Vision, Modeling, and Visualization 2005, Aka / IOS Press, pages 49-56

High-Speed Feature Point Tracking

Timo Zinßer*, Christoph Gräßl*, Heinrich Niemann

Chair for Pattern Recognition
University of Erlangen-Nuremberg
Martensstraße 3, 91058 Erlangen, Germany
{zinsser,graessl}@informatik.uni-erlangen.de

Abstract

Ego-motion estimation with a head-mounted camera requires accurate tracking of a small number of features at a very high frame rate. We propose to solve this task with a hybrid feature tracking approach. First, the frame-to-frame feature translation is estimated with an efficient block matching method. Then, an iterative gradient descent estimation of affine motion between the first frame and the current frame is used to refine the translation estimate, prevent feature drift, and detect outliers.

We evaluated the individual algorithms with respect to performance criteria like basin of convergence, robustness, and accuracy. We also conducted experiments on image sequences of real scenes in order to compare the proposed approach with our existing feature tracking system, which is based on the well known Kanade-Lucas-Tomasi tracker.

1 Introduction

A wide range of algorithms in computer vision rely on input data generated by feature tracking. One recent example is the real-time system for structure and motion reconstruction presented in [6]. Its requirements include the tracking of a large number of features at video frame rate.

Ego-motion estimation for augmented reality is another possible application for feature tracking. In [8], the vision-based part of the system processes video images captured by a helmet-mounted camera. The position and orientation of the user's head is then reconstructed from six or more feature points. With this setup, even small movements of the user's head can cause large movements in the image plane. As inter-frame movements can be reduced by increasing the frame rate of the camera, [5] presented a custom-built high-speed CMOS camera, which is capable of capturing small selectable subregions of the complete image at a rate of more than 2500 frames per second.

For tracking a large number of features like in the first application scenario, we use a tracking system based on the Kanade-Lucas-Tomasi tracker [4, 10]. A description of our tracking system can be found in [12]. After the initial multiscale frame-to-frame translation estimation, our system simultaneously computes both affine motion and linear illumination parameters between the first frame and the current frame [3]. This estimation is performed according to the inverse compositional approach proposed in [1] for increased efficiency.

In contrast to the first application scenario, ego-motion estimation with the custom-built CMOS camera requires the tracking of a small number of features at a very high frame rate. This objective is impeded by the time-consuming computation of the multiscale image gradients for each video frame. Furthermore, the high-speed camera is optimized for capturing small image windows, which conflicts with the multiscale translation estimation. Therefore, we propose to replace the gradient descent translation estimation with an efficient block matching algorithm, which entails considerably less computational overhead per video frame. Moreover, we can easily combine this approach with the existing affine motion estimation, which only requires gradient information when new features are added.

Block matching and gradient descent have already been used together for tracking by [11], but our approach is conceptually simpler and fundamentally different. [7] presents a tracking system based on feature matching, which does not attain subpixel accuracy. An analysis of the equivalence

*This work was partially funded by the European Commission's 5th IST Programme under grant IST-2001-34401 (project VAMPIRE). Only the authors are responsible for the content.

of block matching and gradient descent translation estimation can be found in [2]. Fundamental performance limits in image registration with reference to gradient descent estimation are discussed in [9].

After a short overview of our feature tracking system in the next section, we describe all implemented motion estimation algorithms in Sect. 3. Finally, the results of the in-depth evaluation of the individual algorithms and the complete tracking system are presented in Sect. 4.

2 Feature Tracking System Overview

The requirements for a feature tracking system can be categorized into four performance criteria. In our context, the *basin of convergence* of a motion estimation algorithm describes the maximum feature movement in the image plane that can be reliably estimated. Usually, a higher basin of convergence is achieved at the expense of the *computation speed* of the tracker, which is crucial for real-time systems, but less important for off-line systems. The *robustness* criterion subsumes the ability of the feature tracker to cope with adverse conditions, like image noise or changing illumination. For many applications, the *accuracy* of the estimated feature coordinates determines the accuracy of the final output and should therefore be as high as possible.

The basic structure of our real-time feature tracking system is illustrated in Fig. 1. In the preprocessing component, image noise is reduced by applying a Gaussian filter. This operation is especially beneficial to gradient descent motion estimation. In addition, a Gaussian image pyramid including both intensity and gradient images is computed, if required by the subsequent components of the system.

