ROBUST REGION MERGING FOR MOTION BASED SEGMENTATION USING THE KOLMOGOROV-SMIRNOV TEST

M. BROEKHOVEN*, TH. M. HUPKENS*, E.A. HENDRIKS**, I. PATRAS**

*Royal Dutch Naval College, P.O. Box 10000, 1780 CA Den Helder, The Netherlands
Email: M.Broekhoven@kim.nl

**Delft University of Technology, Information and Communication Theory Group, Delft, The Netherlands

Keywords: Pattern recognition, region based segmentation, motion estimation, Kolmogorov-Smirnov test

ABSTRACT

This paper addresses the problem of segmenting image sequences into moving objects. The proposed method uses a bottom-up approach and uses an indirect method to estimate the motion parameters of the initial regions. These regions have been created using the watershed algorithm. The motion of the regions have been modelled by a four parameter affine model and have been estimated by using a robust method, which diminishes the influence of wrongly estimated displacement or optical flow vectors (outliers). The similarity measure between two adjacent regions is computed through a non-parametric statistical test (the Kolmogorov-Smirnov test) as proposed by Moscheni [1] with some important modifications. The algorithm iterates between a motion estimation step and a region merging step until some stopping criterion has been reached. The results indicate that the method is well suited to obtain a correct segmentation of real image sequences.

1 INTRODUCTION

People are very good at detecting objects. They can do this using several properties of those objects, such as colour, brightness, and texture. Another way of detecting objects is using motion. The ultimate goal of this study is to be able to automatically detect moving objects. This is very important for e.g. surveillance of the neighbourhood of naval ships or other military objects. Another application would be the tracking of objects (such as hostile aeroplanes and missiles), in situations where you cannot or do not want to use active radar.

The methods for segmentation can be grouped into two broad classes: the top-down approach and the bottom-up approach [1][2][3]. Top-down approaches iteratively estimate the parameters of the dominant motion in a sequence. Regions complying with the estimated dominant motion are assumed to belong to the same object and are not considered in the next iteration. A bottom-up approach starts with a set of regions and merges neighbouring regions with similar motion into larger regions. How we come to a definition of ‘similarity’ in this paper will be explained later in this section. In comparison with the top-down approach, the bottom-up approach has two major advantages. First, the bottom-up approach ensures that a complete decomposition of the scene is obtained [2]. Second, the extraction of a given object does not rely on previous extracted objects. For these reasons, the bottom-up approach has been used throughout this work.

The proposed method applies a robust method to estimate the motion models of the regions: First, displacement vectors are estimated. Then these vectors are used to estimate a motion model of each region in which the influence of wrongly estimated vectors (outliers) are diminished. The motion models and the displacement vectors are used to compute a similarity between adjacent regions. This similarity measure is an improvement on the method proposed by Moscheni [1]. Moscheni uses a pondering factor, which (as will be seen) can give wrong results. The way for not taking into account outliers when computing the similarity is also improved.

This paper is organised as follows: in Section 2 the outline of the method is explained. The segmentation method is described in Section 3. Results are discussed in Section 4 and conclusions are drawn in Section 5.

2 OUTLINE OF THE METHOD

We use a bottom-up approach, so we have to initially segment our frame. The initial segmentation is obtained by using the watershed algorithm[4][5]. This method first takes the gradient of the original frame. This gradient image is clipped at a certain level, which results in a number of markers. These markers are used as seeds to come to a segmentation.

In order to be able to estimate the motion behaviour of the initial regions, we have to choose a motion model. The choice of the motion model depends on the applicability and the desired flexibility. In this work, we use a four parameter affine model [6]. This model is quite flexible and has few degrees of motion freedom, which is an advantage when the motion information is not well known. This situation could occur in small regions when the motion is noisy and uncorrelated (for example sea waves).

