INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

MRF model for Motion Detection on Airborne Images

Csaba Benedek — Tamás Szirányi — Zoltan Kato — Josiane Zerubia

N° ????

February 2007

Thème COM

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--????--FR+ENG

INRIA
SOPHIA ANTIPOLIS

MRF model for Motion Detection on Airborne Images

Csaba Benedek[*][†] , Tamás Szirányi[†][*] , Zoltan Kato[‡] , Josiane Zerubia[§]

Thème COM — Systèmes communicants
Projets ARIANA

Rapport de recherche n° ???? — February 2007 — 33 pages

Abstract: In this report, we give a probabilistic model for automatic change detection on airborne images taken with moving cameras. To ensure robustness, we adopt an unsupervised coarse matching instead of a precise image registration. The challenge of the proposed model is to eliminate the registration errors, noise and the parallax artifacts caused by the static objects having considerable height (buildings, trees, walls etc.) from the difference image. We describe the background membership of a given image point through two different features, and introduce a novel three-layer Markov Random Field (MRF) model to ensure connected homogenous regions in the segmented image.

Key-words: change detection, aerial images, camera motion, MRF

[*] Pázmány Péter Catholic University, Department of Information Technology, Budapest, Hungary.
[†] Distributed Events Analysis Research Group of the Computer and Automation Research Institute, Budapest, Hungary
[‡] University of Szeged, Institute of Informatics, Szeged, Hungary
[§] Ariana (joint research group INRIA/I3S), Sophia-Antipolis, France

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

MRF model for Motion Detection on Airborne Images rapport de recherche Inria

Résumé : In this report, we give a probabilistic model for automatic change detection on airborne images taken with moving cameras. To ensure robustness, we adopt an unsupervised coarse matching instead of a precise image registration. The challenge of the proposed model is to eliminate the registration errors, noise and the parallax artifacts caused by the static objects having considerable height (buildings, trees, walls etc.) from the difference image. We describe the background membership of a given image point through two different features, and introduce a novel three-layer Markov Random Field (MRF) model to ensure connected homogenous regions in the segmented image.

Mots-clés : change detection, aerial images, camera motion, MRF

1 Introduction

Change detection is an important early vision task in several computer vision applications. Shape, size, number and position parameters of the moving objects can be derived from the change-mask and used, for example for people or vehicle detection, tracking and activity analysis. This task is more difficult to obtain, if the images to be compared are taken at different camera positions.
The present paper addresses the problem of detecting the accurate silhouettes of moving objects, or at least, object-groups in image pairs taken by moving airborne vehicles in consecutive moments. The shots were focused on urban roads. We consider the presence of static objects in the scene, like short buildings, trees and walls. The time difference between the corresponding images is approximately 1 second, meanwhile the moving objects change their position significantly.
The procedure needs camera motion compensation. Feature correspondence is widely used for this task, where we look for corresponding pixels or other primitives such as edges, corners, contours, shape etc. in the images which we compare [1][5][20][28]. However, these methods are only efficient for image pairs with small differences, and they may fail at occlusion boundaries and within featureless regions, if the chosen primitives or features cannot be reliably detected.
In [27], a motion-based method is presented for automatic registration of images in multi-camera systems, to enable the synthesis of wide-baseline composite views. However, that method needs synchronized video flows recorded by static cameras which are not presented in our case.
According to a different approach, the images are matched via a simpler transformation (similarity [23], affine [18]), for which, we can find existing robust techniques. Although there are sophisticated ways to enhance the accuracy of these mappings [14], the purely similarity or affine matching does not fit to the scene geometry, and causes significant errors, especially at locations of static scene objects with considerable height (this effect is called parallax distortion, see Fig. 1).
In [25], an algorithmic approach is presented to a similar problem, however, the scene assumptions are significantly different. In that paper, very low altitude aerial videos are considered of sparsely cultural scenes, i.e. the "3Dness" of the scene is sparsely distributed, and it contains a few moving objects. The algorithm needs at least three frames from a video sequence. On the other hand, our method assumes that both the 3D static objects and the object motions are densely distributed, but the videos are captured from higher altitude, thus the parallax distortions cause usually errors of a few pixels. We do not expect that a video sequence is available, thus we may have only two images to compare. Hence, [21] can neither be used here, since it exploits a prediction for the camera motion based on previously processed frames.
For the above reasons, we introduce a two stage algorithm which consists of a coarse (but robust) image registration for camera motion compensation, and an error-eliminating step. From this point of view, it is similar to [6], where the authors assume that errors mainly appear near sharp edges. Therefore, at locations where the magnitude of the gradient is

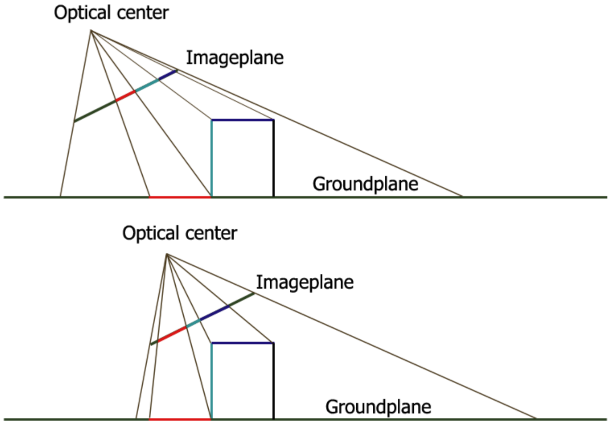

Figure 1: Illustration of the parallax effect, if a rectangular high object appears on the ground plane. We mark different sections with different colors on the ground and on the object, and plot their projection on the image plane with the same color. We can observe that the length ratio of the corresponding sections is significantly different.

large in both images, they consider that the differences of the corresponding pixel-values are caused with higher probability by registration errors than by object displacements. However, this method is less effective, if there are several small objects (containing several edges) in the scene, because the post processing may also remove some real objects, but it leaves errors in smoothly textured areas (e.g. group of trees, corresponding test results are in Section 6). In this paper, we use a Bayesian approach to tackle the above problem. We derive features describing the background membership of a given image point in two independent ways, and develop a three-layer Bayesian labeling model to integrate the effect of the different features. Our model structure is similar to [12]: it has two layers corresponding to the different observations, and one, which presents the final foreground-background segmentation result. However, there are two essential differences: while in [12], the segmentation classes in the combined layer were constructed as the direct product of the classes at the observation layers, we use the same classes in each layer: foreground and background. On the other hand, we define the inter-layer connections also differently. In [12], the observation layers were directly connected only with the segmentation layer, while we define connections between the observation layers also.

2 Image registration

In this section, we define the formal image model. Hereafter, we introduce briefly two approaches on coarse image registration. Finally, we compare the methods on our images and we choose the most appropriate one to be the preprocessing step of our Bayesian labeling model.

2.1 Image model

Denote by X_1 and X_2 the two consecutive frames of the image sequence above the same pixel lattice S. The gray value of a given pixel $s \in S$ is $x_1(s)$ in the first image and $x_2(s)$ in the second one. A pixel is defined by a two dimensional vector containing its x-y coordinates: $s = [s_x, s_y]^T$, $s_x = 1...M$, $s_y = 1...N$. We define a 4-neighborhood system on the lattice:

$$\forall s \in S: \quad \Phi_s = \{r \in S \ : \ ||s - r||_{L1} = 1\}, \tag{1}$$

where we determine the distance between two pixels by the Manhattan (L1) distance.
Formally, the segmentation procedure is a labeling process: a label is assigned to each pixel $s \in S$ from the label-set: $L = \{\text{fg, bg}\}$, corresponding to the two classes: foreground (fg) and background (bg).

2.2 FFT-Correlation based similarity transform (FCS)

Reddy and Chatterji [23] proposed an automatic and robust method for registering images, which are related via a similarity transform (translation, rotation and scaling). In this approach, the goal is to find the parameters of the similarity transform $\mathcal{T}$ for which the correlation between X_1 and $X_2^\dagger = \mathcal{T}(X_2)$ is maximal.
The method is based on the Fourier shift theorem. In the first step, we assume that X_1 and X_2 images differ only in displacement, namely there exists an offset vector o^*, for which $x_1(s) = x_2(s + o^*) : \forall s, s + o^* \in S$. Let us denote with X_2^o the image we get by shifting X_2 with offset o. In this case, $o^* = \text{argmax}_o C_r(o)$, where C_r is the correlation map: $C_r(o) = \text{Corr}\{X_1, X_2^o\}$. C_r can be determined efficiently in the Fourier domain. Let F_1 and F_2 be the Fourier transforms of the images X_1 and X_2. We define the Cross Power Spectrum (CPS) by:

$$\text{CPS}(\eta, \xi) = \frac{F_1(\eta, \xi) \cdot \overline{F}_2(\eta, \xi)}{|F_1(\eta, \xi) \cdot \overline{F}_2(\eta, \xi)|} = e^{j2\pi(o_x\eta + o_y\xi)},$$

where $\overline{F}_2$ means the complex conjugate of F_2. Finally, the inverse Fourier transform of the CPS is equal with the correlation map C_r.[23]
The Fourier shift theorem offers a way also to determine the angle of the rotation. Assume that X_2 is a translated and rotated replica of X_1, where the translation vector is o and the angle of rotation is ϕ. It can be shown that considering $|F_1|$ and $|F_2|$ as images, $|F_2|$ is the purely rotated replica of $|F_1|$ with angle ϕ. On the other hand, rotation in the Cartesian coordinate system is equivalent with a translational displacement in the polar representation [23], which can be calculated similarly to the determination of o^*.
The scaling factor of the optimal similarity transform may be retrieved in an analogous way [23].

To sum up, we can determine the optimal similarity transform $\mathcal{T}$ between the two images based on [23], and derive the (coarsely) registered second image, $X_2^\dagger$. In the following, $x_2^\dagger(s)$ will denote the gray value of pixel s in $X_2^\dagger$.

