Object Oriented Motion Estimation in Color Image Sequences

Volker Rehrmann

University of Koblenz–Landau
Image Recognition Lab
Rheinau 1, D-56075 Koblenz
volker@uni-koblenz.de

Abstract. This paper describes a color region-based approach to motion estimation in color image sequences. The system is intended for robotic and vehicle guidance applications where the task is to detect and track moving objects in the scene. It belongs to the class of feature-based matching techniques and uses color regions, resulting from a prior color segmentation, as the matching primitives. In contrast to other region-based approaches it takes into account the unavoidable variations in the segmentation by the extension of the matching model to multi matches. In order to provide extended trajectories, color regions that could not be matched on the feature level are matched on the pixel level by the integration of a correlation-based mechanism. The usage of color information and the combination of feature-based and correlation-based matching leads to robust and efficient algorithms. The system was applied to a motion segmentation task in vehicle guidance. Experiments on more than 1000 natural color outdoor images, taken from a moving car, show promising results.
Keywords: Motion estimation, motion segmentation, color vision, feature matching

1 Introduction

Motion estimation has long been a field of extensive research (s. [8] for an overview). However, motion estimation in color image sequences has rarely been studied due to different reasons. The devices to store high quality color image sequences at high sampling rates have been very expensive. The computational complexity of motion estimation algorithms in gray value images was already high enough. The classical differential approach gained little when transferred to RGB images. The three color channels are highly correlated, such that the improvements did not justify the increased complexity. Nevertheless, experiences from image segmentation have shown that color is a rich source of information that can significantly improve the robustness of vision algorithms. Our intention is the development of a system for the analysis of color image sequences that is accurate, robust and efficient. In [7] it was stated that most of the existing motion algorithms are either very fast or very accurate. What is needed in real

world applications is a good trade–off between accuracy and efficiency. In this paper we present our OOMECS (Object Oriented Motion Estimation in Color Image Sequences) system that tries to fill this gap and is intended to provide fast and reliable results. Our system belongs to the class of feature-based matching techniques which are known to be more robust than the differential techniques, at least in natural scenes. In contrast to the common corner or edge matching techniques that provide only sparse displacement fields we use complete color regions as matching primitives. Thus, prior to the matching phase each image is independently segmented using a fast and robust color segmentation algorithm. Region-based approaches for motion estimation have been applied in gray value images in [2] and [6]. The image is segmented using various approaches and the correspondence of regions is based on features. The displacement among centroids determines the displacement between regions. In [3] a complex texture-based segmentation of a gray value image is used to get the initial partition into regions. Then affine motion parameters are computed for each region using a robust multi-resolution estimator. Price ([10]) was the first to explore a region-based change detection in color images. In his algorithm the images are segmented using the well-known recursive histogram splitting technique. The resulting regions are described by features and the feature-based descriptions of two images are compared to determine the corresponding regions. In [4] a similar approach was used to track color regions, generated by simple color space clustering, over long image sequences. In [1] the segmentation and motion estimation is simultaneously performed by a clustering process based on color, motion and pixel position.

The simple region-based approaches trying to match one region in an image with exactly one corresponding region in the successive image are successful in simple scenes, where enough contrast exists between regions to stably segment them from frame to frame. However, in natural scenes even the best segmentation algorithm usually can not guarantee that regions will be uniformly stable segmented. More elaborate approaches based on statistical regularization provide accurate results but are far away from fast processing. In our OOMECS system we combine a very efficient feature-based matching algorithm with an accurate correlation-based matching algorithm. As much regions as possible are matched on the feature level. To account for the unavoidable variations in the segmentation we extended our matching model from simple $1:1$ matches to the general case of $n:m$ matches, where sets of color regions are matched. In order to provide extended trajectories, color regions that could not be matched on the feature level are matched on the pixel level using a correlation-based mechanism. The object-oriented correlation provides very accurate results but requires elaborate processing. The combination of both techniques assures that the main design goals, robustness and efficiency, are achieved. The system is intended for robotic and vehicle guidance applications where the task is to detect and track moving objects in the scene. It is not intended for high accuracy measurements like structure from motion, although further algorithms may be based on its results. In the next section we present the color region based motion estimation

algorithm combining matching on feature and pixel level. In section 3 we present results of our experiments with more than 1000 color outdoor scenes from the field of vehicle guidance. The system was applied to the problem of segmenting moving objects from color image sequences. The results show the effectiveness of this approach in such difficult real-world problems. The speed of the algorithm allows for a fast processing rate.

