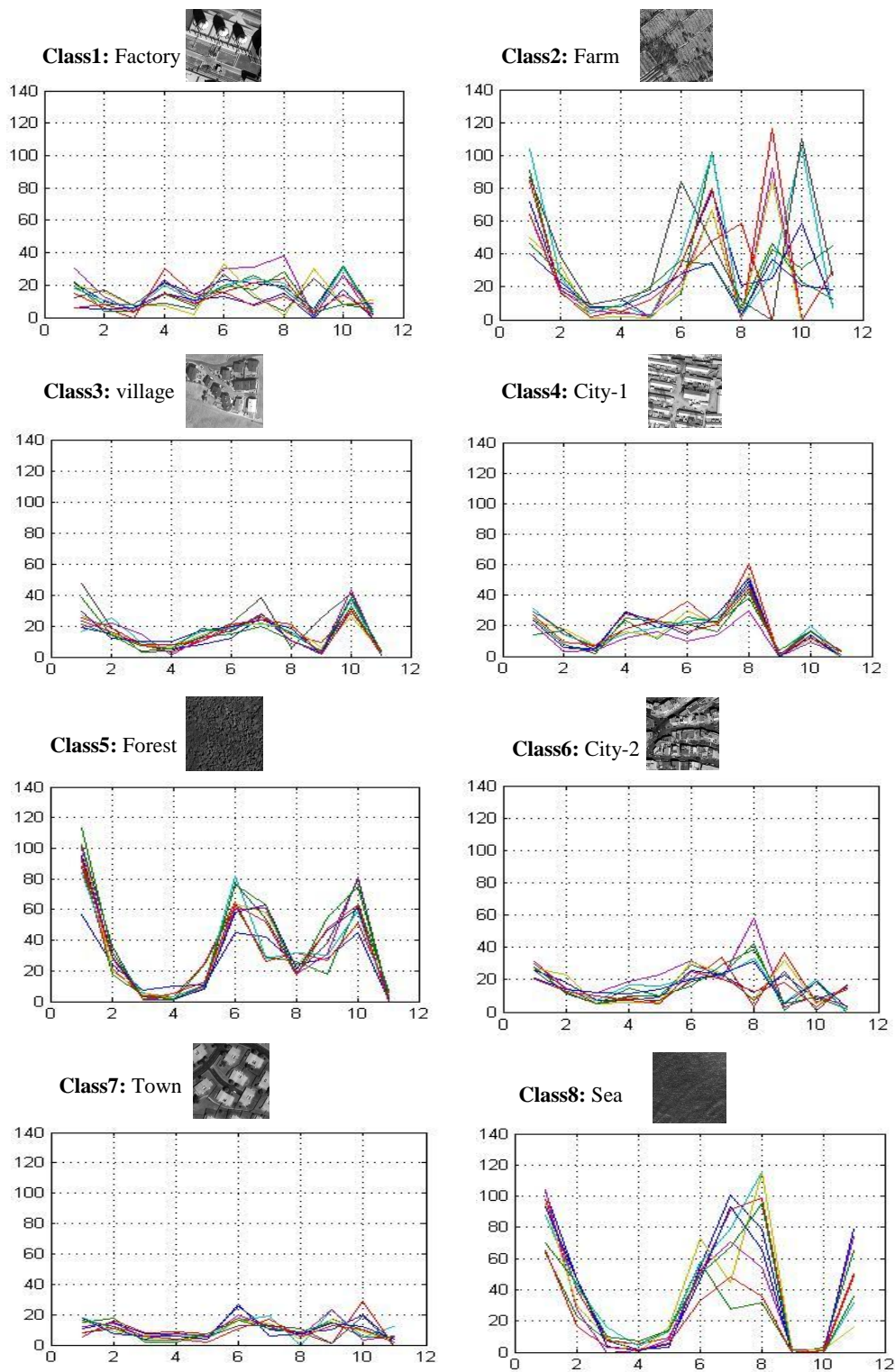| Feature | 8 | 9 | 10 | 11 |
|---------|---|---|----|----|
| Description | *Horizontal* | *Oblique1* | *Vertical* | *Oblique2* |
| Interval | $0^{\circ} \pm 22.5^{\circ}$ | $45^{\circ} \pm 22.5^{\circ}$ | $90^{\circ} \pm 22.5^{\circ}$ | $135^{\circ} \pm 22.5^{\circ}$ |

**Figure 10.5:** Feature vectors which are obtained from the characteristics of lines detected on the basis vectors of contextual patches. These features are obtained for 10 first contextual patches from each class of the *test set*.

Figure 10.5 shows examples of these feature vectors which are obtained for 10 first contextual patches from each class of the *test set*. We could see that the differences between the features which are obtained for different classes are clearer comparing with previous feature vectors which are defined in previous chapters. Specially, the group of classes that contain geometrical objects (class1, class3, class4, class6, class7) could be easily separated from the other group of classes that mostly contain natural landscapes (class2, class5, class8).

## 10.5 Simple clustering for evaluation

To evaluate our features we performed the same clustering which is explained in sub-chapter 7.3 but with the new features. The results of such clustering are summarized in Table 10.2.

**Table 10.2**: Results of simple clustering. We used the feature vectors which are defined from the properties of lines in the basis vectors of a contextual patch.

| Clusters | Class1<br>Factory | Class2<br>Farm | Class3<br>Village | Class4<br>City-1 | Class5<br>Forest | Class6<br>City-2 | Class7<br>Town | Class8<br>Sea |
|---|---|---|---|---|---|---|---|---|
| 1 | **83.3** | 3.3 | 5.5 | 2.2 | 1.1 | 1.1 | 2.2 | 0 |
| 2 | 0 | **73.3** | 0 | 1.1 | 10 | 1.1 | 0 | 8.9 |
| 3 | 3.3 | 0 | **85.6** | 3.3 | 0 | 3.3 | 4.4 | 0 |
| 4 | 6.6 | 1.1 | 3.3 | **84.4** | 0 | 5.5 | 3.3 | 0 |
| 5 | 0 | 12.2 | 1.1 | 0 | **81.1** | 0 | 0 | 7.8 |
| 6 | 4.4 | 1.1 | 2.2 | 5.5 | 0 | **87.8** | 3.3 | 0 |
| 7 | 2.2 | 0 | 2.2 | 3.3 | 0 | 1.1 | **86.7** | 0 |
| 8 | 0 | 8 | 0 | 0 | 7 | 0 | 0 | **83.3** |

## 10.6 Conclusions

In this chapter we proposed an approach for defining features from the properties of line (length, average of gradient magnitude, angle) which are detected inside the basis vectors of a contextual patch. For this goal, we introduced a new method for line estimation in the images.

According to Table 10.2, we see an obvious improvement in the results comparing with all previous methods, except the TICA features. Even if we compare these results with the results of TICA features, somewhere we see a little improvement. Specially,

here we could see a particular property. That is, the group of classes that contain geometrical objects (class1, class3, class4, class6, class7) could be easily separated from the other group of classes that mostly contain natural landscapes (class2, class5, class8). However, we could say that generally the results of this method are in the same level or a little bit better than the results of TICA approach.

But we should take care about the time of computation. In recent method, the average time for obtaining features for a contextual patch is 0.96 sec that is about 8-9 times more than the time which is needed for extracting TICA features.

The last point is the length of feature vector that can represent the complexity of model. We have a vector of 11 features which is less than all previous method except the TICA approach.

# CHAPTER 11

# IMAGE DESCRIPTOR BASED ON LINE SEGMENTS

In chapter 10 we explained the idea of extracting features from the characteristics of line which are detected inside the basis vectors of one contextual image. In this chapter, we are going to use a similar idea for the lines existing inside the contextual image patches. We use the principal idea of ICA basis vectors to develop our feature extraction approach.

## 11.1 Motivation

In previous chapters we explain that ICA basis vectors provide a new space to represent the images. Every window with the size of basis vectors on a contextual image patch could be decomposed onto the set of basis vectors. This is shown in Figure 11.1.
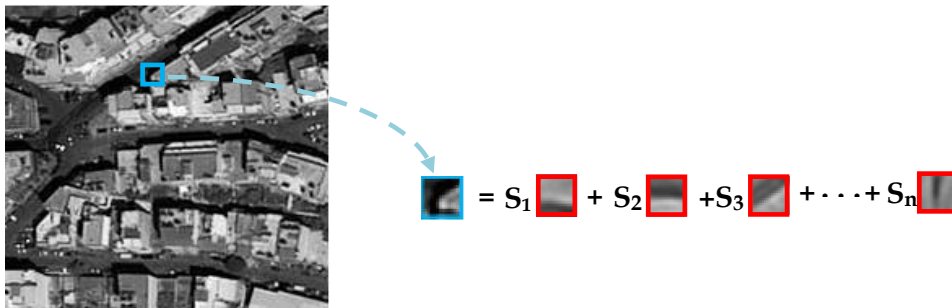


**Figure 11.1:** The windows with the size of basis vectors on the contextual patch could be considered as the linear combination of basis vectors whose important parts could be modeled by different types of lines.

