

Copyright 2008 SPIE and IS&T. This paper was published in the proceedings of IS&T/SPIE 20th Annual Symposium on Electronic Imaging and is made available as an electronic reprint (preprint) with permission of SPIE and IS&T. One print or electronic copy may be made for personal use only. Systematic or multiple reproduction, distribution to multiple locations via electronic or other means, duplication of any material in this paper for a fee or for commercial purposes, or modification of the content of the paper are prohibited.

Mixing Geometric and Radiometric Features for Change Classification

Alexandre Fournier, Xavier Descombes & Josiane Zerubia
ARIANA project team, INRIA/CNRS/UNSA
2004 route des lucioles,
06 902 Sophia Antipolis Cedex, France
firstname.lastname@sophia.inria.fr
<http://www-sop.inria.fr/ariana>

ABSTRACT

Most basic change detection algorithms use a pixel-based approach. Whereas such approach is quite well defined for monitoring important area changes (such as urban growth monitoring) in low resolution images, an object based approach seems more relevant when the change detection is specifically aimed toward targets (such as small buildings and vehicles).

In this paper, we present an approach that mixes radiometric and geometric features to qualify the changed zones. The goal is to establish bounds (appearance, disappearance, substitution ...) between the detected changes and the underlying objects. We proceed by first clustering the change map (containing each pixel bitemporal radiosity) in different classes using the entropy-kmeans algorithm. Assuming that most man-made objects have a polygonal shape, a polygonal approximation algorithm is then used in order to characterize the resulting zone shapes. Hence allowing us to refine the primary rough classification, by integrating the polygon orientations in the state space. Tests are currently conducted on Quickbird data.

1. INTRODUCTION

As the availability of high resolution satellite data is growing at a fast pace, new applications in geoscience for civilian as well as military parties are being developed. Lately, multitemporal data samples have become available in a significant amount, opening new ways to explore in the field of urban growth, target detection and/or tracking.

However, when it comes to change detection, though high definition images allow detection of changes as tiny as vehicle appearances, many parasitic details (such as misregistration errors, surface diffusion differences due to different daylight expositions, shadow directions or parallax errors) are also detected. Change detection approaches heavily relying on pixel-based methods clearly reach their limits, demonstrating the need for more global algorithms.

In this paper, we describe a change classification algorithm based both on radiometric and geometric data. First, we show the preprocessing steps that successively provide a change mask from a couple of panchromatic satellite images and a first (pixel based) classification.

Then, we present a polygonal approximation algorithm designed to characterize the formerly obtained changed zones. Eventually, the polygons are classified according to their orientation, as well as the radiometric couples associated with the zones they characterize.

2. PREPROCESSING STEPS

The goal of this application is to detect and classify the changes between a pair of remotely sensed images of the same scene, but taken at different times (see Figure 1). We focus on man-made objects.

2.1 Qualifying changed/unchanged pixels

Firstly, we try to get rid of the false positives that are due to general illumination differences. To reach that goal, we try to determine a “non change” characteristic between times T_1 and T_2 . (*i.e.* determine an optimal linear transform between the two image radiometries). Simple approaches like Principal Component Analysis (PCA) of the graph representing the intensities at T_1 on the x-axis and at T_2 in y-axis can be biased because of important outlier zones (changes). Indeed, a principal component is the vector \mathbf{w}_1 that maximises the variance of the distribution projection on its axis:

$$\mathbf{w}_1 = \arg \max_{\|\mathbf{w}\|=1} E \left\{ (\mathbf{w}^T \mathbf{x})^2 \right\} \quad (1)$$

where the \mathbf{x} are the population. This leads us to use an iterative approach described in¹¹ :

1. weight each pixel-pair on the histogram according to the distance that separates it from the principal component;
2. recompute the new principal component taking those weights into account.

Though that method yields good results, it does not provide a connexity driven component. However this component is essential in order to link a pixel to its context. Hence we can either regularize the change map via a Markov Random Field, or directly include a MRF model in the algorithm itself as did the author of¹¹ . We choose the first option since the MRF component can lead in some cases to the algorithm divergence. We use a simple Ising model¹ for the *a priori* probability. The optimisation is performed using a simulated annealing algorithm³ .

2.2 Clustering the changed zones

Once the changed zones are detected, the relevant changes are to be discriminated from the irrelevant ones. The obvious criterion that we can use for clustering the changed pixels are their intensity pairs. As the usual classification algorithms (for instance the *k-means* algorithm) require an *a priori* knowledge of the cluster number, we used the *entropy k-means* approach proposed by the authors of⁹ that provides an automatic evaluation of this number.

