Mesh-based Global Motion Compensation for Robust Mosaicking and Detection of Moving Objects in Aerial Surveillance

Marco Munderloh, Holger Meuel, Jörn Ostermann
Institut für Informationsverarbeitung (TNT), Leibniz Universität Hannover, Germany
{munderloh, meuel, ostermann}@tnt.uni-hannover.de

Abstract

Global Motion Compensation is one of the key technologies for aerial image processing e.g. to detect moving objects on the ground or to generate a mosaick image of the observed area. For this task, it is necessary to estimate and compensate the motion of the pixels between the recorded frames evoked by the movement of the camera. As the camera is statically attached to a flying device such as a quadrocopter (also called Micro Air Vehicle, MAV) or a helicopter, the motion of the camera directly corresponds to the plane movements. For simplification, only a planar landscape model is used nowadays to describe the global motion of the scene. However, if objects like buildings or mountains are close to the camera, i.e. the MAV is at a low altitude, this simplification is not valid. Therefore we propose a more complex model by introducing a 2D mesh-based motion compensation technique, also known as image warping, to compensate the global motion. We show the benefits if used for mosaick creation by smaller artifacts due to perspective distortions and smaller drift problems. We also improve a moving object detection system to identify moving objects more reliably. Moreover, the proposed method is also more robust in case of lens distortions.

1. Introduction

Nowadays, aerial surveillance of e.g. disaster areas is becoming more and more important. For example, lately small autonomous drones to be used in a real-time wide area surveillance network have been introduced [11]. These unmanned aerial vehicles (UAV) are intelligent and partially self-organizing, as the infrastructure in a disaster area might be completely destroyed. One of their tasks is the detection of moving objects as cars or persons, or the generation of large mosaick images of the observed area.

To enable this, a global motion compensation has to be performed to evaluate and compensate the motion of the camera during the recording of the sequence. To be able to automatically react on events in the sensor data or use it for navigation decisions, it is necessary to run the surveillance algorithms on-board the UAV itself. As for this scenario the available resources are quite limited, the algorithms have to be easy to compute. For simplification, only a planar landscape model i.e. a projective transform is used nowadays to describe the global motion of the observed ground between frames [11].

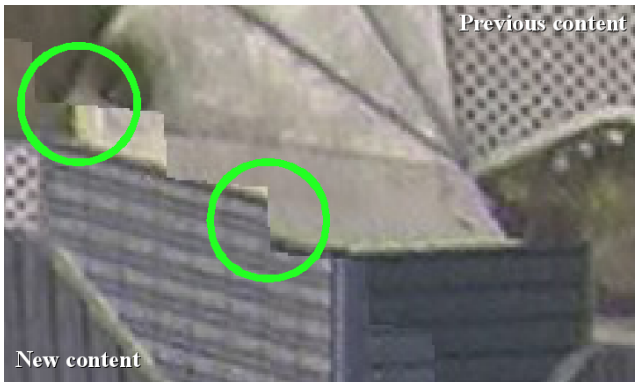

Figure 1. Mapping error due to perspective distortions. The lower left content is added later to the already existing content.

However, if the optical sensor of the UAV is close to the observed scene due to flying at a low altitude this simplification is not valid causing partially misregistration of image content. Figure 1 displays the mapping error caused by the projective transformations of non-planar frames into a mosaick. These disturbing effects are even more problematic if the camera is not pointing directly downwards. To solve this and to keep the computational complexity low, we propose a slightly more complex model by using a 2D mesh motion compensation technique, also known as image warping. In our approach, the planar assumption is applied locally to each of the patches of the mesh resulting in an accurate global motion compensation without the need of computing a full 3D reconstruction [1].

In Chapter 2 we describe the robust motion estimation and the generation of the time varying 2D mesh as well as the motion compensation process. In Chapter 3 we create

a mosaick image using the presented motion compensation algorithm, followed in Chapter 4 by a moving object detection based on the mosaick as a static background model.

2. Mesh-based Motion Compensation

2.1. Motion Estimation

To estimate the motion of the background between frames, one can select a region in one frame and then maximize the correlation by mapping this region to another frame using affine or projective transforms [12]. Another way is to select an amount of representative image points, so called features, in one frame and locate their positions in the following frame. The resulting motion vectors are used to solve an overdetermined linear equation system describing the global motion again as an affine or projective transform. The tracking of the features might be done e.g. by a block matching algorithm or by aligning image gradients.

