Shape Estimation of Articulated 3D Objects for Object–Based Coding Considering Mutual Occlusion

Geovanni Martínez
Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung
Universität Hannover, Appelstraße 9A, D–30167 Hannover , F.R. Germany
Phone: ++49–511–762–5311, email: martinez@tnt.uni–hannover.de

ABSTRACT: *For object based coding of image sequences, the shapes of articulated 3D objects are estimated applying three steps: shape–initialization, object–articulation and shape–adaptation. In this contribution, object–articulation is extended to consider small objects and mutual occlusion. For object–articulation, the rigidity constraint imposed by a rigid component is exploited. According to this constraint, the motion of any of the component's particles can be described by using the same motion parameters. Object–articulation of small objects is now possible because more reliable information is used for evaluation of the rigidity constraint. For evaluation, besides local 3D motion, also local 2D motion and a segmentation of a displacement vector field into regions of homogeneous magnitude of the displacement vector are taken into account. In case of occlusion, the shape of the components in foreground and the shape of the occluded components are modelled separately. Experimental results using the test sequence "Geovanni" show a reduction of the total size of the so called model failure areas from 10.5% to 7.0% without loss of subjective image quality. Only for these model failure areas, color update must be performed.*

1 INTRODUCTION

For coding of moving images at low data rates, object–based analysis–synthesis coding (*OBASC*)[11] subdivides each image of a sequence into moving objects and describes each object by three sets of parameters defining its motion, shape and surface color. The nature of the parameters depends on the applied source model and they have to be estimated automatically. Assuming diffuse illumination, image regions where motion compensation fails because motion and shape parameters could not be estimated successfully are called **M**odel **F**ailure areas (MF–areas). Color parameters are transmitted for MF–areas only. Since the transmission of color parameters is expensive in terms of data rate, the total size of MF–areas should be kept as small as possible.

Ostermann[13] proposes an OBASC scheme based on a source model of "moving rigid 3D objects". According to this source model, model objects are rigid with 3D shape and moving in the 3D space. The motion is defined by a set of 6 parameters which describe the translation (T_x, T_y, T_z) and rotation angles (R_x, R_y, R_z) of the object in the 3D space. The 3D shape is represented by a mesh of triangles which is put up by vertices denoted as control points. Color parameters denote the reflectance of the object surface and are taken by projection of a real image onto the surface of the mesh of triangles.

Since real objects may be articulated, i.e. may consist of flexibly connected rigid 3D object–components, Martínez[7] proposes an OBASC scheme based on the source model of "moving articulated 3D objects". In computer graphics, object–components are also called links. Each object–component has its own set of motion, shape and color parameters. Since the shape of each object–component is defined by its control points, object–components are connected by those triangles with control points belonging to different object–components. Due to these connecting triangles, object–components are flexibly connected. Connecting triangles may enforce spatial constraints on the location of object–components. Spatial constraints between object–components are modelled using joints[1][8].

For shape estimation of articulated objects the following steps are applied in [7]: shape–initialization, object–articulation and shape–adaptation. **Shape–initialization** carries out a change detection between the first two images of a sequence to distinguish temporally changed and unchanged regions. Then, for each changed region one rigid 3D shape represented by a mesh of triangles is generated assuming ellipsoidal shape[13]. Since shape–initialization describes each real articulated object by only one rigid 3D model object, motion compensation will fail if object–components move differently. For **object–articulation**, the rigidity constraint imposed by a rigid object–component is exploited. According to this constraint, the 3D motion of each triangle, which covers the visible surface of one unknown object–component, can be described by using the same 3D motion parameters. Therefore, neighboring triangles which exhibit similar 3D motion parameters during the image sequence are clustered into object–components. For the estimation of the 3D motion parameters of a single triangle *i*, i.e. $\boldsymbol{A}_{3D,i} = (T_{x,i}, T_{y,i}, T_{z,i}, R_{x,i}, R_{y,i}, R_{z,i})^T$, a robust algorithm is applied which evaluates that triangle and its neighborhood. **Shape–adaptation** is used to update the shape of each object–component of an articulated object during the image sequence.

Until now the shape estimation algorithm for articulated objects proposed in [7] has been successfully applied to typical ”head and shoulders” video sequences, where object–components are large and no occlusions occur. In case of small object–components consisting of only a few picture elements object–articulation can fail, because the probability of convergency to correct motion parameters of the applied estimation algorithm becomes low.