The estimation of the four parameter affine motion model has been done using an indirect approach [2][7][8]. This method comprises two steps: the computation of a dense displacement vector field using a non-parametric method, followed by the modelling of the displacement vectors by the parameters of the motion model. The displacement vectors or optical flow vectors *(u, v)* are estimated, using hierarchical block matching (HBM) [6][9][10]. HBM is noise-resistant and is able to estimate relatively large displacements, due to the use of a low-resolution pyramid. In practise, three levels are sufficient for a good estimation. HBM starts with estimating the optical flow vectors *(u, v)* at the lowest resolution level. This estimation is refined when going to the next higher resolution level. To estimate the parameters of the affine model, it is possible to use a method, which minimises the mean square error of the estimated optical flow vectors [6][7]. However, this method is very sensitive to badly estimated optical flow vectors (outliers) [11]. In this work, an iterative re-weighted least square method is used that assigns increasingly low weights to such outliers.

The definition of the similarity measure of two adjacent regions is expressed as a hypothesis test. The assumption to be tested is that two regions are similar. The hypothesis is tested through a statistical test referred to as the Kolmogorov-Smirnov Test [12]. This non-parametric test examines whether the motions of the regions comply with the same distribution; it does not need any assumed model for the distributions. In order to perform the region merging a graph representation of the regions and their similarity is used [3]. The vertices and the edges of the graph represent the regions and the similarity respectively. The structures, present in the graph, are used to merge the regions. A strong and a weak rule are used which deal with erroneous motion information and badly defined regions [1][2][3]. The result of these rules is the creation of new regions for which new affine parameters and a new similarity measure have to be computed. This iteration process stops when a pre-defined lowest threshold has been reached. The outline of the method is shown in Figure 1.

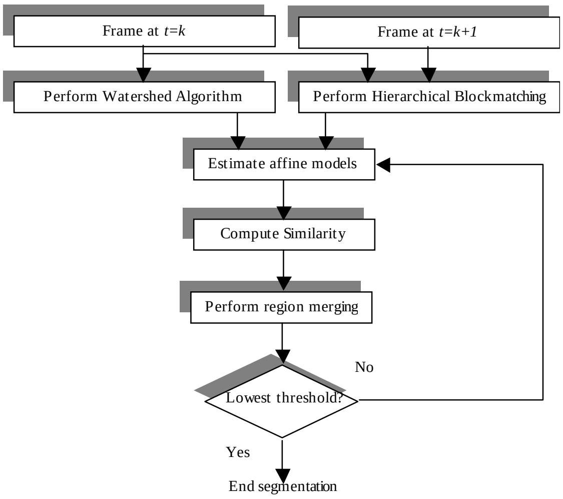

Figure 1: Outline of the method

3 MOTION BASED SEGMENTATION

3.1 MOTION OF A REGION

To estimate the motion of each watershed region a parametric model has to be chosen. This model has to be a good reflection of the real motion behaviour. We use a four parameter affine model [6]:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a_1 + a_2 x + a_3 y \\ a_4 - a_3 x + a_2 y \end{bmatrix}, \qquad \textbf{(1)}$$

where *(x, y)* represents the position within the frame. This model is capable of describing translation, rotation about an axis perpendicular to the image frame and scaling. It offers a good compromise between flexibility and applicability.

To estimate the motion parameters robustly, we use an iterative re-weighted least square scheme with an M-estimator [11]. This scheme gives a more stable solution than a least square [LS] solution, where outliers have a large influence. We convert the estimation problem into a weighted least-squares problem:

$$\arg\min_{\theta} \sum_i \rho(r_i, \sigma) = \arg\min_{\theta} \sum_i \frac{1}{2} w_i r_i^2 \ , \qquad \textbf{(2)}$$

where:

θ : the parameter vector (a_1, a_2, a_3, a_4)
ρ : the M-estimator,
σ : the scale factor, indicating the way we take into account outliers,
i : an index, representing the position within the region,
w_i : the weight at index i,
r_i : the residue, which is defined as:

$$r_i = \sqrt{(\hat{u}_i - \tilde{u}_i(\theta))^2 + (\hat{v}_i - \tilde{v}_i(\theta))^2} \ \textbf{,} \qquad \textbf{(3)}$$

where $\hat{u}_i, \hat{v}_i$ represent the estimated optical flow vector (by the hierarchical block matcher) and $\tilde{u}_i(\theta), \tilde{v}_i(\theta)$ the model generated optical flow vector. We use the Geman and McClure function as an M-estimator [13]:

$$\rho(r_i, \sigma) = \frac{r_i^{\,2}}{\sigma^2 + r_i^{\,2}} . \qquad \textbf{(4)}$$

The first estimation of θ is calculated by solving the given set of linear Equations in the LS sense (Equation (2) with w_i=1 $\forall$ i). Now, a set of weights can be calculated (Equation (2)). Using these weights, a new θ can be computed. We iterate between these steps until a stopping criterion has been reached. This procedure is visualised in Figure 2 (the inner loop). The setting of the stopping criterion is discussed in the next section.