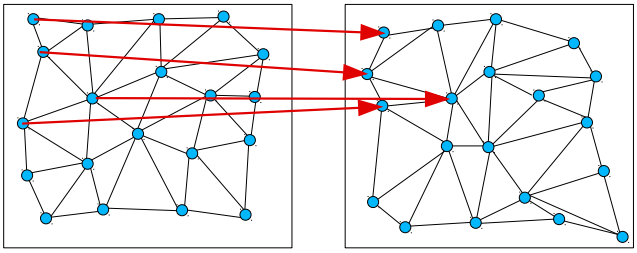

Figure 2. Tracking vector and grid in frame k and $k-1$

Instead of one global solution for the whole frame, the mesh-based motion compensation directly uses the local motion of the estimated feature positions by creating a grid with the feature points as grid nodes (see Figure 2). As the displacement of the grid nodes should reflect the true motion of a small area surrounding each node, the motion of the area between the grid nodes is assumed to be homogeneous, which is more or less true as long as the grid patches are small. The grid structure itself can either be regular or automatically retrieved [9][3] as well as fixed or time varying. Our goal is robustness against a non-planar environment, hence we use automatically derived feature points preferably placed at object corners.

In our approach, the features are selected and tracked using a Kanade-Lucas-Tomasi (KLT) feature tracker [8] which has been extended to provide better tracking results at the image borders. A Harris corner detector is used for automatic feature selection. Figure 3 displays the automatically retrieved feature positions and their trajectories describing their movement over time.

2.2. Outlier Removal

The motion vector field computed by the KLT tracker by following the feature positions has to be cleared of outliers caused by false tracking. To do so, the motion vectors have to be tested against a motion model. The vectors not supported by this model are considered outliers and removed from the motion vector field. Typically the model is an affine or projective transform. By using RANSAC, a linear equation system describing a projective transform is first solved using four random samples of the motion vector field. The resulting motion model is afterwards tested against all remaining vectors. The transform resulting in the most valid motion vectors is used to remove the outliers. Their parameters are refined afterwards using all remaining vectors.

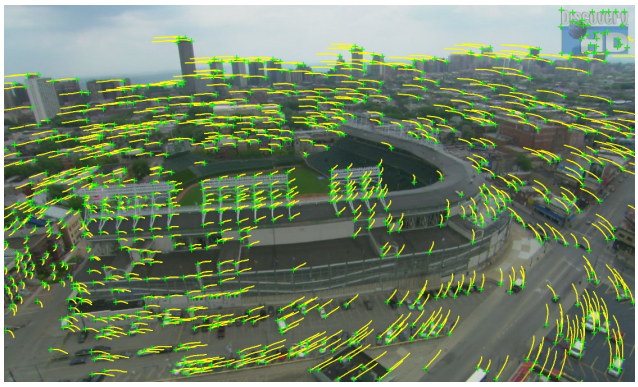

Figure 3. Feature selection (green crosses) and trajectories (yellow lines). Test sequence chicago by Discovery HD.

However, such a model cannot be applied to the mesh-based motion estimation as there is no global solution without a full 3D model available. In our method we solved this problem by using a region growing approach based on motion vector field smoothness [6].

Unclassified motion vectors $\vec{n}_k(x,y)$ neighboring an already classified region are merged into that region if their spatial distance (Equation 1) and their displacement difference to the border of the region (Equation 2) are smaller than a threshold. If one of the thresholds is exceeded, the motion vector remains in the set of unclassified vectors.

$$||\vec{r}_k(x,y) - \vec{n}_k(x,y)|| \quad < \quad t_{d1} \tag{1}$$

$$||\vec{d}_{\vec{r}_k}(x,y) - \vec{d}_{\vec{n}_k}(x,y)|| \quad < \quad t_{d2} \tag{2}$$

$\vec{r}_k(x,y)$ is the position of the closest motion vector of the classified region to the yet unclassified motion vector $\vec{n}_k(x,y)$ at position (x,y) in frame k. $\vec{d}_{r_k}$ and $\vec{d}_{n_k}$ are the displacement vectors of the motion vectors $\vec{(r)}$ and $\vec{(n)}$ in frame k pointing to their positions in frame k-1.