In this contribution, the algorithm for object–articulation proposed in [7] is extended to consider small object–components and mutual occlusion. To consider small object–components, besides the estimation of the 3D motion parameters for each visible triangle i, also the 2D motion parameters $\boldsymbol{A}_{2D,i} = (T_{x,i}, T_{y,i}, R_{z,i})^T$ and the magnitude of the displacement vector $\boldsymbol{A}_{D,i} = (D_{x,i}, D_{y,i})^T$, i.e. $|\boldsymbol{A}_{D,i}| = \sqrt{D_{x,i}^2 + D_{y,i}^2}$, are estimated. For clustering, additionally to the 3D motion parameters of each triangle both the 2D motion parameters as well as the magnitude of the displacement vector of each triangle are also taken into account. For 2D motion estimation, the same robust algorithm used for 3D motion estimation is applied, but only the parameters $T_{x,i}, T_{y,i}, R_{z,i}$ are estimated and no neighborhood is taken into account. The magnitude of the displacement vector of each triangle is taken from a 2D segmentation of a pel–wise displacement vector field (*DVF*) into regions of uniform magnitude of the displacement vector. For 2D segmentation, a maximum likelihood multi–thresholding technique based on population mixture models[6][9][14] is applied. In order to detect mutual occlusion of object–components, a correlation of the silhouettes of a found object–component and of the complete object, as defined in Chapter 3, is calculated. If they are uncorrelated, the found object–component occludes the object. In these cases, the shape of the object–components in the foreground and the shape of the occluded object–components are represented by separated wireframes.

The performance of the developed algorithm is evaluated in the image analysis of an *OBASC*[13] scheme. The performance is measured by the reduction of the total size of MF–areas.

The paper is organized as follows. In Chapter 2, the algorithm for 2D segmentation of a displacement vector field into regions of homogeneous magnitude of the displacement vector is presented. In Chapter 3, the extended algorithm for object–articulation is described. In Chapter 4, experimental results are given. Final discussions are presented in Chapter 5.

2 SEGMENTATION OF A DISPLACEMENT VECTOR FIELD INTO REGIONS OF UNIFORM MAGNITUDE OF THE DISPLACEMENT VECTOR

Multi–thresholding is a fundamental tool for segmentation of grey level images when objects and background pixels can be distinguished by their grey level values. Among the global multi–thresholding methods which determine thresholds from the grey level histogram of an image, the maximum likelihood multi–thresholding based on population mixture models[6] is found to be best[4]. Here, this multi–thresholding technique is used to segment a displacement vector field into regions of uniform magnitude of the displacement vector.

Let $s_k(x, y)$ be the luminance value at point x, y of a moving object–component in frame s_k at time instant k. It is assumed that the moving object–component does not change its luminance from frame to frame. Then, the motion of the object–component generates a displacement vector $\boldsymbol{d}$ with components dx and dy. Using polar form, this displacement vector can be written as $\boldsymbol{d} = |\boldsymbol{d}| \measuredangle \boldsymbol{d}$, where the quantities $|\boldsymbol{d}| = \sqrt{dx^2 + dy^2}$ and $\measuredangle \boldsymbol{d}$ are called the magnitude and phase of the displacement vector, respectively. In this paper, a field in which each pel represents the displacement vector $\boldsymbol{d}$ of the corresponding pel in the image s_k is called a pel–wise displacement vector field. In addition, a field in which each pel represents the magnitude $|\boldsymbol{d}|$ of the corresponding displacement vector in the *DVF* is called a pel–wise displacement magnitude field (*DMF*). The displacement vector field is estimated by hierarchical block matching[2] with 1/8 pel measurement accuracy using the current image s_{k+1} and the previous image s_k.

