# Semantic annotation of satellite images

Jean-BaptisteBordes & Henri Maître

Ecole Nationale Supérieure des Télécommunications (GET - Télécom Paris)
CNRS UMR 5141 LTCI, Paris, France.
CNES-DLR-ENST Competence Center
bordes@enst.fr
⋆

**Abstract.** We describe here a method to annotate satellite images with semantic concepts. The concepts used for the annotation are defined by the user. For each concept, the user provides a set of example images which will be used for learning. The method uses a first step of image processing which computes SIFT descriptors in the image. These feature vectors are quantized using an information theoretic criterion to define the optimal number of clusters. Then, a probabilistic modelization of the generation of this discrete collection of low-level features is set up. Each concept is associated to a mixture of models whoses parameters are learned by an Expectation-Maximization algorithm. The optimal complexity of the mixture is computed using a Minimum Description Length criterion. Finally, given a new image, a greedy algorithm is used to provide a segmentation whose regions are annotated with a concept. The performances of the system are evaluated on Quickbird images at 70cm resolution for a set of high-level concepts.

## 1 Introduction

During the last decades, the imaging-satellite sensors have acquired huge quantities of data. Now, the storage of image archives is getting even more enormous, due to data collected by a new generation of high-resolution satellite sensors. In order to increase the actual exploitation of satellite observations, it is of highest importance to set up systems which are able to selectively and flexibly access the information content of image archives. Most of actual content-based image-retrieval systems directly use symbolic values, and provide rather satisfying results for problems like "query by example images". However, these symbolic features cannot fully satisfy the expectations of the user, because the user thinks in term of semantic concepts ("industrial area", "residential suburb"), and not in terms of extracted symbolic values ("striped texture", "green area"). We aim in this article to cross the gap (often called "semantic gap" in the literature [15]) between these symbolic features and the semantic concepts. Indeed, we are interested in this paper to increase the ability of state-of-art systems to semantically annotate satellite images. Most of actual methods of semantic annotation

⋆ We thank LIAMA who provided us with the images and Mihai Datcu, Olivier Cappe and Michel Roux for fruitful discussions.

use *a priori* knowledge about the different kinds of concepts [11], this results in time-costly adaptations when the user wants to use other concepts. Our aim is to develop a statistical approach using a learning step from a set of example images for each concept. The method uses a first step of image processing which computes low-level features in the image. These low-level features are quantized using an information theoretic criterion to define the optimal number of clusters. Then, a probabilistic modelization of the generation of this discrete collection of low-level features is set up. For each concept, a mixture of models of generation of the low-level features is fitted on a data set consisting in a set of example images given by the user. The optimal complexity of the mixture is defined for each concept using minimum description length criteria. Finally, given a new image, a greedy algorithm is used to find the optimal set of semantic annotations by defining an initial segmentation and then merging regions iteratively. The performances of the system are evaluated on Quickbird images at 70cm resolution for a set of high-level concepts.

## 2 State-of-art of semantic annotation of images using learning

A brief start-of-art of semantic annotations of images is drawn her in two parts. We first present classical modelisations to link the image to the semantic labels, then we describe approaches which use annotation methods coming from text retrieval.

### 2.1 Modelisations of the problem of semantic annotation

The easier way to annotate an image is to link a whole image to a set of keywords meaning if the image checks or not, contains or not, the semantic concepts [16] ( "outside" or "inside", "vegetation", "tiger"). The actual systems proceed by training a classifier annotating automatically an image with semantic keywords. Recent works have been achieved to face the problem generally. The point is to introduce a set of latent variables corresponding to hidden states of the image. During the learning, a set of annotation is provided for each image, the image is segmented in a collection of regions and an unsupervised algorithm processes the whole database to estimate the joint distribution of these words and visual features. Given a new image, vectors of visual features are extracted, the joint probability is computed with these vectors, state variables are marginalised and the set of labels maximizing the joint density of semantic concepts and extracted features is computed.

Other methods aim to annotate only parts of the image, we separate them in two main axes: the methods which first use a segmentation of the image based on the extracted low-level features, and the methods which split the image in tiles which are annoted seperatly. In [10], large satellite images are splited in sub-images of size $64 \times 64$, and Gabor features are extracted from each sub-image. From a manual annotation of the tiles used as learning, a Gaussian Mixture Model is learned for each semantic label ("road","vegetation","parking lots"...).

