Project Acronym:	GRASP
Project Type:	IP
Project Title:	Emergence of Cognitive Grasping through Introspection, Emulation and Surprise
Contract Number:	215821
Starting Date:	01-03-2008
Ending Date:	28-02-2012

Deliverable Number:	D11
Deliverable Title :	Vision based detection, tracking and representation of human hands in action
Type:	PU
Authors	A. Argyros, V. Papadourakis, I. Oikonomidis, D. Michel, S. Gärtner, M. Do, T. Asfour, R. Dillmann
Contributing Partners	FORTH, UniKarl

Contractual Date of Delivery to the EC:	28-02-2010
Actual Date of Delivery to the EC:	28-02-2010

Contents

1 Executive summary **5**

A Attached papers **7**

Chapter 1

Executive summary

Deliverable D11 presents the second year developments within WP1 - Learning to Observe Human Grasping and Consequences of Grasping. According to the Technical Annex, deliverable D11 presents the activities in the context of Tasks 1.3 and 1.4:

- [**Task 1.3**] Observing humans: Definition and development of a system that detects and tracks humans and their movements in particular. Activities in this task will focus on the important problem of acquiring real 3D motion of the arms while the human is interacting with objects. The tracking should be successful also in cases when the robot does not have a frontal view of the human.
- [**Task 1.4**] Observing human grasping: Definition and development of a computational method that detects, tracks and represents human hands in action. The derived representation includes aspects and features in the full 4D spatiotemporal space (3D space and time dimensions). The aim is to extract from a sequence of stereoscopic hand observations, the information that is necessary and sufficient for subsequent (WP2) parsing and interpretation of observed hand activities that, in turn, support future repeats by a robotic hand. Activities within this task will address important subproblems such as figure-ground segmentation (environmental modelling, motion/colour based segmentation, coarse object categorisation) tracking humans/hands in 2D/3D (feature selection, hand models, representation of prior knowledge of motion models, prediction and search strategies), etc.

The work in this deliverable relates to the following second year Milestones:

- [**Milestone 4**] Analysis of action-specific visuo-spatial processing, vocabulary of human actions/interactions for perception of task relations and affordances.
- [**Milestone 6**] Integration and evaluation of human hand and body tracking on active robot heads, demonstration of a grasping cycle on the experimental platforms.

The progress in WP1 is presented in the below summarized scientific publications, attached to this deliverable.

- In Attachment A we propose a new approach for tracking multiple objects in image sequences. The proposed approach differs from existing ones in important aspects of the representation of the location and the shape of tracked objects and of the uncertainty associated with them. The location and the speed of each object is modeled as a discrete time, linear dynamical system which is tracked using Kalman filtering. Information about the spatial distribution of the pixels of each tracked object is passed on from frame to frame by propagating a set of pixel hypotheses, uniformly sampled from the original objects projection to the target frame using the objects current dynamics, as estimated by the Kalman filter. The density of the propagated pixel hypotheses provides a novel metric that is used to associate image pixels with existing object tracks by taking into account both the shape of each object and the uncertainty associated with its track. The proposed tracking approach has been developed to support robust hand tracking.

- In attachment B, we present a method for matching closed, 2D shapes (2D object silhouettes) that are represented as an ordered collection of shape contexts. Matching is performed using a method that computes the optimal alignment of two cyclic strings in sub-cubic runtime. Thus, the proposed method is suitable for efficient, near real-time matching of closed shapes. The method is qualitatively and quantitatively evaluated using several datasets. An application of the method for joint detection in human figures is also presented.

- In Attachment C, we present a novel approach to the problem of establishing the best match between an open contour and a part of a closed contour. At the heart of the proposed scheme lies a novel shape descriptor that also permits the quantification of local scale. Shape descriptors are computed along open or closed contours in a spatially non-uniform manner. The resulting ordered collections of shape descriptors constitute the global shape representation. A variant of an existing DTW matching technique is proposed to handle the matching of shape representations. Due to the properties of the employed shape descriptor, sampling scheme and matching procedure, the proposed approach performs partial shape matching that is invariant to Euclidean transformations, starting point as well as to considerable shape deformations. Additionally, the problem of matching closed-to- closed contours is naturally treated as a special case. Extensive experiments on benchmark datasets but also in the context of specific applications, demonstrate that the proposed scheme outperforms existing methods for the problem of partial shape matching and performs comparably to methods for full shape matching.

- In Attachment D, we present a robust object tracking algorithm that handles spatially extended and temporally long object occlusions. The proposed approach is based on the concept of "object permanence" which suggests that a totally occluded object will re-emerge near its occluder. The proposed method does not require prior training to account for differences in the shape, size, color or motion of the objects to be tracked. Instead, the method automatically and dynamically builds appropriate object representations that enable robust and effective tracking and occlusion reasoning. The proposed approach has been evaluated on several image sequences showing either complex object manipulation tasks or human activity in the context of surveillance applications. Experimental results demonstrate that the developed tracker is capable of handling several challenging situations, where the labels of objects are correctly identified and maintained over time, despite the complex interactions among the tracked objects that lead to several layers of occlusions.

- In Attachment E, we present a method for making the motion of humanoid robots more realistic and human-like in order to increase their acceptance as part of our everyday lives. A proper approach to achieve this requirement is introduced within the scope of this paper by adopting marker-based human motion capture. For this purpose, constraining and mapping of prerecorded motions is applied since robots may have different degrees of freedom (DoFs) as well as a different kinematic structure than a human. Regarding this challenge, the motion is adapted to a given robot while preserving important human-like characteristics of the recorded motion.

Appendix A

Attached papers

[A] H. Baltzakis, A.A. Argyros, Propagation of pixel hypotheses for multiple objects tracking, in Proceedings of the International Symposium on Visual Computing, 2009, pp. 140-149, Las Vegas, USA, Nov 30 Dec 2, 2009.

[B] I. Oikonomidis, A.A. Argyros, Deformable 2D Shape Matching based on Shape Contexts and Dynamic Programming, in Proceedings of the International Symposium on Visual Computing, 2009, pp. 461-469, Las Vegas, USA, Nov 30 Dec 2, 2009.

[C] D. Michel, I. Oikonomidis, A.A. Argyros, Scale invariant and deformation tolerant partial shape matching, submitted to Image and Vision Computing journal, (Elsevier) under review.

[D] V. Papadourakis, A.A. Argyros, Multiple objects tracking in the presence of long term occlusions, Computer Vision and Image Understanding Journal (Elsevier), *in print*.