For feature selection, an interest image is computed with the interest operator described in [10]. Then, a very efficient hierarchical search structure is used to select the features with the best interest values, so that lost features can be replaced regularly even under real-time constraints.

The basic principle of the motion estimation component is illustrated in Fig. 2. First, the frame-to-frame translation of a feature is estimated with an iterative gradient descent algorithm. Starting from the estimated position, we perform an affine motion estimation between the first frame, i.e., the reference frame of the feature, and the current frame. In our experience, this is necessary to better detect outliers and to prevent the features from slowly drifting away from their correct position [12]. In order to increase the robustness of the tracker, we also apply a linear illumination model in this step.

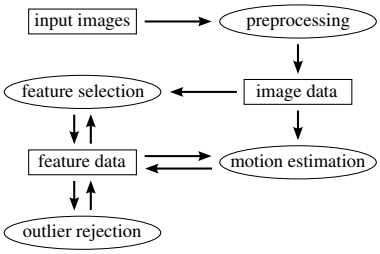

Figure 1: System structure of the tracking system. Rectangles denote different types of data, ellipses denote system components.

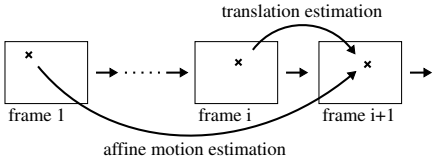

Figure 2: Illustration of motion estimation.

The basin of convergence of the iterative gradient descent translation estimation can be dramatically improved by employing a coarse-to-fine multiscale strategy on a Gaussian image pyramid. In this work, we propose to replace this approach for translation estimation with an efficient block matching algorithm. Although the block matching algorithm is slower than the gradient descent approach per feature, it does not require the initialization of the Gaussian image pyramid and can therefore reach much higher frame rates when tracking a low number of features. Both approaches will be described in more detail in the next section.

Finally, outlier detection enforces a fixed threshold for the maximum sum of squared differences error of corresponding feature window intensities computed by the affine motion estimation. Additionally, we check the singular values of the affine distortion matrix to reject features that are extremely distorted [12]. As the feature tracker is completely data-driven, it can never reliably detect all outliers. One example are features that span across depth discontinuities and therefore do not represent a fixed 3-D feature in the scene.

3 Motion Estimation Algorithms

In this section, we describe the motion estimation algorithms of the proposed tracking system. In our system, the translation estimation algorithms compute the displacement of a feature from the previous frame to the current frame, whereas the affine motion estimation algorithms compute the final feature position with the feature windows of the first frame and the current frame. We also present modifications of the motion estimation algorithms with increased robustness to illumination changes.

3.1 Gradient Descent Motion Estimation

The main idea of the gradient descent algorithms is to iteratively minimize the mean squared difference of the corresponding pixel intensities in two feature windows. Let $f(\boldsymbol{x})$ and $f_c(\boldsymbol{x})$ denote the intensity values of the reference frame and the current frame, respectively. The quadratic reference feature window is represented by a set W of image coordinates $\boldsymbol{x} = (\, x,\, y\,)^T$. Then, the motion parameter update $\Delta\boldsymbol{p}$ is determined in the inverse compositional approach by minimizing

$$\epsilon = \sum_{\boldsymbol{x}\,\in\,W} \left(f(\boldsymbol{g}(\boldsymbol{x},\Delta\boldsymbol{p})) - f_c(\boldsymbol{g}(\boldsymbol{x},\boldsymbol{p}))\right)^2 . \quad (1)$$

In the context of our tracking system, the parameterized warp function $\boldsymbol{g}(\boldsymbol{x},\boldsymbol{p})$ represents either translation

$$\boldsymbol{g}_t(\boldsymbol{x},\boldsymbol{p}_t) = \boldsymbol{x} + \boldsymbol{d}\,, \quad \boldsymbol{d} \in \mathbb{R}^2, \quad \boldsymbol{p}_t = (\, d_1,\, d_2\,)^T$$

or affine motion

$$\boldsymbol{g}_a(\boldsymbol{x},\boldsymbol{p}_a) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{d}\,, \quad \boldsymbol{A} \in \mathbb{R}^{2\times 2}, \quad \boldsymbol{d} \in \mathbb{R}^2,$$

$$\boldsymbol{p}_a = (\, a_{11}-1,\, a_{12},\, a_{21},\, a_{22}-1,\, d_1,\, d_2\,)^T.$$

The parameter vector $\boldsymbol{p}_a$ is defined such that the zero vector yields the identity transformation.

