Geometric and photometric image stabilization for detection of significant events in video from a low flying UAV (Unmanned Aerial Vehicles)

Jiří Matas[1], *Pavel Krsek*[1], *Martin Urban*[1], *Jiří Nohýl*[2]

[1] Czech Technical University, Karlovo nám. 13,121 35 Prague 2, Czech republic
[2] Airforce Research Institute, Mladoboleslavská, 197 06 Prague 9, Czech republic

April 22, 2003

Abstract

Analysis and interpretation of the information present in a video stream from aerial surveys is demanding and time-consuming even for experts. On-line video sequences acquired by cameras on board of a small surveillance plane are very unstable. The brightness and contrast of the images are rapidly changing due to fast changes in illumination and the content of the scene. The movement of the small plane is non-uniform and it depends on the wind and other weather conditions.

As a first step facilitating visual interpretation, a dynamic adaptation of brightness and contrast have been designed implemented. The method supoorts on-line visualisation, i.e. it runs at frame rate. The improved quality of the video-stream may improve on-line decision-making of the operator monitoring the video.

Secondly, stabilisation of camera movement is achieved. We implemented a method based on a fast hierarchical search for point correspondences between consecutive images from video sequence. Next, homographic transformation (planar perspective) of the images is estimated. After stabilisation, moving object are identified. Finally, objects of interest, whose models are automatically built from example images, are recognised and localised. Detected objects could be highlighted (to direct operator's attention in the semi-automatic mode) or subsequences with the objects can be selected from video record (automatic mode) for further inspection.

1 Introduction

We present three image processing methods for enhancement, stabilisation and interpretation of a video stream originating from a low flying surveillance aircraft. The processing is intended to support both off-line ex-post analysis and on-line visualisation and monitoring. Contrast enhancement, brightness stabilisation, compensation of aircraft movements and detection of small moving objects are therefore designed to operate at frame rate. The appearance-based object detection and localisation method is slower and requires that the object of interest is in the field of view for at least a second.

This article is divided into three relatively independent sections:

1. Image contrast and brightness enhancement.
2. Motion compensation and moving object detection.
3. Object of interest detection.

In each section, a problem formulation is given and the adopted method is introduced. A brief description of the implemented algorithm follows. All proposed methods are validated on real data sequences origniating from an aerial surveillance flight.

Paper presented at the RTO SCI Symposium on "Critical Design Issues for the Human-Machine Interface", held in Prague, Czech Republic, 19-21 May 2003, and published in RTO-MP-112.

2 Image contrast and brightness enhancement

Changes of brightness and contrast are typical for image sequences obtained by cameras on board of a small surveying plane. Operator should recognize and track objects of interest in large range of brightness conditions. This task is demanding even for experts. Our goal is to automatically improve the contrast and brightness of image sequences. Changes of illumination should be reduced in result sequences and contrast should be enhanced at low contrast regions to utilize display devices.

2.1 Main ideas

There is many algorithm for improvement of contrast and brightness which are defined for gray scale images. All these algorithms cold be described by brightness transformation function $f(y_W)$. Brightness of each pixel $y_W = Y_W(i,j)$ of input image Y_W is transformed by the function onto brightness of pixel in output image $y'_W = f(y_W)$. The algorithms are called Gray-scale transformations.

Definition of transformation function $f(y_G)$ is important for results of the algorithm. We can identify three basic groups of transformations:

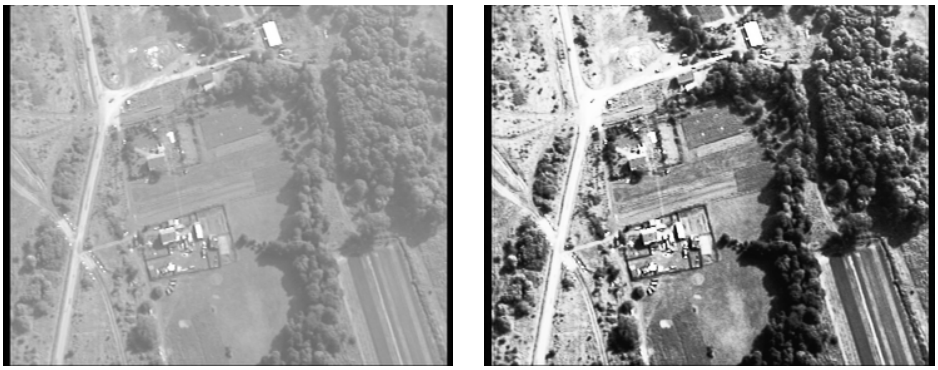
Position dependent corrections has the transformation function dependent on position of transformed pixel. The transformations are used to eliminate systematic deformations of images (such as aperture of lenses for example).

Global corrections has the transformation function defined globally for whole image (all pixels) in accordance to global image properties.

Local corrections have transformation function which is based on neighbourhood parameters of each pixel. The fixed or adaptive neighbourhood could be used. The transformations allow grate modification of the input image but it could produced undesired artifacts in an output sequence.

Our goal is to improve input images by stabilizing different lighting conditions. Especially we would like to eliminate shadows of clouds and diffused light (mist). We should preserve and increase local brightness changes. Due to this goal we focused on local brightness corrections.