[E] S. Gärtner, M. Do, C. Simonidis, T. Asfour, W. Seemann, R. Dillmann, Generation of Human-like Motion for Humanoid Robots Based on Marker-based Motion Capture Data, International Symposium on Robotics (ISR'2010), June 2010 (accepted).

Propagation of Pixel Hypotheses for Multiple Objects Tracking

Haris Baltzakis and Antonis A. Argyros

Institute of Computer Science, Forth
{xmpalt,argyros}@ics.forth.gr
http://www.ics.forth.gr/cvrl/

Abstract. In this paper we propose a new approach for tracking multiple objects in image sequences. The proposed approach differs from existing ones in important aspects of the representation of the location and the shape of tracked objects and of the uncertainty associated with them. The location and the speed of each object is modeled as a discrete time, linear dynamical system which is tracked using Kalman filtering. Information about the spatial distribution of the pixels of each tracked object is passed on from frame to frame by propagating a set of pixel hypotheses, uniformly sampled from the original object's projection to the target frame using the object's current dynamics, as estimated by the Kalman filter. The density of the propagated pixel hypotheses provides a novel metric that is used to associate image pixels with existing object tracks by taking into account both the shape of each object and the uncertainty associated with its track. The proposed tracking approach has been developed to support face and hand tracking for human-robot interaction. Nevertheless, it is readily applicable to a much broader class of multiple objects tracking problems.

1 Introduction

This paper presents a novel approach for multiple object tracking in image sequences, intended to track skin-colored blobs that correspond to human hands and faces. Vision-based tracking of human hands and faces constitutes an important component in gesture recognition systems with many potential applications in the field of human-computer and/or human-robot interaction.

Some successful approaches for hand and face tracking utilize ellipses to model the shape of the objects on the image plane [1–5]. Typically, simple temporal filters such as linear, constant-velocity predictors are used to predict/propagate the locations of these ellipses from frame to frame. Matching of predicted ellipses with the extracted blobs is done either by correlation techniques or by using statistical properties of the tracked objects.

In contrast to blob tracking approaches, model based ones [6–11] do not track objects on the image plane but, rather, on a hidden model-space. This is commonly facilitated by means of sequential Bayesian filters such as Kalman or

G. Bebis et al. (Eds.): ISVC 2009, Part II, LNCS 5876, pp. 140–149, 2009.

particle filters. The state of each object is assumed to be an unobserved Markov process which evolves according to specific dynamics and which generates measurement predictions that can be evaluated by comparing them with the actual image measurements.

Model based approaches are commonly assumed to be more suitable to track complex and/or deformable objects whose image projections cannot be modeled with simple shapes. Human hands, especially when observed from a short distance, fall in this category. Despite the fact that standard Bayesian filtering does not explicitly handle observation-to-track assignments, the sophisticated temporal filtering which is inherent to model based approaches allows them to produce better data association solutions. This is particularly important for multiple objects tracking, where it is common for tracked objects to become temporarily occluded by other tracked or non-tracked objects.

Among model-based approaches, particle filtering [12] has been successfully applied to object tracking, both with edge-based [12] and kinematic [7, 8] imaging models. With respect to the data association problem, particle filtering offers a significant advantage over other filtering methods because it allows for different, locally-optimal data association solutions for each particle which are implicitly evaluated through each particle's likelihood. However, as with any other model-based approach, particle filters rely on accurate modeling, which in most cases leads to an increased number of unknown parameters. Since the number of required particles for effective tracking is exponential to the number of tracked parameters, particle filter based tracking is applicable only to problems where the observations can be explained with relatively simple models.

In this paper we propose a blob-tracking approach that differs significantly from existing approaches in (a) the way that the position and shape uncertainty are represented and (b) the way that data association is performed. More specifically, information about the location and shape of each tracked object is maintained by means of a set of pixel hypotheses that are propagated from frame to frame according to linear object dynamics computed by a Kalman filter. Unlike particle filters which correspond to object pose hypotheses in the model space, the proposed propagated pixel hypotheses correspond to single pixel hypotheses in the observation space. Another significant difference is that, in our approach, the distribution of the propagated pixel hypotheses provides a representation for the uncertainty in both the position and the shape of the tracked object. Moreover, as it will be shown in the following sections, the local density of pixel hypotheses provides a meaningful metric to associate observed skin-colored pixels with existing object tracks, enabling an intuitive, pixel-based data association approach based on the joint-probabilistic paradigm.

The proposed approach has been tested in the context of a human-robot interaction application involving detection and tracking of human faces and hands. Experimental results demonstrate that the proposed approach manages to successfully track multiple interacting deformable objects, without requiring complex models for the tracked objects or their motion.

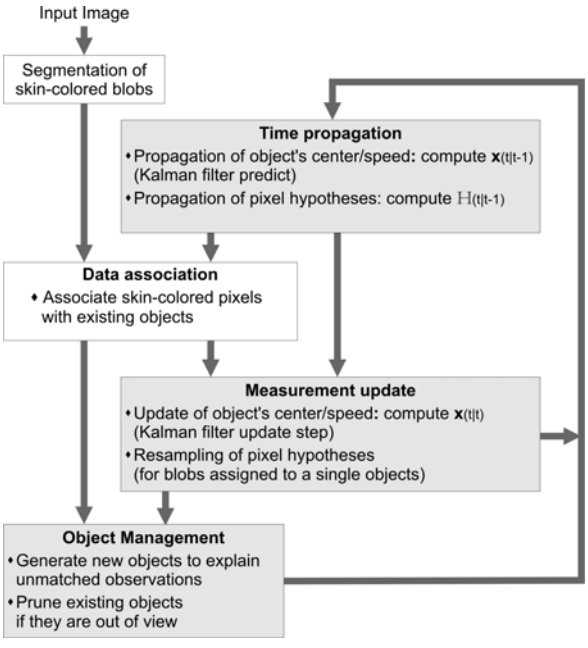


Fig. 1. Block diagram of the proposed approach

2 Problem Description and Methodology

A tracking algorithm must be able to maintain the correct labeling of the tracked objects, even in cases of partial or full occlusions. Typically, this requirement calls for sophisticated modeling of the objects' motion, shapes and dynamics (i.e. how the shape changes over time). In this paper we present a blob tracker that handles occlusions, shape deformations and similarities in color appearance without making explicit assumptions about the motion or the shape of the tracked objects. The proposed tracker uses a simple linear model for object trajectories and the uncertainty associated with them. Moreover, it does not rely on an explicit model for the shape of the tracked object. Instead, the shapes of the tracked objects and the associated uncertainty is represented by a set of pixel hypotheses that are propagated over time using the same linear dynamics as the ones used to model the object's trajectory.