This means that every window on the contextual patch could be considered as a linear combination of basis vectors. In chapter 10 we showed that the most important part of each basis vector could be modeled by different types of lines. Actually, in most of basis vectors, one or several lines compose the principal structure on the basis vector and their other parts don't present important information.



**Figure 11.2:** Many pixels in a square basis vector that presents a line are not necessary. We are able to represent this line by a smaller segment.

We know that the basis vectors, normally, have the square shapes. This means that for representing a line with the length of $n$ pixels we need a basis vector of the size of $n*n$ pixels, approximately. However, this line could be represented by a segment of $n*d$ pixels. In which $d$ is usually between 3 and 5 depending on the width of the line. This means that many pixels in the basis vector that present such a line are not necessary. This is shown in Figure 11.2



$$f_1*(short \text{ line}) + ...+ f_k*(long \text{ line}) + ...+$$
$$f_j*(weak \text{ line}) + ...+ f_i*(strong \text{ line}) + ...+$$
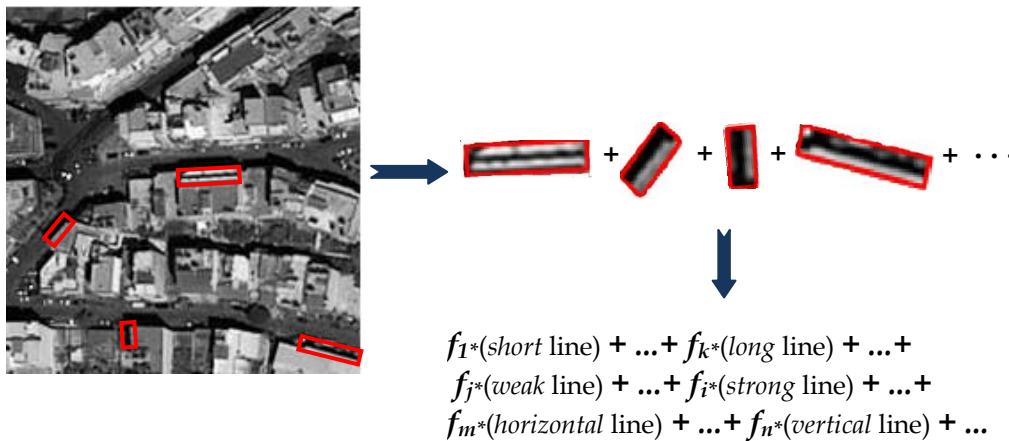$$f_m*(horizontal \text{ line}) + ...+ f_n*(vertical \text{ line}) + ...$$

**Figure 11.3:** The idea for extracting features from satellite images which is extracted from our experiences about ICA. The idea is to detect lines directly in the contextual image patch and extract features from their characteristics

This could lead us to an idea for extracting features from satellite images. This idea is

based on our experiences obtained from using ICA for satellite images. According to the previous chapters we can model our contextual image patch, using ICA basis vectors. Moreover, we know that every basis vector usually presents a kind of structure which can be modeled by few lines. These lines, themselves, could be represented by smaller segments. On the other hand, in chapter 10 we demonstrated that how we can extract features from the characteristics of lines inside the basis vectors.

The idea is to detect lines directly inside the contextual patch and extract features form the characteristics of lines. In fact, each line and the pixels around it could be considered as an important component of a contextual image patch. We only need the pixels around the line which are necessary to compute the gradient for the pixels of line. So we will have narrow segments which contain lines instead of square windows. This is shown in Figure 11.3. In other words, the narrow segments which contain the lines play a role that is similar to the role of basis vectors. There are advantages for this idea in comparison with the ICA basis vectors:

1- If we use this idea we don't need to perform a learning procedure to obtain the basis vectors

2- The length of the line could be different and is not limited with the size of ICA basis vectors. Specially, we could have very long lines comparing with normal size of ICA basis vectors.

3- If a line is represented by an ICA basis vector, many of pixels around the line don't present important information. This means that we have a type of redundancy in the case of ICA basis vectors. This redundancy is eliminated using the new idea.

## 11.2 Lines properties of contextual patch as features

The procedure of feature extraction is similar to the procedure which was explained in chapter 10. However, here we don't need to initially obtain the basis vectors from the contextual patch. Actually, we directly go to the steps of edge detection and line approximation. Then, we estimate a narrow segment around each line and compute three characteristics of it: the length, the angle and the average of gradient of the line. For each characteristic we divide its entire possible interval to some smaller intervals. Below we explain the details of feature extraction.

### 11.2.1 Finding lines inside the contextual patch

The first step of feature extraction is to find the possible lines inside the contextual patch. This is performed using the edge detection and line estimation algorithms which are explained in sub chapters 10.2 and 10.3. We use the Sobel method for computing the gradient in our edge detector. We adjust the parameter of thresholding in edge detector such that we detect all necessary edges and avoid unwanted edges. In addition in our line estimator we are able to determine the minimum length of

lines which must be detected. We decided to detect the lines which are longer than 8 pixels. Figure 11.3 shows the results of edge detector (Binary Matrix) and line estimators which are applied for the contextual patch.
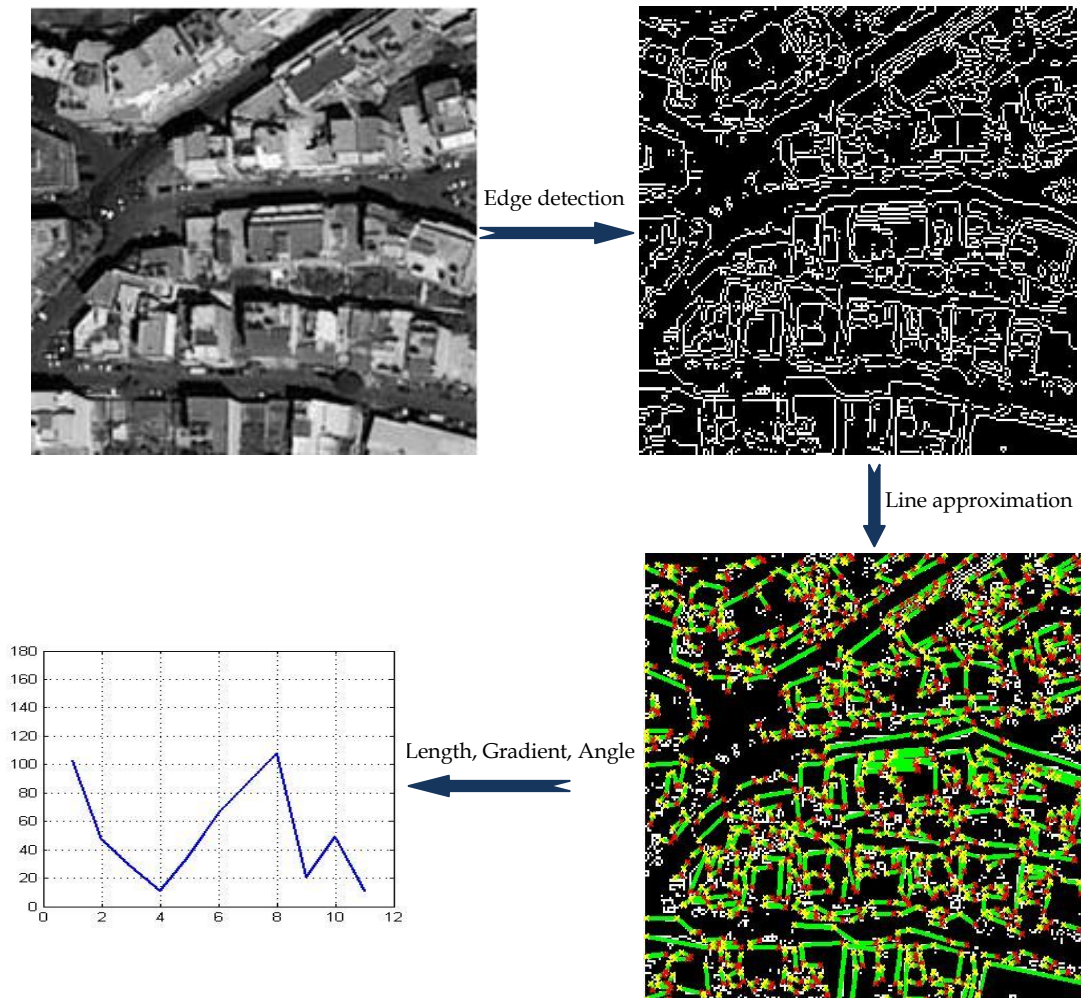


**Figure 11.4:** Feature extraction from a contextual image patch, using its lines properties. There are several steps such as edge detection and line approximation. Finally, for each line, we consider a narrow segment around it and put each of its characteristics (length, average of gradient magnitude and angle) into the corresponding bin.