Let us now consider the displacement magnitude field only inside the silhouette O_s of a rigid model object generated by shape–initialization (DMF_s). The magnitude value of each pel of the DMF_s is uniform quantized in the interval $[0, N\text{–}1]$ with $N = 2^8$ and represented by $|\boldsymbol{d}^s|$. The distribution of the discrete magnitude values in the DMF_s can be represented in the form of a histogram $h(|\boldsymbol{d}^s|)$, $|\boldsymbol{d}^s| = 0, .., N - 1$, which gives the frequency of occurrence of each discrete value $|\boldsymbol{d}^s|$ in the DMF_s. It is convenient to normalize this histogram in the form of $p(|\boldsymbol{d}^s|) = h(|\boldsymbol{d}^s|)/N^s_{pel}$, where $N^s_{pel} = \sum_{|\boldsymbol{d}^s|=0}^{N-1} h(|\boldsymbol{d}^s|)$ is the total number of pels in the DMF_s and $p(|\boldsymbol{d}^s|)$ the probability density function of $|\boldsymbol{d}^s|$ in the case of $N^s_{pel} \rightarrow \infty$.

Now, we suppose that we are classifying the N^s_{pel} pels of the DMF_s into N_C classes C_1, C_2, C_3, ... , C_{N_C} by thresholds at the values th_1, th_2, th_3, ... , th_{N_C-1}. Here C_1 denotes pels with discrete magnitude values in the interval [0, .., th_1], C_2 denotes pels with discrete magnitude values in the interval [th_1+1, ..., th_2] and C_{N_C} denotes pels with discrete magnitude values in the interval [th_{N_C-1}+1, ..., N–1]. Assuming a Gaussian mixture density of size N_C for the probability density function of a discrete magnitude value $|\boldsymbol{d}^s|$, $p(|\boldsymbol{d}^s|)$ can be written as follow

$$p(|\boldsymbol{d}^s|) = \sum_{m=1}^{N_C} \frac{c_m}{\sqrt{2\pi}\,\sigma_m} \exp\left(-\frac{(|\boldsymbol{d}^s| - \mu_m)^2}{2\sigma_m^2}\right) \tag{1}$$

$$p(|\boldsymbol{d}^s|) = \sum_{m=1}^{N_C} c_m \cdot l_{m,|\boldsymbol{d}^s|} \tag{2}$$

where $\mu = (\mu_1, \mu_2, ..., \mu_{N_C})$, $\sigma = (\sigma_1, \sigma_2, ..., \sigma_{N_C})$ and $\boldsymbol{c} = (c_1, c_2, ..., c_{N_C})$ are the means, standard deviations, and weights of Gaussian components densities with $c_m \geq 0$, $m = 1, ..., N_C$, $\sum_{m=1}^{N_C} c_m = 1$ respectively, and the $l_{m,|\boldsymbol{d}^s|}$'s are the likelihood of $|\boldsymbol{d}^s|$ belonging to the mth Gaussian component density in the mixture. When $N_C = 1$, the Gaussian mixture density is reduced to a simple Gaussian density.

Assuming that the discrete magnitude values $|\boldsymbol{d}^s_n|$, $n = 1, ..., N^s_{pel}$, in the DMF_s are statistically independent, the joint log–likelihood of $|\boldsymbol{d}^s|$, μ, σ and $\boldsymbol{c}$ is given by [6][9][14]:

$$l = \sum_{n=1}^{N^s_{pel}} \sum_{m=1}^{N_C} \left[z_{m,|\boldsymbol{d}^s_n|} \cdot \left(\log c_m + \log l_{m,|\boldsymbol{d}^s_n|} \right) \right] \tag{3}$$

where

$$z_{m,|\boldsymbol{d}^s_n|} = \begin{cases} 1, & |\boldsymbol{d}^s_n| \in C_m \\ 0, & |\boldsymbol{d}^s_n| \notin C_m \end{cases} \tag{4}$$

From this log–likelihood, we can obtain the optimal parameter sets μ, σ and $\boldsymbol{c}$ as described in [6]:

$$c_m = \sum_{|\boldsymbol{d}^s| = th_{m-1}+1}^{th_m} \frac{h(|\boldsymbol{d}^s|)}{N^s_{pix}} \tag{5}$$

$$\mu_m = \frac{1}{c_m} \cdot \sum_{|\boldsymbol{d}^s| = th_{m-1}+1}^{th_m} |\boldsymbol{d}^s| \cdot \frac{h(|\boldsymbol{d}^s|)}{N^s_{pel}} \tag{6}$$

$$\sigma_m^2 = \frac{1}{c_m} \sum_{|\boldsymbol{d}^s| = th_{m-1}+1}^{th_m} [|\boldsymbol{d}^s| - \mu_m]^2 \frac{h(|\boldsymbol{d}^s|)}{N^s_{pel}} \tag{7}$$

where $th_0 = -1$ and $th_{N_C} = N_C$. Thus the maximum log–likelihood is given by

$$\begin{aligned} \hat{l} = N^s_{pel} \cdot \sum_{m=1}^{N_C} [c_m \cdot \log c_m] - \frac{N^s_{pel}}{2} \cdot \log(2\pi) - \\ \frac{N^s_{pel}}{2} \cdot \sum_{m=1}^{N_C} [c_m \cdot \log(\sigma_m^2)] - \frac{N^s_{pel}}{2} \end{aligned} \tag{8}$$

