An Image Mosaicing Module for Wide-Area Surveillance

Marko Heikkilä and Matti Pietikäinen
Machine Vision Group
Infotech Oulu and Department of Electrical and Information Engineering
P.O. Box 4500, FI-90014, University of Oulu, Finland
{markot,mkp}@ee.oulu.fi

ABSTRACT

This paper presents a fully automatic image mosaicing method for needs of wide-area video surveillance. A pure feature-based approach was adopted for finding the registration between the images. This approach provides us with several advantages. Our method is robust against illumination variations, moving objects, image rotation, image scaling, imaging noise, and is relatively fast to calculate. We have tested the performance of the proposed method against several video sequences captured from real-world scenes. The results clearly justify our approach.

Categories and Subject Descriptors

I.4 [**Image Processing and Computer Vision**]: Miscellaneous

General Terms

Algorithms

Keywords

Image mosaicing, video surveillance

1. INTRODUCTION

Pan-tilt-zoom cameras are often used in wide-area video surveillance systems. They can maximize the virtual field of view of a single camera without the loss of resolution that accompanies a wide-angle lens. They also allow for active tracking of an object of interest through the scene. Traditional surveillance systems depend solely on human monitoring of video input, which is expensive and unreliable. In order to automatize the process, efficient algorithms are needed. One very crucial low-level module in many automatic wide-area surveillance systems is the image mosaicing. It is the process of stitching together two or more images of the same scene taken from different viewpoints or viewing directions. On top of this module, it is possible to develop higher-level functionality such as moving object detection, recognition, and tracking.

Image mosaicing is a very challenging research topic and there are still many open problems to be solved, especially in case of real-world scenes. The mosaicing method should be robust against illumination variations, multimodality of the scene, moving objects, and imaging noise. Furthermore, invariance to image rotation and camera zoom are usually desirable properties. The speed of the method is of importance in applications such as video surveillance. The method proposed in this paper tries to address all of the above-mentioned issues.

VSSN'05, November 11, 2005, Singapore.

2. RELATED WORK

A large number of different approaches to image mosaicing have been proposed. For a good survey, see [15]. The methods can be roughly divided into two classes: *direct methods* such as [14, 11, 8] and *feature-based methods* such as [2, 5, 1, 10]. Both of these have their pros and cons.

Direct methods: The direct methods usually attempt to iteratively estimate the transformation parameters by minimizing an error function based on the intensity difference in the area of overlap. The advantage of the direct methods is that very accurate registration can be achieved since all the available data is used. Direct methods also have several disadvantages when compared to the feature-based methods. In order to avoid the local minima problem, they usually require a good initial guess for the transformation. Direct methods are not very robust against illumination variations because of the nature of the error function to be minimized. Furthermore, the presence of moving objects in the scene can cause serious problems because all the pixel values are taken into account.

Feature-based methods: Instead of using all the available data, feature-based methods try to establish feature point correspondences between the images to be registered. Many different features have been used in the literature, including region, line, and point features. Most of the existing feature-based mosaicing methods use point features such as corners to find the feature points from the images. After the points have been found, they are matched by utilizing a correlation measure in the local neighborhood. Feature-based methods have many advantages over the direct ones. Unlike direct methods, they do not require the initialization step. By selecting appropriate features, these methods can be done very robust against illumination variations. Also, their tolerance against image rotation and zooming is usually better compared to the direct methods. Furthermore,

the scenes with moving objects can be handled robustly by detecting and removing outlier feature points with appropriate methods.

3. THE METHOD

The method proposed in this paper is a pure feature-based method. The SIFT feature detector [9] was adopted for establishing the feature point correspondences between the images. It has proven to be very powerful in what it does, and it clearly outperforms most of the previous approaches. To the authors' knowledge, the SIFT detector has not been widely utilized in image mosaicing yet. This is maybe because the method is relatively new. In [1], the SIFT was used in an application that recognizes and constructs panoramic image mosaics from a set of images fed to the system. We chose a different approach by designing our method for applications, such as video surveillance, where only one image is fed to the system at a time.

One of the most crucial problems in image mosaicing is the accumulation of the local registration errors. Mosaicing methods that require that the images to be registered are captured in advance, and take the whole set of images as the input, usually utilize some kind of global registration stage in the processing [3, 2, 13, 5, 1, 10]. Global registration reduces the accumulated errors by simultaneously minimizing the misregistration between all overlapping pairs of images. Due to the nature of our approach, we cannot utilize global registration. The proposed method is fully causal, that is, it can only use information in the past to estimate the transformation for the current image. To overcome the restriction, we designed our method to be self-correcting by taking into account the history of the pixel values in the blending stage. Furthermore, the method is able to detect such situations where the image cannot be registered reliably. The images that cannot be registered reliably are discarded by the method.