Histogram equalization is representative of gray-scale transformation algorithm with nonlinear transform function. The aim is to create an output image with equally distributed brightness levels over whole brightness scale. Result of histogram equalization is presented on Fig. 1. Transformation function could be constructed from cumulative histogram of an input image [13]. We must only normalized the cumulative histogram by ratio of size of input image and number of gray levels in an output image. The histogram equalization algorithm is used in both global and local brightness corrections.



(a) (b)

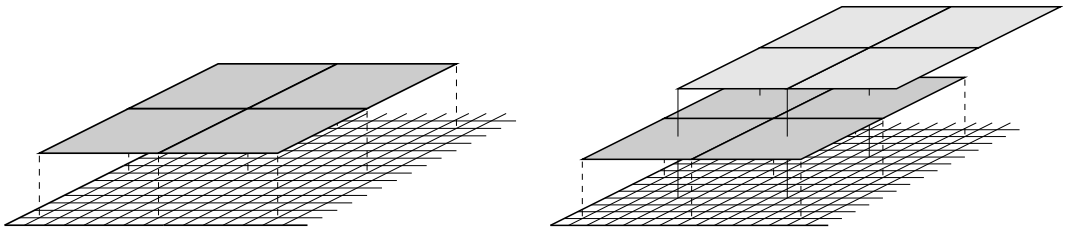
Figure 1: Result of histogram equalization. (a) Original image, (b) Image after histogram equalization.

2.2 Our local histogram equalization algorithm

For solving our task the local histogram equalization was chosen. In the original algorithm the histogram and transformation function is computed for a defined neighbourhood of each point. The neighbourhood is rectangle centered in tested point. The points are modified due to the locally defined transformation function. The algorithm is time consuming because histogram should be computed for each point. This approach is not suitable for on-line modification of video signal. Therefore we star re-implementing the partially local histogram equalization.

In our case brightness transformation function is not establish for each pixel but for the whole block of pixel in the given neighbourhood (Fig. 2a). To obtain smooth transformation function and smooth result image we interpolate the output brightness by bilinear approximation. The brightness level of each point is transformed by transfer function from 4 nearest neighbourhood blocks.

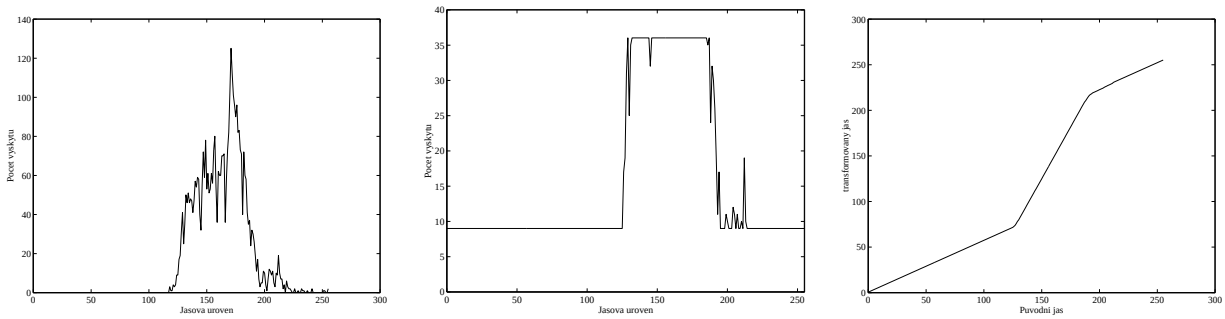
We found that concept of non-overlapping blocks produces observable artifacts on edges in input image. The artifacts are generated by bilinear approximation of nonlinear functions.



(a) (b)

Figure 2: Configurations of histogram equalization blocks. (a) Non-overlapped blocks, (b) Our block overlapping.