An overview of the proposed approach is illustrated in Fig. 1. The first step in the proposed approach is to identify pixels that are likely to belong to tracked objects. In the context of the application under consideration, we are interested in tracking human hands and faces. Thus, the tracker implemented in this paper tracks skin-colored blobs[1]. To identify pixels belonging to such objects we em-

[1] The proposed tracking method can also be used to track blobs depending on properties other than skin color.

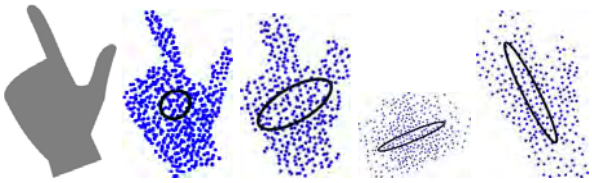


Fig. 2. Object's state representation. (a) Observed blob (b)-(e) Examples of possible states. Ellipses represent iso-probability contours for the location of the object (i.e. the first two components of $\mathbf{x_t}$). Dots represent the pixel hypotheses.

ploy a Bayesian approach that takes into account their color as well as whether they belong to the foreground or not. Image pixels with high probability to belong to hand and face regions are then grouped into connected blobs using hysteresis thresholding and connected components labeling, as in [3]. Blobs are then assigned to objects which are tracked over time. More specifically, for each tracked object two following types of information is maintained:

- The location and the speed of the object's centroid, in image coordinates. This is encoded by means of a 4D vector $\mathbf{x}(t) = [c_x(t), c_y(t), u_x(t), u_y(t)]^{\mathrm{T}}$, where $c_x(t)$ and $c_y(t)$ are the image coordinates of the object's centroid at time t and $u_x(t)$ and $u_y(t)$ are the horizontal and vertical components of its speed. A Kalman filter is used to maintain a Gaussian estimate $\hat{\mathbf{x}}(t)$ of the above-described state vector and its associated 4×4 covariance matrix $\mathbf{P}(t)$.
- The spatial distribution of the object's pixels. This is encoded by means of a set $\mathbb{H} = \{(x_i, y_i) : i = 1 \ldots N\}$ of N pixel hypotheses that are sampled uniformly from the object's blob and propagated from frame to frame using the dynamics estimated by the Kalman filter.

The representation described above is further explained in Fig. 2. Figure 2(a) depicts the blob of a hypothetical object (a human hand in this example). Figures 2(b)-(e) depict four possible states of the proposed tracker.

The distribution of the propagated pixel hypotheses provides the metric used to associate measured evidence to existing object tracks. During the data association step, observed blob pixels are individually processed one-by-one in order to associate them with existing object tracks.

After skin-colored pixels have been associated with existing object tracks, the update phase follows in two steps: (a) the state-vector (centroid's location and speed) is updated using the Kalman filter's measurement-update equations and (b) pixel hypotheses are updated by resampling them from their associated blob pixels. The resampling step is important to avoid degenerate situations and to allow the object hypotheses to closely follow the blobs shape and size.

Finally, track management techniques are employed to ensure that new objects are generated for blobs with pixels that are not assigned to any of the existing tracks and that objects which are not supported by observation are eventually removed from further consideration.

(a) (b) (c) (d)

Fig. 3. Blob detection. (a) Initial image, (b) foreground pixels, (c) skin-colored pixels, (d) resulting skin-colored blobs.

3 The Proposed Tracking Method

In this section we provide a detailed description of the proposed multiple objects tracking method.

3.1 Segmentation of Skin-Colored Foreground Blobs

The first step of the proposed approach is to detect skin-colored regions in the input images. For this purpose, a technique similar to [3, 13] is employed. Initially, background subtraction [14] is used to extract the foreground areas of the image. Then, for each pixel, $P(s|c)$ is computed, which is the probability that this pixel belongs to a skin-colored foreground region s, given its color c. This can be computed according to the Bayes rule as $P(s|c) = P(s)P(c|s)/P(c)$, where $P(s)$ and $P(c)$ are the prior probabilities of foreground skin pixels and foreground pixels having color c, respectively. Color c is assumed to be a 2D variable encoding the U and V components of the YUV color space. $P(c|s)$ is the prior probability of observing color c in skin colored foreground regions. All three components in the right side of the above equation can be computed based on offline training.

After probabilities have been assigned to each image pixel, hysteresis thresholding is used to extract solid skin color blobs and create a binary mask of foreground skin-colored pixels. A connected components labeling algorithm is then used to assign different labels to pixels that belong to different blobs. Size filtering on the derived connected components is also performed to eliminate small, isolated blobs that are attributed to noise.

Results of the intermediate steps of this process are illustrated in Fig. 3. Figure 3(a) shows a single frame extracted out of a video sequence that shows a man performing various hand gestures in an office-like environment. Fig. 3(b) shows the result of the background subtraction algorithm and Fig. 3(c) shows skin-colored pixels after hysteresis thresholding. Finally, the resulting blobs (i.e. the result of the labeling algorithm) are shown in Fig. 3(d).

(a) (b) (c) (d)

Fig. 4. Tracking hypotheses over time. (a), (b) uncertainty ellipses corresponding to predicted hypotheses locations and speed, (c), (d) propagated pixel hypotheses.

3.2 Tracking Blob Position and Speed

The dynamics of each tracked object are modeled by means of a linear dynamical system which is tracked using the Kalman filter [15, 16]. The state vector $\mathbf{x}(t)$ at time t is given as $\mathbf{x}(t) = (c_x(t), c_y(t), u_x(t), u_y(t))^{\mathrm{T}}$ where $c_x(t)$, $c_y(t)$ are the horizontal and vertical coordinates of the tracked object's centroid, and $u_x(t)$, $u_y(t)$ are the corresponding components of the tracked object's speed.

The Kalman-filter described above is illustrated in Figures 4(a) and 4(b) which show frames extracted from the same sequence as the one in Fig. 3. The depicted ellipses correspond to 95% iso-probability contours for the predicted location (smaller, red-colored ellipses) and speed (larger, purple-colored ellipses) of each tracked object's centroid. As can be verified, objects that move rapidly (e.g., object 2 in Fig. 4(a)) or objects that are not visible (e.g., object 2 in Fig. 4(b)) have larger uncertainty ellipses. On the other hand, objects that move slowly (e.g., faces) can be predicted with more certainty.

