Shape-based Posture and Gesture Recognition in Videos

Stephan Kopf, Thomas Haenselmann, Wolfgang Effelsberg
Dept. of Computer Science IV, University of Mannheim, Germany
{kopf,haenselmann,effelsberg}@informatik.uni-mannheim.de

ABSTRACT

The recognition of human postures and gestures is considered to be highly relevant semantic information in videos and surveillance systems. We present a new three-step approach to classifying the posture or gesture of a person based on segmentation, classification, and aggregation.

A background image is constructed from succeeding frames using motion compensation and shapes of people are segmented by comparing the background image with each frame. We use a modified curvature scale space (CSS) approach to classify a shape. But a major drawback to this approach is its poor representation of convex segments in shapes: Convex objects cannot be represented at all since there are no inflection points. We have extended the CSS approach to generate feature points for both the concave and convex segments of a shape. The key idea is to reflect each contour pixel and map the original shape to a second one whose curvature is the reverse: Strong convex segments in the original shape are mapped to concave segments in the second one and vice versa. For each shape a CSS image is generated whose feature points characterize the shape of a person very well.

The last step aggregates the matching results. A transition matrix is defined that classifies possible transitions between adjacent frames, e.g. a person who is sitting on a chair in one frame cannot be walking in the next. A valid transition requires at least several frames where the posture is classified as *standing-up*. We present promising results and compare the classification rates of postures and gestures for the standard CSS and our new approach.

Keywords: shape analysis, posture and gesture recognition, curvature scale space

1. INTRODUCTION

One of the central topics in computer vision is the recognition of human postures and gestures. A major goal is to go beyond traditional human-computer interaction (like mouse or keyboard) and find a more natural means of interaction with computers. Not only natural communication but also the movement and action of a person are highly relevant. Examples include surveillance systems that detect and track a person and recognize special actions (e.g., *theft*). Smart interfaces or smart rooms[1] recognize motion and gestures and can react to and communicate with a person. An example is the KidsRoom,[2, 3] a fully automated and interactive play space for children. The human-computer interaction is based solely on computational perception.

Many approaches[4] detect principal parts of a body[5–7] such as fingers, hands, feet or the head based on color or edge information. Additional parts of the body (e.g., knees) then are derived from the positions of the principal parts. Other methods use three-dimensional models to identify the posture or gesture of a person.[8–10] The images from several cameras are aggregated to gain this 3D information.

We present a generic approach to identify postures and gestures of a person that analyzes the global shape of an individual. A reliable segmentation is required: We assume that the camera is static or that at least half of the visible area in a frame is background. Otherwise it is not possible to differentiate between foreground and background motion nor can segmentation of the person be performed.

The rest of the paper is organized as follows: In the following section the relevant work in the area of object segmentation in combination with the curvature scale space technique is presented. Section 3 gives an overview of the three steps of our recognition process. The subsequent sections describe the details of this approach. The automatic segmentation of people is presented in Section 4. Sections 5 and 6 describe the feature extraction and matching processes. The aggregation of the classification results for individual frames is presented in Section 7. Finally, Section 8 displays the evaluation results and concludes the paper.

This is a preliminary version of an article published in
Proc. of IS&T/SPIE Electronic Imaging (EI), pp. 114 – 124, San José, CA, USA, January 2005
by Stephan Kopf, Thomas Haenselmann, Wolfgang Effelsberg