The proposed mosaicing method can be divided into six stages: *image acquisition, feature detection, feature matching, estimation of geometric transformation, warping, and blending*. The block diagram of the method is shown in Figure 1. Each of the six stages is described in more detail in the following subsections.

3.1 Stage 1: Image Acquisition

In order to prevent the motion parallax effect from occurring, images are acquired by a camera that rotates around the optical center of its lens. To the authors' knowledge, this is the case in most wide-area surveillance applications. The relationship between two overlapping images taken by the camera can be described by a planar perspective projection model

$$\mathbf{x}' = \mathcal{T}(\mathbf{x}) = \frac{\begin{bmatrix} m_0 & m_1 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} m_2 \\ m_5 \end{bmatrix}}{\begin{bmatrix} m_6 & m_7 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + 1}, \qquad (1)$$

which maps the point $\mathbf{x}$ in one image plane to the point $\mathbf{x}'$ in other image plane. Estimation of the model requires a search in eight-dimensional parameter space for a set $\{m_0, ..., m_7\}$. In the proposed method, the mosaic representation is anchored at the first image of a video stream. That is, all subsequent images are warped to a coordinate system of the unwarped first image.

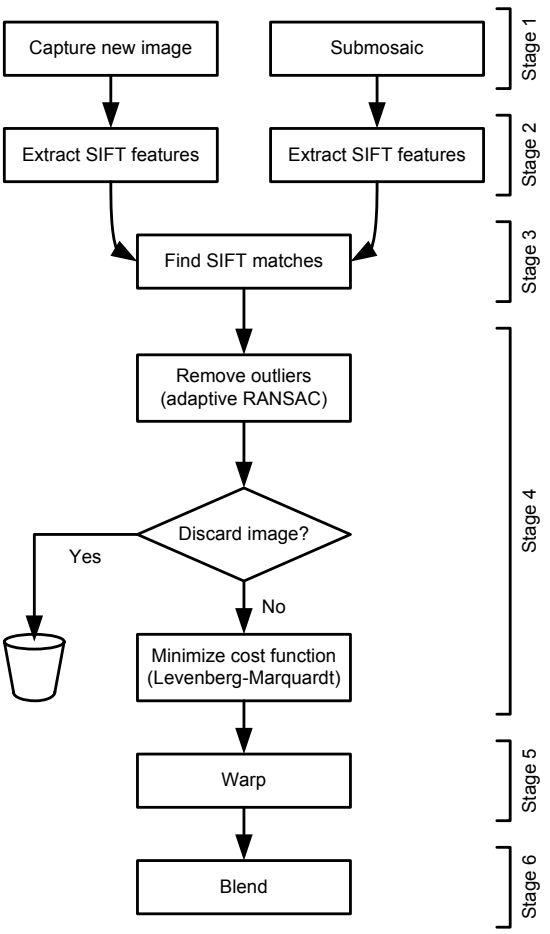

Figure 1: The proposed image mosaicing method.

The submosaic mentioned in Figure 1 is a rectangular area cropped from the mosaic constructed thus far. The details of how the submosaic is selected are explained in the subsection 3.6.

3.2 Stage 2: Feature Detection

After a new image has been captured, the image and the current submosaic are fed to the feature detector. Feature detection is the most critical stage of the proposed method. The method performance mainly depends on how accurately the feature points can be detected and how distinctive they are. Since the method must be able to work reliably in demanding natural environments, there exist several requirements that the detector has to meet. It should be robust against illumination variations, imaging noise, image rotation, image scaling, and if possible, perspective distortions. We tested different approaches presented in the literature including the Kanade-Lucas-Tomasi (KLT) tracker [12] and matching of Harris corners [6] by using a correlation window around each corner, but only one was able to meet our requirements, the SIFT feature detector [9].

The SIFT features are invariant to image scale and rotation, and are shown to provide robust matching across a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination. The features are highly distinctive, in the sense that a single feature can

Figure 2: Upper row: the SIFT features detected from two overlapping images. Lower row: the inlier features selected by RANSAC.

be correctly matched with high probability against a large set of features from other images. The cost of extracting these features is minimized by taking a cascade filtering approach, in which the more expensive operations are applied only at locations that pass an initial test. Following are the major stages of computation used to generate the set of SIFT features. See [9] for details.

1. *Scale-space extrema detection*: The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation.

2. *Keypoint localization*: At each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability.

3. *Orientation assignment*: One or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.