Let \mathbf{x}_j represent the distribution to be clustered, c_i be the clusters and \mathbf{y}_i the centroids associated to the clusters, then the *k-means* algorithm minimises the following energy term:

$$U = \sum_{i=1}^k \sum_{j \in c_i} \frac{T(\mathbf{x}_j - \mathbf{y}_i) \Sigma^{-1}(\mathbf{x}_j - \mathbf{y}_i)}{2} \quad (2)$$

Where Σ is the covariance matrix of the \mathbf{x}_i . The entropy *k-means* algorithm, on the other hand, minimises the following term:

$$U = \sum_{i=1}^k \sum_{j \in c_i} \left[\frac{T(\mathbf{x}_j - \mathbf{y}_i) \Sigma^{-1}(\mathbf{x}_j - \mathbf{y}_i)}{2} - \alpha \cdot \log(p_i) \right] \quad (3)$$

where α decreases geometrically in order to avoid having a single class dominating the whole distribution.⁶ This new term favours the erasing of the clusters that contain too few elements and are close to bigger clusters in the state space. Again, after this step, the results are regularised with a new MRF using the Ising model.

The original images are shown on Figure 1 and the results of these preprocessing steps are shown on Figure2.

2.3 Polygonal approximation

Having clustered the changed data, we have now a series of connex zones that show a change label. The next step is the extraction of their geometrical information.

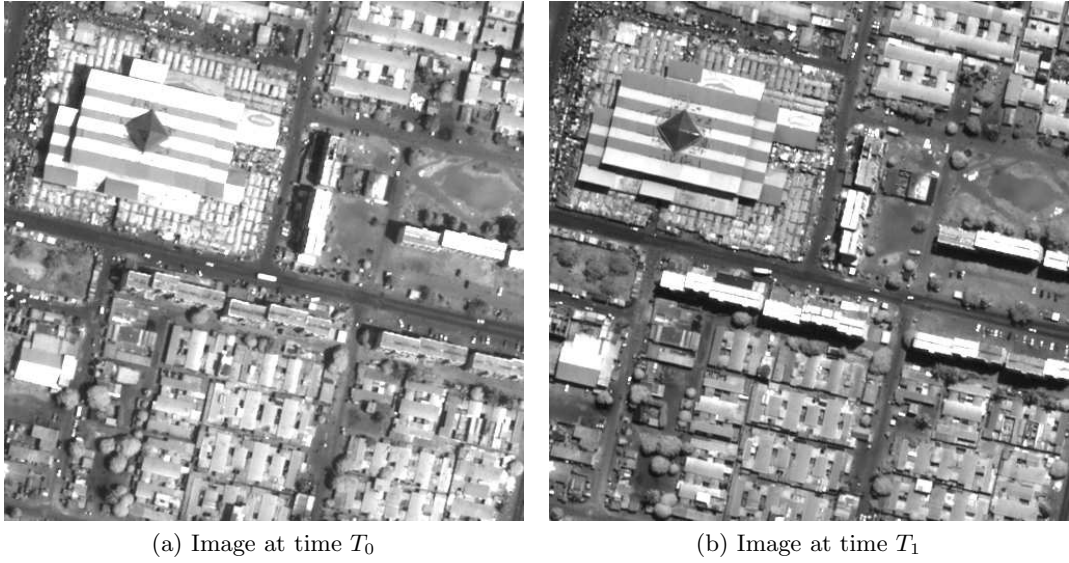
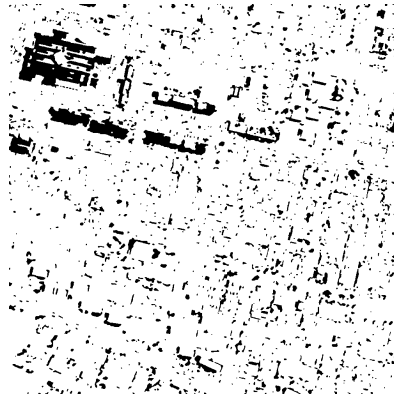


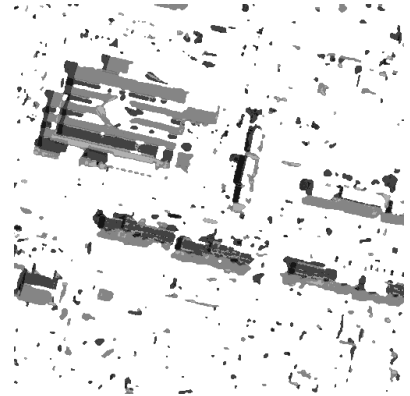
Figure 1: Original images, Quickbird, ©Spot Image, courtesy of the French Defense Agency (DGA).



(a) Change probability from the iterative PCA algorithm



(b) Change label provided by the MRF regularisation



(c) Classes obtained from the *entropy k-means* algorithm

Figure 2: Preprocessing steps

2.3.1 Definitions

Polygonal approximation is linked with two classical problems:⁴

Min- ε problem given a curve \mathcal{C} , and a number n of segments, find the polygonal curve P that minimises the approximation error E .

Min- n problem given a curve \mathcal{C} , and an approximation error ε , find the polygonal curve P with the minimum number of segments n , and with an approximation error that does not exceed ε .

Here, we address the *Min- ε problem*.

Let \mathcal{C} be a closed contour that we try to approximate with a polygon $P = \{P_1, \dots, P_{n+1}\}$ (with $P_{n+1} = P_1$). We define $\Omega_{\mathcal{C}}$ and Ω_P as the inner areas delimited by \mathcal{C} and P , respectively.