3.3 Pixel Hypotheses Propagation

Pixel hypotheses are propagated using the predicted state estimate $\hat{\mathbf{x}}(t|t-1)$ and the predicted error covariance $\mathbf{P}(t|t-1)$ of the Kalman filter discussed in the previous section. More specifically, each pixel hypothesis (x_i, y_i) in $\mathbb{H} = \{(x_i, y_i) : i = 1 \ldots N\}$ is propagated in time by drawing a new sample from

$$N\left(\begin{bmatrix} x_i + \hat{u}_x(t|t-1) \\ y_i + \hat{u}_y(t|t-1) \end{bmatrix}, \mathbf{P_h}(t|t-1)\right) \tag{1}$$

where $\hat{u}_x(t|t-1)$ and $\hat{u}_y(t|t-1)$ are the predicted velocity components (i.e. third and forth element of $\hat{\mathbf{x}}(t|t-1)$ and $\mathbf{P_h}(t|t-1)$ is the top left 2×2 submatrix of $\mathbf{P}(t|t-1)$.

Figures 4(c) and 4(d) depict the predicted pixel locations (i.e. pixel hypotheses) that correspond to the object tracks shown in Figs. 4(a) and 4(b), respectively. As can be verified, tracks with larger uncertainty ellipses correspond to less concentrated pixel hypotheses. On the other hand, propagated pixel hypotheses tend to have higher spatial density for object tracks that are predictable with higher confidence.

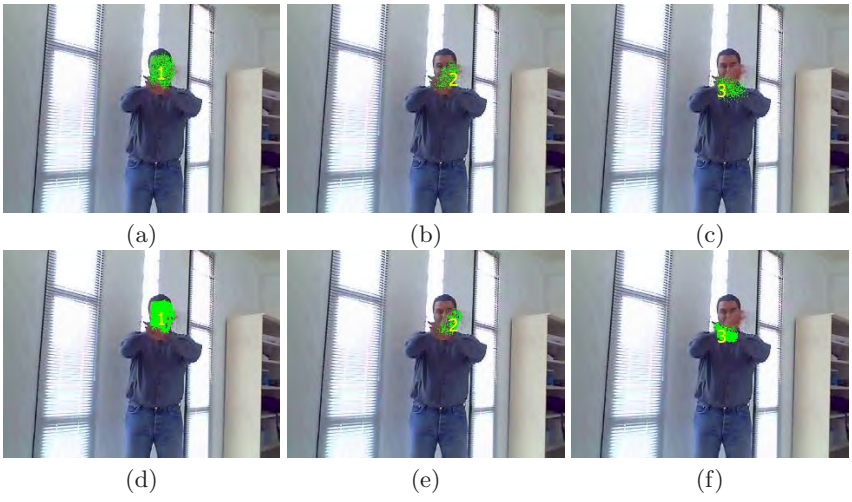


Fig. 5. Three objects merged into a single blob. Predicted pixel locations for each of the three objects (1st row), pixels finally assigned to each object (2nd row).

3.4 Associating Pixels with Objects

The purpose of the data association step is to associate observations with existing object tracks. In this paper, data association is performed on a pixel basis rather than a blob basis; i.e. each observed skin-colored pixel is individually associated to existing tracks. This permits pixels that belong to the same blob to be associated with different object tracks.

The metric used to provide the degree of association between a specific skin-colored pixel with image coordinates (x, y) and a specific object track o_i is assumed to be equal to the local density of the propagated pixels hypotheses of this track at the location of this specific pixel. More specifically, to estimate the degree of association $A(\mathbf{p}, o_i)$ between pixel $\mathbf{p}$ and track o_i, we make use of the following metric:

$$A(\mathbf{p}, o_i) = \alpha_i \frac{C^P_{N(\mathbf{p})}}{C_{N(\mathbf{p})}}, \tag{2}$$

where $N(\mathbf{p}) = \{\mathbf{p_k}, \|\mathbf{p} - \mathbf{p_k}\| \leq D\}$ is a neighborhood of pixel $\mathbf{p}$, $C^P_{N(\mathbf{p})}$ is the number of propagated pixel hypotheses of object track o_i within $N(\mathbf{p})$ and $C_{N(\mathbf{p})}$ is the total number of pixels in $N(\mathbf{p})$. α_i is a normalizing factor ensuring that the sum of all data association weights of (2) remains constant for each track over time. An 8-neighborhood ($D = \sqrt{2}$) has proven sufficient in all experiments.

After pixels have been associated with tracked objects, weighted means (according to $A(\mathbf{p}, o_i)$) are computed for each tracked object and used for the Kalman filter update phase. Pixel hypotheses are also resampled from the weighted distribution of the observed pixels. The above-described data association scheme follows the joint-probabilistic paradigm by combining all potential association candidates in a single, statistically most plausible, update.

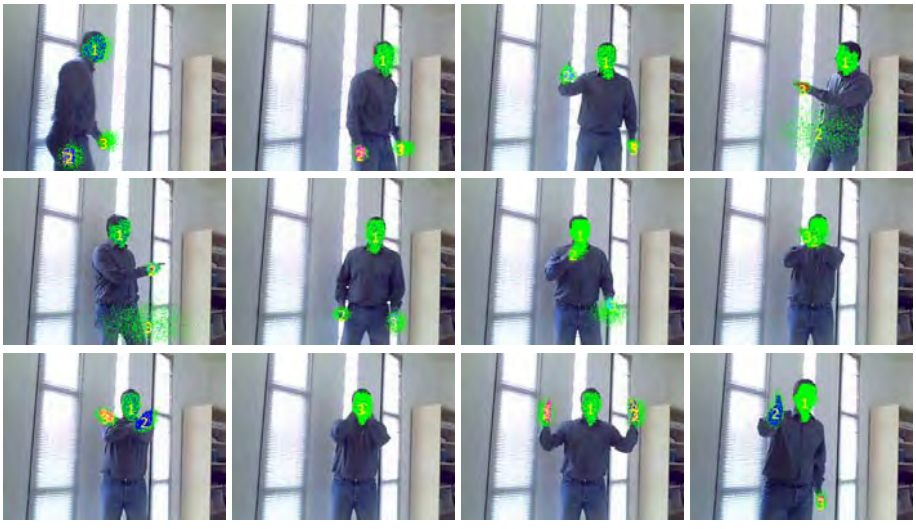


Fig. 6. Tracking results for twelve segments of the office image sequence used in the previous examples. In all cases the algorithm succeeds in tracking the three hypotheses.