## 11.2.2 Length, Gradient and Angle as the important line properties

As it was mentioned, we consider line and the pixels around it as the components which play a role similar to the role of basis vectors. In chapter 10 we demonstrated that the most important characteristic of basis vectors could be modeled by the

properties of its lines. Here, just like in chapter 10, the length, the average of gradient magnitudes and the angle of a line are considered as its most important characteristics.

## 11.2.3 Number of elements in a bin as feature

Similarly to what we did in chapter 10, we consider 4 bins for the length, 3 bins for the average of gradient magnitude and 4 bins for the angle. But the boundaries of intervals are defined differently.

For the length, we define *short* line as the line whose length is between 8 to 12 pixels, *medium* line as the line whose length is between 13 to 15 pixels, *long* line as the line whose length is between 16 to 20 pixels and *very long* line as the line whose length is greater than 20 pixels.

For the average of gradient magnitude we define 3 intervals for *week*, *strong* and *very strong* lines but we shift the boundaries of intervals to higher levels. The reason is that the range of variation in the basis vectors is limited because of initial whitening in the beginning of learning procedure. For the angle of lines we define 4 intervals for the lines whose angles are around 0°, 45°, 90° or 135°. In fact, our 4 intervals for the angle are defined as $0^{\circ} \pm 22.5^{\circ}$, $45^{\circ} \pm 22.5^{\circ}$, $90^{\circ} \pm 22.5^{\circ}$ and $135^{\circ} \pm 22.5^{\circ}$. These are the same intervals for the angle in previous chapter.

The number of elements in each bin could be considered as a feature of our contextual patch. Consequently, here we have a feature vector with 11 features whose first 4 features correspond to the length, the next 3 features correspond to the average of gradient magnitude and the last 4 features correspond to the angel. Table 11.1 contains the descriptions and the intervals for the features.

**Table 11.1**: Features, their description and their intervals

| Feature | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Description | *Short,* | *Medium,* | *Long,* | *Very Long* | *Weak* | *Strong* | *Very Strong* |
| Interval | $8 \leq L \leq 12$ | $13 \leq L \leq 15$ | $16 \leq L \leq 20$ | $21 \leq L$ | $|G| < 42$ | $42 \leq |G| < 60$ | $|G| \geq 60$ |

| Feature | 8 | 9 | 10 | 11 |
|---|---|---|---|---|
| Description | *Horizontal* | *Oblique1* | *Vertical* | *Oblique2* |
| Interval | $0^{\circ} \pm 22.5^{\circ}$ | $45^{\circ} \pm 22.5^{\circ}$ | $90^{\circ} \pm 22.5^{\circ}$ | $135^{\circ} \pm 22.5^{\circ}$ |

Figure 11.5 shows examples of these feature vectors which are obtained for 10 first contextual patches from each class of the *test set*.
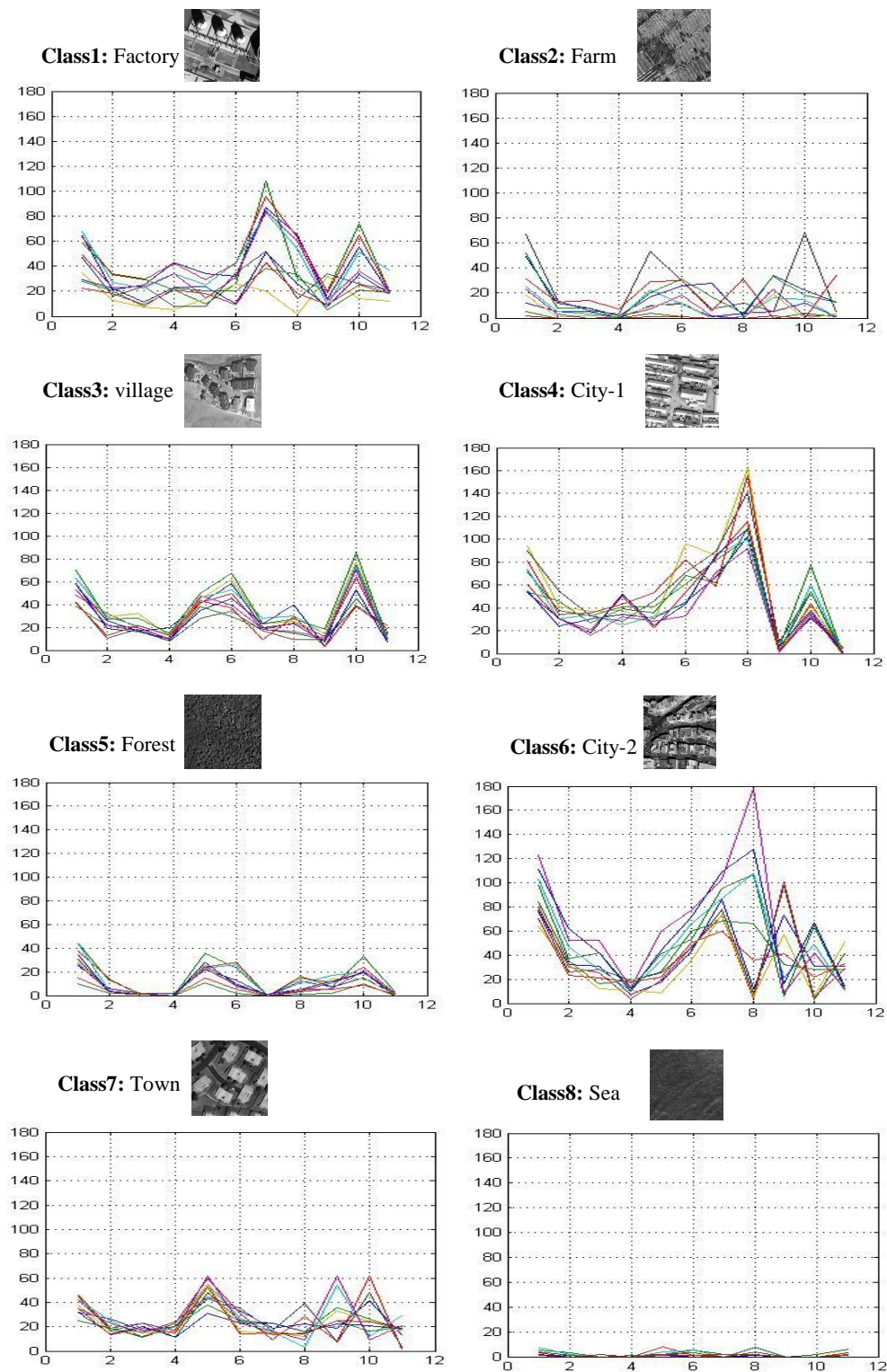
**Figure 11.5:** Feature vectors which are obtained from the characteristics of lines detected on the contextual patches. These features are obtained for 10 first contextual patches from each class of the *test set*.

## 11.3 Simple clustering for evaluation

To evaluate our features we performed the same clustering which is explained in sub-chapter 7.3 but with the new features. The results of such clustering are summarized in Table 11.2.

**Table 11.2**: Results of simple clustering. We used the feature vectors which are defined from the properties of lines inside a contextual patch.

| Clusters | Class1 Factory | Class2 Farm | Class3 Village | Class4 City-1 | Class5 Forest | Class6 City-2 | Class7 Town | Class8 Sea |
|---|---|---|---|---|---|---|---|---|
| **1** | **84.4** | 2.2 | 3.3 | 2.2 | 1.1 | 4.4 | 2.2 | 0 |
| **2** | 0 | **71.1** | 0 | 1.1 | 11.1 | 0 | 0 | 10 |
| **3** | 4.4 | 0 | **88.9** | 2.2 | 0 | 3.3 | 3.3 | 0 |
| **4** | 5.5 | 1.1 | 1.1 | **86.7** | 0 | 4.4 | 4.4 | 0 |
| **5** | 0 | 13.3 | 1.1 | 0 | **80** | 0 | 0 | 6.7 |
| **6** | 3.3 | 1.1 | 0 | 5.5 | 0 | **86.7** | 2.2 | 0 |
| **7** | 2.2 | 0 | 5.5 | 2.2 | 0 | 1.1 | **87.8** | 0 |
| **8** | 0 | 11.1 | 0 | 0 | 6.7 | 0 | 0 | **83.3** |

## 11.4 Conclusions

In this chapter we proposed an approach for defining features from the properties of line (length, average of gradient magnitude, angle) which are detected inside a contextual patch. We took the idea of feature extraction from the idea of ICA basis vectors.