Then, given a new image, the tiles are classified using Maximum Likelihood criteria. In [6], the image is first segmented using features calculated in the image. The learning set consists in images which are annotated globally, meaning that the words are not corresponding to a region. The vectors of low-level features extracted for each regions are then clustered using a "k-means" algorithm, the quantized vectors are called "blobs". A probabilistic modelisation of the generation of a blob for a region given an annotation is then defined and learned using the Expectation-Maximisation algorithm.

### 2.2 Application of text retrieval methods to semantic annotation of images

In [1], the authors propose three hierarchical probabilistic methods of generation of annotated data. The originality is that these methods are close to models traditionnaly used for textual documents. Two datasets are in fact generated: the features of regions which are initially segmented in the image, and the annotation of this data. This idea to use text-retrieval methods is exploited to a further extent in [9] where the authors first extract features in regions of the images and quantize them in order to work on a discrete collection. Then, the histograms of the quantized low-level features in each regions are used to apply "bag-words" methods like *Latent Semantic Analysis* and *Probabilistic Latent Semantic Analysis* in order to annotate the regions.

In this paper, we keep the idea to use methods which have proved to be successfull for text-retrieval to annotate the images. We choose to quantize features extracted in the image so that we may work on a set of discrete low-level features. However, we wish to avoid to apply a segmentation as a first step. Indeed, the regions may not be segmented correctly using directly the extracted low-level features.

## 3 Description of the image by a discrete collection of low-level features

### 3.1 Feature extraction in the image

The first step of the method is to characterize the images using SIFT descriptors. Since its introduction, SIFT (Scale Invariance Feature Transform) descriptor has stirred great enthusiasm among the computer vision community and it is now considered as a competitive local descriptor relatively to other descriptors such as filter banks ([7],[8]). At a pixel where the SIFT descriptor is calculated, four windows $4 \times 4$ are considered: each one is weighted by the gradient magnitude and by a Gaussian circular window with a standard deviation of typical values 4 or 6. Then, the sub-local orientation histograms for each one of these four windows are built: in our case, 4-bins histograms that cover the $[0, \pi]$ range (opposite orientations are considered to describe the same kind of objects). The histograms are then normalized. Each one of the 4 windows is thus described by a histogram of 4 values: the concatenation of these 4 descriptors produces a

local descriptor of size 16: the SIFT. In order to keep this descriptor rotation invariant, the neighbourhood of the detected feature is rotated so that the local gradient has an horizontal direction. Here, the SIFT descriptor is not extracted at Harris points but on a regular grid of step 8 pixels. Indeed, the goal here is not to make object matching but only to have a characterization of the image on a regular grid. Indeed, we assume that the SIFT descriptor extracts geometrical features which provide relevant informations about high resolution images.

### 3.2 Clustering of the SIFT descriptors

The SIFT descriptors thus extracted in the corpus are then quantized. We extract a subset of the feature vectors extracted in the corpus to learn a codebook. In order to determine the optimal number of clusters, we use an approach proposed in [5]. A Gaussian Mixture Model of the features is made and the Minimum Description Length criteria gives access to the optimal complexity of the model. On a dataset of 60000 SIFT vectors, an optimal number of 24 codewords was found on figure 1. We will note $N$ the optimal number of vectors, it will correspond to the size of the vocabulary of the discrete data.

The codewords being calculated, all the features of the corpus of images are quantized but their location is kept. Thus, we have a new set of images whose pixels are the index of the codewords, and whose value is in the set $\{1, ..., N\}$. An illustration is given on figure 2. Notice that we can't compare the values of these pixels, because the closeness between two indexes does not implye a proximity in the feature space.