A notable case that is often encountered in practice, is when all pixels of a single blob are assigned to a single track and vice versa (i.e. no propagated pixel hypotheses are associated with pixels of other blobs). In this case, resampling of pixel hypotheses is performed by uniformly sampling blob pixels. This permits pixel hypotheses to periodically re-initialize themselves and exactly-follow the blob position and shape when no data association ambiguities exist.

Figure 5 demonstrates how the proposed tracking algorithm behaves in a case where three objects simultaneously occlude each other, leading to difficult data association problems. The top row depicts the predicted pixel locations for each of the three valid tracks. The bottom row depicts the final assignment of blob pixels to tracks, according to the density of the predicted pixel hypotheses.

4 Experimental Results

Figure 6 depicts the tracker's output for a number of frames of the image sequence comprising the running example used in Figs 3, 4 and 5. As can be observed, the tracker succeeds in keeping track of all the three hypotheses despite the occlusions introduced at various fragments of the sequence.

The proposed tracker comprises an important building block of a vision-based, hand- and face-gesture recognition system which is installed on a mobile robot. The purpose of the system is to facilitate natural human-robot interaction while guiding visitors in large public spaces such as museums and exhibitions. The performance of the system has been evaluated for a three-weeks time in a large public place. Figure 7 depicts snapshots of three different image sequences captured at the installation site. Despite the fact that the operational requirements

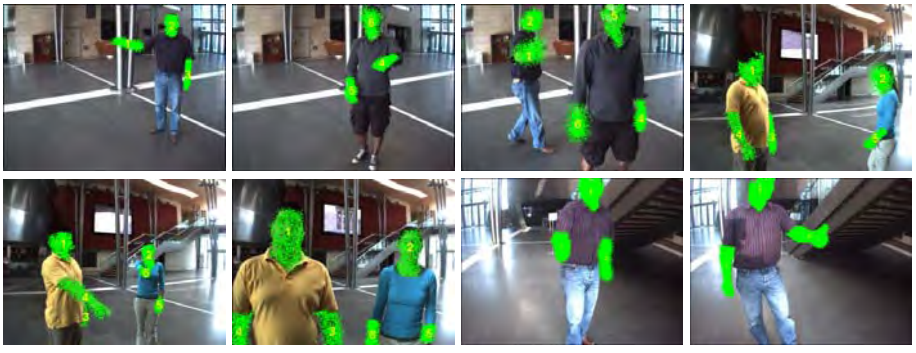


Fig. 7. Tracking results from a real-world application setup

of the task at hand (i.e. unconstrained lighting conditions, unconstrained hand and face motion, varying and cluttered background, limited computational resources) were particularly challenging, the tracker operated for a three weeks time with results that, in most cases, were proved sufficiently accurate to provide input to the hand- and face-gesture recognition system of the robot. During these experiments the algorithm ran on a standard laptop computer, operating at 640×480 images. At this resolution, the algorithm achieved a frame rate of 30 frames per second. Several video sequences obtained at the actual application site are available on the web[2].

5 Conclusions and Future Work

In this paper we have presented a novel approach for tracking multiple objects. The proposed approach differs from existing approaches in the way used to associate perceived blob pixels with existing object tracks. For this purpose, information about the spatial distribution of blob pixels is passed on from frame to frame by propagating a set of pixel hypotheses, uniformly sampled from the original blob, to the target frame using the object's current dynamics, as estimated by means of a Kalman filter. The proposed approach has been tested in the context of face and hand tracking for human-robot interaction. Experimental results show that the method is capable of tracking several deformable objects that may move in complex, overlapping trajectories.

Acknowledgments

This work was partially supported by the EU-IST project INDIGO (FP6-045388) and the EU-IST project GRASP (FP7-IP-215821).

[2] http://www.ics.forth.gr/∼xmpalt/research/handfacetrack_pixelhyps/index.html

References

1. Birk, H., Moeslund, T., Madsen, C.: Real-time recognition of hand alphabet gestures using principal component analysis. In: Proc. Scandinavian Conference on Image Analysis, Lappeenranta, Finland (1997)
2. Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.: Pfinder: Real-time tracking of the human body. IEEE Trans. Pattern Analysis and Machine Intelligence 19, 780–785 (1997)
3. Argyros, A.A., Lourakis, M.I.A.: Real-time tracking of multiple skin-colored objects with a possibly moving camera. In: Proc. European Conference on Computer Vision, Prague, Chech Republic, pp. 368–379 (2004)
4. Argyros, A.A., Lourakis, M.I.A.: Vision-based interpretation of hand gestures for remote control of a computer mouse. In: ECCV Workshop on HCI, Graz, Austria, pp. 40–51 (2006)
5. Usabiaga, J., Erol, A., Bebis, G., Boyle, R., Twombly, X.: Global hand pose estimation by multiple camera ellipse tracking. Machine Vision and Applications 19 (2008)
6. Rehg, J., Kanade, T.: Digiteyes: Vision-based hand tracking for human-computer interaction. In: Workshop on Motion of Non-Rigid and Articulated Bodies, Austin Texas, pp. 16–24 (1994)
7. Deutscher, J., Blake, A., Reid, I.: Articulated body motion capture by annealed particle filtering. In: IEEE Conference on Computer Vision and Pattern Recognition 2000, Proceedings, vol. 2, pp. 126–133 (2000)
8. Sidenbladh, H., Black, M.J., Fleet, D.J.: Stochastic tracking of 3d human figures using 2d image motion. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 702–718. Springer, Heidelberg (2000)
9. Stenger, B., Mendonca, P.R.S., Cipolla, R.: Model-based hand tracking using an unscented kalman filter. In: Proc. British Machine Vision Conference (BMVC), vol. 1, pp. 63–72 (2001)
10. Shamaie, A., Sutherland, A.: Hand tracking in bimanual movements. Image and Vision Computing 23, 1131–1149 (2005)
11. Stenger, B., Thayananthan, A., Torr, P.H.S., Cipolla, R.: Model-based hand tracking using a hierarchical bayesian filter. IEEE Trans. Pattern Analysis and Machine Intelligence 28, 1372–1384 (2006)
12. Isard, M., Blake, A.: Condensation - conditional density propagation for visual tracking. Int. Journal of Computer Vision 29, 5–28 (1998)
13. Baltzakis, H., Argyros, A., Lourakis, M., Trahanias, P.: Tracking of human hands and faces through probabilistic fusion of multiple visual cues. In: Gasteratos, A., Vincze, M., Tsotsos, J.K. (eds.) ICVS 2008. LNCS, vol. 5008, pp. 33–42. Springer, Heidelberg (2008)
14. Grimson, W.E.L., Stauffer, C.: Adaptive background mixture models for real time tracking. In: Proc. IEEE Computer Vision and Pattern Recognition (CVPR), Ft. Collins, USA, pp. 246–252 (1999)
15. Kalman, R.E.: A new approach to linear flitering and prediction problems. Transactions of the ASME–Journal of Basic Engineering 82, 35–42 (1960)
16. Bar-Shalom, Y., Li, X.: Estimation and Tracking: Principles, Techniques, and Software. Artech House Inc., Boston (1993)