Giving this criterion, a region is represented by the motion vectors on its boundary only. We use hard thresholds in order to reliably detect discontinuities in the motion vector field. Probabilistic methods as e.g. markov random fields tend to undersegment [2] in case of larger grid patches and low differential motion or are too computationally intense to

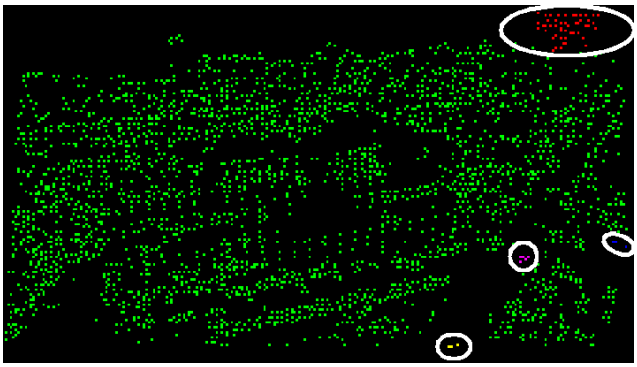

Figure 4. Background region (green) and four other objects.

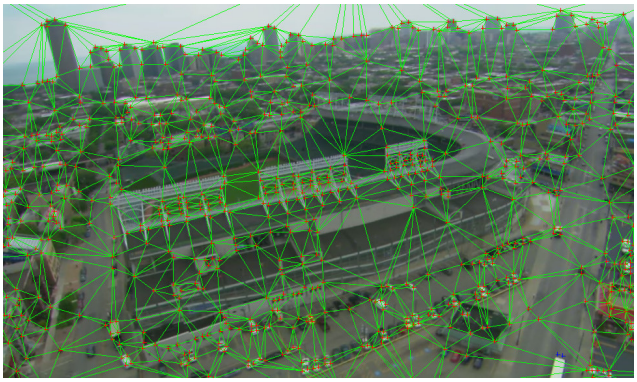

Figure 5. Mesh derived by triangulation of the feature cloud.

be used on-board [10]. Undersegmentation is more harmful to the moving object detection system as an oversegmentation causes moving objects to merge with the background region resulting in being motion compensated and therefore not detected by evaluating image differences.

If no remaining unclassified motion vector is fulfilling these requirements, a new region is created. Regions containing less motion vectors than a threshold t_f are considered outliers and are removed; we used $t_f = 3$. Large regions are treated as background, small regions are considered potentially moving objects and might be used as a priori knowledge in the moving object detection system. Results of the region growing and the outlier removal process are given in Figure 4. All green dots represent motion vectors validated as background. The small colored regions of dots belong to moving cars while the region in the upper right corner is a logo of the broadcasting company. All features on the logo and the moving cars are correctly detected as being inconsistent with the motion model. Moreover, erroneous tracks are completely removed by the region growing approach.

2.3. Motion Compensation

For now, the motion inside the image is known only for those pixels previously selected as a feature. To get a displacement for each pixel of the image, the missing movement information has to be interpolated. In our approach we assume a planar surface in each of the small patches which can therefore be modeled by an affine transform. To get the transform parameters, we first triangulate the feature point cloud of the background region using a Delaunay triangulation (Figure 5).

For each patch of the mesh, the six transform parameters A_{t_k}, B_{t_k}, C_{t_k}, D_{t_k}, E_{t_k}, and F_{t_k} represented by $\mathbf{A_{t_k}}$ and $\mathbf{b_{t_k}}$ are calculated using the three nodes n_{t_k} spanning the triangle t_k in frame k and their positions $n_{t_{k-1}}$ in frame $k-1$. With Equation 3, for each pel (x_{t_k}, y_{t_k}) inside the triangle t_k the associated displaced coordinate $(x_{t_{k-1}}, y_{t_{k-1}})$ is determined.