2 Color Region Matching

We assume color image sequences of natural color outdoor scenes with an arbitrary number of moving objects especially due to a moving camera. As our system is based on matching color regions, in a first step each color image is independently segmented. We use the *CSC* (Color Structure Code) as our segmentation tool (s. [11]). The CSC has proven to be an efficient and reliable scene segmenter in a variety of real-world applications. Figure 7 shows an example of the segmentation results of two consecutive images. However, any other color segmentation tool can be used. The matching quality, of course, increases with the stability of the segmentation results in successive images. The input to the color region matching algorithm is therefore a partition of each image in the sequence into a set of color regions. The goal of the color region matching is to find a definite correspondence between color regions of two consecutive images. We explicitly allow for the fact that a segmented region in one image may split or merge into several regions in the next image. We therefore extended our matching model from simple $1:1$ matches to the general case of $n:m$ matches. A $1:1$ match, where one color region corresponds to exactly one color region (s. Fig. 1 (a)), is the most simple case and the only kind of match accounted for in the previously published papers on region matching. Often a color region splits into two or more regions in the segmentation of the next image. This is a typical $1:n$ match (s. Fig. 1 (b)). The analogous case is the $n:1$ match (s. Fig. 1 (c)). The most general case is the $n:m$ match (s. Fig. 1 (d)): a set of n color regions corresponds to a set of m color regions. Our systems accounts for this general model of $n:m$ matching and is thus more robust against unavoidable variations in the segmentation. Slightly over-segmented images are favorable because objects split into several regions can be composed again in the $n:m$ matching algorithm, whereas differently moving objects merged together into one color region need some special consideration. In this paper we assume that motion boundaries coincide with spatial boundaries. The next version of our system will not be based on this assumption.

The matching phase is divided into two main steps. In a first step we try to match regions solely based on their region attributes. This feature-based matching is very efficient and is best explained in terms of graph algorithms. Color regions that could not be matched on feature level are matched on pixel level in the *iconic color region matching* step. Here color regions are matched using a correlation-based mechanism delivering very accurate results at the expense of higher computational complexity. The combination of both techniques leads to

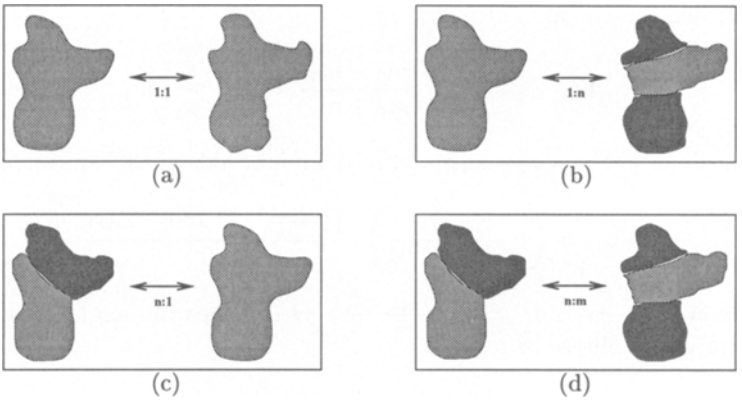

Fig. 1. Four different cases of region matching, from simple $1:1$ matches (a) to complex $n:m$ matches (d).

reliable and robust results while being very efficient. The limits of the system are obviously reached when no homogeneous color regions appear in the image.

2.1 Feature-Based Color Region Matching

The feature-based color region matching relies on the computation of the similarity of two color regions from consecutive images solely based on their features. In order to efficiently handle the complex $n:m$ matches the algorithm operates in four phases, which are described in the following.

Similarity function. In this section we define the similarity function used to compare two color regions. We use the following features for the feature-based matching of color regions: color, area, position (centre of gravity), proportion (of the equivalent ellipse, calculated from the second central moments), orientation (of the equivalent ellipse), minimum bounding rectangle.

In order to reduce the computational complexity of the matching phase we only analyze regions with a minimum size of 40 pixels. The overall similarity EQ between two regions is calculated as the weighted sum of the single features similarity values:

$$EQ = \omega_{color} \cdot EQ_{color} + \omega_{area} \cdot EQ_{area} + \omega_{centre} \cdot EQ_{centre} + \omega_{prop} \cdot EQ_{prop} + \\ + \omega_{ori} \cdot EQ_{ori} + \omega_{mbr} \cdot EQ_{mbr}$$

All single similarity values are normalized to lie in the range [0..1] and are calculated as follows:

$$EQ_{area} = \frac{|A_0 - A_1|}{A_0 + A_1} ,$$

where A_i denotes the area of region i.

$$EQ_{prop} = \frac{1}{2} \cdot \left(\frac{|m_0 - m_1|}{m_0 + m_1} + \frac{|s_0 - s_1|}{s_0 + s_1} \right) ,$$

where m_i denotes the major axis and s_i the minor axis of the equivalent ellipse.

$$EQ_{ori} = \left(1 - \frac{1}{e^{4 \cdot (\Pi - 1)}}\right) \cdot \frac{\left| \left\lfloor \frac{\Delta\Theta + 90}{180} \right\rfloor \cdot 180 - \Delta\Theta \right|}{90} ,$$

with $\Delta\Theta = |\Theta_0 - \Theta_1|$, $\Pi = min(\frac{m_0}{s_0}, \frac{m_1}{s_1})$. Θ_i denotes the angle of orientation of the equivalent ellipse of region i.

$$EQ_{mbr} = \frac{1}{2} \cdot \left(\frac{|w_0 - w_1|}{w_0 + w_1} + \frac{|h_0 - h_1|}{h_0 + h_1} \right) ,$$

where w_i denotes the width and h_i the height of the bounding rectangle.

EQ_{color} and EQ_{centre} are calculated as normalized Euclidean distances between color vectors in the CIE-Lab color space respectively position vectors. The complex equation for EQ_{ori} computes the simple difference between two angles without considering the different quadrants. It is additionally weighted depending on the proportions of the involved regions, as the reliability of the orientation decreases with decreasing proportion (a circle with proportion 1 has no orientation). The weights ω in the overall similarity equation for EQ can be set specific to the application. We use a high weight for color and position and a small weight for the other features in our applications.