When it comes to polygonal approximation, the error is usually defined according to the L_2 norm,^{4,5} which can be addressed using B-splines.¹⁰

$$E = \int_{\mathcal{C}} d(P, M(s)) ds$$

where M belongs to \mathcal{C} , s is its arc length on \mathcal{C} and $d(P, M)$ the minimum distance from M to the polygon P . This definition of the error leads to a very good polygonal approximation of the contour, however, it is very sensitive to outliers or irregularities in the contour. Our goal is to qualify mainly the zone surrounded by the contour than the contour itself. Therefore, we propose an approximation error more robust to outliers. Thus, we introduce the following error approximation measure based on the L_1 norm between surfaces:

$$E = \mu(\Omega_{\mathcal{C}} \text{ XOR } \Omega_P) \tag{4}$$

where μ is the Lebesgue measure and XOR is the exclusive “or” (*e.g.*: A point belonging to $\Omega_{\mathcal{C}}$ and not to Ω_P belongs to $\Omega_{\mathcal{C}} \text{ XOR } \Omega_P$). A more intuitive representation of this approximation error is given in Figure 3.

Since it is rather related to an area than a contour, the use of this distance seems to fit quite well to a polygonal approximation. Moreover, we notice that this distance is not as sensitive as the usual distance to outliers of the contour. Therefore, it seems more adapted to our problem which is shape fitting with a polygon.

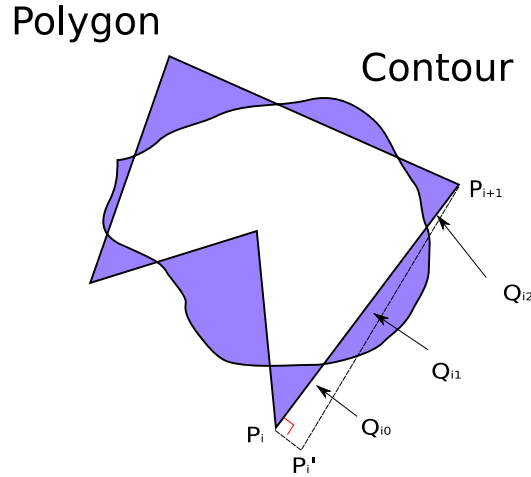


Figure 3: A polygone and its L_1 -norm approximation error. The error is the measure of the colored area.

In the next sub-section, we propose to find a local minimum of this error using a gradient descent algorithm.

2.3.2 Computing the gradient

We expect to converge to a local solution for the polygonal approximation defining a gradient descent over the approximation error. The state space is composed by the 2D coordinates of the N points. However, we show later that for each point, the gradient direction can be computed independently.

First, let us assume that the polygon P is anticlockwise directed. For each $i \in \{1, \dots, N\}$ we define the segment $S_i = [P_i, P_{i+1}]$ and $S_0 = S_N = [P_{N+1}, P_1]$. Moving P_i around its current position only influences segments S_{i-1} and S_i . The energy variation due to moving point P_i can then be subdivided in two terms corresponding to the motions of the two segments. We will focus on dE_R , the error variation due to changes on segment S_i (a similar computation can be easily applied on segment S_{i-1}).

For simplification sake, we consider that S_i has only strict intersections with \mathcal{C} (no tangent points ...). We order those intersections $I_{i,j}, j \in \{1, \dots, p\}$ by their distance to P_i . Let us suppose that P_i is outside $\Omega_{\mathcal{C}}$ and consider moving P_i orthogonally to S_i outwards the contours, to point P'_i . This move creates $p+1$ zones denoted as $Q_{i,k}$ for $k \in \{0, \dots, p\}$, whose influence alternatively increases and decreases the approximation error (see figure 3).

Therefore, if we define $A_{i,j} = \mu(Q_{i,j})$ as the area of $Q_{i,j}$, the total influence on S_i can be expressed by the following equation:

$$\Delta E_R = - \sum_{j=0}^p (-1)^j A_{i,j} \quad (5)$$

On the other hand, if $P_i \in \Omega_{\mathcal{C}}$, then this influence is the exact opposite. We define γ such that $\gamma(P) = 1$ if $P \in \Omega_{\mathcal{C}}$ and $\gamma(P) = -1$ if not.

The expression of ΔE_R becomes:

$$dE_R = \gamma(P_i) \cdot \sum_{j=0}^p (-1)^j A_{i,j} \quad (6)$$

Let us now consider that the move from P_i to P'_i is infinitesimal, then the $Q_{i,k}$ are trapezoids (except for $Q_{i,p}$ which is a triangle) whose area can be directly computed from distance $d(P_i, P'_i)$ and the positions of intersections $I_{i,k}$ of S_i with \mathcal{C} . Let \mathbf{n}_i be the inward normal of segment S_i . Straightforward computation leads to:

$$dE_R = \frac{\gamma(P_i) L_i dy_i}{2} \left(1 + 2 \sum_{j=1}^{p-1} [(-1)^j (1 - x_{i,j})^2] + (-1)^p (1 - x_{i,p})^2 \right) + o(dy_i) \quad (7)$$

$$\text{where } \begin{cases} L_i & \text{the length of segment } S_i \\ x_{i,j} & = \frac{|P_i I_{i,j}|}{L_i} \\ dy_i & = \langle \mathbf{P}_i \mathbf{P}'_i | \mathbf{n}_i \rangle \end{cases}$$