Maximizing this log–likelihood with respect to the thresholds th_1, th_2, th_3, ... , th_{N_C-1} we obtain the optimized thresholds $\hat{th}_1$, $\hat{th}_2$, $\hat{th}_3$, ..., $\hat{th}_{N_C-1}$. For maximization, the Downhill Simplex Method described in [12] is used. At the beginning, the number of classes N_C and consequently the initial position of the thresholds are automatically calculated using the heuristic fast multi–thresholding selection procedure described in [10]. It assumes each desired class in the DMF_s can be represented by an approximately hill–shaped distribution in the histogram. The shape of the original histogram is smoothed by recursively convoluting the histogram with a Gaussian kernel so that the desired peaks and valleys at varying levels of detail can be obtained. The detected valleys in the smoothed histogram indicate the number of classes N_C and the initial position of the thresholds.

Using the optimal thresholds $\hat{th}_1$, $\hat{th}_2$, $\hat{th}_3$, ..., $\hat{th}_{N_C-1}$, the discrete magnitude values $|\boldsymbol{d}^s_n|$ in the DMF_s are classified into N_C classes. After classification, we obtain N_{reg} regions of uniform magnitude of the displacement vector, where $N_{reg} \geq N_C$, because different regions may move with the same magnitude of the displacement vector. In the ideal case, i.e. pure translational movement of the object–components, these regions will represent the complete object–components. However, due to the more complicated real object–components' motion, i.e. translation and rotation in the 3D space, one object–component can be split into several neighboring regions of homogeneous magnitude of the displacement vector. For example, the right arm of the articulated object ”Geovanni” (see Fig. 1.a and 1.b). In this case, due to the arm's rotation around an axis perpendicular to the image plane at the elbow's position, the magnitude of the displacement vector of a pel on the arm depends on its proximity to the elbow. The nearer the pixel, the smaller the magnitude of its displacement vector. Therefore, after classification, the right arm was split into several neighboring regions whose displacement magnitudes decrease depending on its position with respect to the elbow.

Thus, in order to obtain the complete object–component, the N_{reg} regions found using the optimal thresholds are then merged into larger regions[3]. For merging, the regions' size and the rigidity constraint imposed by a rigid object–component are considered. First, each small region is merged to its largest neighboring region. Secondly, neighboring regions which exhibit similar 2D motion parameters are also merged. After merging, the boundaries of each resulted region G_j, j: $1...N_{res}$ with $N_{res} \leq N_{reg}$, is improved by local analysis of image luminance and contours near its boundaries. Finally, a displacement magnitude $|\boldsymbol{D}_j|$ is assigned to each region G_j. $|\boldsymbol{D}_j|$ represents the average of the magnitude of the displacement vectors of the DMF_s inside of G_j:

$$|\boldsymbol{D}_j| = \frac{1}{N_j} \cdot \sum_{\substack{n=1 \\ |\boldsymbol{d}_n^s| \in G_j}}^{N_{pel}^s} |\boldsymbol{d}_n^s| \qquad (9)$$

where N_j is the number of pels of the region G_j.