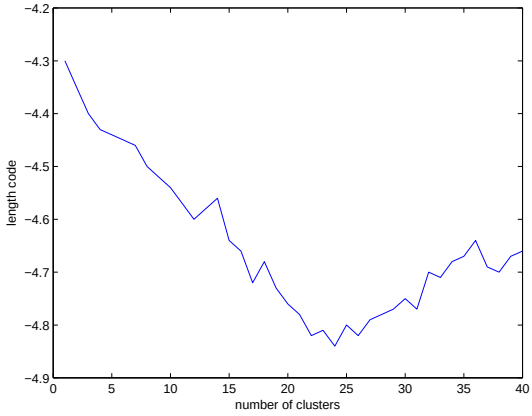
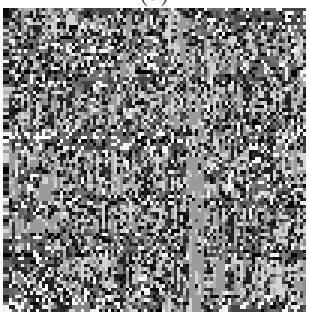



**Fig. 1.** Length of the code of the dataset depending of the number of clusters

(a)



(b)

**Fig. 2.** Result of quantization of the features extracted in the image: (a): raw image of size $1040 \times 1040$, (b): image of size $130 \times 130$ obtained after the SIFT vectors have been extracted and quantized. The grey values of the pixels of this new image correspond to the index of the codeword in the codebook. Pixels whose grey values are closed don't correspond necessarily to SIFT vectors which are closed in the feature space.

## 4 Probabilistic modelisation used

A pixel-like regular grid of discrete low-level features being extracted, we will call "pixels" the quantized low-level features labelled by the index of the codeword, a generative probabilistic modelisation is set up to link these pixels to concepts. Spatial information about the pixels lying inside the quantized pixels themselves, only the histogram of occurrences of the values of the pixels inside this region is modelised. The spatial repartition of the pixels inside a given region is not used. We limit thus the number of parameters in order to have a simple probabilistic modelisation and parameters which are easy to estimate. We sum up all the used notations in 4.

### 4.1 Global generative modelisation of an image

Let $n$ be the number of the different concepts ("dense urban area", "railway node") introduced to annotate the corpus of images we consider. Let also $I$ be an image of the corpus, and $O_I$ the pixels of the image. We define a semantic region $S_l$ by a concept and a 4-connex set of pixels $S_l(O_I)$ which are supposed

| Symbol | meaning |
|---|---|
| $N$ | number of codewords in the codebook |
| $m$ | number of regions found in an image |
| $n$ | number of concepts (defined by the user) |
| $s$ | concept index |
| $K_s$ | number of models in the mixture set-up for the concept $s$ |
| $N_i$ | number of pixels in the image of index $i$ in the dataset of a specific concept |
| $N_{ij}$ | number of pixels whose value is $j$ in the image of index $i$ in the dataset of a specific concept |
| $p_j^{ks}$ | probability of generation of a pixel of value $j$ in the model of index $k$ for the concept $s$ |
| $N_s$: number of pixels in region $S$ | |
| $j$ | index of value of a pixel |
| $O$ | set of pixels in an image whose values are the index of the codewords after quantization of the SIFT descriptors |
| $i$ | index of an image in a dataset of a specific concept |
| $k$ | index of a model in the mixture of a specific concept |
| $S_l$ | semantic region of index $l$ found in an image |
| $Z$ | latent variable corresponding the mixture model |
| $n_s$ | number of images given by the user for the learning of the mixture of the concept $s$ |
| $M_s$ | model of mixture for the semantic $s$ |
| $X_s$ | set of images given by the user for a concept $s$ |
| $X_{si}$ | $i$-th image of the image dataset $X_s$ |

**Fig. 3.** Used notations in the paper

to be generated by this region. We assume here that each pixel must be linked to a single region, it implies that the sets $S_l(O_I)$ have to define a partition of $O_I$.

In order to define the set of semantic regions $G_I = \{S_1, S_2, ..., S_m\}$ which describes best $I$ with the available annotations, $m$ standing for the number of regions found in the image, we choose the set $G_I$ which maximizes the likelihood $P(O_I|G_I)$. The sets of pixels $S_l(O_I)$ being assumed to be independant given the region which generated them, we get the following expression:

$$P(O_I|G_I) = \prod_{l=1}^{m} P(S_l(O_I)|S_l) \tag{1}$$

We don't use the probability $P(G_I)$, which corresponds to the prior of a configuration of regions in an image. Defining such a probability does not appear straightforward, and seems also difficult to learn, as the user does not provide a segmented image as dataset. However, we plan to suggest an expression for this term and to expose a learning procedure in future work.