$$\begin{pmatrix} x_{t_{k-1}} \\ y_{t_{k-1}} \end{pmatrix} = \mathbf{A_{t_k}} * \begin{pmatrix} x_{t_k} \\ y_{t_k} \end{pmatrix} + \mathbf{b_{t_k}} \quad (3)$$

$$\text{with } \mathbf{A_{t_k}} = \begin{bmatrix} A_{t_k} & B_{t_k} \\ C_{t_k} & D_{t_k} \end{bmatrix} \quad (4)$$

$$\text{and } \mathbf{b_{t_k}} = \begin{pmatrix} E_{t_k} \\ F_{t_k} \end{pmatrix} \quad (5)$$

$x_{n_{t_k}}, y_{n_{t_k}}$ specify the coordinates of the pixels inside the triangle t in frame k and $x_{n_{t_{k-1}}}, y_{n_{t_{k-1}}}$ in frame $k-1$, respectively.

Because the coordinate accuracy is not quantized but dependent on the calculation accuracy only, an appropriate interpolation filter must be used to calculate the pixel values. We use a two-stage filter [7]. The half-pel positions are gained using a six tab Wiener filter. The final sub-pel positions are calculated by a bilinear filter.

3. Mosaick Generation

The goal of generating a mosaick image is to provide an overview of the observed area by stitching the single temporal shots together into one big panorama frame. To safe storage space and data link capacity, this panorama frame might also be stored instead of the video sequence. If the camera sequence is recorded by an airplane, this normally results in long stripe-shaped images (Figure 6) containing, in case of non-planar content, distortions as previously displayed in Figure 1.

To create such a mosaick, a global coordinate system has to be defined and a transform for each of the single frames into this global coordinate system must be calculated. Using a mesh-based motion compensation, this is extended to calculating a transform for each of the patches in each of the frames.

To get global coordinates, a fixed amount of features is selected and their movement is tracked over time (see yellow lines in Figure 3). This directly yields to global coordinates, having the first frame defining the coordinate system.

Figure 6. Mosaick panorama image from the Harz mountains.

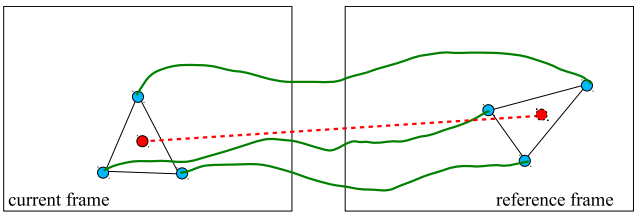

Figure 7. Global coordinates for a newly detected feature (red) are calculated by an affine transform derived by the existing closest features (blue) and their tracked positions (green lines) in the reference frame.

However, each time a track breaks off due to the feature running out of the screen or the feature being occluded, or each time a new feature is newly created in case of too many tracks being broken off, the grid structure slightly changes. Therefore the feature cloud has to be re-triangulated and the structure has to be memorized. If a new feature is created, the first point of the track has to be registered to global coordinates. This is done by again assuming local planarity and applying the affine transform from Equation 3 to the new local feature coordinate. To get the transform parameters $\mathbf{A_{t_k}}$ and $\mathbf{b_{t_k}}$, the local and global coordinates of the three mesh nodes closest to the newly generated feature are used (Figure 7).

All patches are then transformed and mapped into the mosaick coordinate system. If a pixel of more than one patch maps to a pixel of the mosaick, a median filter is applied to reduce noise while maintaining the sharpness. Instead of just using the first frame to define the mosaick coordinate system, any frame of the sequence might be used as base just by applying an offset to the tracked coordinates. This enables to interactively view different angles of non-planar objects inside the 2D mosaick image. In the next chapter we use always the latest recorded frame to define the global coordinate system to create a static background reference frame.

4. Detection of Moving Objects

To detect moving objects, one commonly used method is to investigate the difference between two adjacent frames, whereas one of the frames has to be registered to the other by estimating and compensating the motion between them [5]. The changes between the images are determined by calculating the sum of squared or absolute differences (SSD/SAD) in which moving objects appear as continues regions of high energy. However, if the motion compensation is not perfect, also regions with alignment errors appear in the difference image (stadium and skyscrapers in Figure 8), disturbing any object detection and tracking algorithm following afterwards. Another problem arises if the motion of the moving objects is slow between the two frames: the difference is calculated between two regions of the image containing the same moving object, resulting in a reduced energy level in the difference image and therefore less detection accuracy [4].