Eventually, we measure the influence on S_i of moving P_i by a vector $d\mathbf{P}_i$ in the general reference:

$$dE_R = \langle d\mathbf{P}_i | \mathbf{n}_i \rangle \frac{dE_R}{dy_i} \quad (8)$$

$$dE_R = \frac{L_i \langle d\mathbf{P}_i | \mathbf{n}_i \rangle \gamma(P_i)}{2} \left(1 + 2 \sum_{j=1}^{p-1} [(-1)^j (1 - x_{i,j})^2] + (-1)^p (1 - x_{i,p})^2 \right) \quad (9)$$

Similarly, the influence dE_L of moving P_i on S_{i-1} is defined as follows:

$$dE_L = \frac{L_{i-1} \langle d\mathbf{P}_i | \mathbf{n}_{i-1} \rangle \gamma(P_i)}{2} \left(1 + 2 \sum_{j=1}^p [(-1)^{p+1-j} x_{i-1,j}^2] \right) \quad (10)$$

The \mathbf{t}_i component (with $\mathbf{t}_i = \mathbf{P}_i \mathbf{P}_{i+1} / |\mathbf{P}_i \mathbf{P}_{i+1}|$) of $d\mathbf{P}_i$ results in adding second order terms to E_R , which are not taken into account for computing the gradient. For the same reason, we do not consider the \mathbf{t}_{i-1} component of $d\mathbf{P}_i$ when computing dE_L .

Finally, we get the expression of the P_i -related components of the gradient:

$$dE = dE_L + dE_R \quad (11)$$

$$\begin{aligned} \frac{dE}{d\mathbf{P}_i} = & \frac{\gamma(P_i)}{2} \cdot \left[L_{i-1} \left(1 + 2 \sum_{j=1}^p [(-1)^{p+1-j} x_{i-1,j}^2] \right) \mathbf{n}_{i-1} \right. \\ & \left. + L_i \left(1 + 2 \sum_{j=1}^{p-1} [(-1)^j (1 - x_{i,j})^2] + (-1)^p (1 - x_{i,p})^2 \right) \mathbf{n}_i \right] \end{aligned} \quad (12)$$

2.3.3 Naive initialization

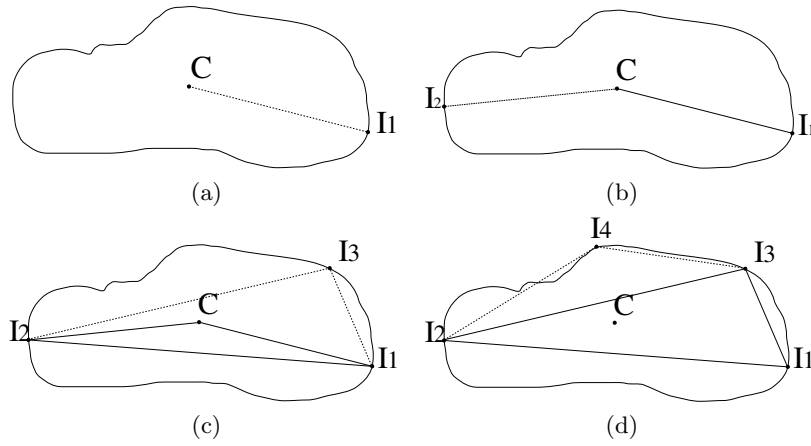


Figure 4: Naive initialization, Points need to be reordered in order to provide a non-crossed polygon

In order to find the minimum of E , we need to initialize the polygon on a pertinent position. We decide to use a naive initialization (see figure 4), that fits most of the cases found in our particular application. Let C be the barycentre of \mathcal{C} . Then:

- P_1 is the farthest point from C in \mathcal{C} ,
- P_2 is the farthest point from the segment $[C, P_1]$ in \mathcal{C} ,
- P_3 is the farthest point from the triangle (C, P_1, P_2) in \mathcal{C}
- P_4 is the farthest point from the triangle (P_1, P_2, P_3) in \mathcal{C}

For this application, we focused on quadrilaterals but it might be adapted for more complex polygons adding the iterative step:

- P_{n+1} is the farthest point from the ordered polygon (P_1, \dots, P_n) in \mathcal{C} .

Nevertheless, this last step needs to be tested yet .

This initialization can be quite sensitive to outliers in the contour, however it returns a good distribution for the P_i . Moreover, this sensitivity is balanced by the robustness to outliers of the norm that is minimised afterwards.

2.3.4 Results

In order to observe the behaviour of the algorithm as well as to check the initialization, we have designed a simple test case on a square image that presents some outliers. The effect of the convergence of the algorithm is presented in Figure 5.