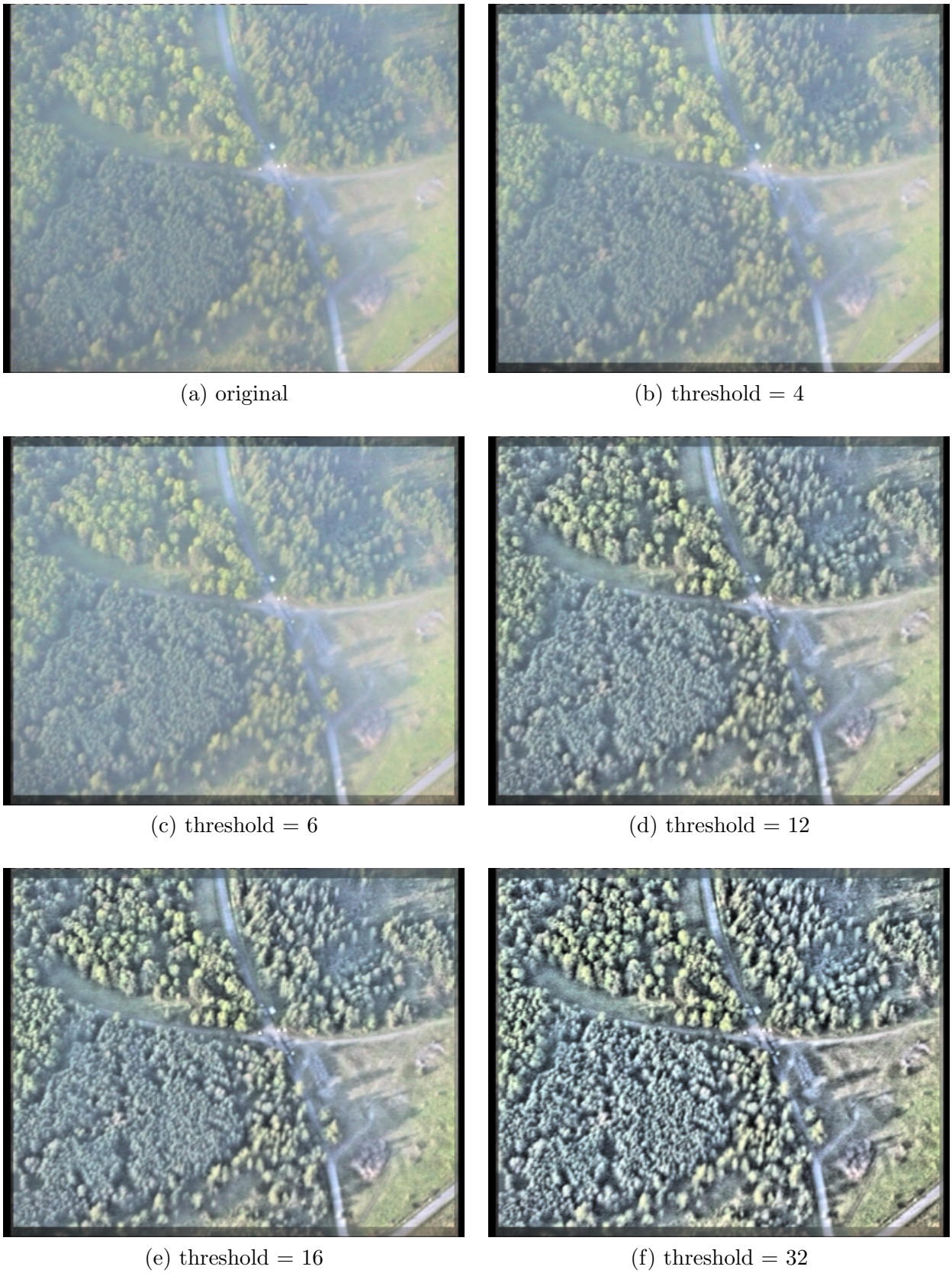
We improve the algorithm by using overlapping blocks as shown on Fig. 2b. The transformation function is computed in each overlapping block. Output brightness of each pixel is given by bilinear approximation from 4 nearest block transfer functions. The 4 neighbourhood transfer functions are functions of 4 blocks which covers a given point. This modification of the algorithm decreases the approximation artifacts.



(a) (b) (c)

Figure 3: Enhanced histogram equalization. (a) Histogram of original image, (b) Modified histogram, (c) Transform function.

Standard histogram equalization algorithm can rapidly change human interpretation of input images. Therefore the limited transformation function of gradient was implemented. The gradient limitation is realized by reducing the number of histogram levels. Transformation function algorithm is documented by Fig. 3. Fig. 3a shows histogram of some local block. The histogram is limited by level of clipping threshold. Then the minimal level of histogram is increased to preserve space of



(a) original (b) threshold = 4

(c) threshold = 6 (d) threshold = 12

(e) threshold = 16 (f) threshold = 32

Figure 4: Results under different clipping threshold. Equalization neighbourhood is 8×8 pixels.

histogram (Fig. 3b). Transformation function is created as normalized cumulative sum of histogram (Fig. 3c).

The threshold has direct influence on results and it's understandability by human operator. The threshold should be dynamically (on-line) set by operator in accordance to his experience, competence and to a type of monitoring scene.

The local equalization algorithm for gray-level images was described in previous paragraphs. However the modification for color RGB images was required also. In such a case the color image $y = (y_R, y_G, y_B)$ is transformed into gray level by equation

$$y_W = 0,299y_R + 0,587y_G + 0,114y_B \ , \tag{1}$$

at first, where y_W is gray level of pixel and y_R, y_G, y_B are color components of the pixel (red,green, blue). The local equalization algorithm is applied on the gray levels. The algorithm could be described as application an function $y'_W = f(y_W)$, where y'_W is transformed gray level of pixel. The function is implemented by amplification ratio. The technique is described by relation

$$y' = [\frac{y'_W}{f(y_W)}y_R, \frac{y'_W}{f(y_W)}y_G, \frac{y'_W}{f(y_W)}y_B] \ , \tag{2}$$

where $fracy'_W f(y_W)$ is amplifying ratio which is applied on pixel color levels and y' is transformed RGB pixel.

2.3 Results of image enhancement

We present results of the mentioned algorithm on real images from aerial survey Fig. 4 (top-left). Image Fig. 4 shows influence of clipping threshold on result. It is seen that the result images has higher contrast, diffused light is suppressed and comprehensibility is preserved.

We implemented our algorithm in C language with partial assembler code. Our implementation uses MMX extension of Intel assembler code to reach faster implementation. The implementation is close to on-line (real-time) usage. The algorithm is able process 20 images per second on one processor Intel P4 1GHz.