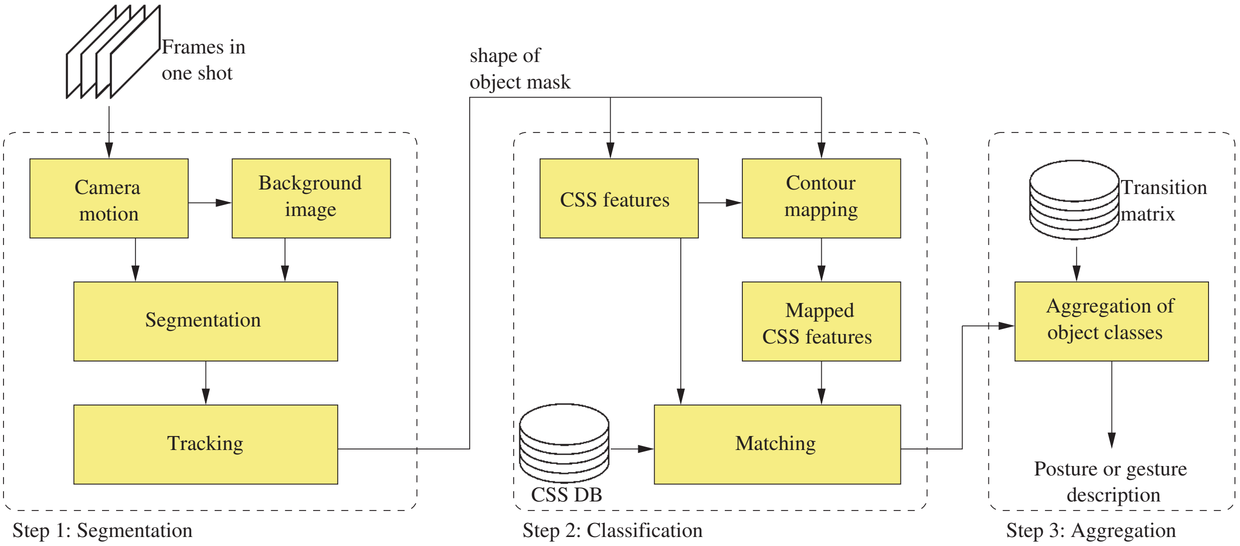


Figure 1. Overview of the recognition process

2. RELATED WORK

A first major step is the separation of the foreground from the background in videos. Many approaches have been presented to separate and segment foreground objects in videos, including the promising optical flow techniques[11–15] and statistical approaches.[16] The general idea is to identify foreground and background (camera) motion. A camera model that uses six or eight parameters is sufficient to describe most camera motions.[17–19] To estimate the modell's parameters feature-based[20] or gradient-based algorithms[21] can be used.

Surveys on the recognition of objects in videos based on shape analysis can be found in[22–24]. One of the reliable and fast shape classification techniques is the curvature scale space (CSS) technique,[25–28] which was one of the features selected to describe objects in the MPEG-7 standard.[29] Many other approaches and combinations of shape descriptors have been proposed in the literature.[30]

The occurrence of ambiguities with regard to concave segments of a shape constitutes a major drawback to the CSS technique. Some methods have been proposed to resolve these ambiguities.[25, 28] A more severe, and as yet unaddressed, problem with the CSS approach, is the poor representation of convex segments of a shape. We present a new approach based on the CSS technique with which even convex shapes can be classified.

3. OVERVIEW

The posture of a person is recognized in a process consisting of three major steps (see Figure 1): Once the person has been segmented (step 1), their shape is classified in each frame according to shape features (step 2). Step 3 aggregates the classification results and provides a reliable description of that person's posture or gesture.

The general idea of *object segmentation* is to estimate the camera motion between consecutive frames in a shot. Based on the camera motion a background image is constructed automatically. Foreground objects are removed from the background image by means of temporal filtering. The differences between each frame and the background image are evaluated and an object mask is calculated.

The *classification* step analyzes the shape of the segmented person. The curvature scale space (CSS) approach analyzes the curvature in each shape pixel and calculates features for significant concave segments of the shape. We present a new approach that we call *contour mapping* to make possible the analysis of convex regions of a shape. The actual matching compares the feature vectors to those of known shapes that are stored in a database. This step uses the features that characterize the concave and convex segments of a shape.

In the last step, the classification results are *aggregated* for each frame in a shot. False classifications of individual frames can be removed. The general idea is to define valid transitions of shapes between adjacent frames since random changes of gestures or postures are very unlikely.

4. SEGMENTATION

The segmentation is based on the analysis of the camera (background) motion and foreground motion. A background image without any objects in the foreground is constructed and the segmentation is performed by comparing the frames to the background image.

We use an eight-parameter camera model to describe the camera motion between two consecutive frames. The position of each pixel (x, y) in a frame i is transformed to a new position (x', y') in frame $i + 1$:

$$x' = \frac{a_{11}x + a_{12}y + t_x}{p_x x + p_y y + 1} \tag{1}$$

$$y' = \frac{a_{21}x + a_{22}y + t_y}{p_x x + p_y y + 1}. \tag{2}$$

The parameters t_x and t_y describe the translation (pan or tilt) in a frame and a_{ij} specifies the additional linear part of the image transformation (rotation, zoom). The parameters of the camera model are calculated by tracking significant feature points, which we gather using the Harris corner detector.[31] Correspondences of corners in consecutive frames are identified. The location of four correspondences between two frames suffices to calculate the eight parameters of the camera model. Since many corners are located and some correspondences describe the motion of foreground objects, we use a robust regression method (least-trimmed squares regression[32]) to identify the correct camera parameters.

A background image is constructed based on the camera motion in adjacent frames. Following the transformation with the camera model all frames are aligned and the position of the background in all matches. We apply a median filter in each pixel position to remove foreground objects. The person is segmented by comparing each frame to the background image. In a final step, the segmented region is validated. Its position, size, aspect ratio and direction of the motion are tracked in the shot in order to eliminate noise and remove segmentation errors. Details of this approach are published in[33].

5. CURVATURE SCALE SPACE REPRESENTATION

We analyze the outer shape of the segmented region and derive curvature features for classification. The standard curvature scale space (CSS) technique[26, 28, 29] is based on the idea of curve evolution. The closed planar curve $\Gamma(u)$ represents the outer shape of the segmented object with the normalized arc length parameter u:

$$\Gamma(u) = \{(x(u), y(u)) | u \in [0, 1]\}. \tag{3}$$

The shape is iteratively deformed by a Gaussian kernel $g(u, \sigma)$ of width σ. $X(u, \sigma)$ and $Y(u, \sigma)$ denote the position of the components after convolution with $g(u, \sigma)$. The deformation of the curve is presented by:

$$\Gamma(u, \sigma) = \{(X(u, \sigma), Y(u, \sigma)) | u \in [0, 1]\}. \tag{4}$$

The curvature κ of an evolved curve is defined as:

$$\kappa(u, \sigma) = \frac{X_u(u, \sigma) \cdot Y_{uu}(u, \sigma) - X_{uu}(u, \sigma) \cdot Y_u(u, \sigma)}{(X_u(u, \sigma)^2 + Y_u(u, \sigma)^2)^{3/2}}, \tag{5}$$

and can be computed with the first and second derivatives of $X(u, \sigma)$ and $Y(u, \sigma)$. A CSS image $I(u, \sigma)$ provides a multi-scale representation of the inflection points of Γ and is defined by:

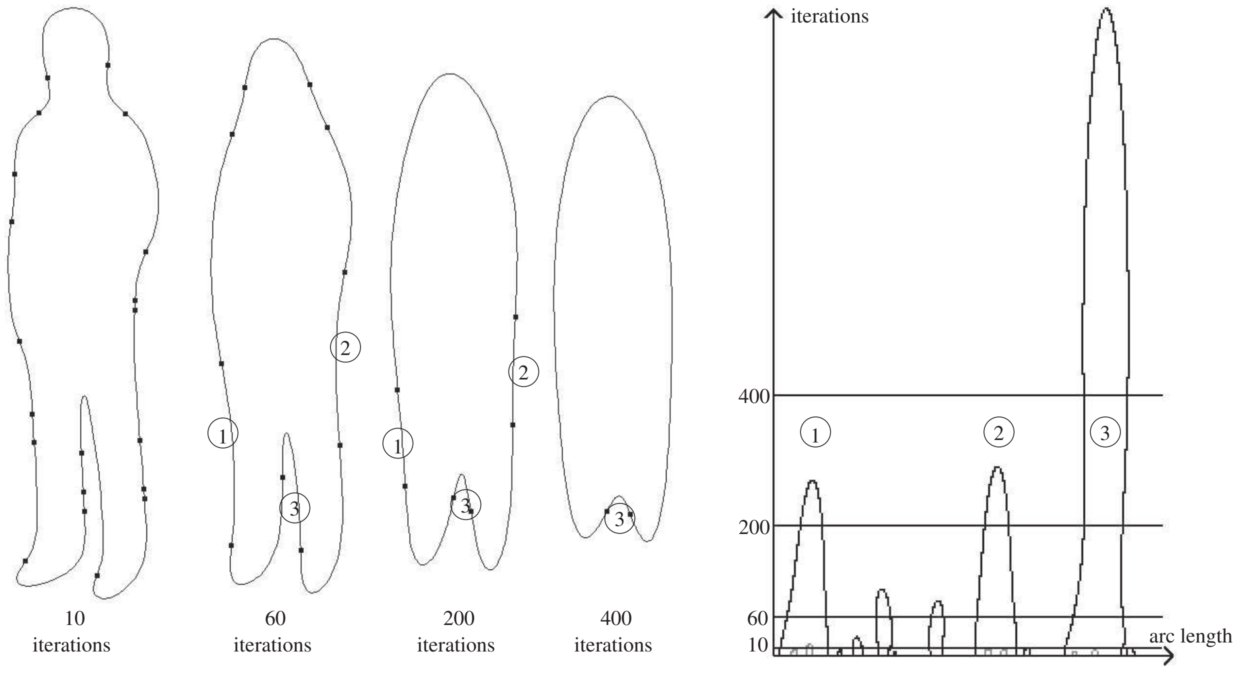


Figure 2. Segmented person and smoothed shapes with inflection points are depicted following 10, 60, 200 and 400 iterations, respectively. The corresponding CSS image is visible on the right side. The positions of three major concave segments are marked in the shapes and the CSS image.

$$I(u,\sigma) = \{((u,\sigma)|\kappa(u,\sigma) = 0\}. \tag{6}$$

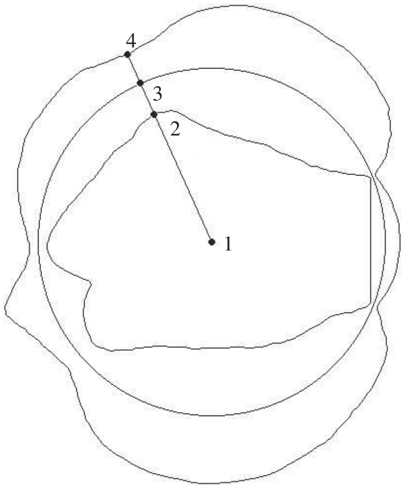
Significant peaks in the CSS images represent major concave segments of the shape. We use the position and size of the peaks as the feature points in the CSS image to describe a shape. Figure 2 depicts the evolution of a shape and the corresponding CSS image.

A major drawback of the standard CSS approach is the inadequate representation of convex segments of the shape. A CSS image represents the position of the inflection points of a shape. Usually, two inflection points define a concave segment of the shape; convex shapes without concave segments cannot be distinguished by this approach.

We use a two-step approach: First, we apply the standard CSS technique to get feature vectors that characterize concave parts of the shape very well. The general idea is now to create a second shape that we call the *mapped shape*. This shape provides additional features for the convex segments of the original shape. Strong convex segments of the original shape become concave segments of the mapped shape.

To create a mapped shape, we enclose the shape by a circle of radius R and identify the point P of the circle closest to each shape pixel. The shape pixel is mirrored on the tangent of the circle in P. A shape and its mapped shape are depicted in Figure 3. Segments of the shape that have a strong convex curvature are mapped to concave segments. The strength of the curvature of the mapped shape depends on the radius R of the circle and on the curvature of the original shape. If a convex curvature is stronger than the curvature of the circle, the segment in the mapped shape will be concave.