2.3 Pixel-correspondence based homography matching (PCH)

This approach consist of two consecutive steps. First, corresponding pixels are collected in the images, thereafter, the optimal coordinate transform is estimated between the elements of the extracted point pairs [29]. Therefore, only the first step is influenced directly by the observed image data, and the method may fail if the feature-correspondence produces poor result. On the other hand, we can obtain a more general transformation in this way than with the FCS.

In our implementation, we search for pixel correspondences for sharp corner pixels with the pyramidal Lucas-Kanade feature tracker [3][17]. The set of the resulting point pairs contains several outliers, which are filtered out by the RANSAC algorithm [8], while the optimal homography is estimated so that the back-projection error is minimized [9].

2.4 Experimental comparison of FCS and PCH

The FCS and PCH algorithms are tested on our test image pairs. Obviously, both gives only a coarse registration, which is inaccurate and is disturbed by parallax artifacts. In fact, FCS is less effective if the projective distortion between the images is significant. The weak point of PCH appears if the object motion is dense, thus a lot of point pairs may be in moving objects, and the automatic outlier filtering may fail, or at least, the homography estimation becomes inaccurate.

In our test database, the latter artifacts are more significant, since the corners of the several moving cars presents dominant features for the Lucas-Kanade tracker. Consequently, if C^* is the number of all the detected corner pixels and C^o is the number of corner pixels on moving objects; while P^*, P^o denote the number of all pixels and pixels corresponding to object displacement, respectively, $\frac{C^o}{C^*} \gg \frac{P^o}{P^*}$ may hold and the FCS method becomes much more robust.

Some results are in Fig. 2. We can observe that using FCS, the error-appearances are limited to the static objects boundaries, while regarding two out of the four frames, the PCH registration is highly erroneous. We note that the Bayesian post processing, which will be proposed in the later part of this report, can remove the FCSs errors, but it is unable to deal with the demonstrated PCH gaps.

For the above reasons, we will use the FCS method for preliminary registration in the following part of this report, however, in other test scenes it can be replaced with PCH in straightforward way.

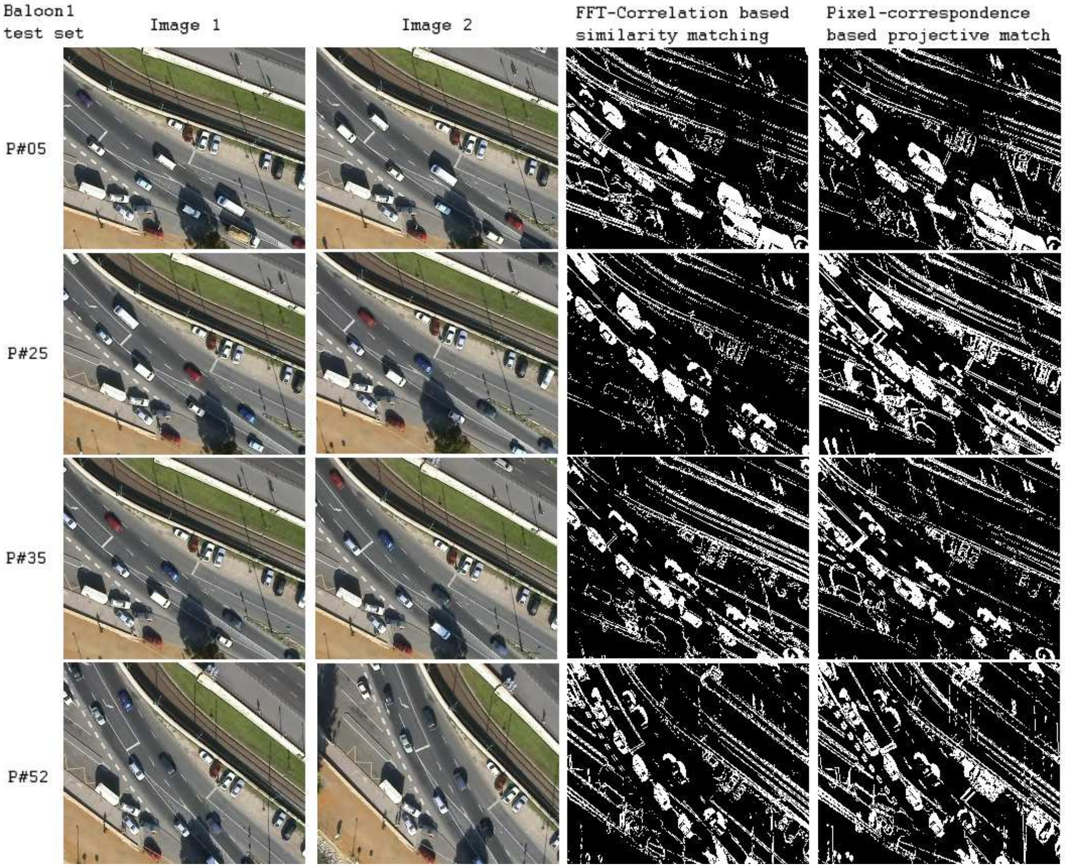

Figure 2: Qualitative illustration of the coarse registration results presented by the FFT-Correlation based similarity transform (FCS), and the pixel-correspondence based homography matching (PCH). In col 3 and 4, we find the thresholded difference of the registered images. Both results are quite noisy, but using FCS, the error-appearances are limited to the static objects boundaries, while regarding P#25 and P#52 the PCH registration is erroneous. Our Bayesian post processing is able to remove the FCSs errors, but it cannot deal with the demonstrated PCH gaps.

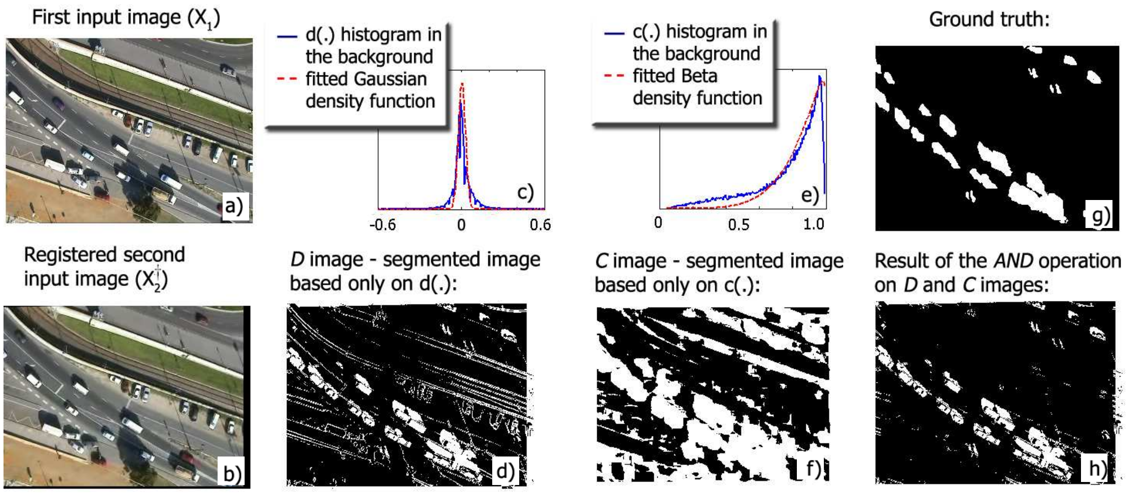

Figure 3: Feature selection. Notations are in the text of Section 3.

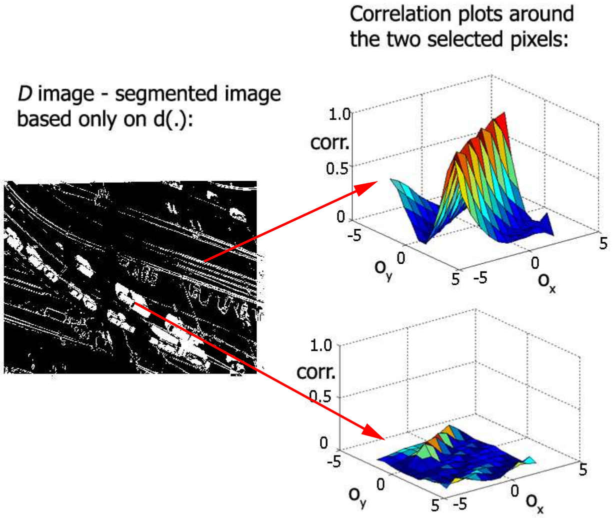

Figure 4: Plot of the correlation values over the search window around two given pixels. The above pixel corresponds to a parallax error in the background, while the below one is part of a real object displacement.

3 Feature selection

In this section, we introduce the feature selection using an airborne photo pair.[1] Taking a probabilistic approach, first we extract features, and then consider the class labels to be random processes generating the features according to different distributions.

3.1 Definition and illustration of the features

The first feature is the gray level difference of the corresponding pixels in the registered images:

$$d(s) = x_2^\dagger(s) - x_1(s).$$

We validate this feature through experiments (Fig. 3c): if we plot the histogram of $d(s)$ values corresponding to manually marked background points, then we can observe that a Gaussian approximation is reasonable:

$$P(d(s)|\mathrm{bg}) = N(d(s), \mu, \sigma) =$$

$$= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(d(s)-\mu)^2}{2\sigma^2}\right). \tag{2}$$

On the other hand, any $d(s)$ value may occur in the foreground, hence the foreground class is modeled by a uniform density:

$$P(d(s)|\mathrm{fg}) = \begin{cases} \frac{1}{b_d - a_d}, & \text{if } \ d(s) \in [a_d, b_d] \\ 0 & \text{otherwise.} \end{cases}$$

Next, we demonstrate the limitations of this feature. After supervised estimation of the distribution parameters, we derive D image in Fig. 3d as the maximum likelihood estimate: the label of s is

$$\mathrm{argmax}_{\psi \in \{\mathrm{fg},\mathrm{bg}\}} P(d(s)|\psi).$$

We can observe here that the registration and parallax errors cannot be filtered out using only $d(.)$, since their $d(s)$ values appear as outliers with respect to the previously defined Gaussian distribution.