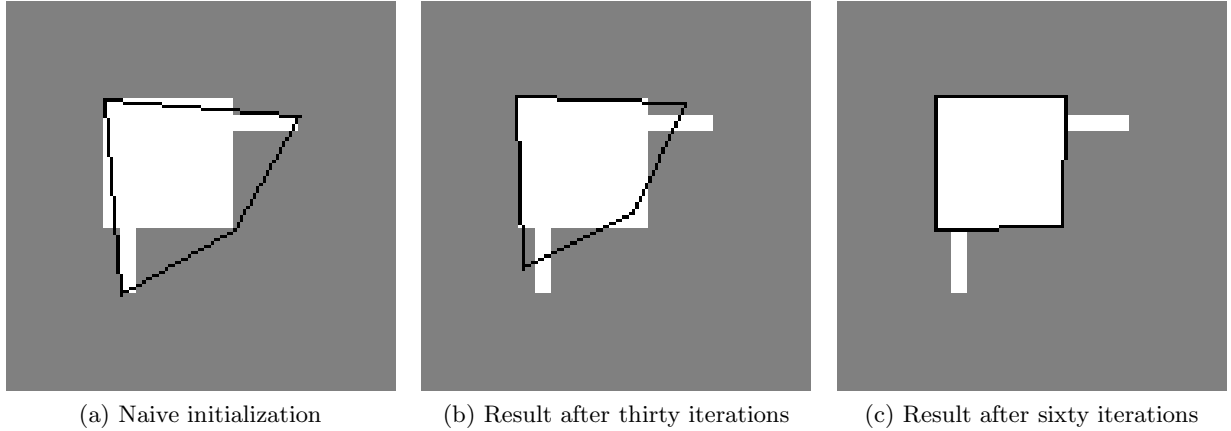


Figure 5: Artificial image

As we can see, the P_i are initialized on the outliers, but after convergence, the polygon fits well to the square.

As we stated before, the final goal is to find the shadows and urban planning directions. We first applied the algorithm on the changes that are related to shadows. The results are shown on Figure 6. As we can see, most of the detected quadrilaterals present two principal directions. The first one corresponds to the building direction (and often corresponds to the city orientation). The second one corresponds to the shadow direction. Notice that the initialization does not let us guess the correct direction for the sun orientation, however, this is corrected after convergence.

3. POLYGONAL CLASSIFICATION

The results yielded by the polygonal approximation algorithm (see Figure 6) show that the quadrilaterals resulting from appearing or disappearing shadows present many common aspects in terms of orientation. We noticed that, in urban areas (mostly in regular cities), two principal directions were found repeatedly. The first one corresponded to one of the city grid direction (this is obviously present only on fairly modern cities) and was present in two sides of the quadrilaterals. The second direction is given by the sun direction. This motivates the use of the detected quadrilaterals orientations as a criteria for a new classification step.

3.1 Metric of the orientation space

The state space is composed of the four quadrilateral orientations, which is $[0, 2\pi[$. Obviously, the euclidian metric can not be adopted for such a state space, but some basic results of directional statistics⁸ can be applied:

In order to adapt a clustering algorithm such as the kmeans algorithm, the minimised energy and therefore consistent concepts of *distance* and *mean* must be defined. Indeed, whereas the “classic” *K-means* algorithm minimises the energy:

$$E_{Euclidian} = \sum_{i=1}^k \sum_{P_j \in S_i} (P_j - \mu_i)^2$$

The underlying euclidian distance does not yield any sense in a circular space (as $[0, 2\pi[$). Therefore, the energy must rely on a more convenient distance.