This similarity function is also applied to compare two sets of color regions ($n : m$ match). The composition of several color regions implies the calculation of the new composed feature set from the features of all involved single color regions. This computation is very efficient because it can be computed directly from the feature values of the involved regions without referring to their single pixels. Thus, we use the same similarity function, no matter if comparing single color regions or composed color regions.

Four Phase Matching. Phase 1 In the first phase a graph is built, connecting all regions in image I_t at time t with all regions in image I_{t+1} at time $t+1$ that are possible candidates for a match. Let $G_{inter} = (V, E)$ be an undirected graph with vertex set V and edge set E. The vertices of G are the color regions in I_t and I_{t+1}. Color regions r_1 from image I_t and r_2 from image I_{t+1} are connected by an edge, if:

1. $EQ_{color}(r_1, r_2) <$ max_color.
2. $EQ_{centre}(r_1, r_2) <$ max_dist.

The graph G_{inter} (s. Fig. 2) is solely based on the two main features color and position. Only regions connected by an edge in G_{inter} can be matched in one of the following phases. A connection in G_{inter} is therefore a necessary condition

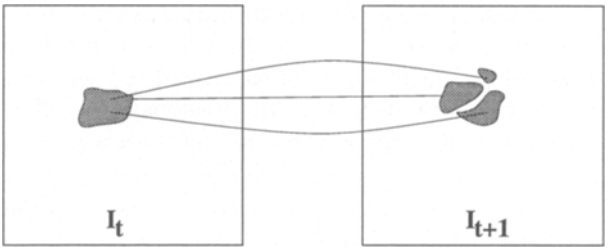

Fig. 2. An example for a subgraph of G_{inter}. A color region in image I_t is connected with three color regions in image I_{t+1}

for a match. The purpose of this graph is mainly the reduction of complexity. It can be viewed as a first check in a hierarchical decision tree. Regions not connected by an edge in G_{inter} are excluded from the subsequent analysis.

Phase 2 The goal of phase 2 is to find all single color regions that perfectly match with each other. Two regions r_1 from image I_t and r_2 from image I_{t+1} perfectly match, if:

1. r_1 and r_2 are connected by an edge in G_{inter}.
2. $EQ(r_1, r_2) < T \;\wedge\; \forall s \in I_t : EQ(r_1, r_2) \leq EQ(s, r_2) \;\wedge$
   $\forall t \in I_{t+1} : EQ(r_1, r_2) \leq EQ(r_1, t)$

where T is an application specific threshold depending on the chosen feature weights. Again, the purpose of this phase is mainly a reduction of complexity. Perfectly matched regions are deleted from G_{inter} and are no longer candidates for a $n : m$ match. The perfect matches are the simple $1 : 1$ matches. Homogeneous regions with distinct contrast to the background can often be perfectly matched. However, in natural scenes this is not always the case.

Phase 3 In this phase those color regions are further analysed that could not be perfectly matched in phase 2. These regions are the candidates for an $n : m$ match. The goal of $n : m$ matching is to determine those *subsets* from two sets of color regions that are best corresponding to each other. As the number of all possible subsets from a set with n elements is 2^n, it is impossible to exhaustively search for the optimal correspondence (usually there are about 100 regions in an image). However, it is possible to decrease the complexity by using the regions color and topological relations. Not all possible subsets of regions are evaluated, but only those subsets where the regions are topologically neighbored and similar in color. Only these regions may merge one with another in the segmentation of the successive image.

In this phase so called clusters of regions are determined independently for each image I. We define the graph G_{intra} as follows:

$G_{intra} = (V, E)$ is an undirected graph with V = {regions in image I}, $E \subseteq VxV$ with $(u, v) \in E :\Leftrightarrow$

1. u and v are similar in color.
2. u and v are topologically neighbored, e.g. $\min_{x \in u, x' \in v} \|x - x'\| <$ top_thresh.
3. u and v move in a similar way (initially always true).

Similarity in color should here be specified analogous to the color similarity measure used in the segmentation phase (but with a higher tolerance). The decision about the similarity of trajectories depends on an appropriate motion model. We use the predicate defined in 3.1. Regions with different trajectories can not belong to one object and are no candidates for a $n : m$ match.

As in phase 1 regions are connected that are topologically neighbored and similar in color. The difference is that in phase 1 a graph is built between the regions of two consecutive images and here a graph is built between the regions within the same image. G_{intra} is usually not connected (s. Fig. 3). A cluster of regions is defined as a maximally connected subgraph of G_{intra}. Candidates for an $n : m$ match are now only the subsets of a cluster. But it makes no sense to consider every subset of a cluster as a possible candidate for an $n : m$ match. Only connected subgraphs of a cluster are permissible candidates as only topologically neighbored regions with similar colors can merge in the segmentation of the next image of a sequence. In order to compute all connected subgraphs of a graph we use an efficient algorithm based on dynamic programming. At first all connected subgraphs with only one vertex are determined; this is exactly the set of all single vertices of the graph. All larger connected subgraphs are determined iteratively using the same algorithm: To get all connected subgraphs with n vertices consider all connected subgraphs with $n-1$ vertices and expand them by exactly one vertex v, if there exists a direct edge from any of the $n-1$ vertices of the subgraph to v.