From another point of view, assuming the presence of errors of a few pixels, we can usually find an $o_s = [o_x, o_y]$ offset vector, for which the rectangular neighborhood of s in X_1 and the same shaped neighborhood of $s + o_s$ in $X_2^\dagger$ is strongly correlated. Correlation of two image parts $A = \{a_1, a_2, \ldots a_n\}$ and $B = \{b_1, b_2, \ldots b_n\}$, where (a_i, b_i) are the values of the corresponding pixels, $\overline{a}$ and $\overline{b}$ are the mean values in the images, is computed by:

$$\mathrm{Corr}(A, B) = \frac{\sum_{i=1}^n (a_i - \overline{a})(b_i - \overline{b})}{\sqrt{\sum_{i=1}^n (a_i - \overline{a})^2 \sum_{i=1}^n (b_i - \overline{b})^2}}. \tag{3}$$

[1] We have also observed similar tendencies regarding the other test images, provided by the ALFA project.

In Fig 4, we plot the correlation values over the search window of the offset o_s around two given pixels (marked with the beginning of the arrows in Fig 4). The upper pixel corresponds to a parallax error in the background, while the lower one is part of a real object displacement. The correlation plot has high peak only in the upper case. We use $c(s)$, the maxima in the local correlation function around pixel s as second feature. By examining the histogram of $c(s)$ values in the background (Fig 3e), we find that it can be approximated with a beta density function:

$$P(c(s)|\mathrm{bg}) = B(c(s), \alpha, \beta),$$

where

$$B(c,\alpha,\beta) = \begin{cases} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} c^{\alpha-1}(1-c)^{\beta-1}, & \text{if} \quad c \in (0,1) \\ 0 & \text{otherwise} \end{cases}$$

$$\Gamma(\alpha) = \int_0^\infty t^{\alpha-1} e^{-t} dt.$$

As for the foreground class we will use a uniform probability $P(c(s)|\mathrm{fg})$ with a_c and b_c parameters. We see in Fig. 3f (C image) that the $c(.)$ descriptor causes also poor result in itself. Even so, if we consider D and C as a Boolean lattice, where 'true' corresponds to the foreground label, the logical AND operation on D and C improves the results significantly (Fig. 3h). We note that this classification is still quite noisy, although in the segmented image, we expect connected regions representing the motion silhouettes. Morphological postprocessing of the regions may extend the connectivity, but assuming the presence of various shaped objects or object groups, it is hardly possible to define appropriate morphological rules. Since the work of Geman and Geman [7], Markov Random Fields (MRFs) offer a powerful tool to ensure contextual classification. However, our case is particular: we have two weak features, which present two different (poor) segmentations, while the final foreground-background clustering depends directly on the labels of the weak segmentations. To decrease noise, we must prescribe, that both the weak and the final segmentations must be 'smooth'. Therefore, we introduce a robust segmentation model in Section 4.

3.2 Justification of the feature selection

Based on the experiments of the previous section, the gray level difference and the local correlation seem to be complementary features which describe together the background class efficiently. This observation has the following intuitive reason:

1. If the gray-level difference $d(s)$ votes for background at s, the correct segmentation class of s is usually background (except in cases of background-colored object points).

2. If the gray level difference $d(s)$ votes for foreground at s we may have two possibilities:

    - s is a real foreground object pixel,

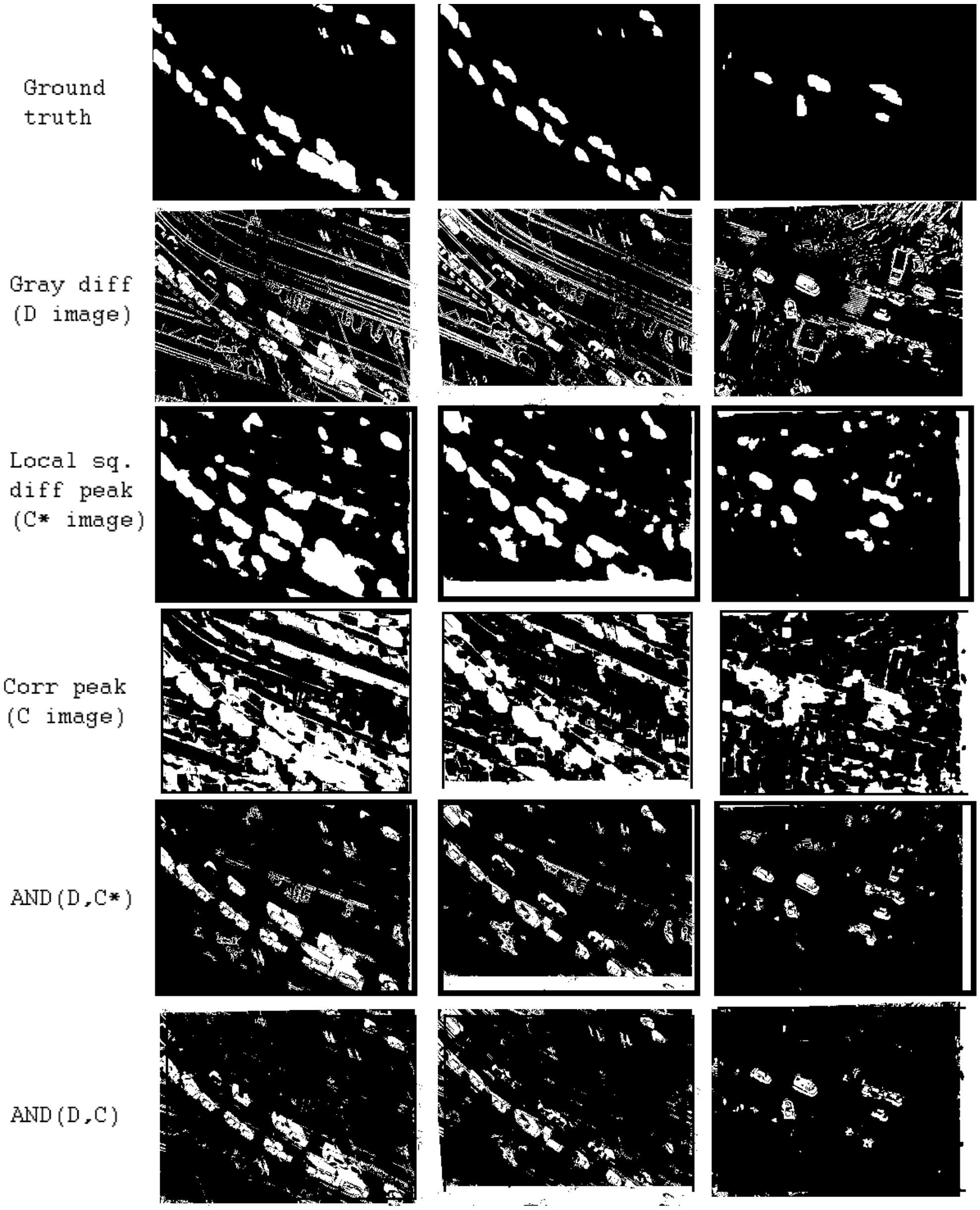

Figure 5: Qualitative comparison of the 'sum of local squared differences' (C^*) and the 'normalized cross correlation' (C) similarity measures with our label fusion model. In itself, the segmentation C^* is significantly better than C, but after fusion with D, the normalized cross correlation outperforms the squared difference.

- s is location of a registration/parallax error. This artifacts occurs mainly in textured 'background' areas and near to the region boundaries. On the other hand, if the background is homogenous in the neighborhood of s, the pixel values in a few pixel distance are similar, so $d(s)$ difference is near to the μ value expected in the background (see eq. 2).

3. If the correlation-peak-feature $c(s)$ votes for background at s, the correct segmentation class is usually background.

4. If the correlation-peak-feature $c(s)$ votes for foreground at s we may have two possibilities:

   - s is a real foreground object point
   - the normalized correlation is erroneously low around s. This artifacts occurs mainly in homogenous 'background' areas: if the variance of the pixel values in the rectangular correlation window is low, eq. 3 becomes quite sensitive to noise.

Therefore, we can summarize that the $d(.)$ and $c(.)$ features may cause quite a lot of false positive foreground points, however, the rate of false negative detection[2] is low in both cases: they appear only at location of background-colored object parts, and they can be partially eliminated by the smoothness constraints of MRF [7]. Moreover, examining $d(s)$ results usually false positive decision if the neighborhood of s is textured, but in that case the decision based on $c(s)$ is usually correct. Similarly, if $c(s)$ votes erroneously, the hint of $d(s)$ is usually correct. This argument agrees with the experimental results of Section 3.1 and supports our decision structure: the class of s is usually background, if and only if at least one of the $d(s)$ or $c(s)$ features votes for background.