After a first-order Taylor expansion of (1) around $\boldsymbol{g}(\boldsymbol{x},\boldsymbol{0})$ and the introduction of

$$\boldsymbol{h}(\boldsymbol{x}) = \left(\nabla\boldsymbol{f}(\boldsymbol{x})\frac{\delta\boldsymbol{g}}{\delta\boldsymbol{p}}\right)^T$$

and

$$\boldsymbol{H} = \sum_{\boldsymbol{x}\,\in\,W} \boldsymbol{h}(\boldsymbol{x})\,\boldsymbol{h}^T(\boldsymbol{x})\,,$$

we finally get

$$\Delta\boldsymbol{p} = \boldsymbol{H}^{-1} \sum_{\boldsymbol{x}\,\in\,W} \boldsymbol{h}(\boldsymbol{x})\,\left(f_c(\boldsymbol{g}(\boldsymbol{x},\boldsymbol{p})) - f(\boldsymbol{x})\right)\,.$$

The rule for updating the motion parameters is

$$\boldsymbol{g}(\boldsymbol{x},\boldsymbol{p}_{\mathrm{new}}) = \boldsymbol{g}(\boldsymbol{g}(\boldsymbol{x},\Delta\boldsymbol{p})^{-1},\boldsymbol{p})\,.$$

The inverse compositional approach is more efficient than the standard approach, because the matrix $\boldsymbol{H}^{-1}$ does not depend on the current frame or the current motion parameters. Consequently, it only has to be computed once per feature for affine motion estimation, and once per feature and frame and resolution hierarchy level for translation estimation. In both cases, a considerable amount of computation time can be saved.

As the presented gradient descent algorithms directly use intensity values in their computations, they are very susceptible to illumination changes. Therefore, we employ a linear model $\alpha f(\boldsymbol{x}) + \beta$ to adapt both contrast and brightness of the feature windows. Combining the linear model with the objective function in (1) results in

$$\epsilon = \sum_{\boldsymbol{x}\,\in\,W} \left(\alpha f(\boldsymbol{g}(\boldsymbol{x},\Delta\boldsymbol{p})) + \beta - f_c(\boldsymbol{g}(\boldsymbol{x},\boldsymbol{p}))\right)^2 .$$

With another first-order Taylor expansion, we get

$$\epsilon = \sum_{\boldsymbol{x}\,\in\,W} \Big(\alpha f(\boldsymbol{x}) + \alpha\nabla\boldsymbol{f}(\boldsymbol{x})\frac{\delta\boldsymbol{g}}{\delta\boldsymbol{p}}\Delta\boldsymbol{p} + \beta - f_c(\boldsymbol{g}(\boldsymbol{x},\boldsymbol{p}))\Big)^2 .$$

After the definition of two vectors and one matrix

$$\boldsymbol{q} = \left(\, \alpha\,\Delta\boldsymbol{p}^T,\ \alpha,\ \beta\,\right)^T,$$

$$\boldsymbol{k}(\boldsymbol{x}) = \left(\nabla\boldsymbol{f}(\boldsymbol{x})\frac{\delta\boldsymbol{g}}{\delta\boldsymbol{p}},\ f(\boldsymbol{x}),\ 1\right)^T,$$

$$\boldsymbol{K} = \sum_{\boldsymbol{x}\,\in\,W} \boldsymbol{k}(\boldsymbol{x})\,\boldsymbol{k}^T(\boldsymbol{x})\,,$$

the least-squares solution can be written as

$$\boldsymbol{q} = \boldsymbol{K}^{-1} \sum_{\boldsymbol{x}\,\in\,W} \boldsymbol{k}(\boldsymbol{x})\,f_c(\boldsymbol{g}(\boldsymbol{x},\boldsymbol{p}))\,.$$

The integration of the linear illumination model increases the number of parameters from two to four for translation estimation and from six to eight

for affine motion estimation. As working with a larger parameter space potentially decreases the basin of convergence of the estimation, we closely evaluate the relative advantages and disadvantages of integrating the linear illumination model in the next section.