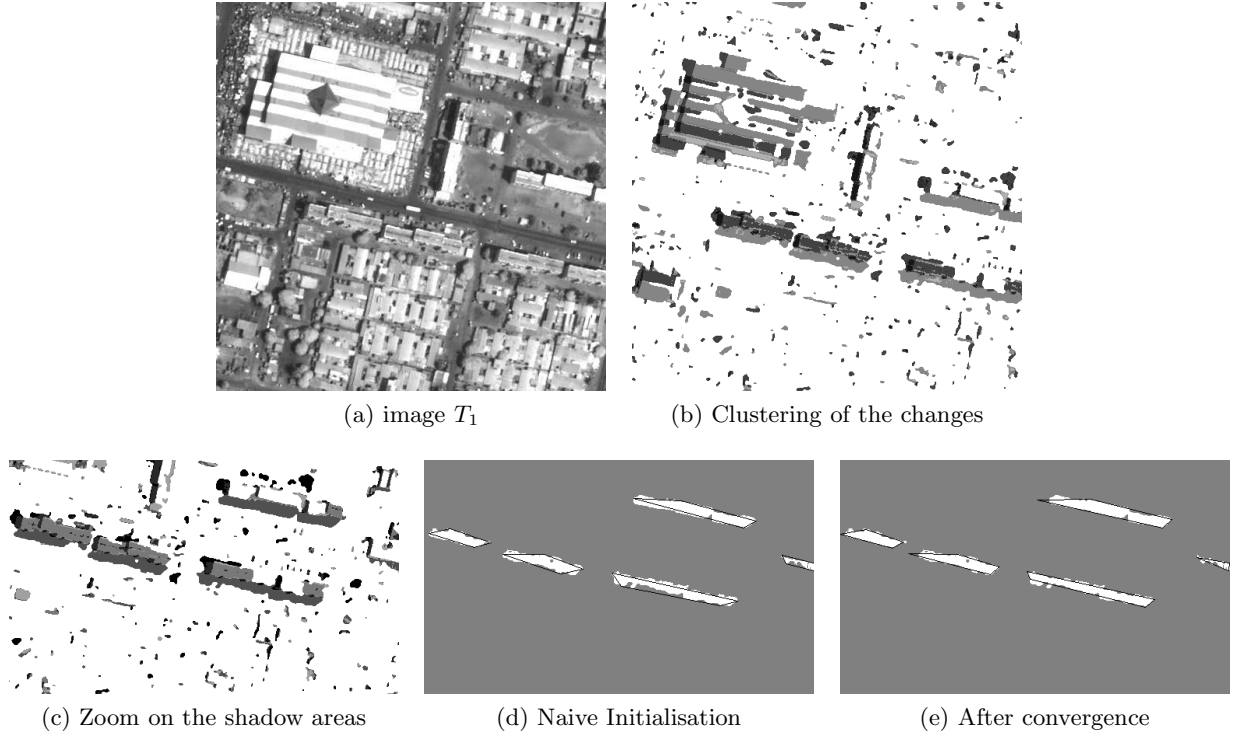


Figure 6: Polygonal approximation applied to real image pairs. At the top, detecting the shadows present at T_0 . At the bottom, detecting the shadows present at T_1

$$E_{Circular} = \sum_{i=1}^k \sum_{P_j \in S_i} d^2(P_j - \mu_i)$$

Directional statistics usually propose the chordal distance (here noted d_c) for directional data analysis (see figure 7), which sticks quite well to the intuitive concept of distance in a circular space.

We define R the function that associates an element of $[0, 2\pi[$ with its representative in the unit circle:

$$R(\alpha) = (\cos(\alpha), \sin(\alpha)) \quad (13)$$

Then, given $(\alpha)_i, i \in T = \{1, \dots, n\}$ a set of elements in $[0, 2\pi[$, the energy term :

$$E(x) = \sum_{i \in T} d_c^2(\alpha_i - x)$$

is minimised^{7,8} by :

$$M(\alpha_i, i \in T) = R^{-1} \left(\overline{\sum_{i \in T} R(\alpha_i)} \right). \quad (14)$$

Where $\bar{\mathbf{y}} = \mathbf{y} / \|\mathbf{y}\|$ ($\|\mathbf{y}\|$ being the Euclidian norm of \mathbf{y}). The expression 14, thus corresponds to the “mean” related to the chordal distance. Therefore, using those new definitions for the distance and the mean, we are now able to redefine a *K-means* algorithm. The two fundamental parts of this algorithm are therefore:

Step A for each sample x_j , associate x_j with the Cluster S_i , which centroid μ_i minimizes the chordal distance to x_j . *i.e.*:

$$x_j \in S_i, \mu_i = \underset{\mu_k}{\operatorname{argmin}} d_c(x_j, \mu_k)$$

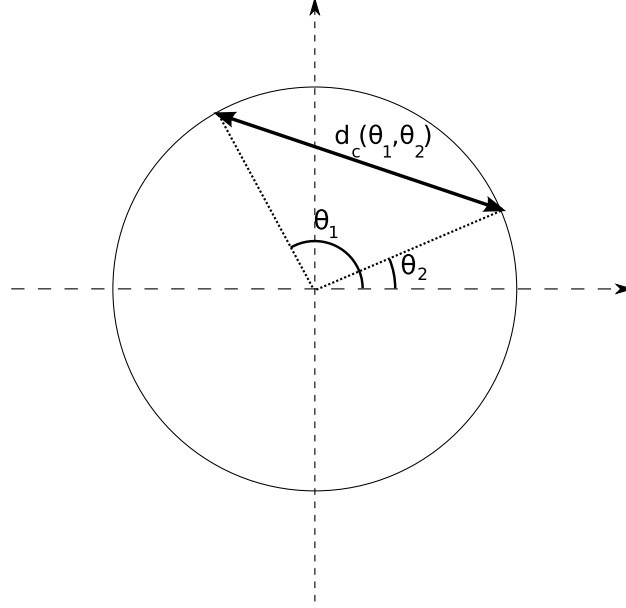


Figure 7: Chordal distance between two angles

Step B for each cluster S_i , move μ_i to the “mean” of the samples that were associated with S_i . *i.e.*:

$$\mu_i := M(x_j, x_j \in S_i)$$

As for the “classic” *K-means* algorithm,² each step minimises the energy E_{d_c} : directly for Step A, and through our definition of the “mean” for step B. Therefore this algorithm also converges to a local minimum of E_{d_c} .