We make two further comments regarding the feature selection. First, the proposed segmentation schema is a label fusion (like [10][16]) of two 'weak' segmentations, instead of observation fusion ([11][12]) of the features $d(.)$ and $c(.)$. Hence, the final segmentation labels depend on the observations indirectly via the 'weak' segmentation labels. We explain briefly why it is it a more natural choice regarding our problem than the observation fusion technique of [16]. Following that approach a two dimensional feature vector $f(s) = [d(s), c(s)]$ is ordered to each pixel s, and the joint distribution of the $f(s)$ values occurring in the background/foreground is estimated in the 2 dimensional feature space, e.g. with a two dimensional Gaussian/uniform density function. However, if s corresponds to a parallax error, and its $d(s)$ value lies far from the desired μ value, $P(f(s)|\text{bg})$ may be erroneously low, even if $c(s)$ fits to the background model perfectly. In other words, observation fusion is more efficient, if the features describe 'completely' but 'noisy' the class which they model, i.e. we find a domain in the feature space which contains most of the occurring feature values corresponding to the background, while the outlier values lie usually near to the background domains boundary (they are just out of the domain because of the noise.) Therefore, we say that $d(.)$ is incomplete descriptor regarding the background class, since it characterizes

[2] Number of pixels corresponding to real object displacements but classified as background.

statistically only one part of the background pixels. Note that the same phenomena appears regarding the $c(.)$ descriptor.
Secondly, the limitation of the $c(.)$ descriptor is caused by the denominator term in the normalized correlation expression (eq. 3). Here, we offer as alternative descriptor a non-normalized similarity factor, namely, the simple squared difference. For $A = \{a_1, a_2, \dots a_n\}$ and $B = \{b_1, b_2, \dots b_n\}$:

$$\mathrm{Sqdiff}(A, B) = \sum_{i=1}^{n} (a_i - b_i)^2, \tag{4}$$

and denote by $c^*(s)$ the minimal Sqdiff value around s, while C^* is the segmented image based on $c^*(.)$. We show some comparative experimental results for C and C^* in Fig. 5. We can observe that in itself, C^* has significantly better quality than C, but $c(.)$ is a better complementary feature of $d(.)$, and the $D-C$ joint segmentation is better than the clustering based on $D - C^*$.

4 Multi-layer segmentation model

In the proposed approach, we construct a Markov random field (MRF) model on a graph $\mathcal{G}$ whose structure is shown in Fig. 6. In the previous section, we segmented the images in two independent ways, and derived the final result by a label fusion using the two segmentations. Therefore, we arrange the sites of $\mathcal{G}$ into three layers S^d, S^c and S^*, each layer has the same size as the image lattice S. We assign to each pixel $s \in S$ a unique site in each layer: e.g. s^d is the site corresponding to pixel s on the layer S^d. We denote $s^c \in S^c$ and $s^* \in S^*$ similarly.
We introduce a labeling process, which assigns a label $\omega(.)$ to all sites of $\mathcal{G}$ from the label-set: $L = \{\text{fg}, \text{bg}\}$. The labeling of S^d/S^c corresponds to the segmentation based on the $d(.)/c(.)$ feature, respectively; while the labels at the S^* layer present the final change mask. A global labeling of $\mathcal{G}$ is

$$\underline{\omega} = \left\{\omega(s^i) | s \in S, i \in \{d, c, *\}\right\}.$$

In our model, the labeling of an arbitrary site depends directly on the labels of its neighbors (MRF condition). For this reason, we must define the neighborhoods (i.e. the edges) in $\mathcal{G}$ (see Fig. 6). To ensure the smoothness of the segmentations, we put edges within each layer between site pairs corresponding to neighboring pixels of the image lattice S.[3] On the other hand, the sites corresponding to the same pixel must interact to proceed the fusion of the two different segmentations' labels in the S^* layer. Hence, we introduce 'inter-layer' edges between sites s^i and s^j: $\forall s \in S$; $i, j \in \{d, c, *\}$, $i \neq j$. Therefore, the graph has doubleton 'intra-layer' cliques (their set is $\mathcal{C}_2$) which contain pairs of sites, and 'inter-layer' cliques ($\mathcal{C}_3$) consisting of site-triples. We also use singleton 'intra-layer' cliques ($\mathcal{C}_1$), which are one-element sets containing the individual sites: they will link the model and the local

[3] We use first order neighborhoods in S, where each pixel has 4 neighbors.

observations. Hence, the set of cliques is $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3$.
Denote the observation process by

$$\mathcal{F} = \{f(s)|s \in S\},$$

where $f(s) = [d(s), c(s)]$.
Our goal is to find the optimal labeling $\widehat{\underline{\omega}}$, which maximizes the a posterior probability $P(\underline{\omega}|\mathcal{F})$ that is a maximum a posteriori estimate [7]:

$$\widehat{\underline{\omega}} = \text{argmax}_{\underline{\omega} \in \Omega} P(\underline{\omega}|\mathcal{F}).$$

where Ω denotes the set of all the possible global labelings. Based on the Hammersley-Clifford Theorem [7] the a posterior probability of a given labeling follows Gibbs distribution:

$$P(\underline{\omega}|\mathcal{F}) = \frac{1}{Z} \exp\left(-\sum_{C \in \mathcal{C}} V_C(\underline{\omega}_C)\right),$$

where V_C is the *clique potential* of $C \in \mathcal{C}$, which is 'low' if $\underline{\omega}_C$ (the label- subconfiguration corresponding to C) is semantically correct, 'high', if not. Z is a normalizing constant, which does not depend on $\underline{\omega}$.
In the following part of this section, we define the clique potentials. We refer to a given clique as the set of its sites (in fact, each clique is a subgraph of $\mathcal{G}$), e.g. we denote the doubleton clique containing site s^d and r^d with $\{s^d, r^d\}$.

The observations affect the model through the singleton potentials. As we stated previously, the labels in the S^d and S^c layers are directly influenced by the $d(.)$ and $c(.)$ values, respectively, $\forall s \in S$:

$$V_{\{s^d\}}\left(\omega(s^d)\right) = -\log P(d(s)|\omega(s^d)),$$
$$V_{\{s^c\}}\left(\omega(s^c)\right) = -\log P(c(s)|\omega(s^c)),$$

where the probabilities that the given foreground or background classes generate the $d(s)$ or $c(s)$ observation, were already defined in Section 3.
On the other hand, the labels at S^* have no direct links with these measurements:

$$V_{\{s^*\}}\left(\omega(s^*)\right) = 0.$$

For presenting smooth segmentation in each layer, the potential of an intra-layer clique $C_2 = \{s^i, r^i\} \in \mathcal{C}_2$, $i \in \{d, c, *\}$ has the following form:

$$V_{C_2} = \theta\left(\omega(s^i), \omega(r^i)\right) = \begin{cases} -\delta^i & \text{if} \quad \omega(s^i) = \omega(r^i) \\ +\delta^i & \text{if} \quad \omega(s^i) \neq \omega(r^i) \end{cases} \tag{5}$$

for a constant $\delta^i > 0$.
As we concluded from the experiments in Section 3, a pixel is likely generated by the

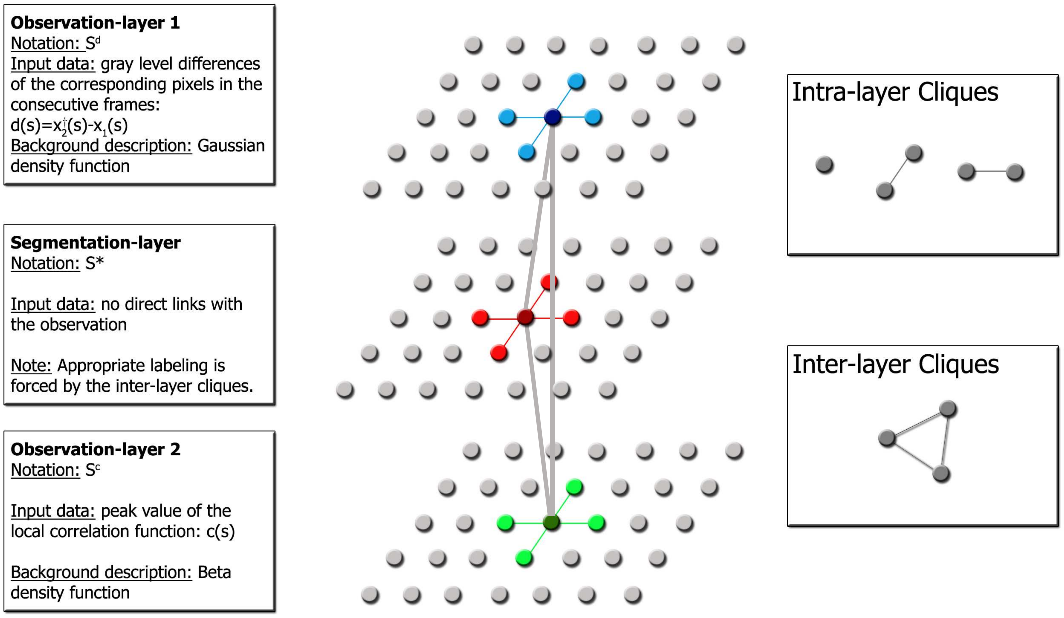
Observation-layer 1
Notation: S^d
Input data: gray level differences of the corresponding pixels in the consecutive frames:
$d(s)=x_2^{\dagger}(s)-x_1(s)$
Background description: Gaussian density function
Segmentation-layer
Notation: S^*
Input data: no direct links with the observation
Note: Appropriate labeling is forced by the inter-layer cliques.
Observation-layer 2
Notation: S^c
Input data: peak value of the local correlation function: c(s)
Background description: Beta density function
Intra-layer Cliques
Inter-layer Cliques

Figure 6: Summary of the proposed three layer MRF model

background process, if and only if in the S^d and S^c layers, at least one corresponding site has the label 'bg'. We introduce the following indicator function:

$$I_{\text{bg}} : S^d \cup S^c \cup S^* \to \{0,\ 1\},$$

where

$$I_{\text{bg}}(q) = \begin{cases} 1 & \text{if} \quad \omega(q) = \text{bg} \\ 0 & \text{if} \quad \omega(q) \neq \text{bg}. \end{cases}$$

With this notation the potential of an inter-layer clique $C_3 = \{s^d, s^c, s^*\}$ is with $\rho > 0$:

$$V_{C_3}(\underline{\omega}_{C_3}) = \zeta(\omega(s^d), \omega(s^c), \omega(s^*)) = \begin{cases} -\rho & \text{if} \quad I_{\text{bg}}(s^*) = \max\left(I_{\text{bg}}(s^d), I_{\text{bg}}(s^c)\right) \\ +\rho & \text{otherwise.} \end{cases} \tag{6}$$

Therefore, the optimal MAP labeling $\widehat{\underline{\omega}}$, which maximizes $P(\widehat{\underline{\omega}}|\mathcal{F})$ (hence minimizes $-\log P(\widehat{\underline{\omega}}|\mathcal{F})$) can be calculated as:

$$\widehat{\underline{\omega}} = \text{argmin}_{\underline{\omega} \in \Omega} - \sum_{s \in S} \log P(d(s)|\omega(s^d)) - \sum_{s \in S} \log P(c(s)|\omega(s^c))$$

$$+ \sum_{C_2 \in \mathcal{C}_2} V_{C_2}\left(\underline{\omega}_{C_2}\right) + \sum_{C_3 \in \mathcal{C}_3} V_{C_3}\left(\underline{\omega}_{C_3}\right). \tag{7}$$

The final segmentation is taken as the labeling of the S^* layer.