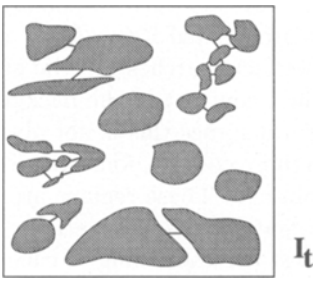

Fig. 3. An example for a graph G_{intra}. There are eight clusters contained in image I_t.

Phase 4 In the last phase all possible $n : m$ matches are computed using the graphs G_{inter} and G_{intra} of the preceding phases. In order to find the optimal $n : m$ matches all connected subgraphs of one cluster C_t in image I_t are compared with **all** connected subgraphs of a corresponding cluster C_{t+1} in image I_{t+1}. Clusters in consecutive images are corresponding if there exists at least one edge in G_{inter} connecting the clusters. This means that there is at least one region in cluster C_t connected with a region in cluster C_{t+1}. For each pair of

composed color regions, computed in this way, the similarity function value EQ is calculated.

Note that this algorithm always finds the optimal matches (in the sense of similarity value EQ) between two sets of color regions, as it compares all allowed subsets within each cluster with each other. Several $n : m$ matches between two corresponding clusters are possible as we delete successfully matched regions from the clusters and again check the remaining regions in the clusters for possible further matches. The key to improved efficiency is holding the clusters small. This is achieved by considering only neighbored regions similar in color. If a cluster consists of many segments we additionally apply the heuristic that usually not more than k (4 to 5 in our applications) regions are involved in one $n : m$ match. Thus, we only evaluate subgraphs with a maximum of k elements avoiding combinatorial explosion. Figure 8 shows some examples of $n : m$ matches from natural scenes found in this way.

Displacement estimation for feature-based region matching. After having established the corresponding color regions based on the similarity of their features the motion parameters for the matched sets of color regions are calculated. We only compute the planar motion which in practice is sufficient assuming a high sampling rate. However, more sophisticated motion models like the affine motion model can directly be applied to region matching approaches (s. [5]).

First of all the displacement vector $\Delta[d]$ between two matched regions r_1 and r_2 is approximated by the displacement of the objects centres of gravity. When regarding $n : m$ matches the centre of gravity refers to the composed color region. The displacement of the centres of gravity is sometimes inaccurate due to small variations in the segmentation of an object in consecutive images. We therefore additionally apply a boundary correlation method as in [9]. The boundary of a region is taken as a template and cross-correlated with the boundary of the corresponding matched region. This boundary correlation is very efficient to compute because it is only a bit-type correlation, only the boundary has to be correlated (not the entire region) and the search width can be restricted to a small size, because $\Delta[d]$ constitutes a good starting point. Using this boundary correlation the displacement vector can be determined rather accurate even if small parts of the object are missing.

2.2 Iconic Color Region Matching

The introduction of the general $n : m$ type matches significantly increased the quality of the matching results. However, there are still some color regions left that could not be matched on the feature level. They belong to one of the following classes (in decreasing order of frequency):

- Small regions, where small variations lead to significant changes of the shape parameters.
- Regions near the image borders, whose size and shape varies significantly because parts of them leave the image in the next frame.

- Occlusions of one segment by another.
- Significant variations of the regions due to changes in segmentation.

In order to match these regions we use an iconic matching technique on pixel level. Iconic matching operates similar to the well-known block-matching techniques. The idea of block-matching is to partition an image I_t at time t into equally sized blocks and to search for each of the blocks the corresponding block in the successive image I_{t+1} at time $t+1$. The similarity function often used in color images is the simple displaced frame difference

$$\begin{aligned} DFD(u,v) = \textstyle\sum_{x,y \in B} & |R(x,y,t) - R(x+u,y+v,t+1)| + \\ & |G(x,y,t) - G(x+u,y+v,t+1)| + \\ & |B(x,y,t) - B(x+u,y+v,t+1)| \end{aligned}$$

The advantage of the common block-matching method is its simplicity and the supply of dense displacement vectors. Its main drawbacks are a high computational complexity, the fixed choice of block size and shape, the missing of a clear minimum of the similarity function within homogeneous regions and the aperture problem due to a pure local view. In contrast to the block-matching technique we do not use fixed block sizes and shapes, but block sizes and shapes that are adjusted to the object boundaries. The color regions resulting from the segmentation phase simply form the blocks. This content-based displacement estimation explicitly accounts for the discontinuities of the displacement vector field occurring at the boundaries of contrary moving objects. Three of the above mentioned block-matching drawbacks no longer hold with iconic color region matching. The displacement vectors are very accurate due to the blocks adjusted to the object shape. The remaining drawback, the high computational complexity, is also strongly weakened when using the iconic region matching in a hybrid manner in the entire system. The iconic color region matching is only applied to those regions that could not be matched on the feature level. This is often only a small part of the image. Especially the large regions in natural scenes (sky, road, meadows) can well be matched on the feature level. Due to the distinct minima of the similarity function all the well-known efficient search techniques for block-matching methods can excellently be applied here. If for example a color region is large we use the hierarchical search technique. Of course, the search width can be restricted if the position of a color region in the successive image can well be predicted based on the trajectory computed so far. We use a simplified Kalman-filter to predict the position of a color regions centre of gravity.