According to Table 11.2, the results are approximately in the same level as the results of previous chapter. This is not strange because we used similar approach to extract features. In other words, both methods are based on the characteristics of lines. Specially, Here we see the same property that exists in the features which are obtained from the lines inside the basis vectors, that is, the group of classes that contain geometrical objects (class1, class3, class4, class6, class7) could be easily separated from the other group of classes that mostly contain natural landscapes (class2, class5, class8).

This method is a little faster than the method presented in previous chapter. Here, the average computation time for extracting features from a contextual patch is 0.59sec

but the computation time in previous chapter was 0.96sec. However, the computation time is strongly depended on the scene presented by image. If the scene includes a lot of lines to be approximated, just like the case that the scene is a piece of urban area, the computation time rapidly increases. We can also guess that if we increase the size of contextual patches, then the recent method gets slower comparing with the method explained in chapter 10. The reason is that the number of lines will grow up.

# CHAPTER 12

# EVALUATION

In previous chapters several approaches were presented to extract features from the contextual image patches. In this chapter, we are going to compare the proposed methods through a supervised classification. This supervised classification is based on the Super Vector Machine (SVM).

## 12.1 Super Vector Machine

Recently, many studies have demonstrated the important potential of SVM-based approaches for classification tasks. See [56] for example.



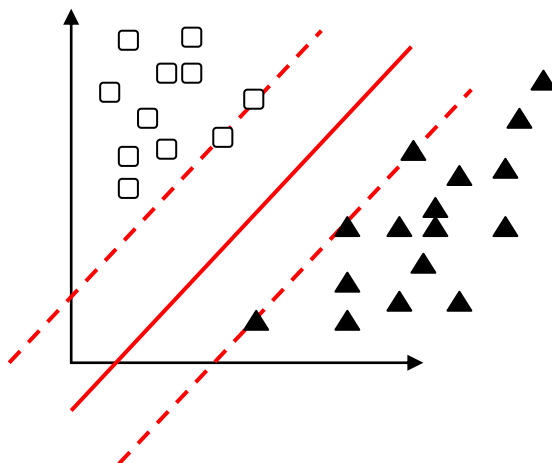**Figure 12.1**: An illustration of SVM classification. The solid line corresponds to the hyperplane ($w.x + b = 0$) and the dashed lines corresponds to binderies of the margin: $w.x + b = \pm1$. Samples on the margin are called the support vectors.

SVM classifiers could operate more effectively in comparison with many other types of classifier. The idea of SVM is to fit a separating hyperplane between two classes such that the training samples that are located at the boundaries of two classes (called the support vectors) will be as far as possible. In other words, it maximizes the margin between positive two groups of samples.

Basics of SVM classification for two classes are illustrated in Figure12.1. We assume that there are some given data points that each of them belongs to one of two classes, and the goal is to put a new sample in one of two classes. The data points are considered as $n$ dimensional vectors and we want to know whether we can separate such points with a $(n-1)$ dimensional hyperplane.

In other words, it is assumed that we have two different classes and a set of training samples of the form $(x_i, y_i)$, in which $x_i$ are the $n$ dimensional vectors and $y_i$ are their labels $(-1,1)$ which indicates to which class the samples belongs. Theoretically, we have two classes which are represented by 1 and −1 but in practice, we usually have just one class and we are going to separate the samples which do belong to this class ($y_i = 1$) from the other samples which do not belong to it ($y_i = -1$). The objective is to find the maximum-margin hyperplane that separates the points having $y_i = 1$ from those having $y_i = -1$. Any hyperplane can be written as the set of points satisfying:

$$w.x + b = 0 \qquad (12.1)$$

The vector $w$ is a normal vector which is perpendicular to the hyperplane and $b$ is the bias. The term $w.x$ is the inner product of $w$ and $x$. A separating hyperplane is supposed to separate the two classes as: $wx_i + b \geq 1$ (for the class $y_i = 1$ ) and $wx_i + b \leq 1$ (for the class $y_i = -1$). These two equations could be combined as:

$$y_i(w.x_i + b) \geq 1 \qquad (12.2)$$

The support vectors of the two classes are the samples that lie on two hyperplanes, which themselves are parallel to the optimal hyperplane and are defined by $w.x + b = \pm 1$. The margin between these planes is $2/\|w\|$ and we aim to maximize this margin through minimizing the $\|w\|$.

This optimization is usually difficult to solve because it depends on the norm of $w$, which involves a square root. Thus, usually, we change the optimization by substituting $\|w\|$ with $\frac{1}{2}\|w\|^2$ :

$$\min\left\{\frac{1}{2}\|w\|^2\right\}$$

(12.3)

When the classes are not linearly separable, some extra variables ($\varepsilon_i$) are defined to compensate the error of samples which are not classified exactly by linear hyperplanes. Equation (12.2) may be rewritten as:

$$y_i(w.x_i + b) \geq 1 - \varepsilon_i$$

(12.4)

And the optimization problem changes to

$$\min\left\{\frac{\|w\|^2}{2} + \sum_i \varepsilon_i\right\}$$

(12.5)

The first part of equation (12.5) aims to maximize the margin between the classes and the second part aims to compensate the error of classification.

## 12.2 Supervised classification based on SVM

Our objective in this chapter is to compare the methods which are introduced in previous chapter through a supervised classification based on SVM. For this purpose we prepared a relevance feedback tool and a database of contextual patches. Then for each contextual patch and for each method we prepared a feature vector.

### 12.2.1 Relevance feedback tool

Our supervised classifier is a visual tool which contains a SVM engine and allows the user to select the desired class during several iterations. In every iteration, we are able to observe a number of classified and unclassified samples (contextual patches) which are placed on the SVM surfaces and we can determine positive and negative samples. Then the SVM engine uses the feature vectors of these positive and negative samples to improve the classification for the next iteration. In our experiments we adjust the parameters of our tool to show 20 positive samples and 20 negative samples in the surfaces of SVM. Figure 12.2 shows a schema of the relevance feedback tool.
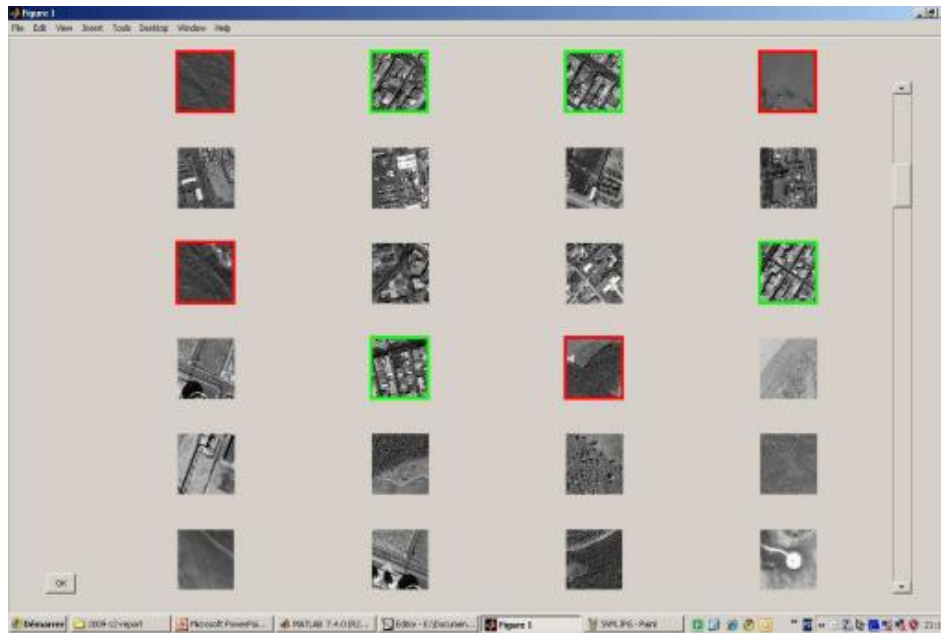
**Figure 12.2**: relevance feedback tool. Positive samples are green and negative sample are red.

## 12.2.2 Contextual patch database

A data base of 20000 contextual image patches is provided to be used in classification. We randomly gathered these contextual patches from the initial satellite images similar to those which are shown in Figure 3.1. The provided contextual patches contain a variety of man-made and natural landscapes.

## 12.2.3 Feature extraction

We compare 5 kinds of descriptor which are presented in previous chapters

- Normal ICA (25 features)
- Topographic ICA (9 features)
- Bag of words (64 features)
- Lines inside basis vectors (11 features)
- Lines inside the image (11 features)

As it is mentioned in previous chapters, for most of feature extraction methods relating to ICA, we transform the image such that its mean value is equal to zero and its norm is equal to one. So, we lose two important characteristics of image. Here, we add these two features to the end of different image descriptors. For each kind of descriptor we normalize the mean value and the norm of image regarding to the

variance of all features existing in the descriptor. The goal is that the variations ranges of these two features would not be very far from other features.