3 Moving object detection

Detection of small and slowly moving objects in a video sequence with strong background motion is hard and demanding task for human operators. The aim of the proposed system is automatic detection of such events. The main requirements posed on the system are detection of small moving objects, marking the corresponding regions of interest and giving a warning to operator.

3.1 Introduction

Motion detection is one of classical tasks of video sequence analysis. The approaches assuming a static camera were studied at first. Most of them are based on direct image comparison [5, 2]. These methods are relatively simple and can be implemented in real-time systems. More sophisticated approaches exploit more complex statistical models (e.g. Bayesian model, Markov Random Fields, etc.) [1, 11]. However the drawback of these algorithms is higher computational cost.

Assuming moving camera the task becomes more difficult since the camera motion should be analyzed at first. Approaches to camera motion estimation can be sorted into three groups. Algorithms from the first group model the image transformation between two frames by a homography or an affine transformation ([9, 3, 12]). Algorithms of the second group track independently the interest points and the camera motion is estimated afterwards from their pose changes ([14, 4]). The third group of algorithms is based on optical flow analysis ([6]). From computational complexity reason the algorithms of the first group are preferred for real-time applications.

3.2 Basic concept

The developed algorithm consists of the following four steps:

1. **Global motion estimation**. The global/camera motion estimation consists in recovering homographies mapping pixels in successive frames.

2. **Global motion compensation.** Camera motion is compensated warping the images according to the estimated homographies.

3. **Suspected region detection.** Images with compensated camera motion are analyzed and the discrepancy regions are labeled as suspected regions.

4. **Motion verification.** The motion of suspected regions is verified by correlation in closed neighborhood.

3.3 Camera motion estimation

Camera motion is modeled by a homography which maps two successive frames in the sequence. Using homogeneous image coordinates the homography model can be written as

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \lambda \mathbf{H} \begin{bmatrix} x \\ y \\ z \end{bmatrix} , \tag{3}$$

where $[x, y, z]$ are homogeneous coordinates in image I, $[x', y', z']$ are coordinates in image J, λ is scale factor and $\mathbf{H}$ is 3×3 homography matrix.

For global illumination changes the following linear modeled is used

$$J_c(r, c) = uJ(r, c) + v, \forall (r, c) , \tag{4}$$

where J,J_c denote intensity in the reference and compensated image respectively, u,v are model coefficients and (r, c) are pixel coordinates.

The final estimation of the geometric and the intensity changes between image I and J is based on a minimization of a quadratic error

$$\sum_{r,c} \Big(I(r, c) - uJ(r', c') + v \Big)^2 , \tag{5}$$

with respect to model parameters (i.e. with respect to $\mathbf{H}$ and u,v). The coordinates (r, c) and (r', c') denote coordinates of corresponding pixels, i.e. their homogeneous coordinates (x, y, z) and (x', y', z') fulfill equation (3).

For solving the mentioned optimization task the tracking software from Imagineer System Ltd is applied. [1] Since the system should detect also slowly moving objects it is impossible to compare just the successive frames in the sequence. More time-distant frames should be analyzed. To achieve high precision and stability, the two-level tracking system was designed (see Fig. 5).

In the first level only preliminary homographies are computed. They are computed frame by frame on lower image resolution.

In the second level the final accurate homographies between more distant frames are estimated. This estimation uses the preliminary homographies as an initial condition. The second level estimation shouldn't be executed for each frame in the sequence. The frequency of calling this routine can be setup with respect to level of background motion. That saves computational cost significantly.

[1] The tracker was lent by Imaginneer System Limited, 40 Occam Road, Surrey Research Park, Guildford, GU2 7YG, UK. The software was lent for laboratory experiments without charges.

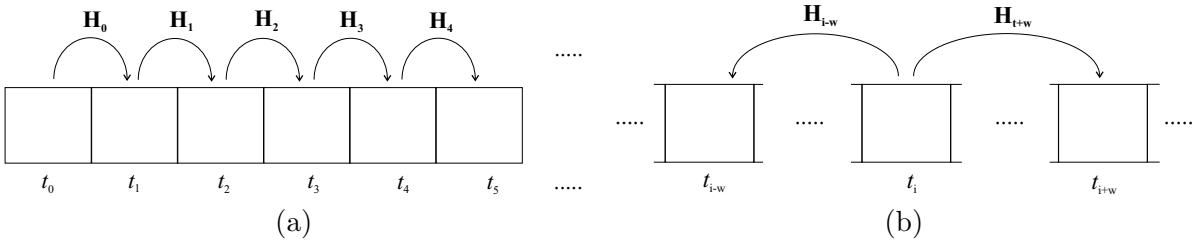


Figure 5: Estimation of global motion, two-level tracking system. (a) The first level: Preliminary estimation frame by frame on lower resolution. (b) The second level: Final accurate estimation of a homography between distant images.

Figure 6: Example of two images from the input sequence.

3.4 Moving object detection

Applying the recovered homographies the stabilized image pairs with compensated global motion are received (see Fig. 7). Thereafter the changes of intensity edges in the stabilized image pairs are analyzed and preliminary selection of suspected regions is created (see Fig. 8).