To demonstrate both of the problems can be efficiently handled by our proposed method, we selected a test sequence with a practical scenario taken by a helicopter. It contains a large 3D building in the front (stadium) and skyscrapers in the back. The 3D buildings violate the planar assumption of the conventional method but should be handled efficiently by the mesh-based approach.

The sequence also contains moving cars, which should be detected by a simple motion detection algorithm based on inter-frame differences. The motion-compensated frame differences are calculated using SSD. A low-pass noise filter is applied afterwards by averaging a 3x3 block around each pixel of this image. Evaluating a threshold t_n gives an image of motion candidate pixels. The result is displayed as yellow dots in Figure 8 for the projective global motion compensation and in Figure 9 for the proposed mesh-based motion compensation. Additionally, regions of blocks containing more yellow dots than a threshold t_b are marked as containing movement by a white outline.

The mesh-based motion compensation approach produces far less motion candidate pixel in the area of the stadium than the conventional method. Our Mesh-based motion compensation system adapts to the 3D structure of the building while maintaining the moving cars on the streets around the stadium. Also the skyscrapers in the back are correctly motion compensated using the mesh-based approach, producing no false alarms. As a result, the mesh-based approach produces over 90% less false positive motion blocks than the conventional system for the tested chicago sequence. The number of true positives and false negatives stays constant.

One remaining problem is that if the displacement of an objects is high between frames, it is detected as two: once where it had been and once where it is now. Moreover, if the displacement is low, only the front and the back of

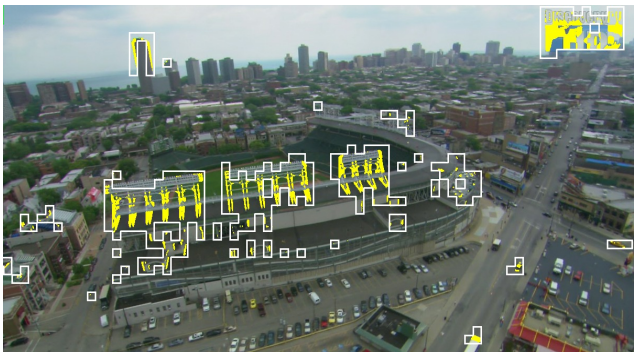

Figure 8. Moving object detection using a projective transform. Motion candidate pixel in yellow, moving areas outlined white.

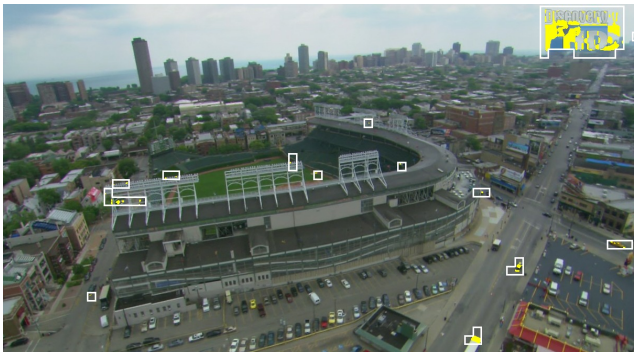

Figure 9. Moving object detection by proposed system. Motion candidate pixel in yellow, moving areas outlined white.

an object is detected as moving as e.g. for cars the roof in the middle is mostly identical in shape and color (see the bus on the road in front of the stadium in Figure 9). To face this problem, we propose to use a motion compensated mosaick image as reference [4] which in our case is created by the mesh-based motion compensation system described in Chapter 2. The previous N frames are registered and mapped into the coordinate system of the current frame, hence a reference image is created having the camera view of the current frame but the image content built from the previous N frames. The pixel values of the preceding frames are weighted dependent on the temporal distance. Regions with detected moving object are skipped during the mapping process, so that the current view represents a still image with all moving objects removed (see Figure 10).

By using the reference still image instead of just the motion compensated previous frame to calculate the differences to the current frame, the amount of motion candidate pixels is increased e.g. in the middle of cars, enabling the use of higher noise filtering thresholds, and resulting in less erroneous regions being detected as moving in Figure 11. In case of slow moving objects not being detected as moving, the usage of a median filter to create the reference still image is able to remove those objects if they are visible for less than $\frac{1}{2}N$ frames at a given pel.