In addition, it is interesting to compare the proposed method with Gabor wavelet features. The motivation is that Gabor wavelet is the most similar method to the ICA in terms of using a set of filters and also in terms of the shape of filters. In Chapter 2, we explained the Gabor features as one of the textural features. In chapter 7 we used a set of 25 ICA filters (basis vectors) for feature extraction. Here, to keep the similar conditions we use a set of 25 Gabor filter (5 scales in $\theta$ angle and 5 scales in $\lambda$) for extracting the Gabor features.
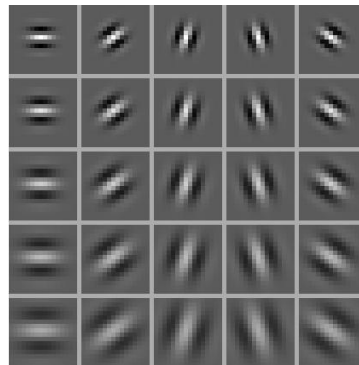


**Figure 12.3**: set of 25 Gabor filter (5 scales in $\theta$ angle and 5 scales in $\lambda$ )

Extracting features from a contextual image patch using a set of Gabor filters is absolutely similar to feature extraction using a set of ICA basis vectors which is explained in chapter 7. That is, we gather a sufficient number of samples from initial image and decompose them into the set of filters. Then, using a root square average upon all coefficients that correspond to one filter we obtain the feature which is related to that filter. Finally, we add the mean value and variance of the contextual image patch to the descriptor.

## 12.2.4 Class detection

Now we are ready to detect the classes using different descriptors of contextual patches. We choose one kind of different descriptors and try to detect the classes among the contextual patches. Detecting one class is done during about 7-15 training iterations depending on the user which determines the boundaries of class and also to the method by which the descriptors are defined. We stop the training when the classified and unclassified samples which are shown by our visual tool stay in a stable situation.  Then we repeat the procedure, 2-4 times, for the same class, to verify if the result of classification for that class stays approximately at the same level. Result of classification for one class is expressed as its related *precision* and *recall*. Precision can be seen as a measure of exactness or fidelity, whereas recall is a measure of completeness.

**Class1**

**Class2**

**Class3**

**Class4**

**Class5**

**Class6**

**Class7**

**Class8**

**Class9**

**Class10**

**Class11**

**Class12**

**Class13**

**Class14**

**Class15**

**Class16**

**Class17**
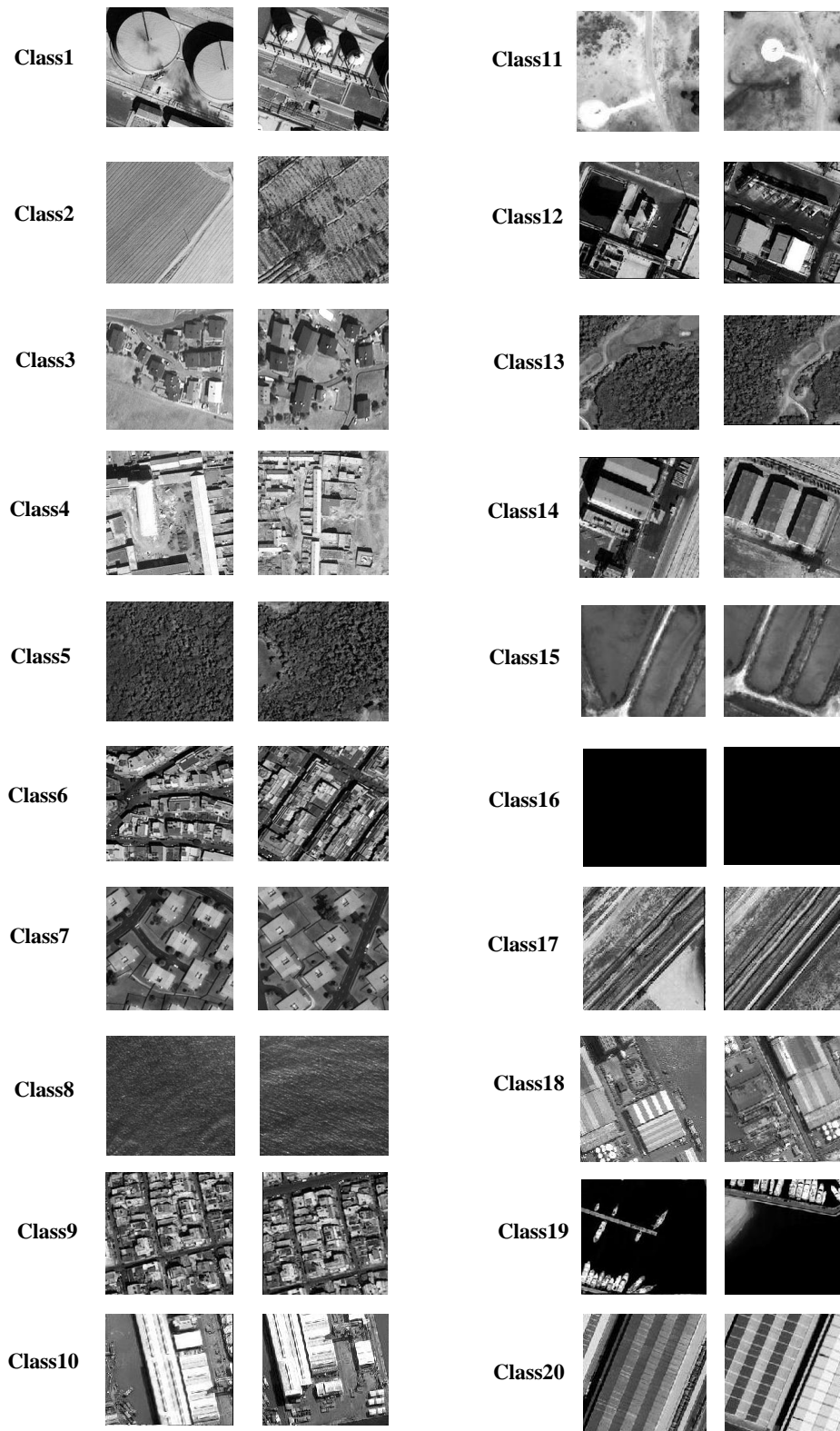
**Class18**

**Class19**

**Class20**

**Figure 12.4:** 20 extracted classes which are obtained from a supervised classification.
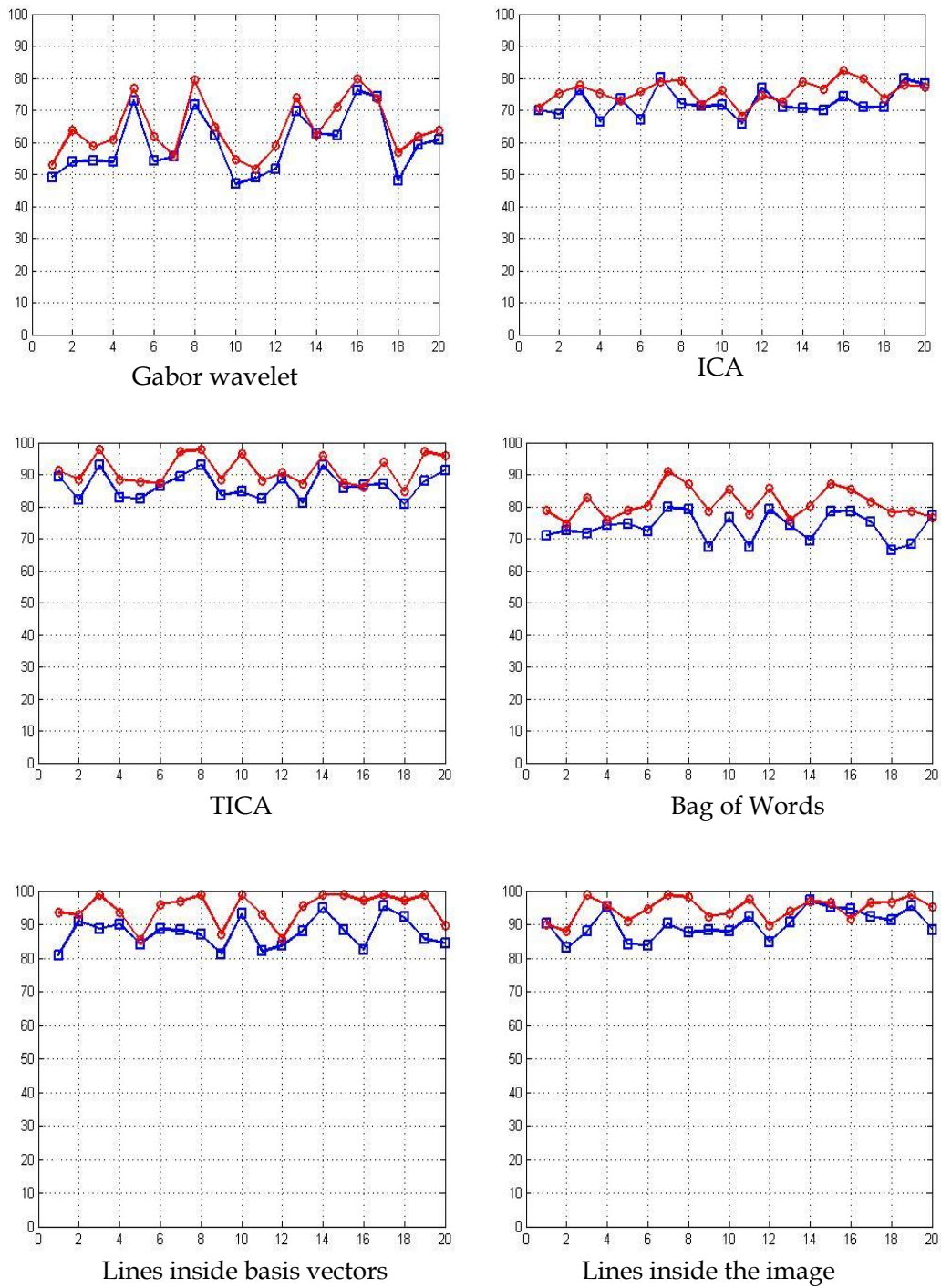
**Figure 12.5**: Precision and recall obtained for 20 classes shown in Figure 12.4 for different kinds of descriptors. Red curves are precisions and the blues are recalls.