### 4.2 Mixture model corresponding to a concept

For a given concept $s$, we define a mixture of models: a latent variable $Z$ is chosen which defines which parameters will be used to compute the probability of generation of the pixels between $K_s$ possible sets. $K_s$ can thus be interpreted as the complexity of the model for this concept. The parameters for this concept $s$ are the probabilities $p_j^{ks}$ of generation of the pixels of value $j$ for the model $k$ and the prior $\pi_{ks} = P(Z = k)$ of the latent variable. The total number of parameters for the concept $s$ is then $(N + 1) \cdot K_s$. More precisely, it is assumed here that a semantic region of index $i$ whose type of concept is $s$ generates the low-level features in the following way:

- The number $N_i$ of pixels in the region is generated with a Poisson's law of parameter $\lambda_s$
- The model $k$ is chosen with probability $\pi_{ks}$
- Each pixel of the region is chosen independantly from the others with probability $p_j^{ks}$, where $j$ stands for the type of the pixel.

Thus, $\{N_{i1}, ...., N_{iN}\}$ being the histogram of the value of the pixels generated by a region of index $i$ corresponding to the concept $s$, the probability of this generation is:

$$P(O = O_i|s, Z = k) = Poiss_{\lambda_s}(N_i)\pi_{ks} \prod_{j=1}^{N} (p_j^{ks})^{N_{ij}} \tag{2}$$

By conditionning on the possible values of the latent variable:

$$P(O = O_i|s) = \sum_{k=1}^{K_s} P(Z = k)P(O = O_i|s, Z = k)$$

By definition, $P(Z = k) = \pi_{ks}$. Moreover, $P(O = O_i|s, Z = k)$ is the probability of generation of the pixels given the concept and the latent variable. Replacing this probability by its expression in Equation 2, we can write:

$$P(O = O_i|s) = Poiss_{\lambda_s}(N_i) \sum_{k=1}^{K_s} \pi_{ks} \prod_{j=1}^{N} (p_j^{ks})^{N_{ij}} \tag{3}$$

The likelihood of a set of observations $O$ is thus expressed as a mixture of models over latent variables.

## 5 Model learning

We assume that, for each concept $s$, the user provides as learning data a set of sub-images $X_s$ corresponding either to semantic regions of type $s$ or to part of the region. We detail here how to learn the best mixture of models $M_s$ to maximize the likelihood of the learning dataset $X_s$ while avoiding overfitting. Notice that as these sub-images may only be a part of a region corresponding to

a concept, it is not possible to estimate the paramater $\lambda_s$ of the Poisson's law from this dataset. We suppose in this paper thus that the user can choose a value of $\lambda_s$ corresponding to three typical scales of regions: intermediate, large, and very-large. We plan to make an unsupervised learning scheme of this parameter in future work.

### 5.1 Expectation-Maximisation algorithm

In this section, we assume that the number of sets of parameters $K_s$ corresponding to a concept $s$ is fixed, the algorithm detailed in this section is used to determine the $K_s$ sets of parameters maximizing the likelihood $P(X_s|M_s)$. The parameters which maximize the likelihood of the learning data are estimated using an Expectation-Maximization algorithm (EM). Indeed, the EM algorithm introduces a hidden variable whose knowledge allows easy computation of the maximum of likelihood [3]. Here, the hidden variable can be chosen naturally as the latent variables $Z$ corresponding to the choice of the model for each image $i$, thus, we can have a tractable computation of the parameters corresponding to a local maximum of the likelihood of the learning data.

We introduce the latent variable $z_{ik}$ whose value is 1 if $Z = k$ for the image $i$ and the quantity $\gamma_k(O_i) = E_{Z|X_{si},M_s}(z_{ik})$, where $k$ stands for the index of the model, and $i$ stands for the index of the image. The interpretation of this quantity is the fitting of the model $k$ to the image $i$ relatively to the other models.