5 Parameter settings

In the following we define a possible grouping of the free parameters in the process: the first group is related to the correlation calculation and the second one to the potential functions.

5.1 Parameters related to the correlation window

The correlation window defined in Section 3 should not be significantly larger than the expected objects to ensure low correlation between an image part which contains an object and one from the same "empty" area. We used a 9×9 pixel window in our experiments for images of size 320×240.

The *maximal offset* of the search window determines maximal parallax error, which can be compensated by the method. We note that in homogenous background, object motions with less than the offset parameter can be falsely detected as parallax errors. Therefore, at the given resolution, we used ±3 pixels for the maximal offset, and detected the moving objects whose displacement was larger.

5.2 Parameters of the potential functions

The singleton potentials are values of conditional density functions as it was defined in Section 3.

The Gaussian mean parameter (μ) corresponds to the average gray value difference between the images caused by quick changes in the lighting conditions or in the camera white balance, the deviation (σ) depends on the noise. These parameters can be estimated by creating a histogram for D difference image, and estimating the parameters of the area close to the main peak of this histogram.

The Beta distribution parameters and the uniform values were determined from one image to another one by trial and error. We used $\alpha = 4.5$, $\beta = 1$ and $a_c = 0, b_c = 1$ for all image pairs (with the assumption that the gray values of the images are between 0 and 1), while the optimal value of a_d and b_d showed significant differences in the images. Using the "2σ-rule" proved to be a good initial approximation, namely $\frac{1}{b_d - a_d} = N(\mu + 2\sigma, \mu, \sigma)$. Here, following the Chebyshev equation:

$$P(|d(s) - \mu| > 2\sigma \mid \omega(s) = \text{bg}) < \frac{1}{4}.$$

The parameters of the intra-layer potential functions, δ^d, δ^c and δ^* influence the size of the connected blobs in the segmented images, while ρ, related to the inter-layer cliques, determines the strength of the relationship between the observation and segmentation layers. For all of these parameters, we used values between 0.7 and 1 for all images.

6 Results

In this section, we validate our method via image pairs from different test sets. We compare the results of the three layer model with three reference methods first qualitatively, then using different quantitative measures. Thereafter, we test the significance of the inter layer connections in the joint segmentation model. Finally, we comment on the complexity of the algorithm.

6.1 Test sets

The evaluations are conducted using manually generated ground truth masks regarding different aerial images. We use three test sets which contain in aggregate 83 (=52+22+9) image pairs. The time difference between the frames to compare is cca 1.5-2 seconds. The 'balloon1' and 'balloon2' test set contain image pairs from a video-sequence captured by a flying balloon, while in 'Budapest', we find different image pairs taken from a plane. For each test set, the model parameters are estimated over 2-5 training pairs and we examine the quality of the segmentation on the remaining test pairs.

6.2 Reference methods and qualitative comparison

We compared the results of the proposed three-layer model to three other solutions. The first reference method (Layer1) is constructed from our model by ignoring the segmentation and the second observation layers. This comparison emphasizes the importance of using the correlation-peak features, since only the gray level differences are used here. The second reference is the method of Farin and With [6]. The third comparison is related to the limits of [14]: the optimal affine transform between the frames (which was automatically estimated in [14]) is determined in our comparative experiments in a supervised way, through manually marked matching points, and a simple Potts-MRF [22] model decreases the registration errors.

Fig. 7 shows the image pairs, ground truth and the segmented images with the different methods. For numerical evaluation, we perform first a pixel based, then an object based comparison.

6.3 Pixel based evaluation

Denote the number of correctly identified foreground pixels of the evaluation images by TP (*true positive*). Similarly, we introduce FP for misclassified background points, and FN for misclassified foreground points.

The evaluation metrics consists of the *Recall* rate and the *Precision* of the detection.

$$\text{Recall} = \frac{TP}{TP + FN} \qquad \text{Precision} = \frac{TP}{TP + FP}$$

The results are presented in Table 1 for each image-sets independently. In the Table 1, we use the F-measure [24] which combines *Recall* and *Precision* in a single efficiency measure (it is the harmonic mean of P and R):

$$F = \frac{2 \cdot R \cdot P}{R + P}. \tag{8}$$

Regarding the 'balloon1'/'balloon2'/'Budapest' test sets, the gain of using our method considering the F-measure is 26/35/16% in contrast to the Layer1 segmentation and 12/19/13% compared to Farin's method. The results of the frames global affine matching, even with manually determined control points, is 5/10/11% worse than what we got with the proposed model.

6.4 Object based evaluation

Although our method does not segment the individual objects, the presented change mask can be the input of an object detector module. It is important to know, how many object-motions are correctly detected, and what is the false alarm rate.

Set		Recall				Precision			
Name	Cardinality	Layer1	Farin's	Sup. affine	**3layer MRF**	Layer1	Farin's	Sup. affine	**3layer MRF**
balloon1	52	0.83	0.76	0.85	**0.92**	0.48	0.74	0.79	**0.85**
balloon2	22	0.86	0.68	0.89	**0.88**	0.35	0.64	0.65	**0.83**
Budapest	9	0.87	0.80	0.85	**0.89**	0.56	0.65	0.65	**0.79**

Table 1: Numerical comparison of the proposed method (3-layer MRF) with the results that we get without the correlation layer (Layer1) and Farin's method [6] and the supervised affine matching. Rows correspond to the three different test image-sets with notation of their cardinality (e.g. number of image-pairs included in the sets).

Set		F-rate			
Name	Cardinality	Layer1	Farin's	Sup. affine	**3layer MRF**
balloon1	52	0.61	0.75	0.82	**0.87**
balloon2	22	0.50	0.66	0.75	**0.85**
Budapest	9	0.68	0.71	0.73	**0.84**

Table 2: Numerical comparison of the proposed and reference methods via the F-rate. Notations are the same as in Table 1.

If an object changes its location, two blobs appear in the binary motion image, corresponding to its first and second positions. Of course, these blobs can be overlapped, or one of them may missing, if an object just appears in the second frame, or if it leaves the area of the image between the two shots. In the following, we call one such blob as 'object displacement', which will be the unit in the object based comparison.

Given a binary segmented image, denote by M_o (missing objects) the number of object displacements, which are not included in the motion silhouettes, while F_o (false objects) is the number of the connected blobs in the silhouette images, which do not contain real object displacements, but their size is at least as large as one expected object. For the selected image pairs of Fig. 7, the numerical comparison to Farin's and the supervised affine method is given in Table 1. A limitation of our method can be observed in the 'Budapest' #2 image pair: the parallax distortion of a standing lamp is higher than the length of the correlation search window side, which results in two false objects in the motion mask. However, the number of missing and false objects is much lower than regarding the reference methods.

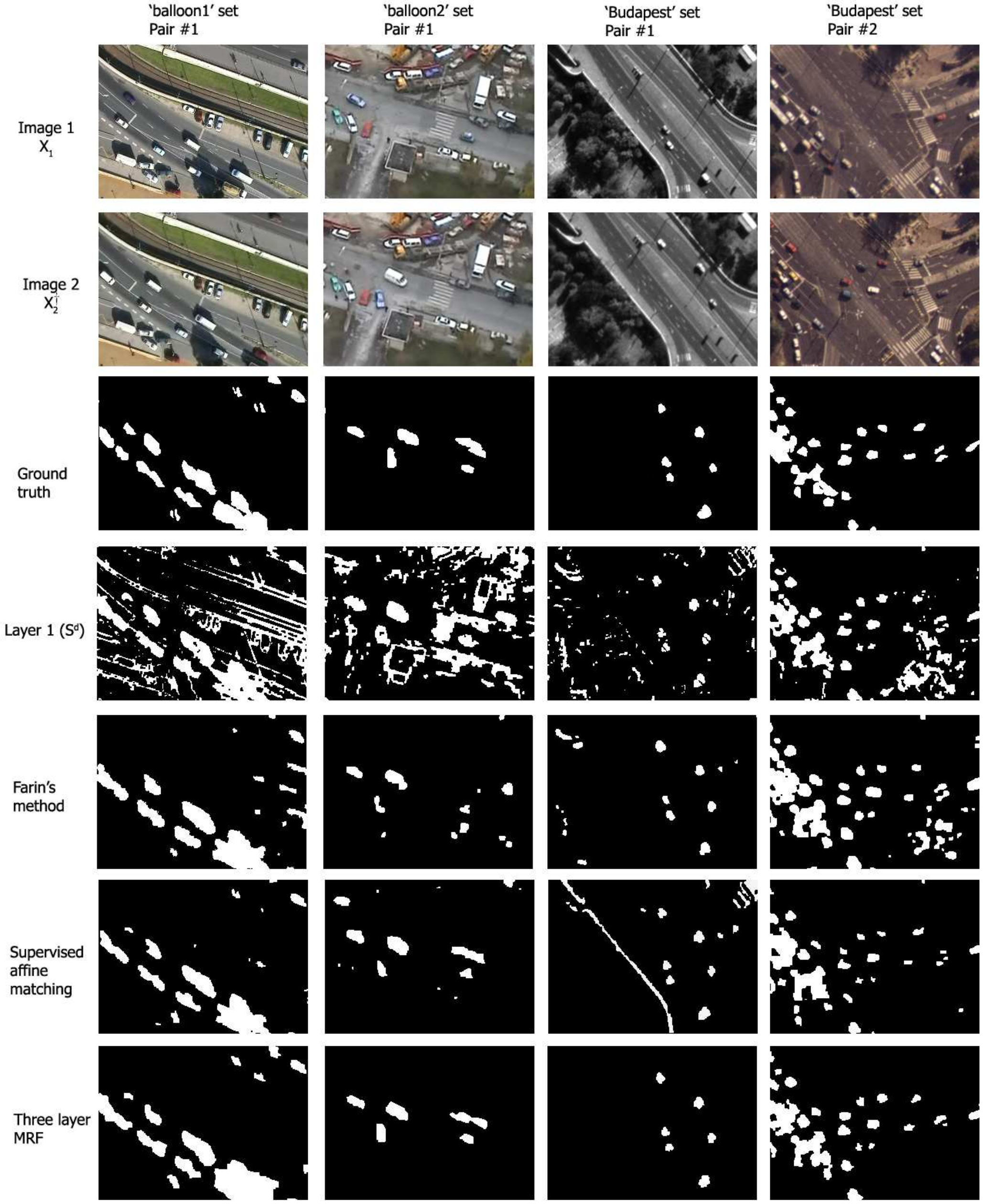

Figure 7: Test image pairs and segmentation results with different methods.