3.2 Block Matching

We propose to replace the gradient descent translation estimation with a block matching algorithm for translation estimation, because it has significant advantages for a number of application scenarios. Most importantly, unlike the gradient descent translation estimation, which requires the computation of a multiscale pyramid for intensity and gradient images, block matching has no computational overhead per video frame. Consequently, it is perfectly suited for high-speed tracking of a small number of features. Furthermore, its basin of convergence can be directly controlled by specifying the size of the search window. This is an advantage for applications where the maximum frame-to-frame displacement of feature points is known a priori. Finally, several similarity measures for block matching are robust against illumination changes, and this robustness is achieved only at the cost of slightly longer computation times.

In order to achieve a more concise notation, we define $f_b(\boldsymbol{x}) = f_c(\boldsymbol{x} + \boldsymbol{d})$ and denote the mean intensity values of the feature windows by $\bar{f}$ and $\bar{f}_b$. We evaluate three similarity measures for the block matching algorithm, which are the *normalized sum of squared differences*

$$s_{\text{ssd}} = \frac{\sum_{\boldsymbol{x}} \left(f(\boldsymbol{x}) - f_b(\boldsymbol{x})\right)^2}{\sqrt{\sum_{\boldsymbol{x}} \left(f(\boldsymbol{x})\right)^2 \sum_{\boldsymbol{x}} \left(f_b(\boldsymbol{x})\right)^2}},$$

the *normalized cross correlation*

$$s_{\text{corr}} = \frac{\sum_{\boldsymbol{x}} f(\boldsymbol{x}) f_b(\boldsymbol{x})}{\sqrt{\sum_{\boldsymbol{x}} \left(f(\boldsymbol{x})\right)^2 \sum_{\boldsymbol{x}} \left(f_b(\boldsymbol{x})\right)^2}},$$

and the *normalized correlation coefficient*

$$s_{\text{coef}} = \frac{\sum_{\boldsymbol{x}} \left(f(\boldsymbol{x}) - \bar{f}\right) \left(f_b(\boldsymbol{x}) - \bar{f}_b\right)}{\sqrt{\sum_{\boldsymbol{x}} \left(f(\boldsymbol{x}) - \bar{f}\right)^2 \sum_{\boldsymbol{x}} \left(f_b(\boldsymbol{x}) - \bar{f}_b\right)^2}},$$

also known as *zero mean normalized cross correlation*.

Figure 3: Illustration of similar feature windows common in man-made environments.

Of course, block matching also has drawbacks compared with gradient descent translation estimation. As mentioned above, for a large number of features, gradient descent is considerably faster. What is more, our efficient algorithm only matches feature windows at integer pixel positions. Although it is generally possible to achieve sub-pixel accuracy by interpolating the intensities of the templates, we use a faster but less accurate approach that applies bi-quadratic interpolation to the similarity values in the neighborhood of the discrete optimum. The obtained accuracy is sufficient for our purposes, because the final translation estimate is determined with the affine motion estimation.

Finally, block matching is prone to mismatching features, especially when the search window is very large and the template size is small. This problem arises frequently when man-made objects, like office buildings, and artificial textures, like block letters (cf. Fig. 3), are part of the video sequence. Gradient descent translation estimation only suffers from this problem when the frame-to-frame displacement is too large, i.e., when a similar feature is at least as close to the expected position as the correct feature.

In order to approximate the behavior of gradient descent estimation, we implemented a mismatch prevention scheme. From the set of all local optima in the search window whose similarity value is below a dynamic threshold, we choose the optimum that is closest to the expected position. Currently, we compute the threshold as the mean value of the 4-neighborhood of similarity values around the global optimum.

4 Experimental Evaluation

In this section, we evaluate the performance of the described algorithms individually and within the complete tracking system. In both cases, the video frames were captured at a resolution of 640×480 pixels with a Sony DFW-VL 500 camera. All com-

putation times were measured on a computer with a Pentium 4 2.4 GHz cpu and 1 GB RAM.

4.1 Evaluation of Individual Algorithms

For the quantitative evaluation of the motion estimation algorithms, we had to generate accurate ground truth data for a high number of features. Consequently, we chose to capture test images of four static scenes with a static camera. For each scene, we captured one base image, one image that was identical to the base image up to image noise, and two images that were exposed slightly brighter and darker than the base image, respectively.