Deformable 2D Shape Matching Based on Shape Contexts and Dynamic Programming

Iasonas Oikonomidis and Antonis A. Argyros

Institute of Computer Science, Forth
and
Computer Science Department, University of Crete
{oikonom,argyros}@ics.forth.gr
http://www.ics.forth.gr/cvrl/

Abstract. This paper presents a method for matching closed, 2D shapes (2D object silhouettes) that are represented as an ordered collection of shape contexts [1]. Matching is performed using a recent method that computes the optimal alignment of two cyclic strings in sub-cubic runtime. Thus, the proposed method is suitable for efficient, near real-time matching of closed shapes. The method is qualitatively and quantitatively evaluated using several datasets. An application of the method for joint detection in human figures is also presented.

1 Introduction

Shape matching is an important problem of computer vision and pattern recognition which can be defined as the establishment of a similarity measure between shapes and its use for shape comparison. A byproduct of this task might also be a set of point correspondences between shapes. The problem has significant theoretical interest. Shape matching that is intuitively correct for humans is a demanding problem that remains unsolved in its full generality. Applications of shape matching include but are not limited to object detection and recognition, content based retrieval of images, and image registration.

A lot of research efforts have been devoted to solving the shape matching problem. Felzenszwalb et al. [2] propose the representation of each shape as a tree, with each level representing a different spatial scale of description. They also propose an iterative matching scheme that can be efficiently solved using Dynamic Programming. Ebrahim et al. [3] present a method that represents a shape based on the occurrence of shape points on a Hilbert curve. This 1D signal is then smoothed by keeping the largest coefficients of a wavelet transform, and the resulting profiles are matched by comparing selected key regions. Belongie et al. [1] approach the problem of shape matching introducing the shape context, a local shape descriptor that samples selected edge points of a figure in log-polar space. The resulting histograms are compared using the x^2 statistic. Matches between corresponding points are established by optimizing the sum of matching costs using weighted Bipartite Matching (BM). Finally, a Thin Plate Spline (TPS) transformation is estimated, that warps the points of the first shape to

G. Bebis et al. (Eds.): ISVC 2009, Part II, LNCS 5876, pp. 460–469, 2009.

the second, based on the identified correspondences. This process is repeated for a fixed number of iterations, using the resulting deformed shape of the previous step as input for the next step. A very interesting work that utilizes shape contexts is presented in [4]. The goal of this work is to exploit the articulated nature that many common shapes possess to improve shape matching. The authors suggest that the distances and angles to be sampled should be measured only inside the closed contour of a figure.

In this work we are interested in the particular problem of matching deformable object silhouettes. The proposed method is based on shape contexts and the work of Belongie [1]. It is assumed that a 2D shape can be represented as a single closed contour. This is very often the case when, for example, shapes are derived from binary foreground masks resulting from a background subtraction process or from some region-based segmentation process. In this context, shape matching can benefit from the knowledge of the ordering of silhouette points, a constraint that is not exploited by the approach of Belongie [1]. More specifically, in that case, two silhouettes can be matched in sub-cubic runtime using a recently published algorithm [5] that performs cyclic string matching employing dynamic programming. The representation power of shape contexts combined with the capability of the matching algorithm to exploit the order in which points appear on a certain contour, result in an effective and efficient shape matching method.

Several experiments have been carried out to assess the effectiveness and the performance of the proposed method on benchmark datasets. The method is quantitatively assessed through the bull's-eye test applied to the MPEG7 CE-shape-1 part B dataset. More shape retrieval experiments have been carried out on the "gestures" and "marine" datasets. Additionally, the proposed shape matching method has been employed to detect the articulation points (joints) of a human figure in monocular image sequences. Specifically, 25 human postures have been annotated with human articulation points. Shape matching between a segmented figure and the prototype postures results in point correspondences between the human figure and its best matching prototype. Then TPS transfers known points from the model to the observed figure.

Overall, the experimental results demonstrate that the proposed method performs very satisfactory in diverse shape matching applications and that the performance of shape matching can be improved when the order of points on a contour is exploited. Additionally, its low computational complexity makes it a good candidate in shape matching applications requiring real-time performance.

The rest of the paper is organized as follows. The proposed method is presented in Sec. 2. Experimental results are presented in Sec. 3. Finally, Sec. 4 summarizes the main conclusions from this work.

2 The Proposed Shape Matching Method

The proposed method utilizes shape contexts to describe selected points on a given shape. A fixed number of n points are sampled equidistantly on the contour

of each shape. For each of these points, a shape context descriptor is computed. To compare two shapes, each descriptor of the 1st shape is compared using the x^2 statistic to all the descriptors of the 2nd, giving rise to pairwise matching costs. These costs form the input to the cyclic string matching, and correspondences between the shapes are established. These correspondences are used to calculate a Thin Plate Splines based alignment of the two shapes. A weighted sum of the cyclic matching cost and the TPS transformation energy forms the final distance measure of the two shapes. The rest of this section describes the above algorithmic steps in more detail.

2.1 Scale Estimation and Point Order

The first step of the method is to perform a rough scale estimation of the input shape. As in [1], the mean distance between all the point pairs is evaluated and the shape is scaled accordingly. Denoting the ith input point as pt_i, the scale a is estimated as

$$a = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{2 \left\| \mathrm{pt}_i - \mathrm{pt}_j \right\|}{n(n-1)} \,. \tag{1}$$

Then, every input point is scaled by $1/a$.

The order (clockwise/counterclockwise) in which silhouette points are visited may affect the process of shape matching. Therefore, we adopt the convention that all shapes are represented using a counter-clockwise order of points. To achieve this, the sign of the area of the polygon is calculated as

$$A = \frac{1}{2} \sum_{i=1}^{n} x_i y_{i+1} - x_{i+1} y_i \,, \tag{2}$$

with $x_{n+1} = x_1$ and $y_{n+1} = y_1$. If A is negative, the order of the input points is reversed.