3 EXTENDED ALGORITHM FOR OBJECT–ARTICULATION

The extended algorithm for object–articulation applies 6 steps for each frame of the image sequence after shape–initialization: by the **first step**, 3D motion estimation and compensation for the whole 3D model object is carried out. By the **second step**, neighboring triangles which exhibit similar 3D motion parameters are clustered into patches. These patches are called patches type 1. By the **third step,** neighboring triangles with similar 2D motion parameters are clustered into patches type 2. By the **fourth step**, neighboring triangles with similar displacement magnitude are clustered into patches type 3. The magnitude of the displacement vector of each triangle is taken from a 2D segmentation of a pel–wise displacement vector field into regions $G_1, G_2, ..., G_{N_{res}}$ of homogeneous displacement magnitude $|\boldsymbol{D}_1|, |\boldsymbol{D}_2|, ..., |\boldsymbol{D}_{N_{res}}|$, respectively. For 2D segmentation, the maximum likelihood multi–thresholding method based on population mixture models as proposed in Chapter 2 is applied. For each triangle, its projection into the image plane of the camera is then calculated. If this projection is inside of a region G_i, $|\boldsymbol{D}_i|$ is assumed to be the triangle's displacement magnitude. Patches type 1, 2 and 3 may share triangles. By the **fifth step**, clustering results obtained by previous frames are updated considering the clustering results obtained by the analysis of the current frame. Therefore, a patch–membership–memory is attached to the triangles of the model object, i.e. the triangle's membership to a patch is stored with each triangle. The patches type 1, 2 and 3 obtained by the second, third and fourth step of the current frame, respectively, are used either to define a new patch in the patch–memory or to update patches stored already in the patch–memory[7]. By the **sixth step**, as soon as a patch in the patch–memory is not changed during more than two successive updates, it is defined as an object–component, if it improves 3D motion compensation.

Let δ be a patch in the patch memory which has just been defined as an object–component and L_δ the length of its silhouette δ_s, see Fig. 2. Let L_δ^O be the length of the shared boundary between the complete model object's silhouette O_s and δ_s, where $L_\delta^O + \overline{L_\delta^O} = L_\delta$. The correlation of the silhouettes δ_s and O_s is measured by the following heuristic correlation factor:

$$C_f = \left(1 - \frac{L_\delta - L_\delta^O}{L_\delta}\right) = \left(1 - \frac{\overline{L_\delta^O}}{L_\delta}\right) \qquad (10)$$

The patch δ is considered to occlude other components if $C_f \leq 0.2$. Otherwise, no occlusion occurs, see Fig. 2. **In case of no occlusion**, the patch δ is considered flexibly connected to the residual model object–components, see the head in Fig. 2.a and 3.b. **In case of occlusion,** an independent wireframe representing the shape of the object–component is generated and placed in front of the occluded object–components, see the person's right arm in the Fig. 2.b and 3.b. Before wireframe generation, a more reliable object–component's silhouette is computed by a 2D segmentation of a DVF inside of δ_s[5]. For 2D segmentation, the same maximum likelihood multi–thresholding method based on population mixture models proposed in section 2 is used.

4 RESULTS

OBASC according to [13] and *OBASC* with the extended algorithm for shape estimation of articulated objects ($OBASC^{ext}$) are applied to the test sequence ”Geovanni” (CIF–10Hz), see Fig. 1.a. This sequence shows a typical articulated object, i.e. a person. Mutual occlusion occurs. For motion estimation of articulated objects, the algorithm proposed by Martínez[8] is used. Experimental results show a decomposition into four model object–components, i.e. head, right arm, left arm and body, see Fig. 3. The image area of model failures obtained by *OBASC* and *OBASC*^{ext} is 10.5% and 7.0%, respectively, measured with the same criterion for MF detection.

5 CONCLUSION

For transmission of moving images at very low bit rates, object–based analysis–synthesis coding using the source model of ”moving articulated 3D objects” is investigated. For coding, the parameter sets describing the object–components have to be estimated. In order to estimate the shape of object–components three steps are applied: shape–initialization, object–articulation and shape–adaptation. In this contribution, the algorithm for object–articulation has been extended to consider small objects and mutual occlusion. For object–articulation the rigidity constraint imposed by a rigid object–component is exploited. For evaluating the rigidity constraint, additionally to the 3D motion parameters of each triangle both, the 2D motion parameters as well as the magnitude of the displacement vector of each triangle are also taken into account. Due to the reduced number of parameters to be estimated, the estimation of the 2D motion and the displacement magnitude for each triangle is more reliable in the case of small objects. If mutual occlusion is detected, the shape of the object–components in foreground and the shape of the occluded object–components are represented by separated wireframes. The wireframes of the object–components in foreground are placed in front of the occluded object–components. Applying the proposed algorithm to the real test sequence ”Geovanni” shows that object–articulation of small object–components is possible also by mutual occlusion of the object–components. For the test sequence ”Geovanni” the average size of the model failures areas decreases from 10.5% to 7.0% of the image area. It can be expected that this reduction of the size of model failures will significantly reduce the bit–rate necessary for coding this image sequence.