The changes of image intensity edges did not have to arise just from the moving object. They could appear also due to inaccuracy of camera motion compensation, strong 3-D character of the scene, local illumination changes etc. Therefore motion verification should follow. The motion verification is accomplished by correlating a given image patch around an expected position. Only regions with significant motions are selected as a moving objects (see 9).

Figure 7: Stabilized image pair.

Figure 8: Suspected regions: intensity edges being found only in one of compensated images.

3.5 Conclusion

The developed algorithm detects moving objects of different speeds. The minimal detected speed corresponds to change in pose around 0.3 pixel per frame. The maximal detected speed is around 3 pixels per frame. The algorithm can be implemented as a real-time application on standard personal computer.

Figure 9: Detected moving objects: red and blue color denote position change of detected object in a given image pair.

4 Real-time decision support in on-line video analysis by event visualization

4.1 Introduction

On-line event detection in a video stream, e.g. coming from and unmanned reconnaissance aircraft, is a demanding task requiring full attention of the operator. Maintaining long-term attention produces fatigue and consequently results in mistakes typically of the 'false negative' type, i.e. some events of interest are overlooked and not reported. In this section we show how an object recognition system based on state-of-the-art computer vision technology can support the decision-making process. In our case, presence of certain objects of interest in the field of view is the event to be detected. The detection system is not intended to take over the control of the decision making process. Its output is used to highlight areas were objects of interest are supposedly present. Operator's attention can be attracted both by a visual and acoustic signal. We will not discuss in any detail strategies for operator–detection system co-operation, like prevention of operator's over-reliance on the decision *support system*. The rest of this section describe in some the detail the capability and structure of the object recognition system. We first present experiments demonstrating its utility of the object recognition system. For the interested reader, details of the operation of the system and reference to relevant literature are given.

Figure 10: Object I.

Figure 11: Object II.

Figure 12: Object III.

4.2 Detecting and highlighting objects of interest

Let us consider the following scenario. Before a reconnaissance flight, we are given a set of prototypical images of some object of interest. Examples of such prototypical images are shown in Figs. 10,11 a 12. The spatial arrangement of the objects may or may not be of importance. During on-line analysis of

the video stream (or later, during off-line analysis) it desirable to detect and localize all instance of the objects seen in the photos. Naturally, the orientation and scale of the images can significantly differ between the example photos and the video stream. So our objective is: given a set of example photos and video sequence to be process, detect all frames in the video that are likely to include instances of the objects of interest,and highlight the area where the objects are located.

Figure 13: Sequence I (terrain).

We have applied the Local Affine Frames method developed by Matas and Obdrzalek [8, 7, 10] to this problem. The output of the system for two video sequences is shown in Figs. 13 and 14. As training material, only the depicted images in Figs. 10, 11 and 12 were given , no manual segmentation of the objects of interest was needed. Looking at Fig. 13 we see that system first localized object I and later object II. The areas where matches between the example photo and the video frame were found are enclosed in a color boundary. Each matched object is highlighted too. The object of interest I is localized already in the second frame of the sequence, when only its part is visible. The behavior demonstrates robustness of the recognition algorithm to partial occlusion. In a second example, object of interest III, a house, is detected and localized in a sequence of images of an urban area (see Fig. 14). Again, the object of interest is detected in all frames. Note that the recognition algorithm correctly ignores numerous similar structures (other houses) in the images and it is not influenced by the rotation of the aircraft.

A snapshot of the operator console is shown in Figs. 15 and 16. The number of the training photo that matched is displayed in the bottom left part of the operator's console together with the confidence measure of the match. The matching photo is retrieved from the database and it is presented to the operator to visually check the match. If no match is found, the 'detected photo' part of the screen is