Test pair		A_0	M_o			F_o		
Set	No.		Far.	Sup. aff.	**3lay. MRF**	Far.	Sup. aff.	**3lay. MRF**
balloon1	#1	19	0	0	**0**	6	1	**1**
balloon2	#1	6	0	0	**0**	3	2	**0**
Budapest	#1	6	1	0	**0**	7	7	**0**
Budapest	#2	32	0	1	**1**	10	6	**3**
	All	63	3	1	**1**	26	16	**4**

Table 3: Object-based comparison of the proposed and the reference methods. A_o means the number of all object displacements in the images, while the number of missing and false objects is respectively M_o and F_o.

Procedure	FCS	PCH	Corr. map	MRF opt.
Time (sec)	0.15	0.04	2.4	2.9

Table 4: Running time of the main parts of the algorithm. The calculation of the correlation map and the MRF optimization are detailed in Appendices A and B, respectively.

6.5 Significance of the joint segmentation model

One of the novelties of the proposed model it that the segmentations based on the $d(.)$ and $c(.)$ features are not performed independently: they interact through the inter-layer cliques. This structure enables to get smooth components in the final change mask. We compare the schema with a sequential model: first, we perform two independent segmentations based on $d(.)$ and $c(.)$ (i.e. we segment the S^d and S^c layers with ignoring the inter layer cliques), thereafter we get the segmentation of S^* by a per pixel AND operation on the D and C segmented images. In Fig. 8, we can observe that the separate segmentation gives noisy results, since in this case, the intra-layer smoothing terms do not take into account in the S^* layer.

6.6 Running speed

With C++ implementation and a Pentium desktop computer (Intel(R) Core(TM)2 CPU, 2GHz), processing 320×240 images takes $5-6$ seconds. For the main parts of the algorithm, we measured the processing times of Table 4. The calculation of the correlation map (i.e. the determination of the $c(.)$ feature in Section 3) and the MRF optimization (finding a good suboptimal labeling according to eq. 7 from Section 4) are detailed in Appendices A and B, respectively.

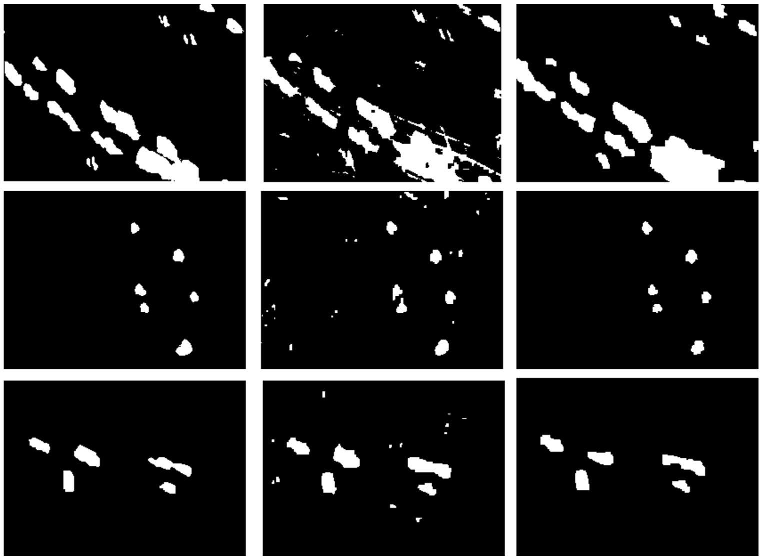

Figure 8: Illustration of the benefit of the inter layer connections in the joint segmentation. Col 1: ground truth, Col 2: results after separate MRF segmentation of the S^d and S^c layers, and deriving the final result with a per pixel AND relationship. Col 3. Result of the proposed joint segmentation model

7 Applications

The proposed model can be inserted into different high level applications being developed by ongoing research projects.
The *Shape Modelling E-Team of the EU Project MUSCLE* is interested in learning shapes and recognizing shapes as a central part of image database indexing strategies. Its scope includes shape analysis and learning, prior-based segmentation and shape-based retrieval. In shape modelling, however, accurate silhouette extraction is crucial preprocessing task.
The primary aim of the *Hungarian R&D Project ALFA* is to create a compact vision system that may be used as autonomous visual recognition and navigation system of unmanned aerial vehicles. In order to make long term navigational decisions the system has to evaluate the captured visual information without any external assistance. The civil use of the system includes large area security surveillance and traffic monitoring, since effective and economic solution to these problems is not possible using current technologies. The *Hungarian GVOP (3.1.1.-2004-05-0388/3.0)* attacks the problem of semantic interpretation, categorizing and indexing the video frames automatically. For both applications, object motion detection provides significant information.

8 Conclusion

This paper address the problem of exploiting accurate change masks from image pairs taken by a moving camera. A novel three-layer MRF model has been proposed, which integrates the information from two different observations. The efficiency of the method has been validated through real-world aerial images, and its behavior versus three reference methods has been quantitatively and qualitatively evaluated.

9 Acknowledgement

This work was partially supported by the EU project MUSCLE (FP6-567752) and the Hungarian R&D Project ALFA. The authors would like to thank to the MUSCLE Shape Modelling E-Team for support and to Xavier Descombes for his remarks and advices.

10 Appendices

A Calculation of the correlation map

In this appendix, we introduce the efficient determination of the correlation map used by the $c(.)$ feature (Section 3, eq. 3). The algorithm uses box filtering technique with the integral image trick similarly to eg. [26]. However, our method does not assume accurate epipolar matching, therefore, the region where we search for pixel correspondences is a rectangle instead of a line.

A.1 Integral image

Given and image $\Lambda \leftarrow S$, its integral image $\mathcal{I}_\Lambda \leftarrow S$ is defined by the following:

$$\mathcal{I}_\Lambda(x,y) = \sum_{i=1}^{x}\sum_{j=1}^{y}\Lambda(i,j).$$

With notation $\zeta(x,0) = 0$ and $\mathcal{I}_\Lambda(0,y) = 0$, $x = 1 \dots S_x$,$y = 1 \dots S_y$:

$$\zeta(x,y) = \zeta(x,y-1) + \Lambda(x,y),$$

$$\mathcal{I}_\Lambda(x,y) = \mathcal{I}_\Lambda(x-1,y) + \zeta(x,y),$$

the integral image can be computed in one pass over the original image.

With the integral-trick:

$$\sum_{i=a}^{c}\sum_{j=b}^{d}\Lambda(i,j) = \mathcal{I}_\Lambda(c,d) - \mathcal{I}_\Lambda(a-1,d) - \mathcal{I}_\Lambda(c,b-1) + \mathcal{I}_\Lambda(a-1,b-1).$$

A.2 Correlation

Let Υ_1 and Υ_2 two $l_w \times l_h$ sized 2 dimensional real arrays, with mean values $\overline{\Upsilon}_1$ and $\overline{\Upsilon}_2$, respectively. Their normalized cross correlation is defined by:

$$\text{Corr}(\Upsilon_1, \Upsilon_2) = \frac{\sum_{x=1,y=1}^{l_w,l_h}(\Upsilon_1(x,y) - \overline{\Upsilon}_1)(\Upsilon_2(x,y) - \overline{\Upsilon}_2)}{\sqrt{\sum_{x=1,y=1}^{l_w,l_h}(\Upsilon_1(x,y) - \overline{\Upsilon}_1)^2 \sum_{x=1,y=1}^{l_w,l_h}(\Upsilon_2(x,y) - \overline{\Upsilon}_2)^2}}$$

A.2.1 Local correlation map

Denote by $\mathcal{P}$ the set of images over S. Denote by $\Lambda_1, \Lambda_2 \in \mathcal{P}$ two images, w_x, w_y, l_w and l_h scalars. $t_{\text{win}} = (2l_w + 1)(2l_h + 1)$ is the size of the comparison window.

Denote by $\Upsilon_1^{x,y}$ a $(2l_w+1)\times(2l_h+1)$ sized subimage of Λ_1, whose center is at $[x,y]$. For simpler notation, we use also negative indices for identifying the elements of $\Upsilon_1^{x,y}$. Hence,

$$\Upsilon_1^{x,y}(i,j) = \Lambda_1(i+x, j+y),$$

$$-l_w \leq i \leq l_w,\ -l_h \leq j \leq l_h.$$

$\overline{\Upsilon_1^{x,y}}$ denotes the average of the elements in $\Upsilon_1^{x,y}$. $\Upsilon_2^{x,y}$ is defined similarly.