2.2 Rotation Invariant Shape Contexts

For the purposes of this work, rotation invariance is a desirable property of shape matching. As mentioned in [1], since each shape context histogram is calculated in a log-polar space, rotation invariance can be achieved by adjusting the angular reference frame to an appropriately selected direction. A direction one can use for imposing rotation invariance in shape contexts, is the local tangent of the contour. In this work this direction is estimated using cubic spline interpolation. First, the 2D curve is fitted by a cubic spline model. Cubic splines inherently interpolate functions of the form $f : \mathbb{R} \to \mathbb{R}$. It is easy to extend this to interpolate parametric curves on the plane (functions of the form $\gamma : \mathbb{R} \to \mathbb{R}^2$), by concatenating two such models. The next step is to compute the derivatives of the two cubic spline models at each point of interest. For each such pair of derivatives, the local tangent is computed by taking the generalized arc tangent function with two arguments. This method has the advantage that the computed

angles are consistently aligned not only to a good estimate of the local derivative, but also to a consistent direction. The estimated local contour orientation is then used as the reference direction of shape contexts towards achieving descriptions that are rotationally invariant.

2.3 Cyclic Matching

The comparison of a pair of shape contexts can be performed with a number of different histogram comparison methods. In this work, the x^2 statistic is selected as in [1]:

$$\chi^2(h_1, h_2) = \frac{1}{2} \sum_{k=1}^{K} \frac{[h_1(k) - h_2(k)]^2}{h_1(k) + h_2(k)}, \tag{3}$$

where h_1 and h_2 are the compared histograms, each having K bins. The comparison of two shapes is performed by considering a 2D matrix C. The element (i, j) of this matrix is the x^2 statistic between the ith shape context of the first shape and the jth shape context of the second shape. Any such pair is a potential correspondence. Belongie et al. [1] use Bipartite Matching to establish a set of 1-to-1 point correspondences between the shapes. However, by exploiting the order that is naturally imposed by the contour, the search space can be significantly reduced.

For the purpose of matching, we adopt the method presented in [5]. The matrix C of x^2 shape context comparisons forms the matching costs matrix needed for the cyclic matching. Along with the matching pairs, a matching cost c_m is calculated as the sum of costs of all the aligning operations that were used. Thus, c_m can be used as a measure of the distance between the two shapes.

2.4 Thin Plate Spline Computation

The final step of the presented shape matching method is the computation of the planar deformation that aligns two shapes. The alignment is performed using Thin Plate Splines. The input to this stage is the result of the previous step, i.e. a set of pairs of correspondences between two 2D shapes. The output is a deformation of the plane, as well as a deformation cost. This cost can be properly weighted along with the cost of the previous step to form the final matching cost or distance between the shapes.

The regularized version of the TPS model is used, with a parameter λ that acts as a smoothness factor. The model tolerates higher noise levels for higher values of λ and vice versa. Since the scale of all shapes is roughly estimated at the first step of the method, the value of λ can be uniformly set to compensate for a fixed amount of noise. For all experiments, λ was fixed to 1, as in [1].

Besides the warping between the compared shapes, a total matching cost $\mathcal{D}$ is computed as

$$\mathcal{D} = l_1\, c_m + l_2\, c_b\ . \tag{4}$$

$\mathcal{D}$ is a weighted sum of the cyclic matching cost c_m and the TPS bending cost c_b. While c_b has the potential to contribute information not already captured by c_m,

in practice it proved sufficient to ignore the c_b cost, and use only the c_m cost as the distance $\mathcal{D}$ between shapes (i.e. $l_1 = 1$ and $l_2 = 0$). For all the following, this convention is kept. It should be also noted that the TPS might be needed for the alignment of matched shapes, regardless of whether the c_b cost contributes to the matching cost $\mathcal{D}$. Such a situation arises in the joints detection application described in Sec. 3.2.

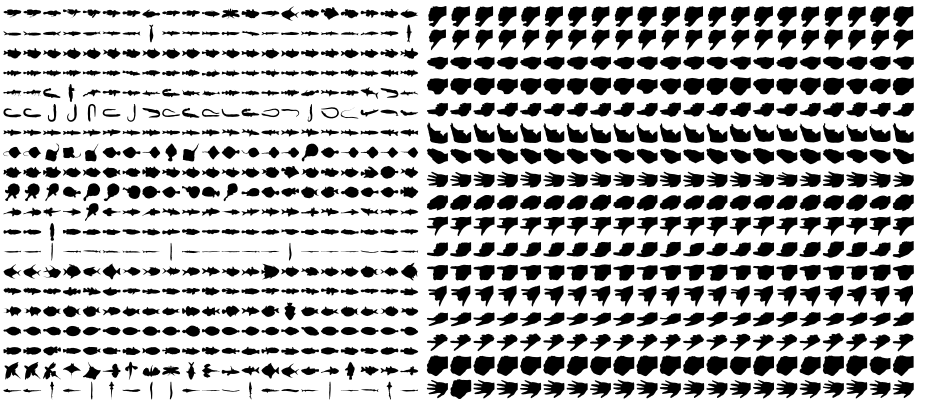
3 Experimental Results

Several experiments have been carried out to evaluate the proposed method. The qualitative and quantitative assessment of the proposed method was based on well-established benchmark datasets. An application of the method for the localization of joints in human figures is also presented. Throughout all experiments $n = 100$ points were used to equidistantly sample each shape. For the MPEG7 experiment (see Sec. 3.1) this results in an average subsampling rate of 13 contour pixels with a standard deviation of 828 pixels. This large deviation is due to the long right tail of the distribution of shape lengths. Shape contexts were defined having 12 bins in the angular and 5 bins in the radial dimension. Their small and large radius was equal to 0.125 and 2, respectively (after scale normalization). The TPS regularization parameter λ was set equal to 1 and the insertion/deletion cost for the cyclic matching to 0.75 (the χ^2 statistic yields values between 0 and 1).

3.1 Benchmark Datasets

The proposed shape matching method has been evaluated on the "SQUID" [6] and the "gestures" [7] datasets. In all the experiments related to these datasets, each shape was used as a query shape and the proposed method was employed to rank all the rest images of the dataset in the order of increasing cost $\mathcal{D}$. Figures 1(a) and 1(b), show matching results for the "SQUID" and the "gestures" datasets, respectively. In each of these figures, the first column depicts the query shape. The rest of each row shows the first twenty matching results in order of increasing cost $\mathcal{D}$. The retrieved shapes are, in most of the cases, very similar to the query.