We extracted a total of 2000 features windows from the base images of the four scenes shown in Fig. 4. The motion estimation algorithms were evaluated by tracking these features from the base images to one of the other images of the same scene. As the correct feature positions in these images are identical to the feature positions in the base images, we simulated feature movement by telling the tracker to start its search at a specified displacement from the correct position. Although the performance of the algorithms for distorted feature windows cannot be determined in this way, we think that the accuracy of the available ground truth data outweighs this disadvantage. Additionally, the evaluation of the complete tracking system in the next subsection was performed on more realistic video data, so that any shortcoming of the individual algorithms related to feature distortion or resampling artifacts would be detected there.

If not explicitly stated otherwise, the experiments in this subsection were conducted with the following settings. All images were preprocessed with a 3×3 Gaussian filter, the features were detected using a window size of 5×5, and the feature window size was 11×11. In many experiments, we evaluated the basin of convergence of the individual algorithms. In this regard, a tracking trial was considered a success if the position estimate of an algorithm was within one pixel of the true position.

In order to improve readability, we use abbreviations in the figures of this section. These abbreviations are *bm* for block matching, and *ssd*, *corr*, and *coef* for its respective similarity measures. Furthermore, we write *gd* for gradient descent, *trans* for translation estimation, *affine* for affine motion estimation, and *ill* for the linear illumination model described in the last section.

Figure 4: Base images for evaluation of individual motion estimation algorithms.

Figure 5: Basin of convergence of motion estimation algorithms for maximum feature displacement of ten pixels, brighter intensities denote higher percentage of successful tracking. Algorithms from left to right: bm coef, gd trans, gd trans + ill, gd affine, gd affine + ill.

In our first experiment, we evaluated the basin of convergence with respect to the 2-D feature displacement. We tracked each of the 2000 features from its base image to the corresponding image with fixed illumination, simulating the displacements as described above. In Fig. 5, the percentage of successful tracking attempts is encoded as a gray value, with pure white representing 100 percent. The initial feature displacement is encoded in the pixel positions of the shown images. It is zero in their center, and as high as ten pixels at their edges. A search range of eight pixels was chosen for the block matching approach.

The main purpose of this experiment is to demonstrate the shape of the different basins of convergence. It resembles the rectangular search window for the block matching approach and is approximately circular for the gradient descent algorithms. It can also be seen that adding the linear illumination model reduces the size of the basin of convergence of the gradient descent algorithms.

As the basins of convergence are very regular, we only consider the distance of the feature displacements for the remaining experiments of this subsection. To this end, we displace the features along the coordinate axes and their diagonals, which yields eight trials for each displacement distance for each

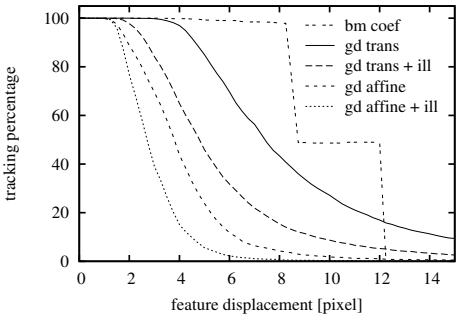

Figure 6: Basin of convergence of motion estimation algorithms for fixed illumination.

Figure 7: Test images for evaluation of robustness to illumination changes for one of four scenes.

feature. In Fig. 6, the results of the first experiment are plotted according to this scheme. In the range of eight to twelve pixels, the block matching approach successfully tracked the features displaced along the diagonals of the coordinate axes, because only they were still within the search window. The basin of convergence is generally larger for translation estimation than for affine motion estimation, which has more unknown parameters.

The remaining experiments evaluate the algorithms on images with varying illumination. One example for the changes in illumination, which were obtained by changing the aperture of the camera, can be seen in Fig. 7. The middle image is the base images providing the reference features. As the illumination changes are fairly large, the following experiments represent more or less the worst case of what has to be expected for frame-to-frame tracking on real video sequences.