At the beginning of the process the estimator has to be tolerant to outliers. so we must take all optical flow vectors into account. Therefore, the initial scale factor σ chosen for has to be large (equal to the largest r_i). With this initial scale factor, an estimation is done as described in the previous section. Then, as the accuracy of the dominant motion estimation improves, outliers are better identified and should be rejected. Therefore, σ is lowered by a certain factor. In this work, this factor is set to 0.9. Again, the affine parameters are computed. This is repeated until a certain stopping criterion has been reached. In order to define a stopping criterion, we need a measure of the overall change of the affine parameters. If simply the norm of the difference vector is used, the influence of the rotation *(a_3)* and zoom component *(a_2)* would be overestimated with respect to the translation component *(a_1* and *a_4).* Instead, we should use a norm in which all parameters get a certain weight depending on the influence of the individual parameter. This norm is defined as follows:

$$k^{(f)} = \sum_{j=1,2,3,4} m_j (\Delta \hat{a}_j^{(f)})^2 , \qquad \textbf{(5)}$$

where f represents the iteration number, $\Delta \hat{a}_j^{(f)}$ is the difference between one of the estimated affine parameters after iteration f and iteration f - 1. m_j indicates the way we take into account the affine parameters. The coefficients m_j are determined as follows: If $\vec{V}_{\Delta \hat{a}_j^{(f)}}$ indicates the optical flow vectors supplied only by $\Delta \hat{a}_j^{(f)}$, setting the other parameters to zero, we can write:

$$\frac{1}{area(W)} \sum_i \left\| \vec{V}_{\hat{a}_j^{(f)}} \right\| = m_j \left| \Delta \hat{a}_j^{(f)} \right| . \qquad \textbf{(6)}$$

The size of the region W under test is indicated by area(W). Equation (6) leads to:

$$m_1 = m_4 = 1 . \qquad \textbf{(7)}$$

$$m_2 = m_3 = \frac{1}{area(W)} \sum_i \sqrt{x_i^2 + y_i^2} . \qquad \textbf{(8)}$$

The stopping criterion has been chosen as $k^f < 0.5$. The total estimation process is shown in Figure 2.

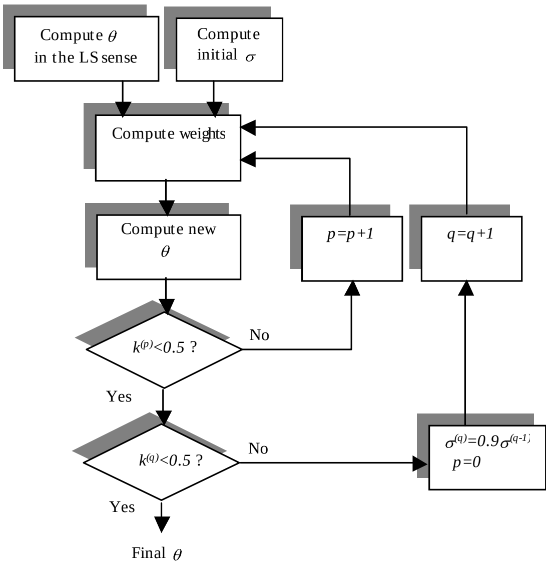

Figure 2: Overview of the estimation of θ

3.2 THE SIMILARITY MEASURE

To effectively perform the region merging, a representation of the regions with their similarities has to be defined. A graph G is used to perform this task [2][14]. Each vertex in the graph represents one of the regions in the set $R = \{R_1, \ldots, R_Z\}$, where Z is the number of initial regions. Every edge represents the similarity between the regions connected by the edge.