The EM algorithm draws the following scheme to find a local maximum of the likelihood:

- E step: computation of $\gamma_k(X_{si})$, for every model $k$ and every image $i$ by the Bayes inversion rule.

$$\gamma_k(X_{si}) = \frac{\pi_k \prod_{j=1}^{N} (p_j^{ks})^{N_{ij}}}{\sum_{m=1}^{K_s} \pi_m \prod_{j=1}^{N} (p_j^{ms})^{N_{ij}}}$$

  This expression is written as the probability of generation conditionnaly to the model $k$ over the likelihood of the observations in the image $i$. It seems logical as $\gamma_k(X_{si})$ stands for the interpretation of this quantity is the fitting of the model $k$ to the image $i$.
- M step: maximisation of $E_{Z|X_s,M_s}(logP(X_s, z|M_s))$. The Lagrange multipliers method is used to maximize this quantity. The following upgrade formula is thus found :

$$p_j^{ks} = \frac{\sum_{i=1}^{n_s} \gamma_k(X_{si}) N_{ij}}{\sum_{i=1}^{n_s} \gamma_k(X_{si}) N_i}$$

$$\pi_{ks} = \frac{\sum_{i=1}^{n_s} \gamma_k(X_{si})}{n_s}$$

  Notice that the estimation of $p_j^{ks}$ corresponds logically to the ratio of the occurrences of the pixels $j$ in all the images by the total number of pixels, weighted by the quantities $\gamma_k(X_{si})$. The prior has also a very intuitive expression as the ratio of the sum of the quantities $\gamma_k(X_{si})$ over the number of images in the dataset for the concept $s$.

### 5.2 Minimisation of stochastic complexity

We wish to find the optimal complexity of the model for the fitting of the learning data, meaning the optimal value of $K_s$. Indeed, the more complex is the model, the highest is the likelihood, but it may result in overfitting. The model selection is a classical problem in pattern recognition, and a lot of criteria have been suggested. In this paper, we select the model by using the algorithm of minimisation of the stochastic complexity [12]. This principle, introduced by Rissanen, assumes that the best model is the one which enables the shortest coding of the learning data. If $M_s$ is noted as the model used to describe the learning data $X_s$ for the concept $s$, the length of the code can be separated in two terms:

$$C(X_s, M_s) = C(X_s|M_s) + C(M_s) \tag{4}$$

$M_s$ being a set of real parameters, the length of the code to code should be infinite, but, as the parameters are estimated with a finite number if samples, Rissanen suggests in [13] the following expression for $C(M_s)$:

$$C(M_s) = \sum_{i=1}^{n_s} \frac{\alpha}{2} log(N_i)$$

where $\alpha$ stands for the number of parameters to code.

As for the probability of generation of the pixels, the property: $\sum_{i=1}^{N} p_i^{js} = 1$ is checked for each $j \epsilon \{1, ..., k_s\}$ since $p_i^{js}$ is a set of probabilities. Thus, $N - 1$ parameters need to be coded for each model of index $k$ for the generative probabilities of the mixture.

And as for the prior $\pi$, which stands for the prior of the values of the latent variables, we have the relation: $\sum_{i=1}^{K_s} \pi_{is} = 1$. Thus, $K_s - 1$ parameters have to be coded.

For the term $C(X_s|M_s)$, Shannon proposes the following formula, linking directly the length of coding of a sequence to its probability of apparition ([14]):

$$C(X_s|M_s) = -logP(X_s|M_s)$$

By using the expression of $P(X_{si}|M_s)$ detailed in Equation 3, the Equation 4 can be written as:

$$C(X_s, M_s) = -\sum_{i=1}^{n_s} \sum_{j=1}^{k_s} \pi_{js} \prod_{i=1}^{N_i} p_{m(l_i)}^{js} + \sum_{i=1}^{K_s} \frac{N-1}{2} log(\sum_{k=1}^{n_s} \gamma_k N_k) + \frac{K_s - 1}{2} log(n_s) \tag{5}$$

We have to minimize this expression on the models which have less than $n_s$ sets of parameters.