The results of the experiment of Fig. 6, but now conducted on images with varying illumination, are presented in Fig. 8. As can easily be seen, the performance of the gradient descent algorithms without illumination compensation suffers significantly. The block matching with the normalized correlation coefficient and the gradient descent algorithms with illumination compensation are obviously not

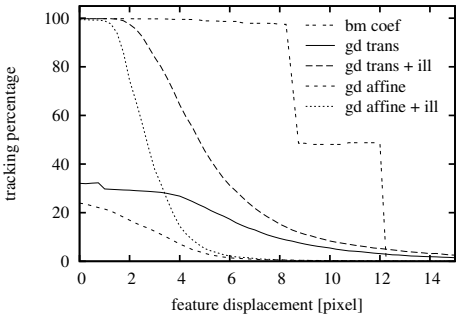

Figure 8: Basin of convergence of motion estimation algorithms for varying illumination.

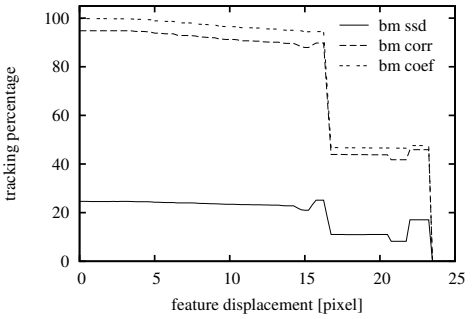

Figure 9: Basin of convergence for block matching with different similarity measures.

affected by the illumination changes, as their results are almost identical to those of Fig. 6.

In the third experiment, we evaluated the performance of block matching with the different similarity measures described in the previous section. Here, we used a larger search range of 16 pixels. The results of the normalized sum of squared differences are the worst. With the normalized cross correlation, block matching performs significantly better. Many features that were classified as mismatches were only off by one or two pixels. This phenomenon does not occur with the normalized correlation coefficient. Thus, it is the best similarity measure for our purposes.

Previous experiments have shown that the maximum feature displacement is rather small for gradient descent translation estimation. This problem can be solved by using a coarse-to-fine multiscale strategy on a Gaussian image pyramid. As illus-

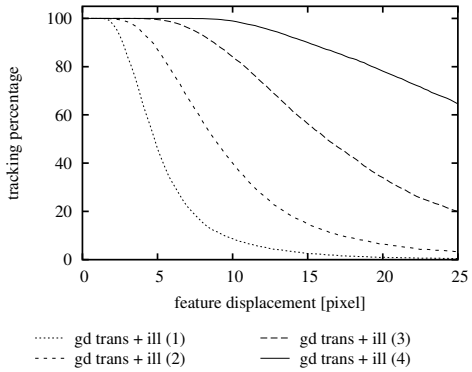

Figure 10: Basin of convergence for gradient descent translation estimation with different numbers of hierarchy levels.

Figure 11: From left to right: frames 1, 50 and 100 of the test video sequence.

	gd trans		bm coef	
# features	atl	time	atl	time
10	91.6	6.2	91.6	**2.8**
30	97.2	8.3	97.2	**6.2**
100	96.0	**15**	95.5	16
300	92.0	**35**	91.7	43

Table 1: Average trail length (atl) and computation time per frame in milliseconds for four different numbers of features.

trated in Fig. 10, the basin of convergence approximately doubles for every additional level. With four hierarchy levels, displacements of up to ten pixels can be reliably tracked, and the top-level image has a size of only 80×60 pixels. Consequently, using a higher number of hierarchy levels is not advisable.

The accuracy of motion estimation algorithms is another important performance criterion. Experiments conducted on the images with illumination changes yielded an average translation error of 0.18 pixels for gradient descent translation estimation with illumination compensation, and 0.17 pixels for block matching with the normalized correlation coefficient, and 0.22 pixels for gradient descent affine motion estimation with illumination compensation. Even better results have to be expected under less demanding conditions.

Several conclusions can be drawn from the experiments presented in this subsection. Firstly, gradient descent translation estimation should always be used with a coarse-to-fine multiscale strategy to improve its basin of convergence. Secondly, although the linear illumination model decreases the basin of convergence of gradient descent motion estimation, we strongly encourage its use for affine motion estimation. The accuracy of all evaluated translation estimation algorithms is so high that the starting position for affine motion estimation lies well inside its basin of convergence. Lastly, both basin of convergence and estimation accuracy of gradient descent translation estimation and block matching are comparable, so that a decision for one or the other can be based on the computation speed, which will be analyzed in the next subsection.