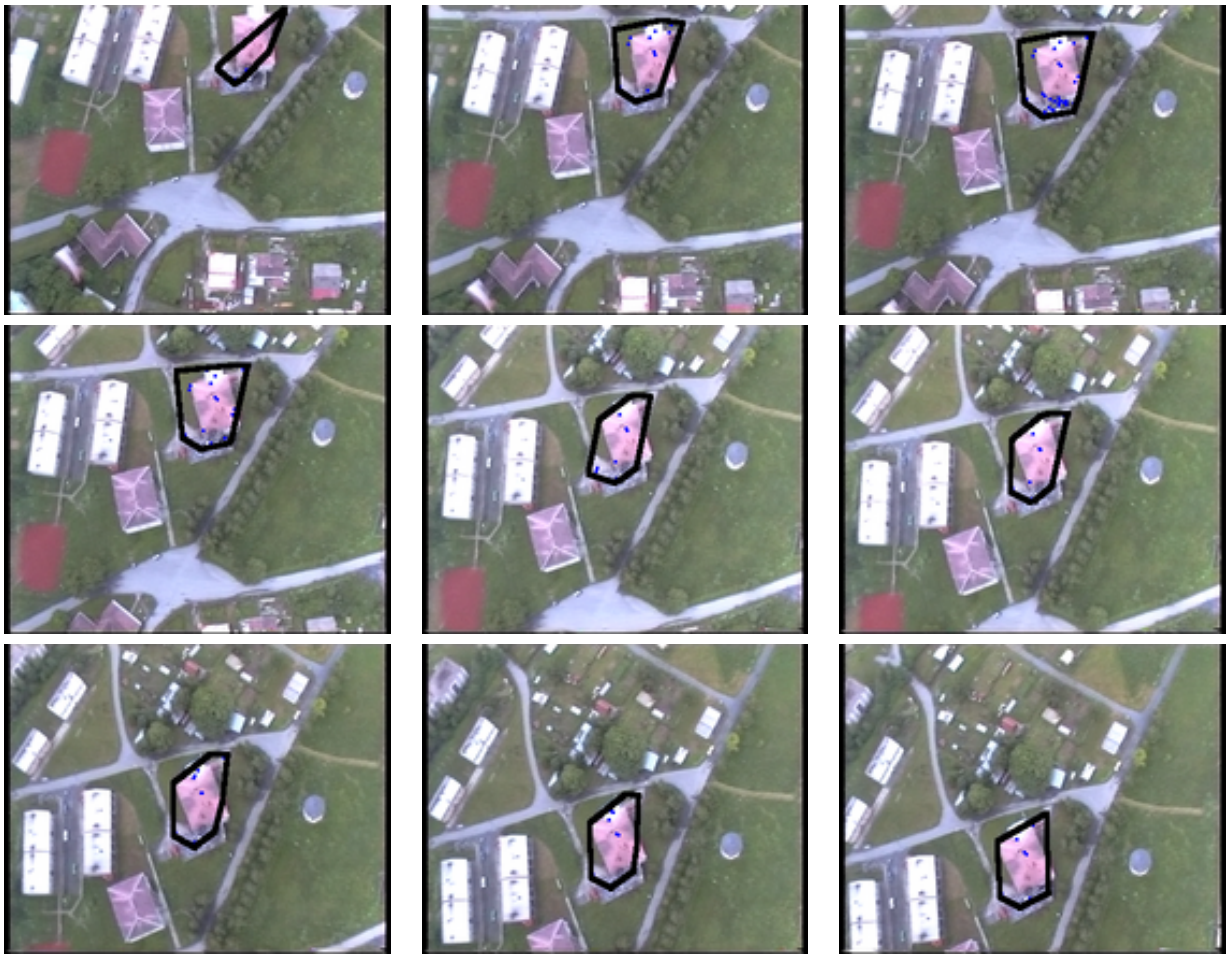


Figure 14: Sequence II (city).

left blank, see Fig. 16. The number of objects the system can watch for is theoretically not limited, but the speed of the system is inversely proportional to the number of objects. At present, it is practical to have about ten photos in the 'watch list'.

From the operator's point of view, the recognition system has the following properties: 1. fully automatic preparation of prototype photos, 2. scale and rotation invariant detection, 3. insensitivity to background and illumination changes. 4. it is intended to support operators decisions by visualizing potentially interesting areas of the image.

4.3 Recognition by matching of Local Affine Frames on Distinguished Regions

The structure of the approach is schematically shown in Figure 17. The method can be described here only superficially, interested reader is referred to [8, 7, 10]. As a starting point, the algorithm is given a 'database' image the example photo in our case, and a 'query image' say some frame of the surveillance video. The addressed problem can be stated as: is there a part of the database image that matches part of query image? Of course, as clearly seen in in Figure 17, the query and database image can be projectively distorted (anisotropic change of scale, rotation, translation and perspective effects), some parts maybe occluded and illumination may differ. The processing of the two images is identical, i.e. the process is symmetric. In step (1), the so called distinguished regions are found. Distinguished regions are parts of the image that can be reliably segmented irrespective of the viewing and illumination conditions. We see in Figure 17, that the distinguished region corresponding to the head of the sheep is segmented in the same way both in the query and database image, regardless of the big (more than threefold) scale change. In step (2), three point local affine frames are found

on each distinguished region by processes that are invariant to affine transformation of the region. Putting the triplets of points into canonical positions (step 3), normalized local patches are obtained. Such normalized patches can be matched by standard correlation technique. The localization of the query is achieved as shown in part (4) of Figure 17. Areas where multiple local affine frames are matched is found and highlighted.