The mapped shape can be calculated quickly. Each shape pixel at position u of the closed planar curve $(x(u); y(u))$ is mapped to a curve $(x'(u); y'(u))$. The center of the circle $(M_x; M_y)$ of radius R is calculated as the average position of shape pixels ($M_x = 1/N \sum_{u=1}^{N} x(u), M_y = 1/N \sum_{u=1}^{N} y(u)$).



1: center of original shape and circle $(M_x; M_y)$

2: sample pixel of original shape (x(u); y(u))
(convex segment)

3: closest point to the circle (P)

4: new (mapped) shape pixel (x'(u); y'(u))
(concave segment)
(x(u),y(u)) and (x'(u),y'(u)) have the
same distance to P

Figure 3. Original shape of a fist, enclosing circle and mapped shape

$$\begin{aligned} D_{x(u),y(u)} &= \sqrt{(M_x - x(u))^2 + (M_y - y(u))^2} && (7) \\ x'(u) &= \frac{2 \cdot R - D_{x(u),y(u)}}{D_{x(u),y(u)}} \cdot (x(u) - M_x) + M_x && (8) \\ y'(u) &= \frac{2 \cdot R - D_{x(u),y(u)}}{D_{x(u),y(u)}} \cdot (y(u) - M_y) + M_y && (9) \end{aligned}$$

$D_{x(u),y(u)}$ specifies the distance between the center of the circle and the current shape pixel. If the position of a shape pixel and the center of the circle coincide, they cannot be mapped. If this is the case, the shape pixel will be interpolated from adjacent shape pixels of the mapped shape.

In principle, the mirroring of shapes is not limited to enclosing circles. However, using other shapes would cause difficulties: Angular shapes like rectangles would create discontinuous shapes. Ellipses bear the disadvantage that the point P (where the shape pixel is mirrored) is not always unique. E.g., in the case of ellipses that are parallel to the X- and Y-axes, the mirroring is undefined for all points on these axes.

We apply the standard CSS approach to the mapped shape and gain additional feature points. To indicate the classification of convex segments in the original shape, we represent this new CSS image by means of negative values. Figure 4 depicts extended CSS images for two shapes of a hand. Positive values represent the original CSS images, negative values those of the mapped shapes. Although the original CSS images of the hands in Figure 4 are quite similar, the extended CSS approach allows them to be reliably distinguished.

6. CLASSIFICATION

The segmented shape is sampled using a fixed number of samples. The CSS image is calculated, and peaks are extracted. It is sufficient to extract the significant maxima (above a certain noise level). The position on the shape and the value (iteration or Gaussian kernel width) are stored for each peak, each of which characterizes a convex region.

The sampled shape pixels are transformed to the mapped shape, and a second CSS image is created. The mapped feature vectors are stored as negative values. Up to ten feature vectors are stored for each CSS image (original and mapped). To match an unknown shape, all CSS peaks are compared to those of the shapes that are stored in a database.[28]

The general idea behind the matching algorithm is to compare the peaks in the CSS images according to the height and position of the arc. This is done by first determining the best position at which to compare the peaks. It might be

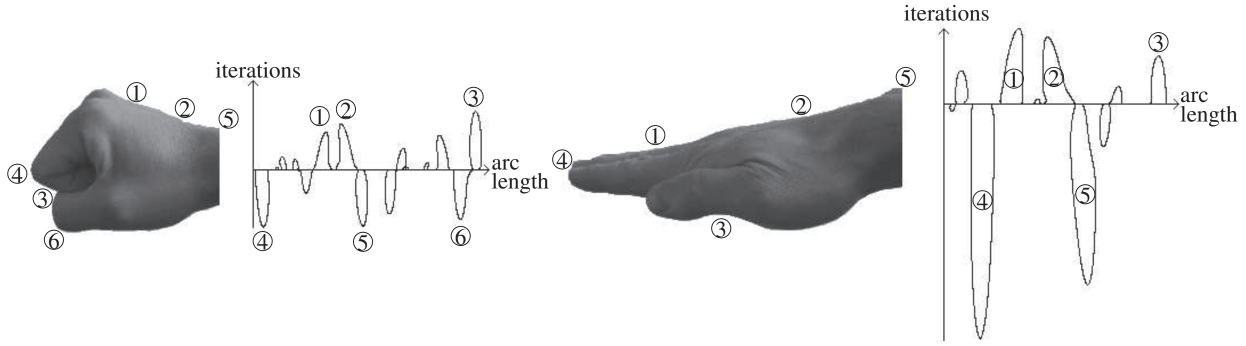


Figure 4. Two extended CSS images of a hand. Positive values in the CSS image represent concave segments of a shape, negative values characterize significant convex segments. A reliable distinction between the left and the right shape is not possible if only positive CSS peaks are compared. Both CSS images are significantly different when positive and negative peaks are regarded. The positions of the major concave and convex segments are numbered.

necessary to rotate one of the CSS images to best align them. Shifting the CSS image to the left or right corresponds to rotation of the original shape in the image. Each shape is stored only once in the database, and the horizontal moves compensate missing rotations.