3.2 Multi-orientation Vector

Since a detected quadrilateral presents 4 sides, its state space in terms of orientation is actually $[0, 2\pi]^4$. We directly adapt our mean and distance to the 4 dimensional case. We define the distance between two elements \mathbf{x} and \mathbf{y} , lying in $[0, 2\pi]^n$, as:

$$d_{mc}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n d_c(x^i, y^i)$$

x^i being the i^{th} coordinate of \mathbf{x} . Therefore, the element μ that minimises $\sum_j d_{mc}^2(\mathbf{x}_j - \mu)$ of a series $(\mathbf{x})_j$ of elements lying in $[0, 2\pi]^n$ is defined as:

$$\mu = (M(x_{(1)}^1), M(x_{(1)}^2), \dots, M(x_{(1)}^n))$$

3.3 Mixed radiometric-orientation Vector

This last stage of the construction is similar to the former except that 2 more coordinates are added to the angle information. They represent the pair of radiometric values (in the two input images) beared by the zone that the polygon qualifies. The (Euclidian) mean for those coordinates is obviously indepently computed. However, though the distance can be added to the “multi-orientation”, weights must be given for the radiometric and geometric influence on the distance computation. The distance for the final state space is then as follows:

$$d_f^2(\mathbf{x}, \mathbf{y}) = (1 - \rho) \cdot d_{mc}^2(\mathbf{x}_G, \mathbf{y}_G) + \rho \cdot (\mathbf{x}_R - \mathbf{y}_R)^2 \quad (15)$$

$$\text{where } \begin{cases} \rho \in [0, 1] \\ \mathbf{x}_G = (x^1, \dots, x^n) \text{ and } \mathbf{y}_G = (y^1, \dots, y^n) \\ \mathbf{x}_R = (x^{n+1}, x^{n+2}) \text{ and } \mathbf{y}_R = (y^{n+1}, y^{n+2}) \end{cases}$$

4. RESULTS

4.1 Test case: synthetic images

We first test our polygonal classification on a test case involving different quadrilaterals showing similar orientations. The test case would present 3 different classes. This classification was uniquely based on the geometric features. The algorithm could correctly classify the objects as shown on Figure 8.

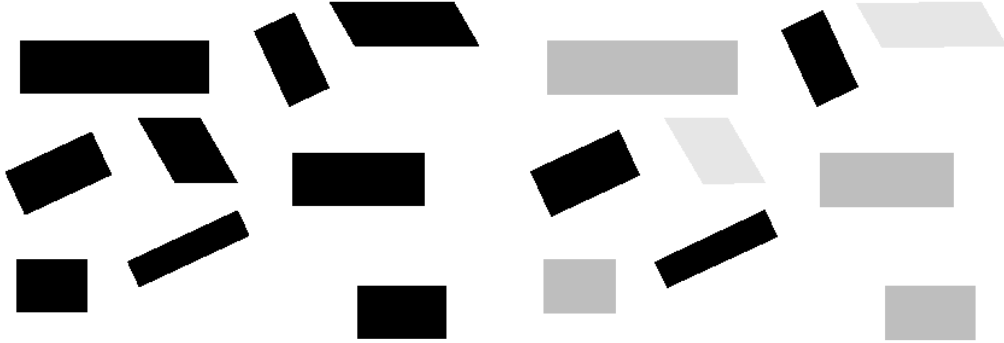


Figure 8: Quadrilaterals presenting different orientations (left), classified by the kmeans algorithm (right, each class has a different color)

4.2 Image Pairs

We tested the proposed classification algorithm on the polygons extracted with the method described in section 2.3. We only consider polygons with a big enough surface in order to avoid too small objects to present workable polygonal characteristics. Results are shown on Figure 9.

If the classification shows fair results for the shadow classes, it is clear that the results are not optimal for discriminating contrast changes and changes due to parallax. Another source of misclassification is linked to the first steps drawbacks (see section 2.1) . It is clear that if an object appears in one of the images in the pair, then it is most likely detected as a connex zone. However, parallax errors often induce intersections of the two pictures level sets, thus causing an over-segmentation. This issue is currently addressed with a new change detection model, based on the level sets of both images.

5. CONCLUSION

We have presented a method to detect and classify changed zones in high resolution remotely sensed urban image pairs, based on geometric and radiometric data. Up to now, according to the urban context, we have focused on the orientations for classification, However, depending on the studied scene, other geometrical characteristics could be of interest.

Future work will focus on the preprocessing step, in order to take into account a more global information such as the level sets and give better defined connex zones as an input to the polygonal approximation algorithm. Then, we will study the feasibility of a classification algorithm allowing polygons with non-fixed numbers of sides.