Definition 1 ***(Local correlation map)*** *The local correlation map asserts a* $(2w_x+1)\times(2w_y+1)$ *array,* $C^{x,y}$ *to each pixel* $s=[x,y]$*:*

$$C^{x,y}(m,n) = \mathrm{Corr}(\Upsilon_1^{x,y}, \Upsilon_2^{x+m,y+n}),$$

$$-w_x \leq m \leq w_x, -w_y \leq n \leq w_y.$$

For efficient computation, we introduce some notes:
For a given image Λ, denote by Λ^{sq} the "squared image":

$$\Lambda^{\mathrm{sq}}(x,y) = \left[\Lambda(x,y)\right]^2.$$

Denote by $\Lambda^{m,n}$ the "offset image":

$$\Lambda^{m,n}(x,y) = \Lambda(x+m, y+n).$$

Denote by $\mathcal{M}:\mathcal{P}\times\mathbb{N}\times\mathbb{N}\rightarrow\mathbb{R}$ the local average functional of a given image over S:

$$\mathcal{M}\{\Lambda,x,y\} = \frac{1}{t_{\mathrm{win}}}\sum_{i=-l_w}^{l_w}\sum_{j=-l_h}^{l_h}\Lambda(x+i,y+j).$$

If the $\mathcal{I}_\Lambda$ integral image is available, $\mathcal{M}\{\Lambda,x,y\}$ can be computed with 3 addition and one division operations:

$$\mathcal{M}\{\Lambda,x,y\} = \frac{1}{t_{\mathrm{win}}}[\mathcal{I}_\Lambda(x+l_w,y+l_h)+\mathcal{I}_\Lambda(x-l_w-1,y-l_h-1)-$$

$$-\mathcal{I}_\Lambda(x-l_w-1,y+l_h)-\mathcal{I}_\Lambda(x+l_w,y-l_h-1)].$$

We introduce the following notations:

$$M_1(x,y)=\mathcal{M}\{\Lambda_1,x,y\},\quad M_2(x,y)=\mathcal{M}\{\Lambda_2,x,y\},$$

$\Lambda_*^{m,n}$ image is introduced by

$$\Lambda_*^{m,n}(x,y) = \Lambda_1(x,y)\Lambda_2^{m,n}(x,y),\qquad \forall[x,y]\in S,$$

and

$$M_*^{m,n}(x,y) = \mathcal{M}\{\Lambda_*^{m,n},x,y\}.$$

$$\mathcal{B}(\Lambda, x, y) = \sum_{i=-l_w}^{l_w} \sum_{j=-l_h}^{l_h} \left(\Lambda(x+i, y+j) - \mathcal{M}\{\Lambda, x, y\}\right)^2 =$$

$$= \sum_{i=-l_w}^{l_w} \sum_{j=-l_h}^{l_h} \Lambda^{\mathrm{sq}}(x+i, y+j) - 2\mathcal{M}\{\Lambda, x, y\} \sum_{i=-l_w}^{l_w} \sum_{j=-l_h}^{l_h} \Lambda(x+i, y+j) + t_{\mathrm{win}}[\mathcal{M}\{\Lambda, x, y\}]^2 =$$

$$= t_{\mathrm{win}} \left(\mathcal{M}\{\Lambda^{\mathrm{sq}}, x, y\} - [\mathcal{M}\{\Lambda, x, y\}]^2\right)$$

On the other hand,

$$\mathcal{A}(x, y, m, n) =$$

$$= \sum_{i=-l_w}^{l_w} \sum_{j=-l_h}^{l_h} (\Lambda_1(x+i, y+j) - M_1(x,y))(\Lambda_2(x+m+i, y+m+j) - M_2(x+m, y+m)) =$$

$$= \sum_{i=-l_w}^{l_w} \sum_{j=-l_h}^{l_h} \Lambda_1(x+i, y+j)\Lambda_2(x+m+i, y+m+j) -$$

$$-M_1(x,y) \sum_{i=-l_w}^{l_w} \sum_{j=-l_h}^{l_h} \Lambda_2(x+m+i, y+m+j) -$$

$$-M_2(x+m, y+m) \sum_{i=-l_w}^{l_w} \sum_{j=-l_h}^{l_h} (\Lambda_1(x+i, y+j)) + t_{\mathrm{win}} M_1(x,y) M_2(x+m, y+m) =$$

$$= t_{\mathrm{win}} \left(M_*^{m,n}(x,y) - M_1(x,y) M_2(x+m, y+m)\right).$$

With these notations, the local correlation map is determined by:

$$C^{x,y}(m,n) = \frac{\mathcal{A}(x,y,m,n)}{\sqrt{\mathcal{B}(\Lambda_1, x, y) \cdot \mathcal{B}(\Lambda_2, x+m, y+n)}}.$$

Finally, the steps of the algorithm which calculates the correlation map, and the $c(.)$ feature (defined in Section 3) are listed in Table 5.

A.2.2 Complexity

Denote by $W = (2w_x + 1) \times (2w_y + 1)$ the size of the search window, $t_{\mathrm{win}} = (2l_h + 1) \times 2(l_w + 1)$ is the size of the correlation window, $\mathcal{S}$ is the size of the image. With the naive solution, the process needs $10\mathcal{S} \cdot W \cdot t_{\mathrm{win}} + 2\mathcal{S} \cdot W$ operations, while the improved version uses $10\mathcal{S} \cdot W + 37\mathcal{S}$ operations. Hence, the complexity of the improved method does not depend on the *correlation* window size t_{win}. For some *search* window sizes (W), we show the processing time in Table 6.

In the tests of Section 6, we have used $W = 7 \times 7$ pixel search windows. If larger W

```
1.  For $-w_x \leq m \leq w_x,\ -w_y \leq n \leq w_y$:
    • Calculate $\Lambda^{m,n}$
    • Calculate $\Lambda_*^{m,n}$
    • Calculate the integral image of $\Lambda_*^{m,n}$.
2.  Calculate the integral images of $\Lambda_1$, $\Lambda_2$, $\Lambda_1^{\mathrm{sq}}$ and $\Lambda_2^{\mathrm{sq}}$.
3.  For all $x$,$y$:
    • Calculate $M_1(x,y)$ and $M_2(x,y)$.
    • Calculate $\mathcal{B}(\Lambda_1,x,y)$ and $\mathcal{B}(\Lambda_2,x,y)$.
4.  For all $x$,$y$:
    • Calculate $C^{x,y}(m,n)$ for all $-w_x \leq m \leq w_x,\ -w_y \leq n \leq w_y$.
    • Store the maximal correlation value (over $m$, $n$): with $s = [x,y], c(s) = \max_{m,n} C^{x,y}(m,n)$
```

Table 5: Algorithm for efficient determination of the correlation feature $c(.)$. Notations are defined in Section 3 and A.

Window size (W)	3×3	5×5	7×7	9×9	11×11
Time (sec)	0.5	1.1	2.4	4.2	6.3

Table 6: Processing time of the correlation map calculator algorithm of Table 5 as function of the search window sizes (W), using 320×240 images, C++ implementation and a Pentium desktop computer (Intel(R) Core(TM)2 CPU, 2GHz)

is necessary, we can speed up the method with multi-resolution techniques [15]. If the fundamental matrix can be extracted (i.e. the PHC method works), the $(2w_x+1) \times (2w_y+1)$ pixel rectangular search window is restricted to a section in the corresponding epipolar line [8] (see also Fig. 9).

B MRF optimization

In MRF applications, the quality of the segmented images depends on:

- the appropriate model structure and the probabilistic model of the classes,
- the optimization technique which finds a good global labeling considering eq. 7 (Section 4). It is a key point, since the global optimum can be reached usually by computationally expensive methods [19] only.

In the tests (Section 6), we focus on the validation of our model instead of the comparison of various optimization techniques which has been already done in [4][13]. We use the Modified Metropolis (MMD) [13] algorithm, since we have found it similarly efficient but significantly

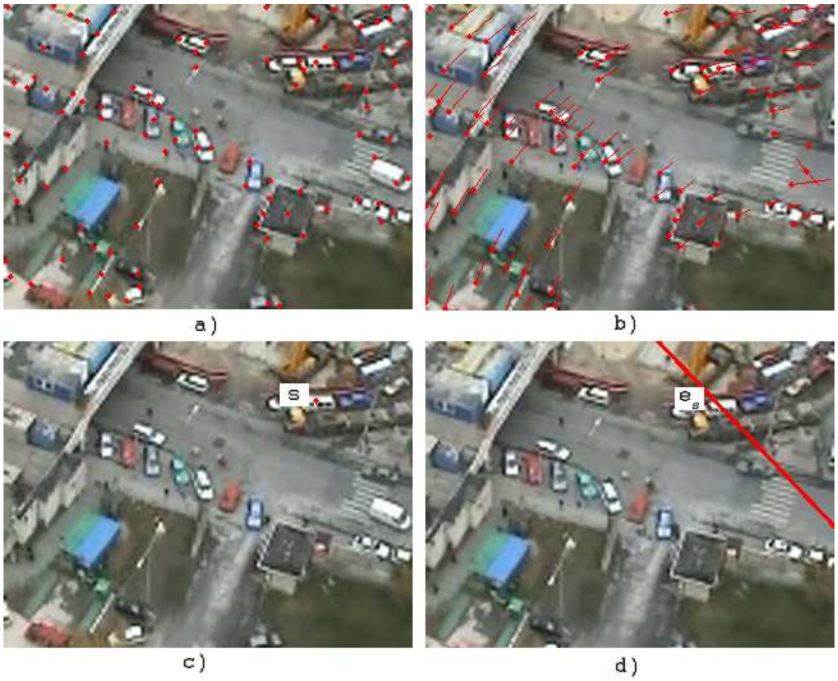

Figure 9: Illustration on how the PCH algorithm can restrict the correlation search window to a line. a) first input image (X_1) with the detected corner points b) result of the feature tracker [3] in X_2 for the previous corner pixels. The global motion is estimated based on the 2D displacement vectors corresponding the to corner points: the fundamental matrix, and the epipoles are calculated [8][9]. c) a selected pixel s in X_1 and d) the corresponding epipolar line e_s in X_2. For a given pixel s in X_1, the corresponding pixel in X_2 must be in line e_s. Note: as stated in Section 2.4, the PCH may fail for some inputs, however, as demonstrated here, it is efficient for test set 'balloon2', where the number of object motions is lower.

quicker than the original Metropolis [19]. We give the detailed pseudo code of the MMD adopted to the three layer segmentation model in Table 7. We note that a course but real-time MRF optimization method is the ICM algorithm [2]. If we use ICM with our model, its processing time is negligible compared to the other parts of the algorithm, in exchange for some degradation in the segmentation results.