For each peak in the CSS image of the unknown shape a matching peak is determined. If one is found, the Euclidean distance between the two peaks (height, position) is calculated and added to the difference between both CSS images. Otherwise, the height of the peak in the first, unknown, image is multiplied by a penalty factor and added to the total difference. Negative CSS peaks cannot be matched to positive ones (the concave segments in the original and in the mapped shape).

7. AGGREGATION

The matching technique described above calculates the best database match for each automatically segmented shape. Since the database entries are labeled with an appropriate shape name, the shape in the video can be classified accordingly. Each shape is aggregated to a class of similar shapes, e.g. to a class that describes the posture of a person. Typical classifications for a person are *standing*, *walking*, *sitting* and *standing up*. Examples for gestures of a *hand* are stored in the database as *fist*, *side (open)* or *side (closed)*.

The shape class assigned to a segmented object mask can change over time due to object deformations or matching errors. The probability of an object's changing from one object class to another depends on the respective classes. It is improbable that a person who is sitting in one frame will be walking in the following frames. Some transition frames where the person is standing-up are expected in real video shots. For each change of class we assign additional matching costs. Plausible transitions like *sitting* → *standing up* or *standing up* → *walking* will have very low additional matching costs (transition costs).

Let $d_k(i)$ denote the CSS distance between an input object mask at frame i and an object class k from the database. Furthermore, let $w_{k,l}$ denote the cost of transition from class k to l. Then, we seek the classification vector c, assigning an object class to each input object mask for all N frames, which minimizes

$$\min_c \sum_{i=1}^{N} d_{c(i)}(i) + w_{c(i),c(i-1)}. \tag{10}$$

This optimization problem can be solved by a search for the shortest path as depicted in Figure 5. With respect to the figure, $d_k(i)$ corresponds to costs assigned to the nodes, while $w_{k,l}$ corresponds to costs at the edges. The optimal path can be computed efficiently using a dynamic programming algorithm. The object behavior can be extracted easily from the nodes along the minimum-cost path.

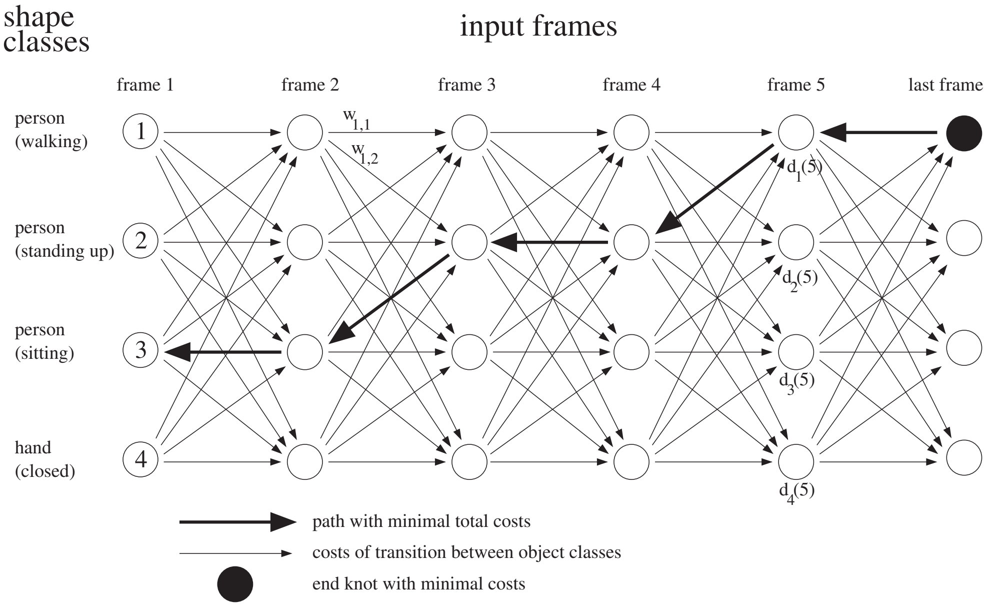


Figure 5. Example of the transition matrix including the path with the minimum costs. The transition costs $w_{k,l}$ from the shape class "person (walking)" to the class "hand (closed)" are very high. $d_k(i)$ denotes the matching costs between the segmented object in frame i and the best match in object class k.

8. RESULTS AND OUTLOOK

We have selected 400 automatically and manually segmented objects and built twelve object classes. In addition to the classes that describe the posture or gesture of a person, images of four other object classes were added, e.g., the classes *car (frontal view)* or *car (side view)*. Objects from these additional classes do not occur in the selected video sequences. Rather the classes were added to test the robustness of the recognition results.

We have analyzed five short video sequences illustrating different postures and gestures. Static and hand-held cameras were used in these examples. Figure 6 presents sample images and classification results for two video sequences. Due to the low noise level in these sequences and good segmentation results, the standard CSS approach combined with the aggregation was sufficient to classify a large number of postures. Figure 7 depicts an example in which the standard CSS technique fails. The extension with the mapped shapes and aggregation enables a correct classification. Figure 8 depicts an example with a moving camera. The segmentation errors are significantly larger than in the previous examples.