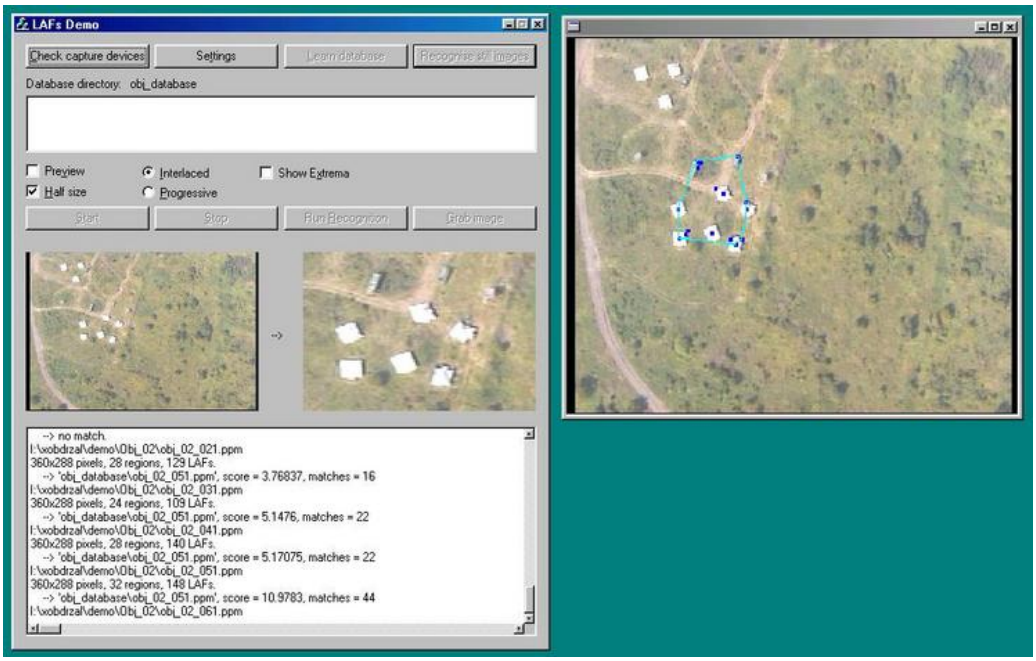


Figure 15: Snapshot of operator's console. Object detected and highlighted.

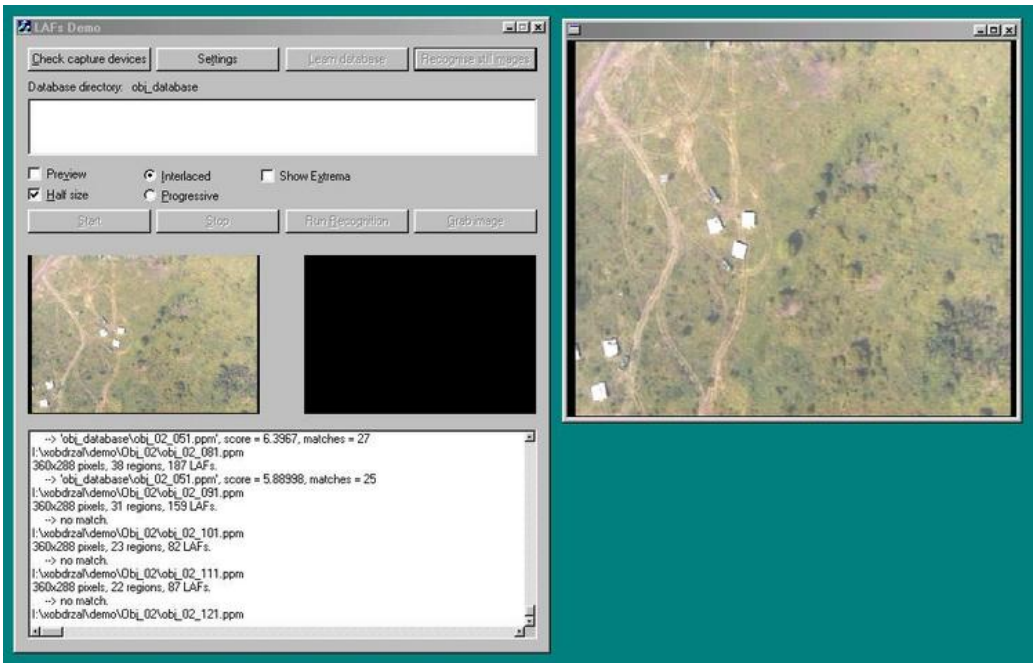


Figure 16: Snapshot of operator's console. No object detected.