References

[1] S. T. Barnard, W. B. Thompson, "Disparity analysis of images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 2, pp. 333-340, 1980.

[2] J. Besag, "On the statistical analysis of dirty images," *Journal of Royal Statistics Society*, 48

[3] J-Y. Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker: Description of the Algorithm", *Technical Report, Intel Corporation,* 1999.

[4] Y. Boykov, V. Kolmogorov, "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 26, no. 9, pp. 1124-1137, Sept. 2004

[5] J. K. Cheng, T. S. Huang, "Image registration by matching relational structures," *Pattern Recognition*, vol.17, pp.149-159, 1984.

[6] D. Farin and P. With, "Misregistration Errors in Change Detection Algorithms and How to Avoid Them," *Proc. International Conference on Image Processing (ICIP)*, vol. 2 p. 438-441, Genoa, Italy, Sep 2005.

[7] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images", *IEEE Trans. Pattern Analysis and Machine Intelligence,* pp. 721-741, 1984.

[8] R. Hartley and A. Zissermann, "Multiple View Geometry in Computer Vision," *Cambridge University Press*, pp. 11, Cambridge, 2000.

[9] Intel Corporation, "OpenCV documentation,"
`http://www.intel.com/technology/computing/opencv/index.htm`

[10] P-M. Jodoin and M. Mignotte, "Motion Segmentation Using a K-nearest-Neighbor-Based Fusion Procedure, of Spatial and Temporal Label Cues," in *Proc. of ICIAR*, Toronto, Canada, 2005.

[11] S. Khan and M. Shah, "Object based segmentation of video color, motion and spatial information", *Proc. of CVPR*, Hawaii, USA, pp. 746–751, December 2001.

```
1. Pick up randomly an initial configuration $\underline{\omega}^{[0]}$, with $k = 0$ and $T = T_0$
2. Using a uniform distribution, pick up layer $i \in \{d, c, *\}$, a pixel $s \in S$ and a new label for site $s^i$: $\vartheta \in \{\mathrm{fg}, \mathrm{bg}\}$.
3. Let $\underline{\widetilde{\omega}}$ be the global state which differs from $\underline{\omega}^{[k]}$ only in the label of $s^i$, namely, for each site $q$ of the three layer model,
       $\widetilde{\omega}(q) = \begin{cases} \vartheta & \text{if} \quad q = s^i, \\ \omega^{[k]}(q) & \text{if} \quad q \neq s^i. \end{cases}$
4. Compute $\Delta U_1$ by the following:
       $\Delta U_1 = \begin{cases} \log P\left(d(s)|\omega^{[k]}(s^d)\right) - \log P\left(d(s)|\vartheta\right) & \text{if} \quad i = d, \\ \log P\left(c(s)|\omega^{[k]}(s^c)\right) - \log P\left(c(s)|\vartheta\right) & \text{if} \quad i = c, \\ 0 & \text{if} \quad i = *. \end{cases}$
5. Calculate $\Delta U_2$ as
       $\Delta U_2 = \sum_{r \in \Phi_s} \theta\left(\widetilde{\omega}(s^i), \widetilde{\omega}(r^i)\right) - \theta\left(\omega^{[k]}(s^i), \omega^{[k]}(r^i)\right) =$
       $= \sum_{r \in \Phi_s} \theta\left(\vartheta, \omega^{[k]}(r^i)\right) - \theta\left(\omega^{[k]}(s^i), \omega^{[k]}(r^i)\right).$
6. Calculate $\Delta U_3$ as
       $\Delta U_3 = \zeta\left(\widetilde{\omega}(s^d), \widetilde{\omega}(s^c), \widetilde{\omega}(s^*)\right) - \zeta\left(\omega^{[k]}(s^d), \omega^{[k]}(s^c), \omega^{[k]}(s^*)\right).$
7. Let be
       $\Delta U = \Delta U_1 + \Delta U_2 + \Delta U_3.$
8. Update the configuration:
       $\underline{\omega}^{[k+1]} = \begin{cases} \underline{\widetilde{\omega}} & \text{if} \quad \Delta U \leq 0, \\ \underline{\widetilde{\omega}} & \text{if} \quad \Delta U > 0 \text{ and } \log \tau \leq -\frac{\Delta U}{T}, \\ \underline{\omega}^{[k]} & \text{otherwise.} \end{cases}$
   where $\tau$ is a constant threshold ($\tau \in (0, 1)$).
9. Set $T = T_{k+1}$, $k := k + 1$ and goto step 2, until convergence.
```

Table 7: Pseudo code of the Modified Metropolis algorithm used for the current task. Corresponding notations are in Section 2, 3, 4 and B. In the tests, we used $\tau = 0.3$, $T_0 = 4$, and an exponential heating strategy: $T_{k+1} = 0.96 \cdot T_k$

[12] Z. Kato, T. C. Pong, and G. Q. Song, "Multicue MRF Image Segmentation: Combining Texture and Color", *Proc. of International Conference on Pattern Recognition*, vol. 1, Quebec, Canada, pp. 660-663, August 2002.

[13] Z. Kato, J. Zerubia, and M. Berthod, "Satellite Image Classification Using a Modified Metropolis Dynamics", *Proc. International Conference on Acoustics, Speech and Signal Processing,* vol. 3, San-Francisco, USA, pp. 573-576, March 1992.

[14] S. Kumar, M. Biswas and T. Nguyen, "Global motion estimation in spatial and frequency domain", *IEEE International Conference on Acoustics,Speech,and Signal Processing,* Montreal, Canada, May 2004.

[15] S. Kumar and U.B. Desai, "New algorithms for 3D surface descriptopn from binocular stereo using integration", *Journal of the Franklin Institute,* 331B(5):531–554, 1994.

[16] A. Kushki, P. Androutsos, K.N. Plataniotis, A.N. Venetsanopoulos, , "Retrieval of images from artistic repositories using a decision fusion framework," *IEEE Transactions on Image Processing*, vol. 13, no. 3, pp. 277–292, 2004.

[17] B. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," Proc. of 7th International Joint Conference on Artificial Intelligence (IJCAI), pp. 674-679, 1981.

[18] L. Lucchese, "Estimating Affine Transformations in the Frequency Domain," *Proc. Int. Conf. on Image Processing*, Thessaloniki, Greece, Sept. 2001.

[19] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," *J. of Chem. Physicsn*, vol. 21, pp. 1087-1092, 1953.

[20] I. Miyagawa and K. Arakawa, "Motion and shape recovery based on iterative stabilization for modest deviation from planar motion," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* Vol. 28, Issue 7, pp: 1176 - 1181, 2006.

[21] J. M. Odobez, P. Bouthemy, "Detection of multiple moving objects using multiscale MRF with camera motion compensation," *Proc. Int. Conf. on Image Processing*, vol 2, pp 257-261, 1994.

[22] R. Potts, "Some generalized order-disorder transformation," *Proceedings of the Cambridge Philosophical Society,* 48(106), 1952.

[23] B. Reddy and B. Chatterji, "An FFT-based technique for translation, rotation and scale-invariant image registration", *IEEE Trans. on Image Processing*, vol. 5, no. 8, pp. 1266–1271, 1996.

[24] C. J. Van Rijsbergen, "Information Retrieval," 2nd edition, London, Butterworths.

[25] H.S. Sawhney, Y. Guo, R. Kumar, "Independent Motion Detection in 3D Scenes", *IEEE Trans. Pattern Analysis and Machine Intelligence*,Vol. 22, No. 10, pp. 1191-1199, 2000.

[26] C. Sun, "Fast Stereo Matching Using Rectangular Subregioning and 3D Maximum-Surface Techniques", *International Journal of Computer Vision*, vol. 47. no. 1, 99-117, 2002.

[27] Z. Szlávik, T. Szirányi, L. Havasi, "Stochastic view registration of overlapping cameras based on arbitrary motion", *IEEE Trans. Image Processing*, to appear , 2007.

[28] J. Weng, N. Ahuja, T. S. Huang, "Matching two perspective views," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, pp. 806-825, 1992.

[29] Z. Zhang, R. Deriche, O. Faugeras, Q-T. Luong, "A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry," Artifical Intellignece, vol. 78. pp. 87–119, 1995.

Contents

1 Introduction 3

2 Image registration 4
2.1 Image model 5
2.2 FFT-Correlation based similarity transform (FCS) 5
2.3 Pixel-correspondence based homography matching (PCH) 6
2.4 Experimental comparison of FCS and PCH 6

3 Feature selection 9
3.1 Definition and illustration of the features 9
3.2 Justification of the feature selection 10

4 Multi-layer segmentation model 13

5 Parameter settings 16
5.1 Parameters related to the correlation window 16
5.2 Parameters of the potential functions 17

6 Results 17
6.1 Test sets 17
6.2 Reference methods and qualitative comparison 18
6.3 Pixel based evaluation 18
6.4 Object based evaluation 18
6.5 Significance of the joint segmentation model 21
6.6 Running speed 21

7 Applications 23

8 Conclusion 23

9 Acknowledgement 23

10 Appendices 24

A Calculation of the correlation map 24
A.1 Integral image 24
A.2 Correlation 24
A.2.1 Local correlation map 24
A.2.2 Complexity 26

B MRF optimization 27

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399