In order to track color regions over several frames and thus yield extended trajectories, the displacement vectors found by iconic region matching are not sufficient. A displacement vector found in this way describes the displacement of a color region from an image I_t to the consecutive image I_{t+1}. This information has to be transmitted to the corresponding color region or regions in image I_{t+1}. The entire system is based on the fact that a color region from frame to frame transfers its motion history to the corresponding color regions in the consecutive image. Only in this way a maximum of the information contained in a sequence of images can be used for consecutive analysis phases like motion segmentation.

Therefore, let A' denote a color region in image I_t at time t and $d_{A'} = (x_{A'}, y_{A'})$ the displacement vector of A' found by iconic color region matching. Let A be the color region displaced by $d_{A'}$, $A = \{(x + x_{A'}, y + y_{A'}) \mid (x, y) \in A'\}$, projected into the successive image I_{t+1} (s. Fig. 4). Let $F \in CR(I_{t+1}) = \{\text{color regions in image } I_{t+1}\}$. In $(F \cap A)$ are all those pixels of I_{t+1} that as well belong to the color region F as are covered by A. Let B be the color region of I_{t+1} which covers the maximum part of A (s. Fig. 4), that is

$$area(B \cap A) > area(F \cap A)\ \forall F \in CR(I_{t+1})$$

where $area(F)$ denotes the number of pixels of a color region F.

We distinguish the following 4 cases:

1. $\frac{area(A \cap B)}{area(A)} \approx 1$ and $\frac{area(A \cap B)}{area(B)} \approx 1 \Longrightarrow$ match A with B (s. Fig. 5 (a)).
2. $\frac{area(A \cap B)}{area(A)} \approx 1$ and $\frac{area(A \cap B)}{area(B)} \ll 1 \Longrightarrow$ match A with B.
   If B is already matched $\longrightarrow$ multi match (s. Fig. 5 (b))
3. $\frac{area(A \cap B)}{area(A)} \ll 1$ and $\frac{area(A \cap B)}{area(B)} \approx 1 \Longrightarrow$ match A with B.
   If B is already matched, delete match.
   If A is already matched $\longrightarrow$ multi match (s. Fig. 5 (c)).
4. $\frac{area(A \cap B)}{area(A)} \ll 1$ and $\frac{area(A \cap B)}{area(B)} \ll 1 \Longrightarrow$ no match (s. Fig. 5 (d)).

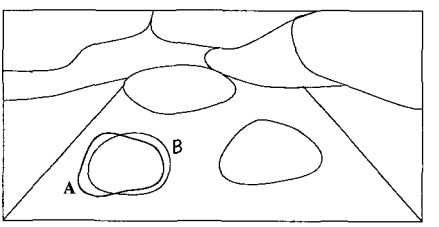

Fig. 4. The displaced color region A of image I_t projected into image I_{t+1}.

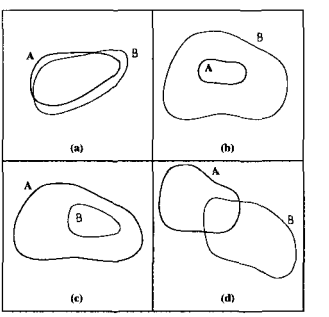

Fig. 5. Four possible cases, how A and B may overlap.

Hence, the results of the iconic color region matching are integrated in the entire system in the same way as the results of the feature-based color region matching. The matching results are represented in a graph data structure describing the correspondence of color regions from frame to frame. In addition to the classical region attributes like color, area, etc. each color region possesses a motion trajectory resulting from the matching phase. This motion trajectory clearly describes the path a color region has taken through time (s. Fig. 9).

3 Results

Our system has been applied to real world scenes, especially to a typical situation in vehicle guidance. The images were taken from a vehicle travelling along a highway with a 3-CCD color camera and recorded on a digital video recorder (Y:U:V=4:2:2, 720x560 pixels, 25 frames/sec). Due to the interlaced mode we only used half resolution images in our experiments. The color image sequences were kindly provided by the Daimler-Benz Research Institute in Ulm. The following statistics refer to eight sequences with altogether 1000 images. 85 color regions appeared on average in each image. About 89 % of them successfully were matched. The area covered by the successfully matched color regions is about 99 % of the whole image area. This implies that mainly very small regions were not matched. Visual judgement by human observers showed that nearly all of the matches are correct, which is explained by the usage of strict thresholds for allowable matches. About 35 % of the color regions are involved in $n : m$ matches. The expansion to $n : m$ matches has therefore led to a significant increase in performance. Described in percentage of the whole image area, only 11 % of the whole image area were matched using iconic region matching. This small amount is important for the efficiency of the entire system. The successful matching rate led to extended motion trajectories, which are very useful for subsequent tasks like motion segmentation. The trajectories are accurate due to the boundary and region correlation used to estimate the displacement.

The average processing time was 800 msec per image on a SUN ULTRA SPARC I (166 MHz). Thus, even today it is possible to operate at rates of 10 frames/sec using the latest processor technology on symmetric multiprocessor architectures. The share of color segmentation and feature extraction is about 55 % of the total processing time. The feature-based matching part requires 22 % and the iconic region matching part 23 % on average (search width 10 pixels). The time required for color segmentation is nearly constant, whereas the required time of the matching phase depends on the complexity of the scene, e.g. the number of regions, the size of the clusters and the number of regions matched using iconic region matching.