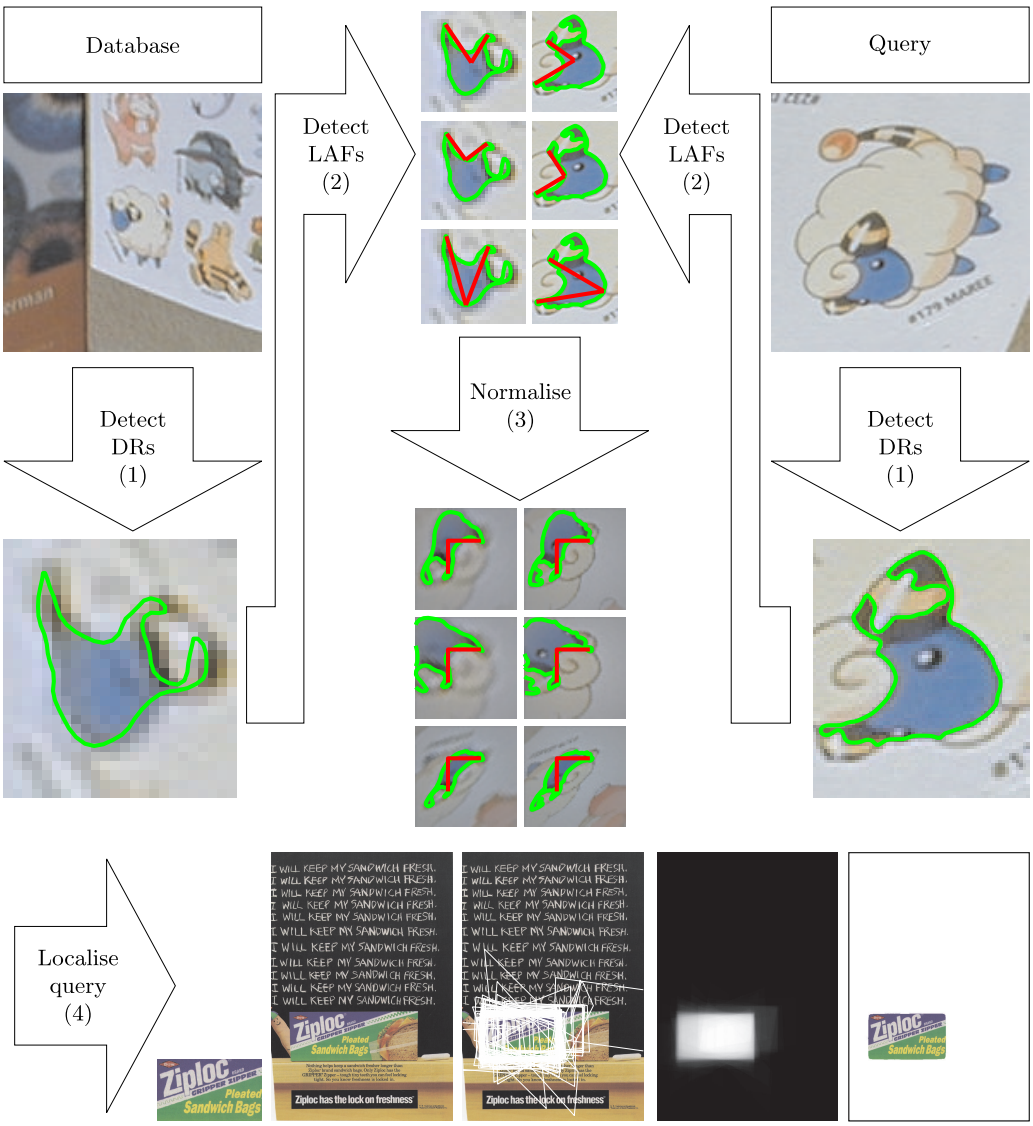


Figure 17: Structure of the Local Affine Frames method.

5 Conclusion

We have presented methods for contrast enhancement, brightness stabilisation, compensation of aircraft movements and detection of small moving objects and an appearance-based object detection and localisation method. All methods were implemented and tested on experimental realistic data acquired during aerial surveillance. The contrast enhancement and motion stabilisation procedures were assessed by professional image interpreters who confirmed their usefullness. With the exception of appearance-based recognition and localisation, the process run in real-time and thus support on-line visualisation as well as post flight analysis.

Acknowledgement

This work was supported by the Czech Ministry of Education under Project LN00B096. The results are part of research project of ACR.

References

[1] T. Aach and A. Kaup. Bayesian algorithms for adaptive changes detection in image sequences using markov random fields. *Signal Processing: Image Communication*, 7(2):147–160, 1995.

[2] P. Bouthemy and P. Lalande. Detection and tracking of moving objects based on a statistical regularization method in space and time. In *ECCV'90*, pages 307–311, 1990.

[3] J. Cohen. A computational approach ti edge detection. *USC Computer Vision*, 1999.

[4] Richard I. Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000.

[5] S.M. Haynes and R.C. Jain. Detection optical flow. *CVGIP*, 21(3):345–367, 1983.

[6] K. P. Horn and B. G Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[7] Jiří Matas, Ondřej Chum, Martin Urban, and Tomáš Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In Paul L. Rosin and David Marshall, editors, *Proceedings of the British Machine Vision Conference*, volume 1, pages 384–393, London, UK, September 2002. BMVA.

[8] Jiří Matas, Štěpán Obdržálek, and Ondřej Chum. Local affine frames for wide-baseline stereo. In R. Kasturi, D. Laurendeau, and Suen C., editors, *ICPR 02: Proceedings 16th International Conference on Pattern Recognition*, volume 4, pages 363–366, CA 90720-1314, Los Alamitos, US, August 2002. IEEE Computer Society.

[9] P.F. McLauchlan, I.D. Reid, and D.W. Murray. Recursive affine structure and motion from image seqences. In *ECCV'94*, volume A, pages 217–224, 1994.

[10] Štěpán Obdržálek and Jiří Matas. Local affine frames for image retrieval. In Michael S. Lew, Nicu Sebe, and John P. Eakins, editors, *CIVR'02: Proceedings of International Conference The Challenge of Image and Video Retrieval*, volume 1, pages 318–327, Berlin, Germany, July 2002. Springer-Verlag.

[11] N. Paragios, P. Perez, G. Tziritas, C. Labit, and P. Bouthemy. Adaptive detection of moving objects using multiscale techniques. In *ICIP'96*, 1996.

[12] Jianbo Shi and Carlo Tomadi. Good features to track. In *CVPR'94*, pages 593–600, 1994.

[13] Milan Šonka, Václav Hlaváč, and Roger D. Boyle. *Image Processing, Analysis and Machine Vision*. Chapman and Hall, London, UK, first edition, 1993.

[14] P. H. S. Torr and A. Zisserman. Robust computation and parameterization of multiple view relations. In *Proc. 6th International Conference on Computer Vision, Bombay*, pages 727–732, January 1998.