For each class which is obtained through a classification, precision is defined as the number of relevant samples retrieved by classification divided by the total number of samples retrieved by that classification, and recall is defined as the number of relevant samples retrieved by classification divided by the total number of existing relevant samples (which should have been retrieved). To compute the precision and recall for each class, we built a visual tool to observe all the classified non-classified samples, so the user can detect the false positive and false negative samples.

Depending on the boundaries of classes, it is possible to extract different numbers of classes. If we choose very specialized classes, it is possible to increase the number of extracted classes. We extracted 20 classes from different man-made and natural landscapes. Some samples of these 20 classes are shown in Figure 12.4. Extracted classes contain different number of samples from about 120 samples to 1500 samples.

After obtaining precision and recall for all classes using one kind of descriptors, we do the same procedure to extract the same 20 classes using other kinds of descriptors. Results of classification are summarized in Figure 12.5.

## 12.3 Conclusion

Looking to the different diagrams in Figure 12.5 we are able to compare the capabilities of different methods. As the first conclusion we could say that the methods "TICA", "Lines inside the basis vectors" and "lines inside the images" have the best results and Gabor features don't present a suitable result. The results of ICA features are less than TICA features but higher than Gabor features. The bag of words model is placed between the ICA features and the three best methods.

**Table 12.1**:  Comparison of methods

|  | Average of precision and recall | Average of time for obtaining features | Length of feature vector |
|---|---|---|---|
| **Gabor features** | P=64.14% R=59.41% | 0.15 sec | 27 |
| **ICA features** | P=75.79% R=72.29% | 0.15 sec | 27 |
| **TICA features** | P=91.39% R=86.57% | 0.21 sec | 11 |
| **Bag of words features** | P=81.01% R=73.68% | 0.82 sec | 66 |
| **Lines inside basis vectors** | P=94.87% R=87.54% | 0.96 sec | 13 |
| **Lines inside images** | P=93.37% R=88.63% | 0.59 sec | 13 |

 The important point is observed in the results of natural landscapes (classes 5-8-13-16-17). We see that Gabor features work with an acceptable accuracy for such classes. But when our class contains the geometrical objects, their results are not so good. This is normal because Gabor features are used to model the textures and we know that the natural landscapes can be usually described by texture like features. However, other methods present approximately the same level of quality for natural landscapes and man-made classes. This shows the capability of presented descriptors for characterisation of geometrical structures comparing with Gabor features.

# CHAPTER 13

# CONCLUSIONS AND PERSPECTIVES

In this chapter, based on our work in the thesis some conclusions are presented. In addition, the perspectives of thesis including the probable future works and different directions in which this research can be developed are demonstrated.

## 13.1 Conclusions

In this thesis we tried to present a methodology to study the statistical nature of satellite images and to extract their statistical signatures. The satellite images are considered as some multi-variable random signals such that each pixel could be an individual random variable and the objective is to investigate the statistical dependencies between this random variable (pixel) and the other random variables (pixels). Independent Component Analysis was used as the theoretical core of the thesis to study the statistical dependencies inside the satellite images.

In Figure 13.1 we see a graphical schema of the main thesis contributions. The first contribution of the thesis was a study about the scale and dimensionality behavior of an ICA system when it is used for satellite image characterization. We found that the optimum dimensionality of such ICA system corresponds to a reduction factor of about 0.1. Also, we chose the size of 16*16 pixels as the optimum size for ICA basis vectors. This study helped us to choose the framework of ICA systems for the goal of feature extraction from satellite images.

Feature extraction methods presented in this thesis can be divided into two main groups: methods that use the ICA sources to define features and methods who extract features from ICA basis vectors. First, we proposed an approach in which different samples of ICA sources related to one satellite image patch are integrated to a feature vector. This feature vector could be considered as the signature of the image. In this part, we also proposed an approach to improve the set of ICA basis vectors which leads to a better descriptor for the case that objective is to separatr two different classes.
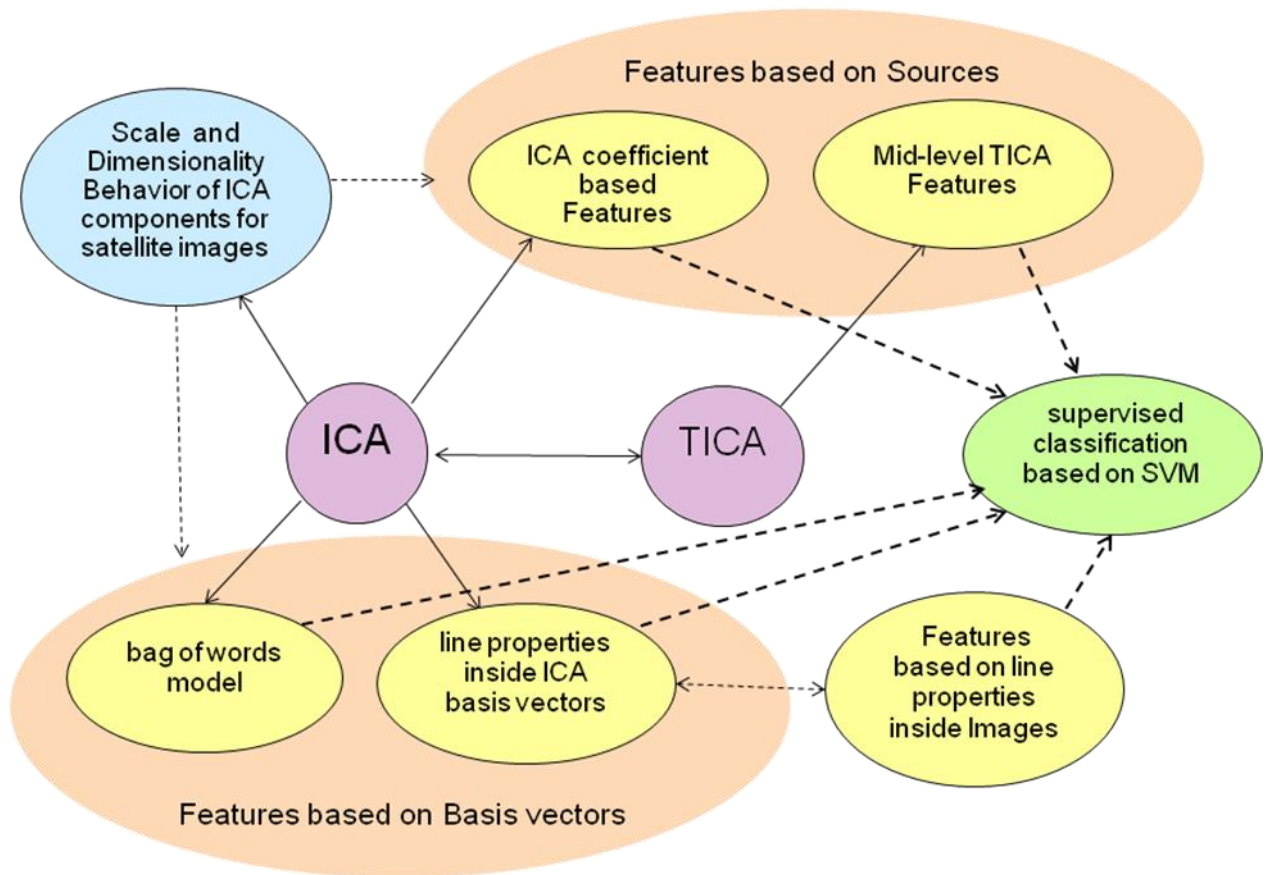
**Figure 13.1**: Graphical schema of the thesis contributions.