4. *Keypoint descriptor*: The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination. Each keypoint is represented by a 128 element feature vector.

See Figure 2 for an example of feature detection using the SIFT detector. After the features have been detected, they are forwarded to the feature matching stage.

3.3 Stage 3: Feature Matching

For each feature in the new image, we seek for two closest matches in the submosaic image by using the Euclidean distance as the distance measure. If the two distances are too close to each other, the matching cannot be done reliably and the feature is discarded. Otherwise, the closest match is included to the match set. This procedure effectively removes the duplicate matches. In our experiments, we ignored the feature if the ratio of the two closest distances was bigger than 0.6. The feature matching stage outputs a set of feature matches between the new image and the submosaic.

3.4 Stage 4: Estimation of Geometric Transformation

After the feature matching stage, we have a set of feature correspondences between the new image and the submosaic. Most of the duplicate features are removed during the matching process, but there is still a possibility that some outliers, such as mismatched feature points and features that fall on moving objects, are included in the set. In order to achieve a reliable estimate for the projection model, these outliers need to be removed. We chose to use a very well known and robust algorithm, the RANdom SAmple Consensus (RANSAC) [4]. In our experiments, we used an adaptive version of the RANSAC presented in detail in [7]. In our case, the algorithm for removing the outliers and finding a first estimate for the projection model goes like:

- Repeat for N samples:
    1. Select a random sample of 4 correspondences from the match set and compute the projection model, that is, $\{m_0, ..., m_7\}$.
    2. Warp the feature points from the new image into the submosaic image by using the computed projection model and calculate the geometric distances between the warped and true positions by using the Euclidean distance.
    3. Compute the number of inliers consistent with the projection model by the number of correspondences for which the distance is within a user selected threshold.
    4. Adapt N as explained in [7].
- Choose the projection model with the largest number of inliers.

After the RANSAC has done its processing, we have a set of inlier matches and a first estimate for the projection model. According to the experiments, in most cases, the estimate is already relatively accurate. See Figure 2 for an example of inlier detection result.

In order to optimize the performance of our method, we added a stage where we iteratively fine-tune the model parameters $\{m_0, ..., m_7\}$ for better accuracy. This is done by using the Levenberg-Marquardt algorithm for minimizing a geometric cost function for the matches classified as inliers by the RANSAC. Since measurement errors occur in both the images, it is preferable that errors be minimized in both images. The cost function used in the experiments was the *symmetric transfer error*

$$\sum_i [d(\mathbf{x}_i, \mathcal{T}^{-1}(\mathbf{x}'_i))^2 + d(\mathbf{x}'_i, \mathcal{T}(\mathbf{x}_i))^2]. \quad (2)$$

The first term in this sum is the transfer error in the new image, and the second term is the transfer error in the submosaic image. We use the notation $d(\mathbf{a}, \mathbf{b})$ to represent the

Euclidean distance between the inhomogeneous points represented by **a** and **b**. After the fine-tuning, we have a perspective projection model ready for the warping stage where the new image is transformed into the coordinate system of the mosaic.

There is a possibility that the method faces a situation where the projection model estimation cannot be done reliably or at all. This situation occurs, for example, when there is no overlap between the images to be registered. Due to the nature of the projection model used by the method, at least four correspondences are needed between the images. Also, when there is a large number of outlier features present, the probability that the estimation process will fail is relatively high. We used a simple scheme to deal with the situation: the image under registration is discarded and a new image is read from the camera if

$$\frac{\text{Matches after RANSAC}}{\text{Matches before RANSAC}} < T. \quad (3)$$

In our experiments, T was given a value of 0.5.

3.5 Stage 5: Warping

The perspective projection model established during the previous stages can now be used to transform the new image into the submosaic image. To achieve the final projection model between the new image and the mosaic, the projection model is combined with a translation. We realized the transformation in a backward manner. In this way neither holes nor overlaps can occur in the resulting mosaic image. The registered image data from the new image are determined using the coordinates of the target pixel and the inverse of the estimated projection model. The image interpolation takes place in the new image on the regular grid. Many different interpolation methods have been investigated in the literature. In the experiments, we used bilinear interpolation. Even though the bilinear interpolation is outperformed by higher-order methods in terms of accuracy and visual appearance of the transformed image, it offers probably the best trade-off between accuracy and computational complexity.

3.6 Stage 6: Blending

We have now registered the new image with the current mosaic. If new areas were conquered, the pixels belonging to these areas are assigned values directly from the warped new image. Due to various reasons, such as illumination changes, it is possible that there occur intensity differences in the area of overlap. This may cause visible discontinuities to the resulting mosaic image. Therefore, the area of overlap is handled differently from the new areas. In order to seamlessly merge the new image into the mosaic, the blending stage was attached to the method.