### 5.3 Complete algorithm

The EM algorithm described above determines the maximum of likelihood for a fixed number of models. In order to find the optimal mixture for a number of models ranging from 1 to $n_s$, we apply the EM algorithm for a number of models varying from 1 to $n_s$, and we compute the stochastic complexity defined in 5. The chosen model $M_s$ is the one which corresponds to the minimal stochastic complexity:

$$M_s = argmin_M[C(X_s, M_s)]$$

## 6 Procedure of semantic annotation

### 6.1 Method of semantic annotation of a test image

$I$ being an image to annotate with a given set of concepts, and the parameters of the mixtures for each concept being calculated, finding the optimal set $G_I = \{S_1, ..., S_{m_I}\}$ in the set $\mathbf{G}$ of all the possible configurations of semantic regions in the image is a very complex problem. Indeed, an exhaustive search of all possible configurations is impossible, because of the huge cardinal of $\mathbf{G}$. Thus, we detail here a tractable algorithm which explores a path in the set $\mathbf{G}$ in the following way: we start from an initial and complex configuration, and then we simplify it by merging regions iteratively, choosing at each step the best fusion at the sense of the maximum likelihood until there remains only one region for the whole image. This is a greedy algorithm as the best configuration is chosen at each step but no backtrack is allowed, and the algorithm may be stuck in a local maximum of the likelihood.

More precisely, we proceed with the following steps:

- Initialisation of the algorithm: each pixel $l$ of the image $I$ is linked to a concept considering its type and its neighbourhood $NE(l)$ by choosing the concept $s$ minimizing the quantity (cf Equation 3):

$$P(NE(l)|S = s) = \sum_{j=1}^{k_s} \pi_{js} \prod_{lf \epsilon NE(l)} p^{js}_{m(lf)}$$

  To define the neighbourood, we consider a square centered on $l$ whose edge is of size $t$ and take $NE(l)$ as the set of pixels wich are in this square. Then, semantic regions are created as 4-connex sets of pixels to make an initial set of semantic regions $G_0$. The likelihood $P(X_I|G_0)$ si calculated (cf Equation 1). We notice that the larger the value of $t$ is, the fewer regions there are in $G_0$, and so, the faster is the algorithm.
- Let $i$ be the number of the step of iteration the loop, as long as the number of regions is more than 1:
  - We consider all the possible merging of adjacent semantic regions.
    - For each possible merging, we consider the $n$ cases where the merged region has the concept of index $s$, $s \epsilon \{1, ..., n\}$. Then, for each case, if semantic regions are adjacent and have the same concept, they are merged. We compute the likelihood for all these resulting configurations.

- The configuration maximizing the likelihood is kept and noted $G_i$.

– We keep the configuration maximizing the likelihood among all the found $G_i$ at each step.

The number of iterations of the loop is less than $card(G_0)$, the number of semantic regions found during the "initial guess". Indeed, for each iteration, at least two regions are merged, we thus have: $card(G_i) \leq card(G_i) - 1$, this ensures that the algorithms ends within a finite number of iterations.

### 6.2 Experiments

We made evaluations of this method on a database of Quickbird images of Beijing at 0,7cm of resolution. The database contains 16 images of size $16000 \times 16000$ pixels and covers thus a square whose edge is of size 11 kilometers. We use the followings concepts: dense urban area, residential housing area, industrial area, railway node, residential area, commercial area, working area, wasteland, fields, greenhouses, water.

**Classification of sub-images** We made classifications of extracted sub-images to have quantitative results. The database contains around 150 sub-images as shown on figure 4 corresponding each one to a concept among those listed above. We proceeded by cross-validation with 80% of learning and 20% of test. The size of the images vary from one $400 \times 400$ to $1000 \times 1000$. We get a result of 96,4% of good classification. We intend to evaluate this algorithm on a bigger databases of different cities in future work.

**Segmentation of large images** We performed semantic annotation of test sub-images using the chosen concepts. The models are learnt for each concept using our database of examples sub-images. For a test image of typical size $6000 \times 6000$, the whole process of feature extraction, quantization and annotation lasts around 5 minutes on a 3,2GHz processor. The results are satisfying as we can see on figure 5. Notice that as the algorithm has to annotate the whole image, the concepts have to cover all the possible kinds of areas in the database. We plan to add a reject class for areas found in the test images not correponding to any kind of example images given in the learning dataset.