We developed the initial ICA approach through a Topographic ICA system to obtain Mid-level TICA features that are combined from some low level features and work absolutely more effective with respect to the initial low level features.

On the other side, we developed two methods that use characteristics of ICA basis vectors to define features. Using the Bag of Words model we produced some descriptors that shows the level of similarity between dictionary words and the ICA basis vectors related to each document. Also, we proposed another approach who extracts the features from the line properties of ICA basis vectors related to each image patch. Based on our experiences about using ICA for satellite images, we found that there is some strong links between ICA and gradient properties of the image. So, we proposed another method which detects the lines directly inside a satellite image and define features from gradient and line properties inside it.

The objective of the thesis is to present descriptors for high resolution satellite images that are more precise than textural features and simpler with respect to the local descriptors. Presented descriptor are placed somewhere between textural and local approaches. From one side, they give a global interpretation of the scene and don't present details of objects inside the image patch. But from the other side they

deal with the edge and gradient properties that are important in the structures of geometrical objects.

The descriptors presented in the thesis were verified through a supervised classification method based on Super Vector Machine (SVM). The classification was performed on a database of 20000 satellite image patch with a resolution of about one meter. Results of classification were presented in the form of precisions and recalls obtained for 20 different detected classes of landscapes. Based on these results we can conclude that presented descriptors are suitable for describing a variety of landscapes especially those who contain geometrical structures.

We compared all presented methods in terms of the average of precision and recall, computational time and the length of the feature vector. A feature vector based on Gabor-wavelet filters (as a typical textural approach) was also compared with our presented approaches. In terms of precision and recall, we showed that "TICA", "Lines inside the basis vectors" and "lines inside the images" are the most accurate approaches. The "Gabor" approach does not present an accurate result but it works with an acceptable accuracy for natural classes. In terms of computational time, we found that "Gabor ", "normal ICA" and "TICA" are faster than other methods. Finally, in terms of length of feature vector, we showed that "TICA", "Lines inside the basis vectors" and "lines inside the images" present shorter feature vectors. Generally, regarding to all of these criteria we can conclude that among all presented methods, "TICA" is the most efficient one.

## 13.2 Perspectives

Presented approaches for defining descriptors are some global approaches. In other words, they are not dependent to the content, resolution and type of image. However, we used them for the high resolution satellite images. Features extraction algorithms presented during the thesis can be verified with the satellite images from other sensors and with other resolutions. This could be one of the future works predicted for this thesis. They also can be verified with other types of images such as medical images, natural images, images in the field of astronomy, etc.

Presented descriptors can be developed and can be combined with each other and also with other features extracted from other methods to improve the efficiency of the descriptors. In the thesis we added mean value and variance of the image patches to the feature vectors to improve their efficiencies. They also can be combined with some of well-known textural or local features.

We proposed a supervised classification to evaluate the features. Supervised classifiers have some advantages. For example, they let us to detect samples corresponding to a desired class. In other words, we can detect one class among a lot of relevant or non-relevant samples. But they also have some disadvantages. For example they are strongly dependent to the user's point of view and the number of training iterations for detecting one class. Thus, one of perspective of the thesis could be providing a standard framework for the evaluation of the descriptors in order to reduce effects of other parameters such as user's point of view. As a proposed solution, the experiments can be performed by a number of users and the final results can be obtained through averaging the results of different users.

Different descriptors presented in the thesis can be used for Image Information Mining methods, classification algorithms and segmentation methods. They also can be used for a variety of applications such as urban area detection, Geographic Information System, image search engines, etc. Generally, any application that needs a description or interpretation of high resolution satellite images can befits these descriptors.

# BIBLIOGRAPHY

[1] H. Barlow. "Sensory Communication", *chapter Possible principles underlying the transformation of sensory messages,* pages 217-234. MIT press, 1961.

[2] A. J. Bell and T. J. Sejnowski. "The independent components of natural scenes are edge filters". *Vision Res*, Vol. 37, No. 23. (December 1997), pp. 3327-3338.

[3] B. Olshausen and D. Field. "Emergence of simple-cell receptive field properties by learning a sparse code for natural images". *Nature*, 381:607-609, 1996.

[4] T-W. Lee, M. S. Lewicki, and T. J. Sejnowski. "Unsupervised classification with non-gaussian mixture models using ICA". *In Advances in Neural Information Processing Systems*, 1999.

[5] A. Hyvärinen and E. Oja. "A Fast Fixed-Point Algorithm for Independent Component Analysis". *Neural Computation,* 9(7):1483-1492, 1997

[6] A. Hyvärinen and E. Oja. "Independent Component Analysis: Algorithms and Applications" *Neural Networks*, 13(4-5):411-430, 2000

[7] Zhang, X. and C. H. Chen, "New independent component analysis method using higher order statistics with application to remote sensing images", *Opt. Eng,* vol.41, 2002.

[8] Jin-xia Zhang, Yen-wei Chen, Zensho Nakao and Tomoko Tateyama; "Independent Components Analysis for classification of remotely sensed images". *International Journal of Innovative Computing, Information and Control* ;Volume 2, Number 3, June 2006

[9] Quintiliano and Santa-Roza; "Detection of Streets Based on KLT using Ikonos Multispectral Images", *GRSS/ISPRS Joint Workshop Urban,*2003

[10]  Dogrusoz and Aksoy; "Modeling Urban Structures Using Graph-Based Spatial Patterns" , *Proceedings of the IEEE International Geoscienceand Remote Sensing Symposium, Barcelona, Spain,*June 2007

[11]  Yi Yang and Shawn Newsam; "Comparing SIFT Descriptors and Gabor Texture Features for Classification of Remote Sensed Imagery",*IEEE International Conference on Image Processing*, 2008

[12] B. Sırmaçek and C. Ünsalan, "Urban area and building detection using SIFT keypoints and graph theory", *IEEE Transactions on Geoscience and Remote Sensing*

[13] S. Newsam and Y. Yang, "Integrating Gazetteers and Remote Sensed Imagery," *ACM International Conference on Advances in Geographic Information Systems (ACM GIS),* 2008.

[14] Lior Weizman and Jacob Goldberger,"Detection of urban zones in satellite images using visual words".. *SPIE Int. Geoscience and Remote Sensing Symposium (IGARSS),* 2008

[15] J.X. Zhang, Y.W. Chen, Z. Naka and T. Tateyama, "Independent component analysis for classification of remotely sensed images", *Int. J. Innov. Comput. Inf. Cont. 2 ,*2006

[16]  Mavrantza and  Argialas, "Identification of urban Features using object-oriented image analysis", *PIA07. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36

[17] D. Tuia, F. Pacifici, A. Pozdnoukhov, C. Kaiser, D. Solimini, W.J. Emery, "Very-high resolution image classification using morphological operators and SVM", *IGARSS'08*, Boston, MA, U.S.A., Jul. 2008

[18] V. Karathanassi, C. Iossifidis, and D. Rokos, "A texture-based classification method for classifying built areas according to their density", *International Journal of Remote Sensing*, vol. 21, no. 9, pp. 1807-1823, 2000.

[19] J. A. Benediktsson, M. Pesaresi, and K. Arnason, "Classification and feature extraction for remote sensing images from urban areas based on morphological transformations," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 9, pp. 1940-1949, 2003.

[20] C. Unsalan and K. L. Boyer, "Classifying land development in high resolution panchromatic satellite images using straight line statistics," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 42, no. 4, pp. 907-919, 2004.

[21] C. Unsalan and K. L. Boyer, "A theoretical and experimental investigation of graph theoretical measures for land development in satellite imagery", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 27, no. 4, pp. 575-589, 2005.

[22] C. Unsalan, "Measuring land development in urban regions using graph theoretical and conditional statistical features", *IEEE Transactions on Geoscience and Remote Sensing,* vol. 45, no. 12, pp. 3989-3999, 2007.

[23] L. M. Fonte, S. Gautama, W. Philips, and W. Goeman, "Evaluating corner detectors for the extraction of man made structures in urban areas", in *IEEE International Geoscience and Remote Sensing Symposium,* 2005

[24] S. Bhagavathy and B. S. Manjunath, "Modeling and detection of geospatial objects using texture motifs," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 12, pp. 3706-3715, 2006.

[25] L. Bruzzone and L. Carlin, "A multilevel context-based system for classification of very high spatial resolution images," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 44, no. 9, pp. 2587-2600, 2006.

[26] P. Zhong and R. Wang, "A multiple conditional random fields ensemblemodel for urban area detection in remote sensing optical images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 12, 2007.