To define a similarity measure for two adjacent regions A and B, we use the Kolmogorov-Smirnov test. This test compares two distributions and estimates the degree to which they are similar, without knowing any a priori information about the distributions [2][12]. We compute two residual distributions. Residual distribution $h_1(r_i)$ is obtained by compensating each optical flow vector within region A with θ_A. If we compensate the optical flow vectors within region A with θ_B, residual distribution $h_2(r_i)$ will be created. The residual distribution represents the discrepancy between the affine model and the optical flow of the region. If these distributions show a lot of dissimilarity the regions A and B should not be merged. However, if they do show enough similarity, we should merge them.

The Kolmogorov-Smirnov Test defines the realisation q_{ks}, of the test statistic Q_{KS} as follows [12]:

$$q_{ks} = \max_{r_i} \left| F_1(r_i) - F_2(r_i) \right| , \qquad \textbf{(9)}$$

where:

$$F_1(r_i) = \sum_{c=0}^{r_i} h_1(c)\ , \tag{10}$$

$$F_2(r_i) = \sum_{c=0}^{r_i} h_2(c) \tag{11}$$

are the cumulative residual distributions. In [1] Moscheni uses a pondering factor. This factor indicates the difference in θ_A and θ_B and is used to correct q_{ks}. This could lead to unwanted situations. If we compare a translation with a rotation along a point far away, both optical flow fields would be approximately the same. The affine parameters will however differ considerably.

In order to cope with outliers, Moscheni uses an M-estimator with a fixed σ, but this is very inaccurate. Instead, we use the weights that were computed when estimating the affine model. We want $h_1(r_i)$ to have most residues close to zero. The influence of the outliers can be neglected by only taking into account residues with a weight larger than a certain value. To set this value, the distribution of the weights can be used. In Figure 3, a typical distribution of the weights can be seen. Most residues have weights close to zero or close to one, so it is sufficient to use a threshold of 0.5 .

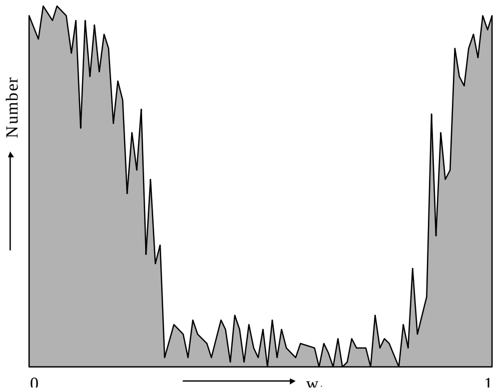

Figure 3: A typical distribution of the weights

The similarity T_{AB} is defined as a function of q_{ks} [13]:

$$T_{AB} = 2\sum_{j=1}^{\infty} -1^{j-1} e^{-2j^2\delta^2}\ , \tag{12}$$

where:

$$\delta = \left(\sqrt{N_e} + 0.12 + \frac{0.11}{\sqrt{N_e}} \right) q_{ks}\ , \tag{13}$$

and

$$N_e = \frac{area(A) \cdot area(B)}{area(A) + area(B)}\ . \tag{14}$$

A strong and weak rule as described by Moscheni [2] are used. The strong rule searches for cycles in the graph and merges regions with well defined motion. This rule is carried out first. Most of the regions obtained after applying the strong rule already are good approximations of the objects present in the scene, but there may be some regions with erroneously estimated motion information. The latter regions are processed by the weak rule. The weak rule is derived from the Greedy Merging Algorithm [2]. It merges adjacent regions with the highest similarity. The graph is initially thresholded at the maximum value that would still allow the strong and weak rule to carry out a merging. This threshold (specified for both the strong (T_{sr}) and the weak rule (T_{wr})) decreases with a predefined step (in this work set to 2) until a predefined lowest threshold (T_{lsr}, T_{lwr}) is reached. This lowest threshold has been set to T_{lsr} = T_{lwr} = 90% throughout the experiments. After each step the graph is updated. This means that for the newly created clusters a new similarity with its neighbours is computed. The strategy is visualised in Figure 4.