3.1 Motion segmentation

In order to demonstrate the effectiveness of our approach we applied the system to partition each image of a sequence into regions having different motion characteristics. We developed a simple algorithm to compute the motion segments based on the motion trajectories determined in the matching phase. All color regions that are topologically neighbored and possess similar motion trajectories are combined to one motion segment (s. Fig. 9). To decide about the similarity of motion trajectories, a proper motion model is necessary. Here we restrict ourselves to the case of simple 2D-motion, which already leads to promising results in our application. In order to tolerate local faults, the trajectories are split into overlapping, piecewise linear approximations (s. Fig. 6).

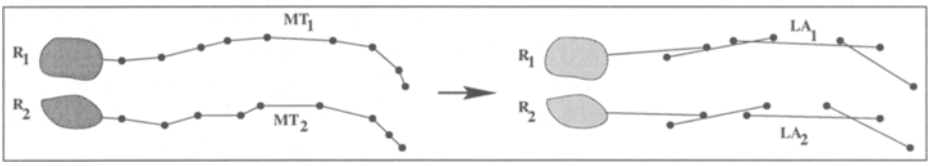

Fig. 6. Two regions R_1 and R_2 with their motion trajectories MT_1 and MT_2 and their corresponding linear approximations LA_1 and LA_2.

A trajectory is divided into sections of three contiguous displacement vectors. Two trajectories are now called similar, if their corresponding linear approximations are similar. We only use the last 12 images for the comparison, which means we use not more than the last five sections of the linear approximation. Let $LA_i = (v_{i1}, v_{i2}, v_{i3}, v_{i4}, v_{i5})$ $for\ i = 1, 2$ be the linear approximations of two motion trajectories t_1 and t_2. The similarity d of two trajectories t_1 and t_2 is defined by the predicate:

$$d(t_1, t_2) = d(v_{11}, v_{21}) \wedge d(v_{12}, v_{22}) \wedge d(v_{13}, v_{23}) \wedge d(v_{14}, v_{24}) \wedge d(v_{15}, v_{25}) \text{ with}$$
$$d(v_{1j}, v_{2j}) = (|vect\_length(v_{1j}) - vect\_length(v_{2j})| < LENGTH\_THRESH) \wedge$$
$$(vect\_angle(v_{1j}, v_{2j}) < 30^\circ)$$

where *vect_length* denotes the length of a vector and *vect_angle* the angle between two vectors. The threshold $LENGTH\_THRESH$ is dynamically defined as a linear function depending on the velocity of the slower color region. Figures 10 to 13 show some examples of the motion segmentation application. A fast approximation of the convex hull of a motion segment (a regular polygon with 24 edges) is plotted in green (s. Fig. 9). The trajectory of a motion segment is drawn in yellow. There are several more motion segments than the one plotted in the example. To better visualize the results, only one manually chosen motion segment is shown per image. Of course, not all motion segments in all sequences are as accurate as in the shown examples. But the quality is comparable to them. Problems occur sometimes with parts of the road that merge into the motion segment due to a *pseudo* motion of the part. This has to be solved using knowledge about the specific application. Note that no model of the scene or application has been used. The computed motion segments are accurate and stable even e.g. in Figure 12 where the car is entering the shadow of the van. The above simple similarity measure works well because the extended trajectories resulting from the robust matching allow for a clear discrimination of the independent motions in the scene.

4 Conclusions

In this paper we introduced an original approach to motion estimation in color image sequences based on color region matching. The main design criteria of

the system have been efficiency and robustness. The robustness was achieved by using a region-based approach, exploiting color information and extending the matching model to $n : m$ matches. The combination of the feature-based color region matching and the iconic color region matching contributes to the robustness and provides the basis for the efficiency of the approach. With feature-based region matching, extended to $n : m$ matches, nearly 90 % of the image area can be matched in an efficient and accurate way. The iconic region matching, which requires more elaborate processing, is thus only applied to a small fraction of the image area. It yields very accurate displacement vectors and most of the time finds the corresponding color regions. The combined approach delivers therefore dense displacement fields and extended trajectories. The experiments on natural color images proved the suitability of this approach to difficult tasks like motion segmentation in the field of vehicle guidance. Future work lies in the improvement of the motion segmentation based on the matched extended trajectories and the extension of the system to more sophisticated motion models.