## 7 Conclusion

We presented in this paper a probabilistic approach to semantically annotate images of a corpus with concepts using a learning step. For each concept, the user provides a data set and a mixture of models is fitted on this data by adapting the complexity of the model to this dataset. Inside a region annotated by a given concept, only the histogram of the value of the pixels is used. This approach can be linked to probabilistic methods using "bag-of-words" description of text documents [4] [2]. This is a strong hypothesis because the spatial relationships between pixels is overlooked and only the spatiality between the pixels

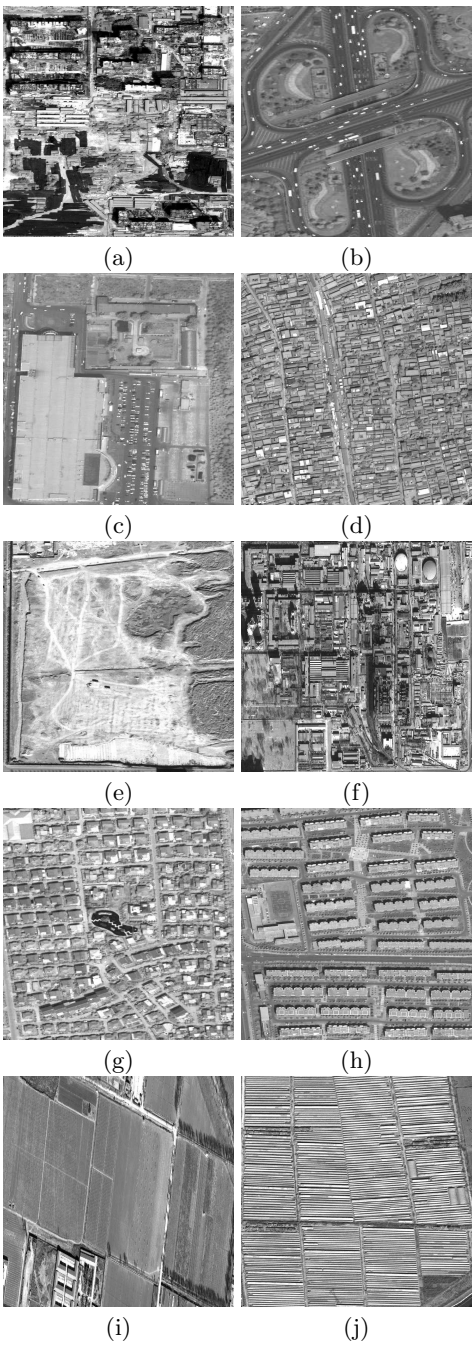



**Fig. 4.** Examples of semantics considered for evaluation: (a): building site, (b): railway node, (c): commercial place, (d): dense urban area, (e): wasteland, (f): industrial place, (g): housing area, (h): residential big buildings area