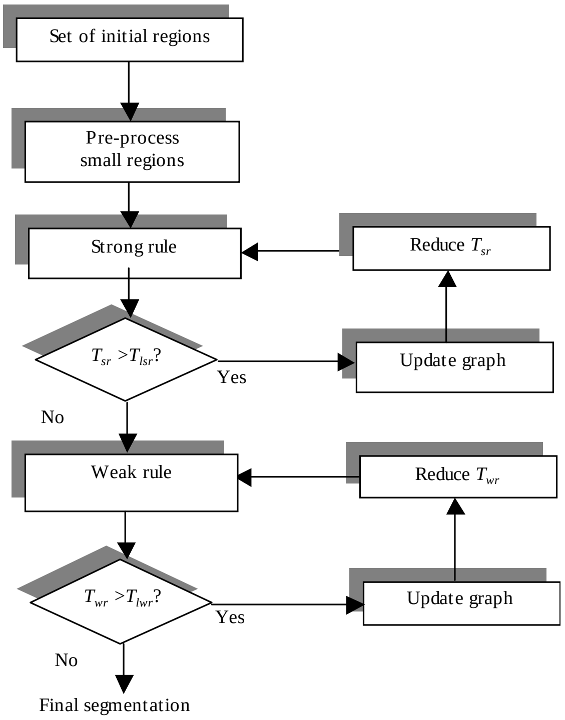

Figure 4: The region merging strategy

Because outliers are of large influence in small regions, we created an option to eliminate such regions. Regions smaller than a certain value (the smallest region parameter λ) are merged with the adjacent neighbour with the highest spatial similarity S_{AB}. In [4] a similarity-distance D_{AB} between adjacent regions A and B is defined as:

$$D_{AB} = (\mu_A - \mu_B)^2 N_e, \tag{15}$$

where $\mu_{A,B}$ represents the mean value of the grey values in regions A and B. From (15), it is clear we have to look for the smallest D_{AB}. The setting of λ is quite arbitrary. In the experiments, we will vary this parameter to see its effect. This process is repeated until no regions exist with an area smaller than λ.

4 RESULTS AND DISCUSSION

A typical result obtained for a real infrared sequence is shown in Figure 5(a). The ship (the HNLMS Tydeman, a Dutch navy ship) travels from right to left. The boundaries of the watershed regions have been projected onto the original (Figure 5(b)). The frame is segmented into four parts (Figure 5(c)). The region, bounded by the mast, the horizon and the upper deck of the ship is merged to the ship. This is due to the size of the block, which is used in the HBM (this size is set to 11 x 11 throughout the experiments) and the lack of structure in it. Decreasing the size of the block is no option, because experiments taught us that this gave rise to a lot of mismatches.

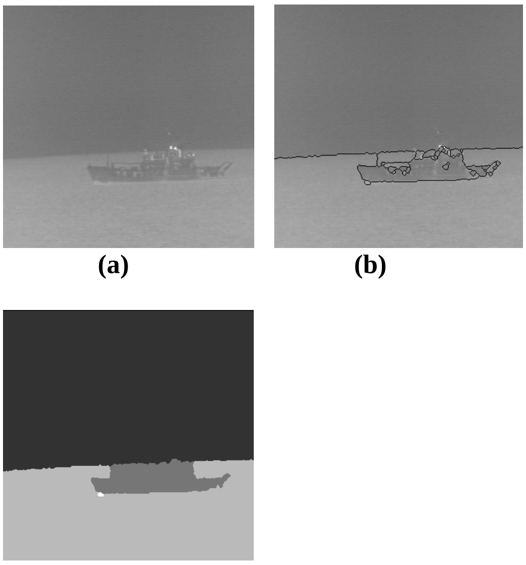

Figure 5: The "Tyd32" sequence. The original (a), the watersheds (b) and the final segmentation (c)

To illustrate the effect of very small regions, we use the "Tyd51" sequence (see Figure 6 (b)). As can be seen in Figure 6(b), a lot of small regions are present. The resulting segmentation is rather bad (see Figure 6(c) The result after the pre-process step (where λ is set to 100 pixels) is shown in Figure 6(d). The overall result is good, however the front mast has vanished. This is not so much because of the merging algorithm but rather because the watershed algorithm has trouble finding the correct initial regions of the front mast. This is because very little texture is present in that area.