In the proposed method, the blending is a process of finding the updated pixel values in the area of overlap by applying a blending function $b(\mathbf{x})$ that outputs a weight between 0 and 1 for each pixel in the new image. The updated pixel values are now generated as follows:

$$I'(\mathbf{x}') = b(\mathbf{x})I(\mathbf{x}) + (1 - b(\mathbf{x}))I'(\mathbf{x}'), \quad (4)$$

where I and I' stand for the pixel values of the new image and mosaic, respectively. A blending function that decreases near the boundary of an image will effectively prevent visible discontinuities from occurring. In the experiments, we used a Gaussian-style blending function which is visualized in Figure 3.

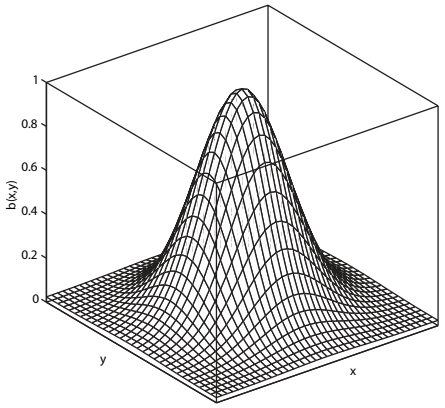

Figure 3: The Gaussian-style blending function used by the proposed image mosaicing method.

Figure 4: The proposed method is able to correct small registration errors. Left image: an erroneous area from a mosaic. Right image: the area after revisiting it.

In the proposed method, the blending is not only used to remove the visual discontinuities, but can be identified as an efficient way of making the method more robust against the accumulation of small registration errors. This is evident, because also the history of the pixel values is taken into account. Small errors can be removed by revisiting the erroneous area. See Figure 4 for an example.

As the final step, we select the submosaic that is used to register the next image captured by the camera. This is because the registration with the whole mosaic would be too expensive to calculate, especially, when the mosaic grows very large. The submosaic is a rectangular area bounded by the bounding box of the warped new image in the mosaic.

4. EXPERIMENTS

We have tested our mosaicing method with several video sequences captured from real-world scenes. See Figures 5-9 for some examples of the results.

The sequences used in Figures 5-7, were taken from [10]. There are no moving objects present in these sequences. As can be seen from the results, our method was capable of generating very accurate mosaics for these sequences. In [10], accurate results were also achieved, but the authors clearly demonstrated that their method will not work robustly if the global registration stage is skipped. This means that their approach does not fit well to applications, such as video

surveillance, where the global registration cannot be easily utilized. The results for our approach show that an accumulation of registration errors has been effectively prevented.

In order to utilize a mosaicing method in applications such as wide-area video surveillance, the method must be tolerant to moving objects that appear in the scene. The mosaic shown in Figure 8 was constructed by our method from a video sequence of 2000 images. The results clearly demonstrate that even if the moving objects are relatively large, the registration can be done reliably. This is because the method effectively detects and ignores the feature matches that fall on moving objects. Results for another sequence containing moving objects are shown in Figure 9.

We also measured the speed of the method. For the sequences used in the tests, the processing of one image took an average of one second by using a standard PC with a AMD Athlon 64 2800+ CPU and 512 MB of memory. All the code is written in C++ programming language.

Ongoing work includes integrating the proposed mosaicing module into a system that detects and tracks moving objects by using a pan-tilt-zoom camera. The preliminary results demonstrate the usability of the proposed mosaicing module. See Figure 10 for some detection results. More results and a description of the methodology will be published in the near future.

5. CONCLUSION

In this paper, we presented a fully automatic image mosaicing method for needs of wide-area video surveillance. The method uses a pure feature-based approach for registering the images. The SIFT feature detector was chosen as the feature detector due to its several desirable properties, including robustness against illumination changes, imaging noise, image rotation, and camera zoom. Experiments with several real-world video sequences, including scenes with moving objects, have shown the efficiency of our approach. We have integrated the proposed mosaicing module into a system that detects and tracks moving objects by using a pan-tilt-zoom camera. Some preliminary detection results were presented here.

6. REFERENCES

[1] M. Brown and D. Lowe. Recognising panoramas. In *Ninth IEEE International Conference on Computer Vision*, volume 2, pages 1218–1225, 2003.

[2] D. Capel and A. Zisserman. Automated mosaicing with super-resolution zoom. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 885–891, Santa Barbara, CA USA, 1998.