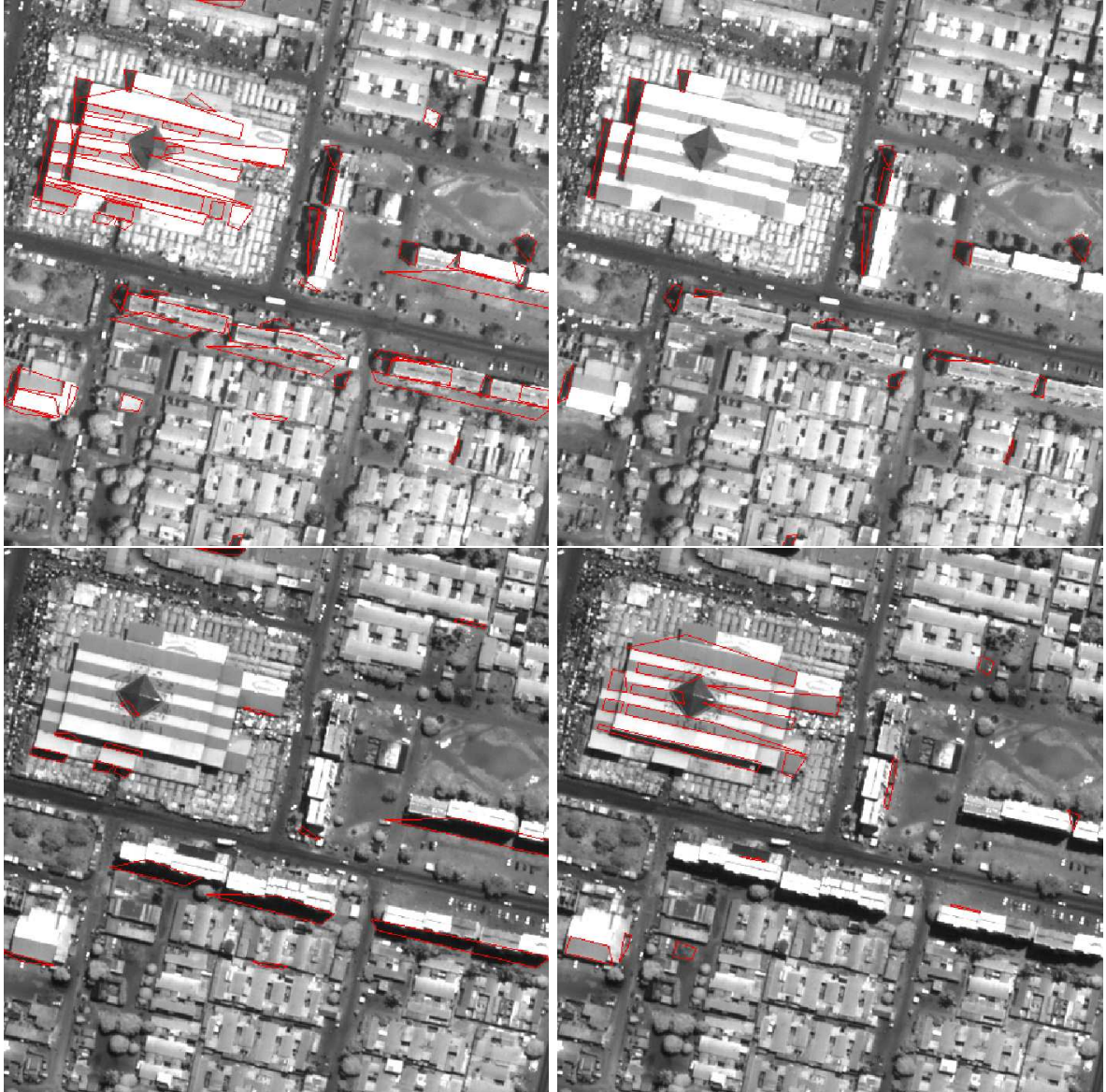


Figure 9: Classification of the polygons obtained. Top left: All detected polygons, Top Right: Shadows having appeared in the second Image, Bottom Left: Shadows having disappeared from the first image, Bottom Right: contrast changes between the two images

Acknowledgements

The work of the first author is partially funded by the French Defense Agency (DGA) through a PhD fellowship.

REFERENCES

1. J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):192–236, 1974.
2. L. Bottou and Y. Bengio. Convergence properties of the K -means algorithms. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 585–592. The MIT Press, 1995.
3. S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions and Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.
4. A. Kolesnikov. *Efficient algorithms for vectorization and polygonal approximation*. PhD thesis, University of Joensuu, Finland, October 2003.
5. A. Kolesnikov and P. Fränti. Polygonal approximation of closed discrete curves. *Pattern Recognition*, 40(4):1282–1293, 2007.
6. 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*, 36(3):221–236, 2000.
7. K.V. Mardia and P.E. Jupp. *Directional Statistics*. Wiley, 2000.
8. K.V. Mardia, J.T. Kent, and J.M. Bibby. *Multivariate Analysis*. Academic Press, 1979.
9. G. Palubinskas, X. Descombes, and F. Kruggel. An unsupervised clustering method using the entropy minimization. In *Proc. International Conference on Pattern Recognition (ICPR)*, Brisbane, Australia, August 1998.
10. H. Prautzsch, W. Boehm, and M. Paluszny. *Bezier and B-Spline Techniques*. Springer-Verlag New York, USA, 2002.
11. R. Wiemker. An iterative spectral-spatial Bayesian labeling approach for unsupervised robust change detection on remotely sensed multispectral imagery. In *Proc. Computer Analysis of Images and Patterns (CAIP)*, Kiel, Germany, September 1997.