References

1. M. Etoh and Y. Shirai. Segmentation and 2d motion estimation by region fragments. In *Proc. of IEEE Int. Conf. on Computer Vision*, pages 192–199, 1993.
2. C.-S. Fuh and P. Maragos. Region-based optical flow estimation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 130–135. IEEE, 1989.
3. M. Gelgon and P. Bouthemy. A region-level graph labeling approach to motion-based segmentation. Research report 3054, INRIA, http://www.inria.fr/RRRT/RR-3054.html, 1996.
4. B. Heisele and W. Ritter. Obstacle Detection based on Color Blob Flow. In *Proc. of the Intelligent Vehicles Symposium*, pp. 282–286. IEEE, 1995.
5. D.S. Kalivas and A.A. Sawchuk. A Region Matching Motion Estimation Algorithm. *Computer Vision, Graphics, and Image Processing*, 54(2):275–288, 1991.
6. D.P. Kottke and Y. Sun. Motion estimation via cluster matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(11):1128–1132, 1994.
7. H. Liu, T.-H. Hong, M. Herman, and R. Chellappa. Accuracy vs. Efficiency Tradeoffs in Optical Flow. In B. Buxton and R. Cipolla, editors, *Proc. 4th European Conference on Computer Vision*, pages 174–183. Springer Verlag, 1996.
8. A. Mitiche and P. Bouthemy. Computation and analysis of image motion: A synopsis of current problems and methods. *International Journal of Computer Vision*, 19:29–55, 1996.
9. H.-H. Nagel. Formation of an object concept by analysis of systematic time variations in the optically perceptible environment. *Computer Graphics and Image Processing*, 7:149–194, 1978.
10. K. Price. *Change Detection and Analysis in Multi-Spectral Images.* PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1976.
11. V. Rehrmann and L. Priese. Fast and Robust Segmentation of Natural Color Scenes. In *Proc. of the 3rd Asian Conf. on Computer Vision, Special Session on Advances in Color Vision*, Vol. I, pp. 598-606. Springer Verlag, 1998.
12. V. Rehrmann. Object Oriented Motion Estimation in Color Image Sequences. http://www.uni-koblenz.de/~lb/lb_publications.e.html, 1998.

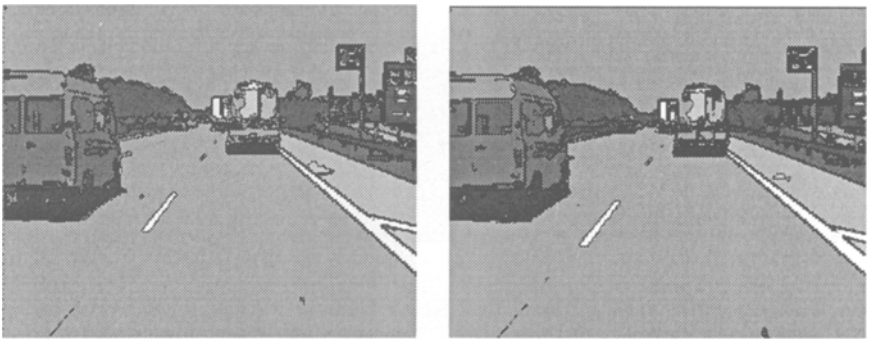

Fig. 7. An example for the color segmentation of two successive images. See [12] for the original color image.

Fig. 8. Three examples for $n : m$ matches. Left: $1 : 3$ match. Middle: $2 : 1$ match. Right: $2 : 3$ match. See [12] for the original color image.

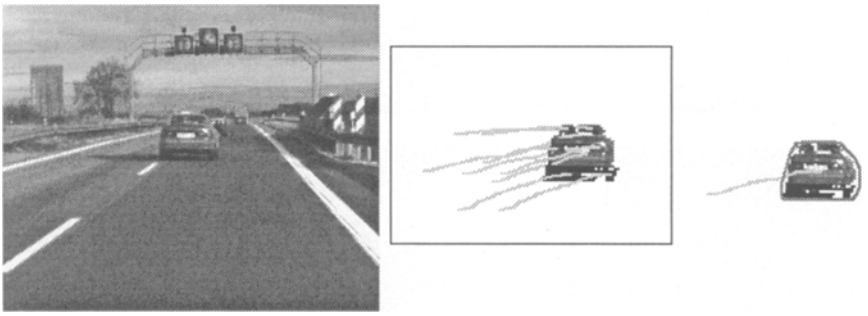

Fig. 9. Left: Original scene. Middle: 9 color regions with similar trajectories. Right: The resulting motion segment, drawn in green the approximation of the convex hull. See [12] for the original color image.

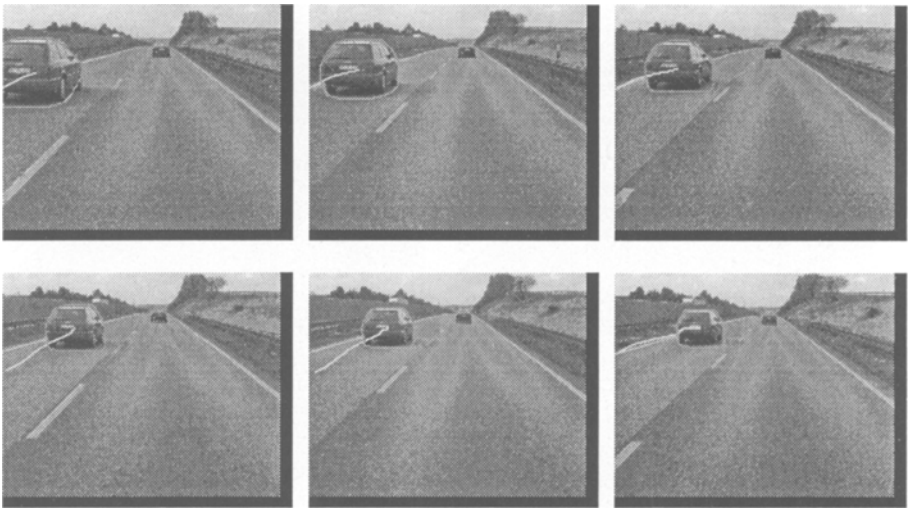

Fig. 10. Example of the color region based motion segmentation. See [12] for the original color image.