[3] J. Davis. Mosaics of scenes with moving objects. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 354–360, Santa Barbara, CA USA, 1998.

[4] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[5] A. Fusiello, M. Aprile, R. Marzotto, and V. Murino. Mosaic of a video shot with multiple moving objects. In *International Conference on Image Processing*, volume 2, pages 307–310, 2003.

[6] C. Harris and M. Stephens. A combined corner and edge detector. In *4th Alvey Vision Conference*, pages 147–151, Manchester, UK, 1988.

[7] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[8] M. Irani, P. Anandan, and S. Hsu. Mosaic based representations of video sequences and their applications. In *5th International Conference on Computer Vision*, pages 605–611, Cambridge, MA, 1995.

[9] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[10] R. Marzotto, A. Fusiello, and V. Murino. High resolution video mosaicing with global alignment. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 692–698, 2004.

[11] J. A. Robinson. A simplex-based projective transform estimator. In *International Conference on Visual Information Engineering*, pages 290–293, 2003.

[12] J. Shi and C. Tomasi. Good features to track. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 593–600, Seattle, WA, 1994.

[13] H. Y. Shum and R. Szeliski. Construction of panoramic image mosaics with global and local alignment. *International Journal of Computer Vision*, 36(2):101–130, 2000.

[14] R. Szeliski. Image mosaicing for tele-reality applications. In *Proceedings of the Second IEEE Workshop on Applications of Computer Vision*, pages 44–53, Sarasota, FL USA, 1994.

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

Figure 5: A mosaic (605×688) constructed from a video sequence of 131 images (340×282). Note: The figure is scaled for a better fit to the page.

Figure 6: A mosaic (540×2050) constructed from a video sequence of 87 images (340×282). Note: The figure is rotated 90 degrees and scaled for a better fit to the page.

Figure 7: A mosaic (928×436) constructed from a video sequence of 146 images (340×282). Note: The figure is scaled for a better fit to the page.

Figure 8: A mosaic (930×291) constructed from a video sequence of 2000 images (320×240). The sequence contains several moving objects. Two images from the sequence are shown below the mosaic. There are no moving objects present in the final mosaic because of the blending. Note: The figures are scaled for a better fit to the page.

Figure 9: A mosaic (1839×437) constructed from a video sequence of 336 images (364×268). The sequence contains several moving objects. Note: The figure is scaled for a better fit to the page.

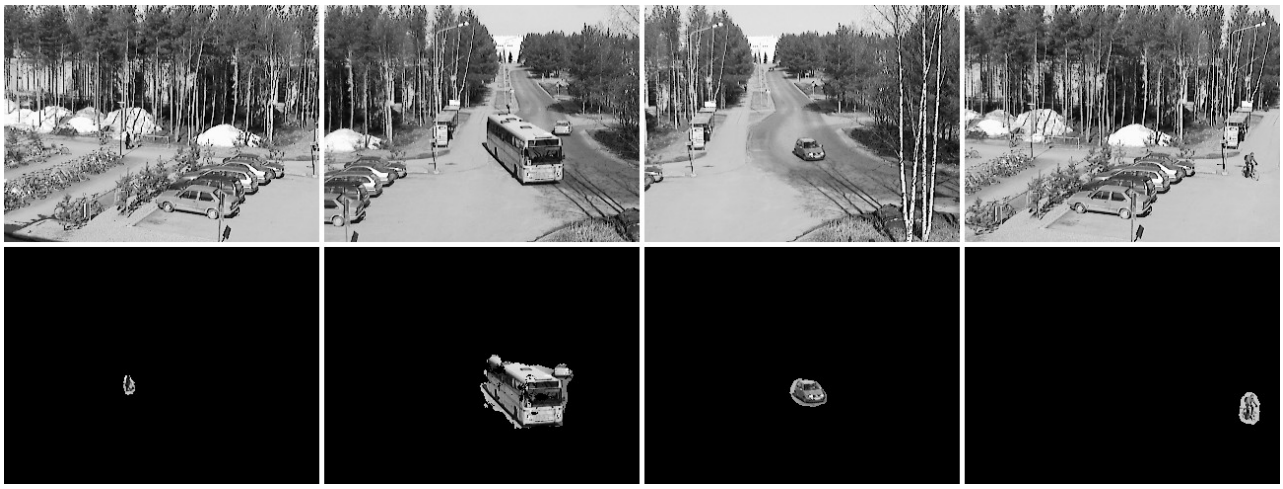

Figure 10: We have integrated our mosaicing module into a system that detects moving objects by using a pan-tilt-zoom camera. Some detection results are shown here.