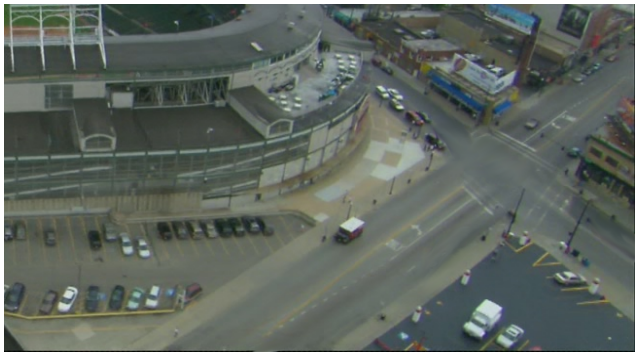

Figure 10. Motion compensated background mosaick still image, moving objects removed

Figure 11. Moving object detection using a still image reference created by the mesh-based motion compensation.

5. Conclusions

In this paper we presented a 2D mesh-based motion compensation system to be used for e.g. mosaick creation or moving object detection. It uses automatically selected features on image corners and tracks their positions over time. To compensate the motion, each patch of the triangulated vector field is mapped using an individual affine transform build up by the corner points of each patch. This gets over the limitations of a single transform for the hole frame while still being less computationally intense and more robust than a full 3D reconstruction. For mosaick image generation, the proposed algorithm can handle the problem of discontinuities caused by inaccurate compensation due to violations of the model of a planar scene efficiently. Moreover, it is able to cope with the erroneous regions in the difference image used for moving object detection and achieves over 90% less false positives. By computing a mosaick still image built up by the N latest frames and using it as the background reference, the detection accuracy is further improved.

References

[1] T. S. Huang and A. N. Netravali. Motion and structure from feature correspondences: a review. *Proc. of IEEE*, 82:252–268. 1

[2] F. Z. B. Kettaf and J. P. D. Asselin de Beauville. A comparison study of image segmentation by clustering techniques. In *3rd International Conference on Signal Processing*, volume 2, Beijing, 1996. 2

[3] W.-H. Lee, S.-Y. Jeong, and D.-S. Jeong. Motion compensation method using active-mesh structure. In *Proc. of IEEE International Symposium on Circuits and Systems (ISCAS)*, volume 2, pages 1305–1308, June 1997. 2

[4] R. Mech. Robust 2d shape estimation of moving objects considering spatial and temporal coherency in one map detection rule. *IEEE Trans. on Image Processing*, 2000. 4, 5

[5] R. Mech and M. Wollborn. A noise robust method for 2d shape estimation of moving objects in video sequences considering a moving camera. *IEEE Signal Processing Magazine*, 66(2):203 – 217, 1998. 4

[6] M. Munderloh, S. Klomp, and J. Ostermann. Mesh-based decoder-side motion estimation. In *Proc. of IEEE International Conference on Image Processing*, pages 2049–2052, Sept. 2010. 2

[7] I. E. G. Richardson. *H.264 and MPEG-4 Video Compression*, chapter 6.5.1.4. John Wiley & Sons Ltd., West Sussex, England, 2003. 3

[8] J. Shi and C. Tomasi. Good features to track. In *Proc. of Comp. Society Conf. on Computer Vision and Pattern Recognition*, pages 593–600, Seattle, WA, June 1994. 2

[9] Y. Wang and O. Lee. Active mesh - a feature seeking and tracking image sequence representation scheme. *IEEE Trans. on Image Processing*, 3(5):610–624, Sept. 1994. 2

[10] J. Xiao, H. Cheng, H. Sawhney, C. Rao, and M. Isnardi. Bilateral filtering-based optical flow estimation with occlusion detection. In *Proc. of European Conference on Computer Vision (ECCV)*, volume 1, pages 211–224, 2006. 3

[11] S. Yahyanejad, D. Wischounig-Strucl, M. Quaritsch, and B. Rinner. Incremental mosaicking of images from autonomous, small-scale uavs. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 329 –336, Sept. 2010. 1

[12] B. Zitová and J. Flusser. Image registration methods: a survey. *Image Vision Computing*, 21(11):977–1000, 2003. 2