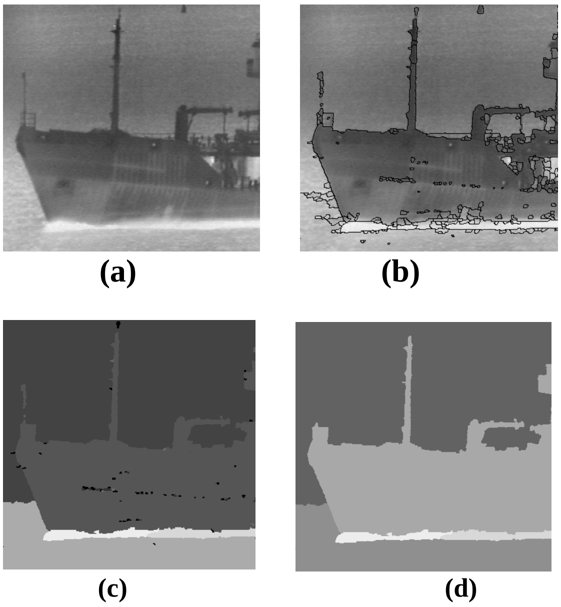

Figure 6: The "Tyd51" sequence. The original (a), the watersheds (b) the resulting segmentation without preprocessing (c) and with pre-processing (d)

5. Conclusions

We have presented a method for motion based segmentation, based on a bottom-up approach and a robust method to estimate the motion-parameters. The proposed method works well on real infrared images. However, the watershed algorithm has some drawbacks. It only produces a good set of initial regions, when enough texture and intensity differences are present in a frame. If this is not the case it is very hard to get a good initial segmentation. This greatly depends on the setting of the clipping level, which is not automatically. Therefore research should be done to set this parameter properly. Another problem of the watershed algorithm is the production of very small initial regions. This is not desirable because outliers have a large negative influence in such regions. In the proposed method, this problem has been solved by using a pre-process step. From many experiments, it was found that we could use as a rule of thumb a smallest region parameter λ of 100 pixels.

It has been experienced that the estimation of the affine models is very accurate when the initial regions are large enough. However, the consequence of these accurate affine models together with the elimination of outliers when computing the similarity is the transformation of the region merging strategy into a binair problem: regions belong to each other or they don't. In most experiments done, therefore, it was sufficient to carry out only the strong rule, the weak rule did not contribute to a better result.

REFERENCES

1. F. Moscheni, S. Bhattacharjee and M. Kunt, Spatiotemporal segmentation based on region merging, *IEEE transactions on pattern analysis*

and machine intelligence, vol. 20(9), 1998, 897-915.

2. F. Moscheni, *Spatiotemporal segmentation and object tracking an application to second generation video coding*, Ph.D thesis, 1997.
3. F. Moscheni, S. Bhattacharjee, Robust region merging for spatio-temporal segmentation, *International Conference on Image Processing*, vol. 1, 1996, 346-350, 1996.
4. P.J. Mulroy, Video content extraction: Review of current automatic segmentation algorithms, *Workshop on Image Analysis for Multimedia Interactive Services*, 1997, 45-50.
5. K.R. Castleman, *Digital Image Processing*, Prentice Hall, 1996.
6. S. Wu, J.Kittler, A differential method for simultaneous estimation of rotation, change of scale and translation, *Signal Processing Image Communication*, vol. 2, 1990, 69-80.
7. A. Tekalp, *Digital video processing*, Prentice Hall, 1995.
8. C. Stiller, J. Konrad, Estimating motion in image sequences: A tutorial on modelling and computation of 2D motion, *IEEE Signal Processing Magazine*, vol. 16, 1999, 70-91.
9. B. Furht, J. Greenberg, R. Westwater, *Motion estimation algorithms for video compression*, Kluwer Academic Publisher, 1997.
10. M.I. Sezan, R.L. Lagendijk, *Motion Analysis and Image sequence processing*, Kluwer Academic Publishers, 1993.
11. J. Odobez, P. Bouthemy, Robust multiresolution estimation of parametric motion models, *Journal of visual communication and image representation*, vol. 6(4), IRISA/INRIA Rennes, 1995, 348-365.
12. W. Press, S. Teukolsky, W. Vetterling, B.P. Flannery, *Numerical recipes in C*, Cambridge, Mass. Cambride University Press, 1992.
13. P. Meer, D. Mintz, A. Rosenfeld, Robust regression methods for computer vision: A review, *Int. J. of computer vision*, vol. 6(1), 1991, 59-70.
14. R. Johnsonbaugh, *Discrete Mathematics*, Macmillan Publishing Company, 1993.