The recognition results for the five sequences are listed in Table 1. In general, the most reliable approach is the extended CSS with aggregation. Most recognition results of the extended approaches are superior to those of the standard CSS approaches. Sequence 4 is an exception: The recognition results of the standard CSS approach are slightly better. This can be explained by the fact that concave regions are visible in each segmented object. The recognition results cannot be improved by additional convex features due to many segmentation errors based on the poor quality of the difference image. The aggregation step improves the results if more than fifty percent of the shapes are successfully identified. In the case of sequence 1, the quality of the recognition using the standard CSS approach was slightly lower with additional aggregation.

In this paper, we have presented a new algorithm to automatically classifying the posture or gesture of a person. We implemented the curvature scale space approaches and provided a solution to overcoming one of the major deficiencies of the standard approach: the classification of convex shapes. We use a dynamic programming algorithm to aggregate the classification results in a shot.

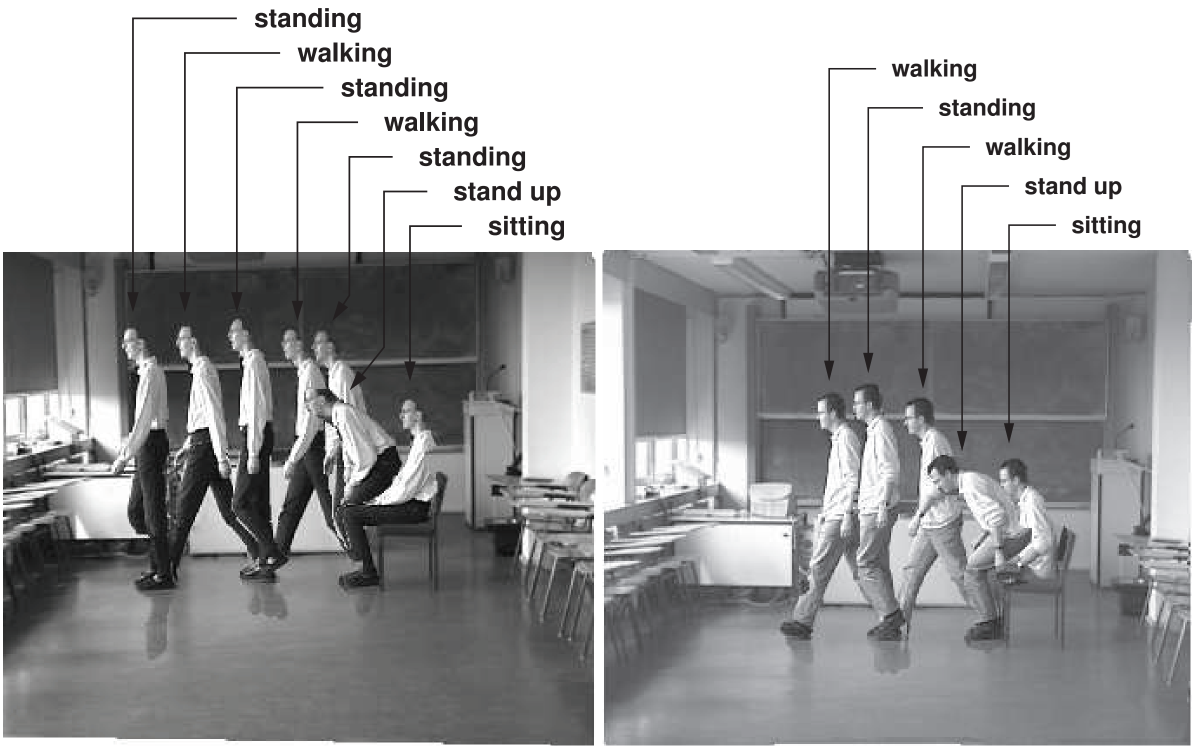
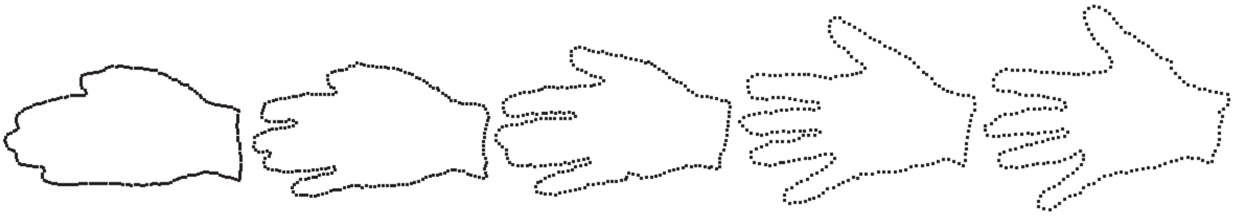


Figure 6. Automatically extracted postures of moving people (left: sequence 2, right: sequence 3).