4.2 Tracking System Experiments

For the experiment described in this subsection, we evaluated two translation estimation algorithms in the context of our tracking system. We used a test video sequence with 100 frames, three of which can be seen in Fig. 11. We combined both translation estimation algorithms with the gradient descent affine motion estimation with illumination compensation. Due to the good quality of the video, we did not apply a preprocessing filter. For the gradient descent translation estimation, we used the multiscale approach with three levels. For block matching, we set a search range of eight pixels.

The results of our tracking system experiment are summarized in Tab. 1. In our case, an average trail length of 100 means that all features were successfully tracked through the complete sequence. But this is not possible, because features are occluded and leave the field of view, which happens in almost every video sequence with moving objects. As the average trail length achieved with both algorithms is roughly the same, the tracking results are more or less equal. This fact was also endorsed by manual inspection of the tracking results. Although the illumination changes between the frames shown in Fig. 11 are rather dramatic, the much smaller frame-

to-frame changes can obviously be handled by the pure translation estimation algorithm.

A much larger difference between the algorithms can be observed when analyzing their computation times. For ten features, the block matching approach is more than twice as fast as the gradient descent algorithm. Although block matching requires more computation time per feature, the large overhead for computing the Gaussian image pyramid considerably slows down the gradient descent algorithm. The turning point is reached at 100 features, where both configurations are capable of tracking at a rate of more than 50 frames per second.

5 Conclusion

We proposed to combine an efficient block matching approach with gradient descent estimation of affine motion to form a feature tracking system that is specialized on high-speed tracking of a small number of features. In order to make the tracker more robust to illumination changes, we detailed how to efficiently combine gradient descent motion estimation with a linear illumination model.

In our experiments, we first evaluated the individual algorithms for motion estimation. Testing different similarity measures for block matching, we found the normalized correlation coefficient to be ideal for our purposes. The comparison of block matching and gradient descent translation estimation indicated that both approaches perform similarly with respect to their basin of convergence, robustness, and estimation accuracy. Finally, tests with our complete tracking system showed that the proposed combination of block matching and gradient descent improves the computation speed of our very efficient standard system by more than a factor of two, when only a small number of features have to be tracked.

References

[1] S. Baker and I. Matthews, "Lucas-Kanade 20 Years On: A Unifying Framework", *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221-255, 2004.

[2] C. Davis, Z. Karu, and D. Freeman, "Equivalence of Subpixel Motion Estimators Based on Optical Flow and Block Matching", *Proceedings of the IEEE International Symposium on Computer Vision*, pp. 7-12, 1995.

[3] H. Jin, P. Favaro, and S. Soatto, "Real-Time Feature Tracking and Outlier Rejection with Changes in Illumination", *Proceedings of the IEEE International Conference on Computer Vision*, pp. 684-689, 2001.

[4] B. D. Lucas, T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision", *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pp. 674–679, 1981.

[5] U. Mühlmann, M. Ribo, P. Land, and A. Pinz, "A New High Speed CMOS Camera for Real-Time Tracking Applications", *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 5195-5200, 2004.

[6] D. Nister, "An Efficient Solution to the Five-Point Relative Pose Problem", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756-770, 2004.

[7] D. Nister, O. Naroditsky, J. Bergen. "Visual Odometry", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 652-659, 2004.

[8] M. Ribo, H. Ganster, M. Brandner, P. Lang, C. Stock, and A. Pinz, "Hybrid Tracking for Outdoor AR Applications", *IEEE Computer Graphics and Applications Magazine*, vol. 22, no. 6, pp. 54-63, 2002.

[9] D. Robinson and P. Milanfar, "Fundamental Performance Limits in Image Registration", *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1185-1199, 2004.

[10] C. Tomasi and T. Kanade, "*Detection and Tracking of Point Features*", Technical Report CMU-CS-91-132, Carnegie Mellon University, 1991.

[11] Q. Zheng and R. Chellappa, "Automatic Feature Point Extraction and Tracking in Image Sequences for Arbitrary Camera Motion", *International Journal of Computer Vision*, vol. 15, pp. 31-76, Kluwer Academic Publishers, 1995.

[12] T. Zinßer, Ch. Gräßl, and H. Niemann, "Efficient Feature Tracking for Long Video Sequences", *Pattern Recognition, 26th DAGM Symposium*, pp. 326-333, Springer-Verlag, 2004.