6 REFERENCES

[1] R. Holt, A. Netravali, T. Huang and R. Qian, ”Determining Articulated Motion from Perspective Views: A Decomposition Approach”, Proc. IEEE Workshop on Motion of Non–Rigid and Articulated Objects, Austin, Texas, Nov. 1994, pp. 126–137.

[2] M. Bierling, ”Displacement Estimation by Hierarchical Blockmatching”, 3rd SPIE Symposium on Visual Communications and Image Processing, Cambridge, USA, Nov. 1988, pp. 942–951.

[3] J. G. Frerichs, Verfahren zur Detektion kleiner bewegter Objektkomponenten bei teilweiser gegenseitiger Verdeckung für die Bewegtbildcodierung, Studienarbeit, University of Hannover, Germany, 1996.

[4] C. A. Galsbey, ”An Analysis of Histogram–Based–Thresholding Algorithms”, CVGIP: Graphical Models and Image Processing, Vol. 55, Nov. 1993, pp. 532–537.

[5] J. Goldbeck, Automatische Modellierung bewegter 3D–Objekte in Bildfolgen unter Berücksichtigung möglicher Verdeckungen, Studienarbeit, University of Hannover, Germany, 1993.

[6] T. Kurita, N. Otsu and N. Abdelmalek, ”Maximum Likelihood Thresholding Based on Population Mixture Models”, Pattern Recognition, Vol. 25, No. 10, March 1992, pp. 1231–1240.

[7] G. Martínez, ”Shape Estimation of Articulated 3D Objects for Object–Based Analysis–Synthesis Coding (OBASC)”, accepted for publication in Signal Processing: Image Communication.

[8] G. Martínez, ”3D Motion Estimation of Articulated 3D Objects for Object–Based Analysis–Synthesis Coding (OBASC)”, International Workshop on Coding Techniques for Very Low Bit–rate Video, Tokyo, Japan, Nov. 1995, paper No. G–1.

[9] G. McLachlan and K. Basford, Mixture Models, Marcel Dekker Inc., USA, 1988, Cap. 1–2, pp. 1–70.

[10] Y. W. Lim, S. U. Lee, ”On the Color Image Segmentation Algorithm Based on Fuzzy C. Means Techniques”, Pattern Recognition, Vol. 23, No. 9, 1990, pp. 935–952.

[11] H.G. Musmann, M. Hötter, J. Ostermann, ”Object–Oriented Analysis–Synthesis Coding of Moving Images”, Signal Processing: Image Communication, Vol. 3, No. 2, Nov. 1989, pp. 117–138.

[12] J.A. Nelder, R. Mead: ”A Simplex Method for Function Minimization”, No. 7, July 1964, pp. 308–313.

[13] J. Ostermann, ”Object–Based Analysis Synthesis Coding Based on the Source Model of Moving Rigid 3D Objects”, Signal Processing: Image Communication, Vol. 6, No. 2, May 1994, pp. 143–161.

[14] S. L. Sclove, ”Population Mixture Models and Clustering Algorithms”, Commun. Statist.–Theor. Meth. A6(5), 1977, pp. 417–434.

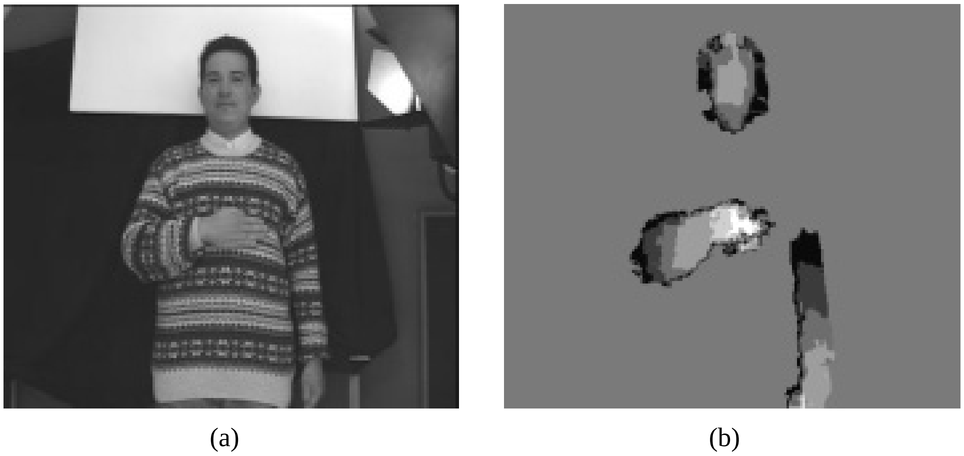

Fig. 1 (a) Frame 5 of the image sequence ”Geovanni”, CIF–10Hz. (b) Regions found after classification of the discrete magnitude values of the DMF_s .