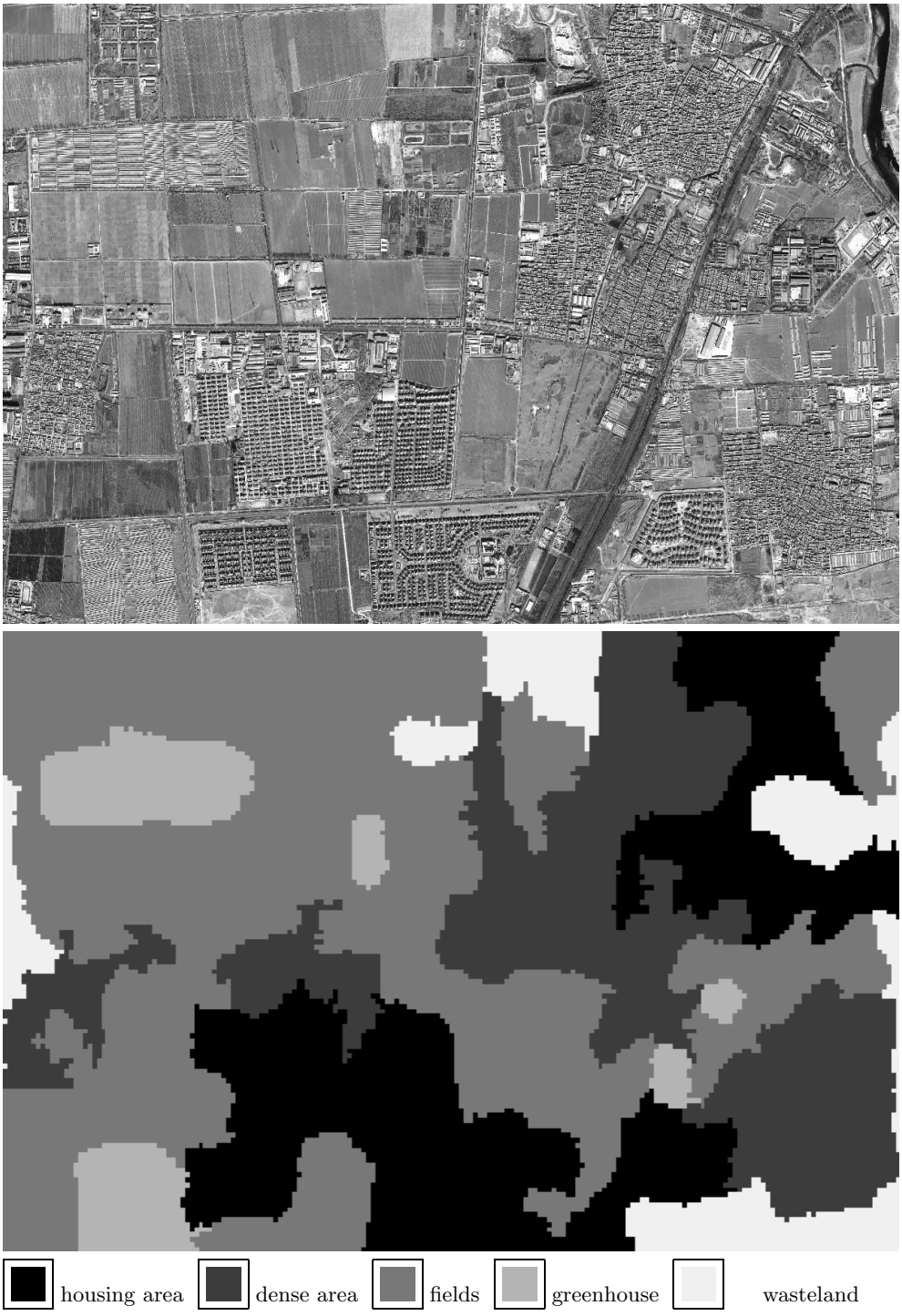



**Fig. 5.** Results of segmentation on a Quickbird image of Beijing of size 7000 × 5000 pixels

of the raw image captured during the feature extraction is thus considered. This simplification choice enables us to limit the number of parameters in order to have a simple probabilistic modelisation and parameters which are easy to estimate. This method has proved to be efficient for segmentation of large images by experimental results.

## References


1. D. Blei and M. Jordan. Modeling annotated data. 2002.
2. D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
3. A. P. Dempster, N. M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 1(39):1–38, 1977.
4. T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR-99, pp 35-44*, 1999.
5. H. Maitre I. Kyrgyzov and M. Campedel. Kernel mdl to determine the number of clusters. submitted to MLDM 2007.
6. J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of the 26th international ACM SIGIR Conference*, pages 119–126, 2003.
7. David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
8. K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *CVPR*, 2003.
9. F. Monay and D. Gatica-Perez. On image auto-annotation with latent space models. In *Proceedings ACM International Conference on Multimedia, Berkeley*, pages 271–274, November 2003.
10. S. Newsam, L. Wang, S. Bhagavathy, and B.S. Manjunath. Using texture to analyze and manage large collections of remote sensed image and video data. *Applied optics*, 43(2):210–217, Jan 2004.
11. A. Puissant and C. Weber. Une démarche orienté-objets pour extraire des objets urbains sur des images thr. *Bulletin de la Société Francaise de Photogrammétrie et Télédétection*, 43(3):993–1022, 2004.
12. J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.
13. J. Rissanen. *Stochastic complexity in statistical inquiry*. World Scientific, 1989.
14. C. Shannon. A mathematical theory of communication. *Bell Syst Technology*, 27:379–423, 1948.
15. Jonathon S.Hare, Paul H. Lewis, Peter G.B Enser, and Christine J. Sandom. Mind the gap: another look at the problem of the semantic gap in image retrieval. *Management and retrieval*, 6073, 2006.
16. Nuno Vasconcelos and Gustavo Carneiro. Formulating semantic image annotation as a supervised learning problem. *CVPR*, 5:163–168, 2005.