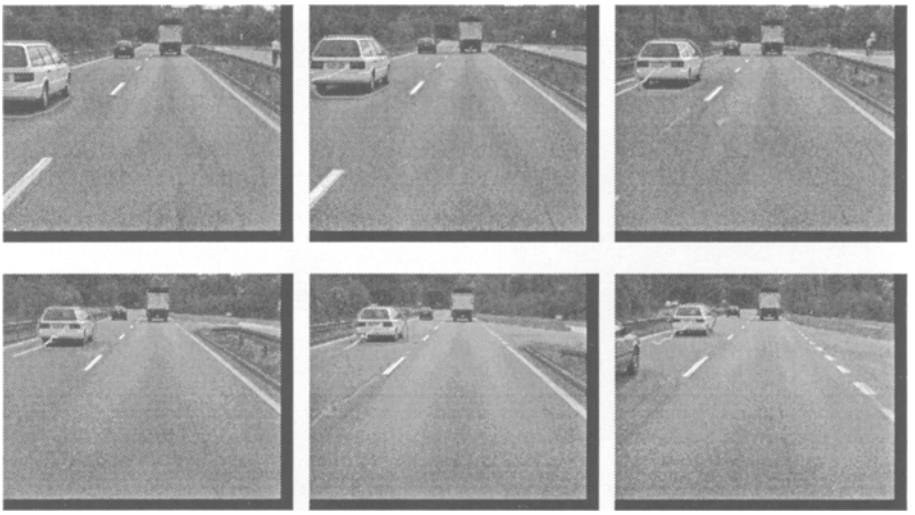

Fig. 11. Example of the color region based motion segmentation. See [12] for the original color image.

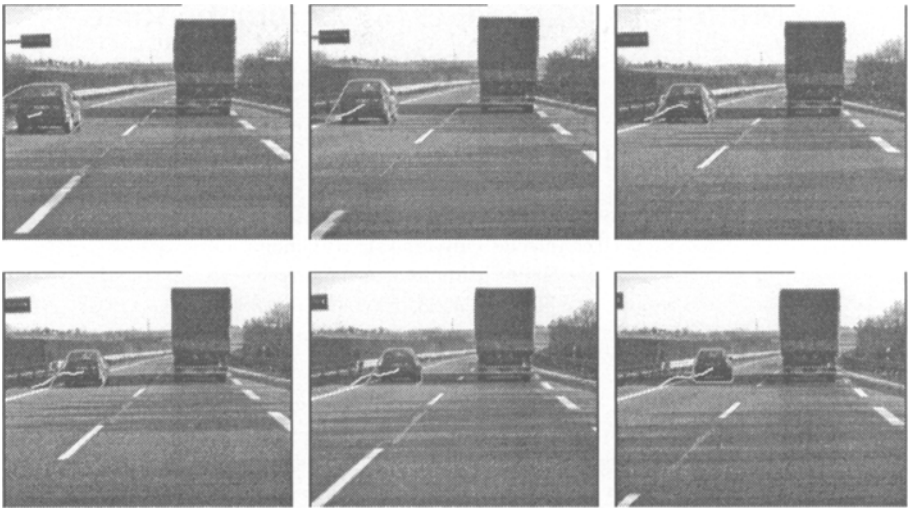

Fig. 12. Example of the color region based motion segmentation. See [12] for the original color image.

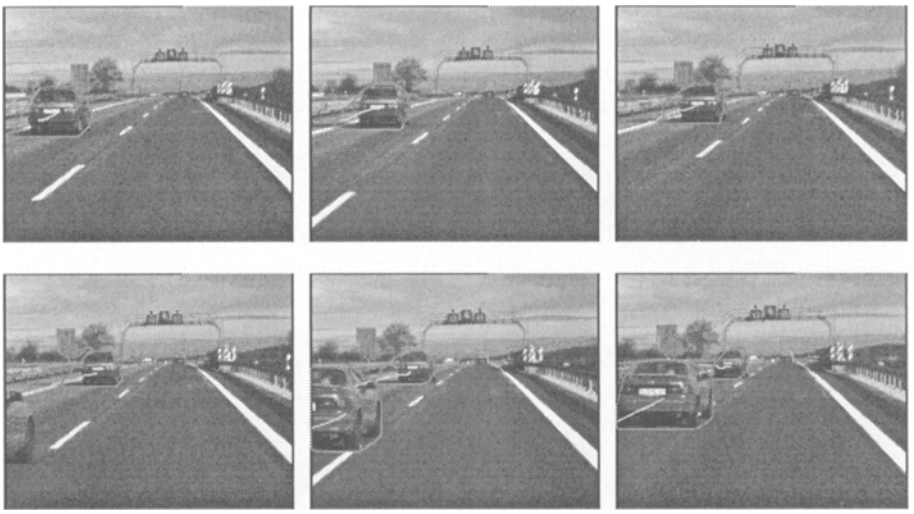

Fig. 13. Example of the color region based motion segmentation. See [12] for the original color image.