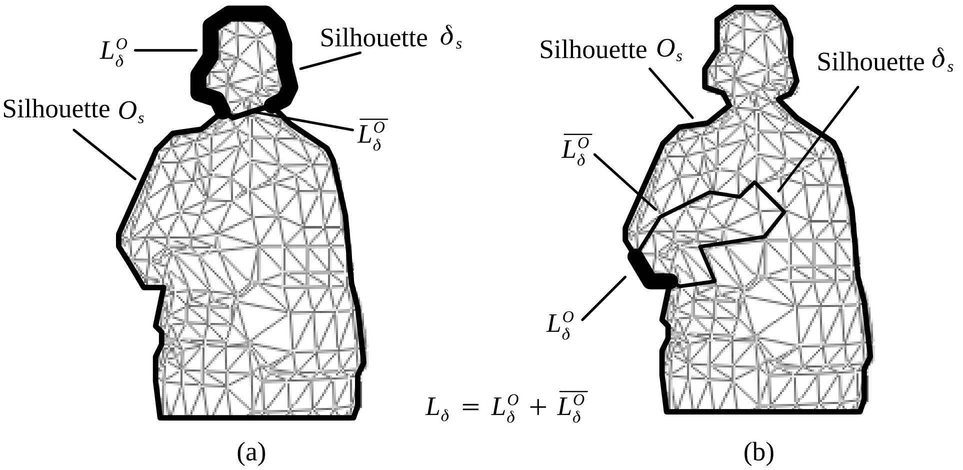

Fig. 2 Lengths evaluated for occlusion detection. δ_s is the silhouette of a patch δ in the patch–memory considered as a model object–component. O_s represents the silhouette of the complete model object. a) Since $C_f = 0.75$ i.e. $\overline{L^O_\delta} \ll L^O_\delta$ occlusion is not asummed (see Eq. 10). b) Since $C_f = 0.18$ i.e. $\overline{L^O_\delta} \gg L^O_\delta$ occlusion is assumed (see Eq. 10).

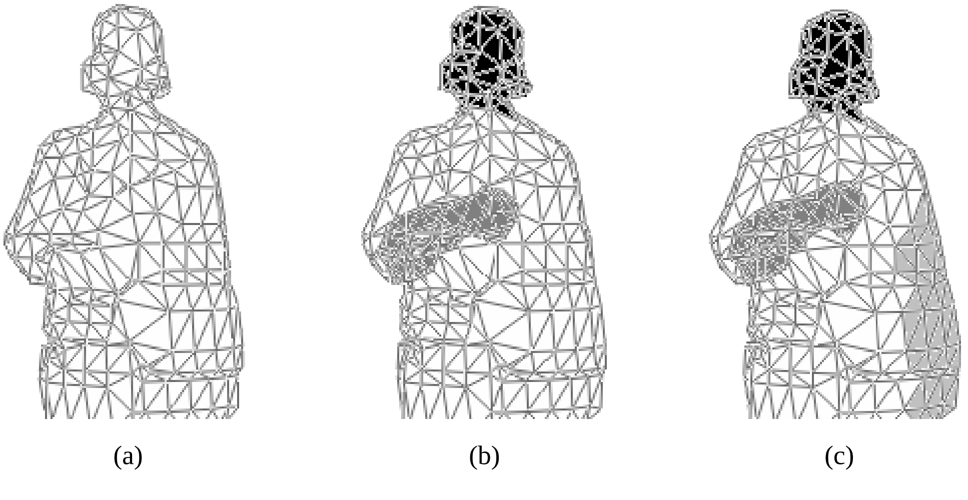

Fig. 3 Wireframe of the model object of the test sequence ”Geovanni” after (a) 3, (b) 6 and (c) 8 frames. The object–components are represented with different grey levels.