(a) hand (side, closed)	person (standing)	person (walking)	hand (side, open)	hand (side, open)
(b) hand (side, closed)	hand (side, closed)	hand (side, open)	hand (side, open)	hand (side, open)

Figure 7. Automatically segmented and classified gestures in sequence 1: (a) standard CSS approach, (b) extended CSS approach with aggregation of the results

	Seq. 1	Seq. 2	Seq. 3	Seq. 4	Seq. 5	Average (Total)
Number of frames	46	301	259	55	189	850
CSS	46 %	61 %	75 %	56 %	72 %	67 %
CSS with aggregation	43 %	70 %	86 %	76 %	87 %	78 %
Extended CSS	76 %	67 %	83 %	53 %	81 %	75 %
Extended CSS with aggregation	100 %	75 %	92 %	71 %	95 %	86 %

Table 1. Correctly classified posture in frames in five video segments

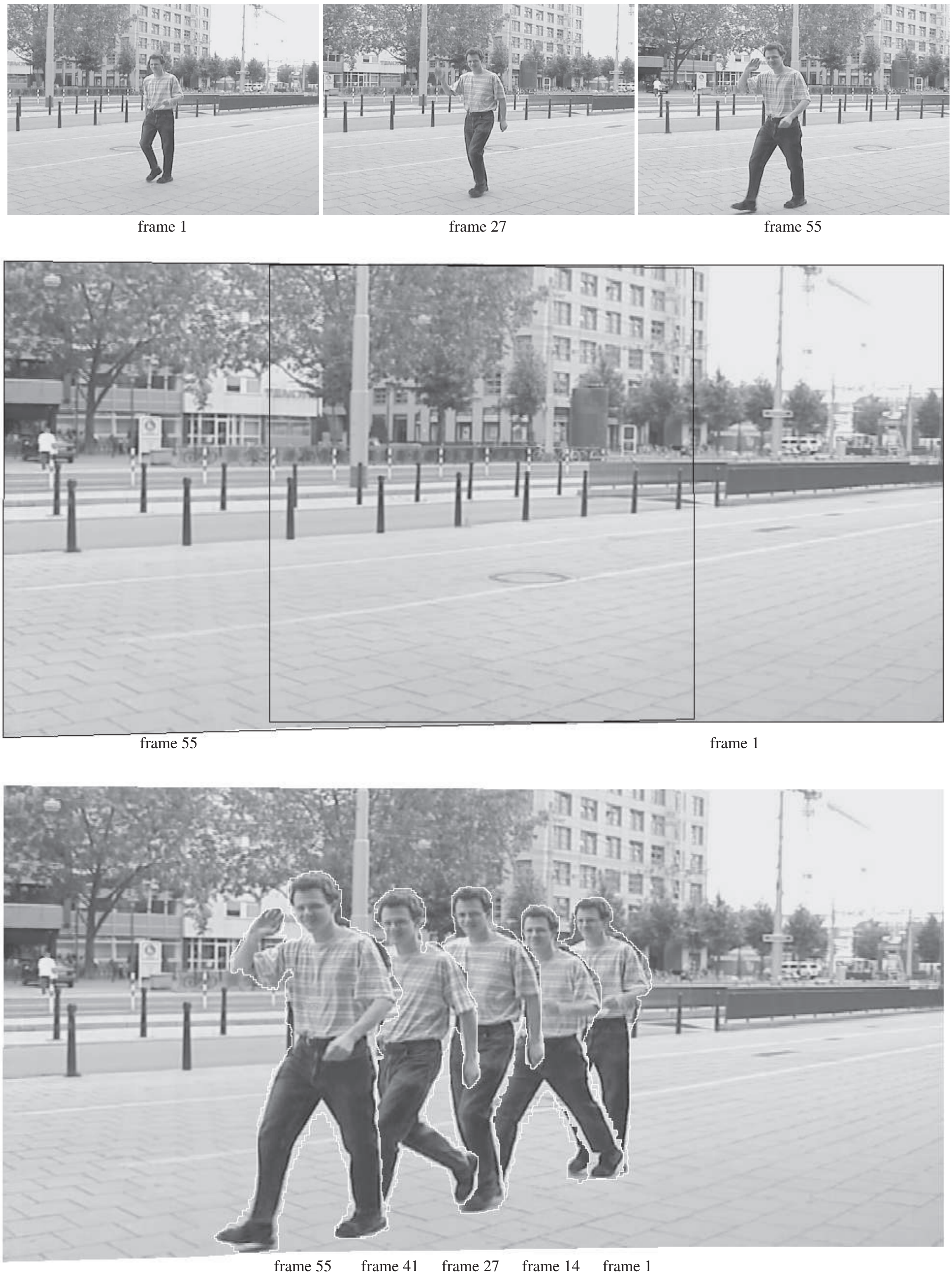


Figure 8. Three selected frames of sequence 4 with camera motion (top), automatically generated background image (center) and segmented person (bottom). The position of the first and last frame is highlighted in the background image.