[27] A. Lorette, X. Descombes, , and J. Zerubia, "Texture analysis through a markovian modelling and fuzzy classification: Application to urban area extraction from satellite images," *International Journal of Computer Vision*, vol. 36, no. 3, pp. 221–236, 2000.

[28] T. Kim and J. P. Muller, "Development of graph-based approach for building detection," *Image and Vision Computing*, vol. 17, no. 1, pp. 3–17, 1999.

[29] K. Segl and H. Kaufmann, "Detecting small objects from high-resolution panchromatic satellite imagery based on supervised image segmentation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 9, pp. 2080–2083, 2001.

[30] C. Unsalan and K. L. Boyer, "A system to detect houses and residential street networks in multispectral satellite images," *Computer Vision and Image Understanding*, vol. 98, pp. 432–461, 2005.

[31] S. D. Mayunga, Y. Zhang, and D. J. Coleman, "Semi-automatic building extraction utilizing Quickbird imagery," in Proceedings of the ISPRS Workshop CMRT, 2005, pp. 131–136.

[32] J. Peng, D. Zhang, and Y. Liu, "An improved snake model for building detection from urban aerial images," *Pattern Recognition Letters*, vol. 26, pp. 587–595, 2005.

[33] X. Huang, L. Zhang, and P. Li, "An adaptive multiscale information fusion approach for feature extraction and classification of Ikonos multispectral imagery over urban areas," *IEEE Geoscience and Remote Sensing Lett*ers, vol. 4, no. 4, pp. 654–658, 2007.

[34] P. Gamba, F. D. Acqua, G. Lisini, and G. Trianni, "Improved VHR urban area mapping exploiting object boundaries," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 8, pp. 2676–2682, 2007.

[35] F. Del Frate, G. Schiavon and C. Solimini, "Use of High Resolution Satellite Data for Change Detection in Urban Areas". *ESA-EUSC 2005: Image Information Mining*

[36] S. Voigt, T. Kemper, T. Riedlinger, R. Kiefl, K. Scholte, and H. Mehl , "Satellite Image Analysis for Disaster and Crisis-Management

Support." *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, 6: 1520-1528, 2007

[37] P. Gamba, F. Dell'Acqua, and G. Trianni , "Rapid Damage Detection in the Bam Area Using Multitemporal SAR and Exploiting Ancillary Data." *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, 6: 1582-1589, 2007

[38] J. A. Benediktsson., J. A. Palmason and  J. R. Sveinsson, "Classification of Hyperspectral Data from Urban Areas Based on Extended Morphological Profiles." *IEEE Transactions on Geoscience and Remote Sensing*, vol 42: 480-491,2005

[39] M. Schröder, H. Rehrauer, K. Siedel, and M. Datcu "Spatial Information Retrieval from Remote Sensing Images-Part II: Gibbs-Markov Random Fields". *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, 5:1446-1455,1998

[40] P. Gong, D. J. Marceau, and P. J. Howarth  " A Comparison of Spatial Feature Extraction Algorithms for Land-Use Classification with SPOT HRV Data.", *Remote Sensing of Environment*, 40:137-151.,1992

[41] M. Datcu et al, "Information Mining in Remote Sensing Image Archives: System Concepts.",  *IEEE Trans. Geosci.  Remote Sensing*, vol. 41, 12:2923-2936.

[42] R. Baeza-Yates and B. Ribeiro-Neto , "Modern Information Retrieval", *ACM Press*. 1999

[43]  Aapo  Hyvärinen, Patrik  O. Hoyer, Mika  Inki: "Topographic Independent Component Analysis". *Neural Computation* 13(7): 1527-1558 ,2001

[44]  Haralick, R., Shanmugam, K. & Dinstein, I . "Textural features for image classification" , *CMetImAly77, pp. 141–152.*, 1977

[45] Vetterli,M. (1986). "Filter banks allowing perfect reconstruction", *Signal Process. 10(3): 219–244.,1986*

[46] Daugman, J. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by 2d visual cortical filters, JOSA-A 2(7): 1160–1169., 1986

[47] Lowe, David G. "Object recognition from local scale-invariant features". *Proceedings of the International Conference on Computer Vision. 2. pp. 1150–1157.*doi:10.1109/ICCV.1999.790410

[48] Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool "SURF: Speeded Up Robust Features", *Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346--359, 2008*

[49] Hyvärinen A. ‚"New approximations of differential entropy for independent component analysis and projection pursuit". *Advances in Neural Information Processing Systems, volume 10,* 1998

[50] Duda, R. O. and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Comm. ACM, Vol. 15, pp. 11–15 , 1972*

[51] D. Ziou and S. Tabbone , "Edge detection techniques: An overview", *International Journal of Pattern Recognition and Image Analysis*, 1998

[52] J. Canny "A computational approach to edge detection", *IEEE Trans. Pattern Analysis and Machine Intelligence, vol 8, pp.679-714.,*1986

[53] R. Haralick, (1984) "Digital step edges from zero crossing of second directional derivatives", *IEEE Trans. on Pattern Analysis and Machine Intelligence, 6(1): pp:58–68.*

[54] R. Kimmel and A.M. Bruckstein, "On regularized Laplacian zero crossings and other optimal edge integrators", *International Journal of Computer Vision, 53(3) pp. 225-243.* 2003

[55] *R. Deriche* "Using Canny's criteria to derive an optimal edge detector recursively implemented", *Int. J. Computer Vision, vol 1, pp 167–187,*1987

[56] Corinna Cortes and V. Vapnik, "Support-Vector Networks", *Machine Learning, 20,* 1995.

[57] Blei, David M.; Ng, Andrew Y.; Jordan, Michael I . "Latent Dirichlet allocation". *Journal of Machine Learning Research 3: pp. 993–1022.,* 2003

# Modélisation et Extraction des Descripteurs Intrinsèques des Images Satellite à Haute Résolution: Approches Fondées sur l'Analyse en Composantes Indépendantes

**RESUME :** Les images satellites haute résolution contiennent des informations très détaillées comme la forme des bâtiments, les zones industrielles, etc. Leur contenu d'information est hyper riche et très compliqué a extraire. Parmi les paysages différents, les zones urbaines et des structures géométriques sont les paysages plus compliques pour les différant domaines de recherches. Nous allons extraire les indices intrinsèques des images satellite et proposer les descripteurs robustes. En utilisant ces descripteurs, nous serions capables de reconnaitre une variété des paysages, en particulier, les structures géométriques au sein des images satellite. L'analyse en composantes indépendantes (l'ACI) est la base théorique de cette thèse. La première contribution de thèse est une investigation sur l'effet de la taille de l'échelle et la dimension d'un système de l'ACI qui est utilisé pour caractérisation des images satellite. Cela nous aide à choisir le framework de notre modèle de l'ACI pour extraire des caractéristiques. On propose deux groupes des descripteurs pour les images satellites haute résolution. Le premier groupe contient deux types des descripteurs qui sont basés sur les coefficients (les sources) de l'ACI ordinaire ou l'ACI topographique et le deuxième contient deux types des descripteurs qui sont basés sur les propriétés des vecteurs de base de l'ACI. En se basant sur notre expérience en l'ACI nous proposons un autre descripteur qui extrait les caractéristiques des lignes dans les images satellites. Finalement, les capacités des descripteurs proposés sont comparés grâce a une classification supervisée basée sur la machine à vecteurs de support.

**Mots clés :** Image Satellite, Descripteur, l'Analyse en Composantes Indépendantes

# Modeling, Extracting and Description of Intrinsic Cues of High Resolution Satellite Images: Independent Component Analysis Based Approaches

**ABSTRACT :** Sub-meter resolution satellite images, capture very detailed information, as for example, shape of buildings, roads, etc. The main purpose of the thesis is to propose descriptors for sub-meter resolution satellite images especially for those who contain geometrical or man-made structures. Independent component analysis (ICA) is a good candidate for this purpose, since previous studies demonstrated that the resulted basis vectors contain some small lines and edges, the important elements in the characterization of geometrical structures. As a basic analysis, a study about the effects of scale size and dimensionality of ICA system on indexing of satellite images is presented and the optimum dimensionality and scale size are found. There are two view points for feature extraction based on ICA. The usual idea is to use the ICA coefficients (ICA sources) and the other is to use the ICA basis vectors related to every image. Based on the first point of view, an ordinary ICA source based approach is proposed for feature extraction. This approach is developed and modified through a topographic ICA system to extract middle level features which leads to a significant improvement in results. Based on the other point of view, two methods are proposed. One of them uses the bag of words idea which considers the basis vectors as visual words. Second method uses the lines properties inside the basis vectors to extract features. Also, using the lines properties idea, another method is developed which directly detects the line segments in the images. Finally, the capabilities of proposed descriptors are compared through a supervised classification based on support vector machine (SVM).

**Keywords :** Satellite Images, Descriptor, Independent Component Analysis