REFERENCES

1. B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer, “Easyliving: Technologies for intelligent environments,” in *Handheld and Ubiquitous Computing*, September 2000.
2. A. F. B. et al., “The KidsRoom: a perceptually-based interactive and immersive story environment,” in *Teleoperators and Virtual Environments*, **8(4)**, pp. 367–391, Aug. 1999.
3. A. F. B. et al., “Perceptual user interfaces: the kidsroom,” in *Communications of the ACM archive*, **43 (3)**, ACM Press, March 2000.
4. J. J. LaViola, “A survey of hand posture and gesture recognition techniques and technology,” Tech. Rep. CS-99-11, Department of Computer Science, Brown University, June 1999.
5. I.Haritaoglu, D.Harwood, and L.Davis, “W4: Who, when, where, what: A real time system for detecting and tracking people,” in *Third Face and Gesture Recognition Conference*, pp. 222–227, 1998.
6. S. Iwasawa, K. Ebihara, J. Ohya, and S. Morishima, “Real-time estimation of human body posture from monocular thermal images,” in *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR ’97)*, 1997.
7. C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, “Pfinder: Real-time tracking of the human body,” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19 (7)**, pp. 780–785, July 1997.
8. Q. Delamarre and O. Faugeras, “3D articulated models and multi-view tracking with silhouettes,” in *Proceedings of the International Conference on Computer Vision*, **2**, IEEE Computer Society, 1999.
9. T. B. Moeslund and E. Granum, “3D human pose estimation using 2D-Data and an alternative phase space representation,” in *IEEE Workshop on Human Modeling, Analysis and Synthesis (HuMAnS 2000)*, June 2000.
10. A. Imai, N. Shimada, and Y. Shirai, “3-D hand posture recognition by training contour variation,” in *Proc. of 6th Int. Conf. on Automatic Face and Gesture Recognition*, pp. 895–900, 2004.
11. J. L. Barron, D. J. Fleet, and S. S. Beauchemin, “Performance of optical flow techniques,” *International Journal on Computer Vision* **12**(1), pp. 43–77, 1994.
12. S. S. Beauchemin and J. L. Barron, “The computation of optical flow,” *ACM Computing Surveys* **27**(3), pp. 433–467, 1995.
13. G. Kühne, J. Weickert, O. Schuster, and S. Richter, “A tensor-driven active contour model for moving object segmentation,” in *Proc. IEEE International Conference on Image Processing (ICIP)*, **II**, pp. 73–76, October 2001.
14. G. Kühne, S. Richter, and M. Beier, “Motion-based segmentation and contour-based classification of video objects,” in *Proc. ACM Multimedia 2001*, pp. 41–50, 2001.
15. A. Mitiche and P. Bouthemy, “Computation and analysis of image motion: A synopsis of current problems and methods,” *International Journal of Computer Vision* **19**(1), pp. 29–55, 1996.
16. N. Paragios and R. Deriche, “Geodesic active contours and level sets for the detection and tracking of moving objects,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**, pp. 266–280, March 2000.
17. O. D. Faugeras, *Three-Dimensional Computer Vision : A Geometric Viewpoint*, MIT Press, Cambridge, CA, second ed., 1996.
18. R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, Cambridge University Press, Cambridge, 2001.
19. B. K. P. Horn, *Robot Vision*, MIT Electrical Engineering and Computer Science, Cambridge, MA, first ed., 1986.
20. P. Torr and A. Zisserman, “Feature based methods for structure and motion estimation,” in *Vision Algorithms: Theory and Practice, International Workshop on Vision Algorithms*, B. Triggs, A. Zisserman, and R. Szeliski, eds., *Lecture Notes in Computer Science* **1883**, pp. 278–294, Springer, (Berlin, Heidelberg), 1999.
21. M. Irani and P. Anandan, “About direct methods,” in *Vision Algorithms: Theory and Practice, International Workshop on Vision Algorithms, held during ICCV ’99, Corfu, Greece, September*, B. Triggs, A. Zisserman, and R. Szeliski, eds., *Lecture Notes in Computer Science* **1883**, pp. 267–277, Springer, (Berlin, Heidelberg), 1999.
22. L. da Fontoura Costa and R. M. Cesar, Jr., *Shape Analysis and Classification*, CRC Press, Boca Raton, FL, September 2000.
23. S. Loncaric, “A survey of shape analysis techniques,” *Pattern Recognition* **31**, pp. 983–1001, August 1998.
24. T. Pavlidis, “Review of algorithms for shape analysis,” *Computer Graphics and Image Processing* **7**, pp. 243–258, April 1978.

25. S. Abbasi, F. Mokhtarian, and J. Kittler, “Enhancing CSS-based shape retrieval for objects with shallow concavities,” *Image and Vision Computing* **18**(3), pp. 199–211, 2000.
26. F. Mokhtarian, S. Abbasi, and J. Kittler, “Efficient and robust retrieval by shape content through curvature scale space,” in *Proc. International Workshop on Image DataBases and MultiMedia Search*, pp. 35–42, 1996.
27. F. Mokhtarian, S. Abbasi, and J. Kittler, “Robust and efficient shape indexing through curvature scale space,” in *British Machine Vision Conference*, 1996.
28. S. Richter, G. Kühne, and O. Schuster, “Contour-based classification of video objects,” in *Proceedings of SPIE, Storage and Retrieval for Media Databases*, **4315**, pp. 608–618, January 2001.
29. F. Mokhtarian and M. Bober, *Curvature Scale Space Representation: Theory, Applications, and MPEG-7 Standardization (Computational Imaging and Vision, 25)*, Kluwer Academic Publishers, Dordrecht, The Netherlands, first ed., 2003.
30. F. Mokhtarian and S. Abbasi, “Robust recognition based on fusion of multiple shape descriptors,” in *IASTED International Conference on Signal and Image Processing*, pp. 473–478, 2003.
31. C. Harris and M. Stephens, “A combined corner and edge detector,” in *Proc. Alvey Vision Conference*, pp. 147–151, 1988.
32. P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*, John Wiley, New York, 1987.
33. D. Farin, W. Effelsberg, and P. H. N. de With, “Robust clustering-based video-summarization with integration of domain-knowledge,” in *IEEE International Conference on Multimedia and Expo (ICME)*, **1**, pp. 89–92, 2002.