The quantitative assessment of the proposed method was performed by running the bull's-eye test on the MPEG7 CE-shape-1 part B dataset [8]. This dataset consists of 70 shape classes with 20 shapes each, resulting in a total of 1400 shapes. There are many types of shapes including faces, household objects, other human-made objects, animals, and some more abstract shapes. Given a query shape, the bull's-eye score is the ratio of correct shape retrievals in the top 40 shapes as those are ranked by the matching algorithm, divided by the theoretic maximum of correct retrievals, which for the specific dataset is equal to 20. The bull's eye score of the proposed method on the MPEG7 dataset is 72.35%. The presented method does not natively handle mirroring, so the minimum of the costs to the original and mirrored shape is used in shape similarity comparisons. By post-processing the results using the graph transduction method [9]



(a) (b)

Fig. 1. Matching results for (a) the "SQUID" and (b) the "gestures" datasets

with the parameter values suggested therein, the score is increased to 75.42%. For comparison, the state of the art reported scores on this dataset are 88.3% for the Hilbert curve method [3] and 87.7% for the hierarchical matching method [2] (for more details, see Table 2 in [3]).

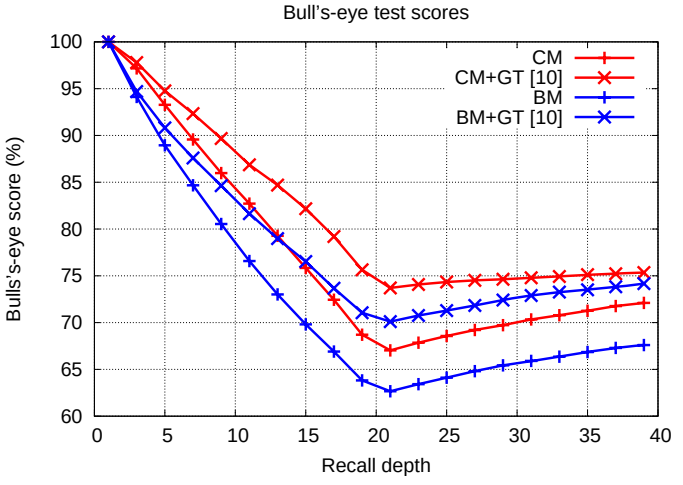
An extended investigation of the results of the bull's-eye test is graphically illustrated in Fig.2(a). This graph essentially turns the rather arbitrary choice of the forty best results into a variable. The horizontal axis of the graph is this recall length variable, and the vertical axis is the percentage of correct results among the examined ones. The experimental results demonstrate that the cyclic string matching performs better than Bipartite Matching. Additionally, graph transduction improves both methods but does not affect the superiority of the cyclic matching compared to Bipartite Matching.

The essential advantage of cyclic matching over Bipartite Matching is the reduction of the search space: while Bipartite Matching searches among all possible permutations between two shapes, cyclic matching only considers the matchings that obey the ordering restrictions imposed by both shape contours. This effectively speeds up the matching process while yielding intuitive results. Sample[1] shape retrieval results on the MPEG7 dataset are shown in Fig.2(b).

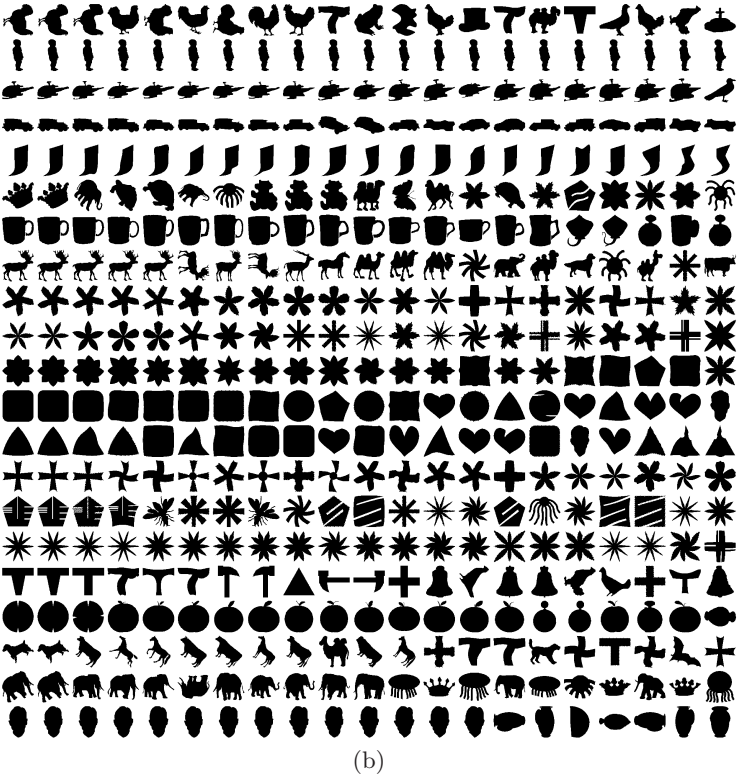
3.2 Detecting Joints in Human Figures

Due to its robustness and computational efficiency, the proposed method has been used for the recovery of the joints of a human figure. For this purpose, a set of synthetic human model figures were generated. Two model parameters control the shoulder and elbow of each arm. Several points (joints and other points of interest) are automatically generated on each model figure. Figure 3

[1] The full set of results for the reported experiments is available online at http://www.ics.forth.gr/∼argyros/research/shapematching.htm



(a)



(b)

Fig. 2. Results on the MPEG7 data set. (a) The bull's-eye test scores on the MPEG7 dataset as a function of the recall depth, (b) sample shape retrieval results.

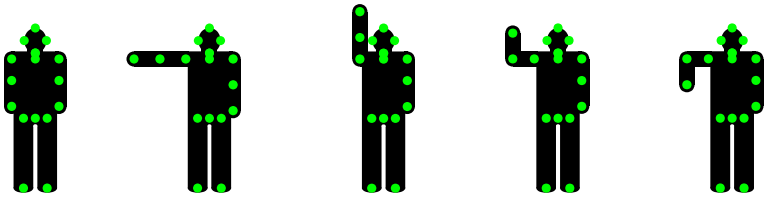


Fig. 3. The five configurations for the right arm. The contour of each figure is used as the shape model; Marked points are the labeled joints.

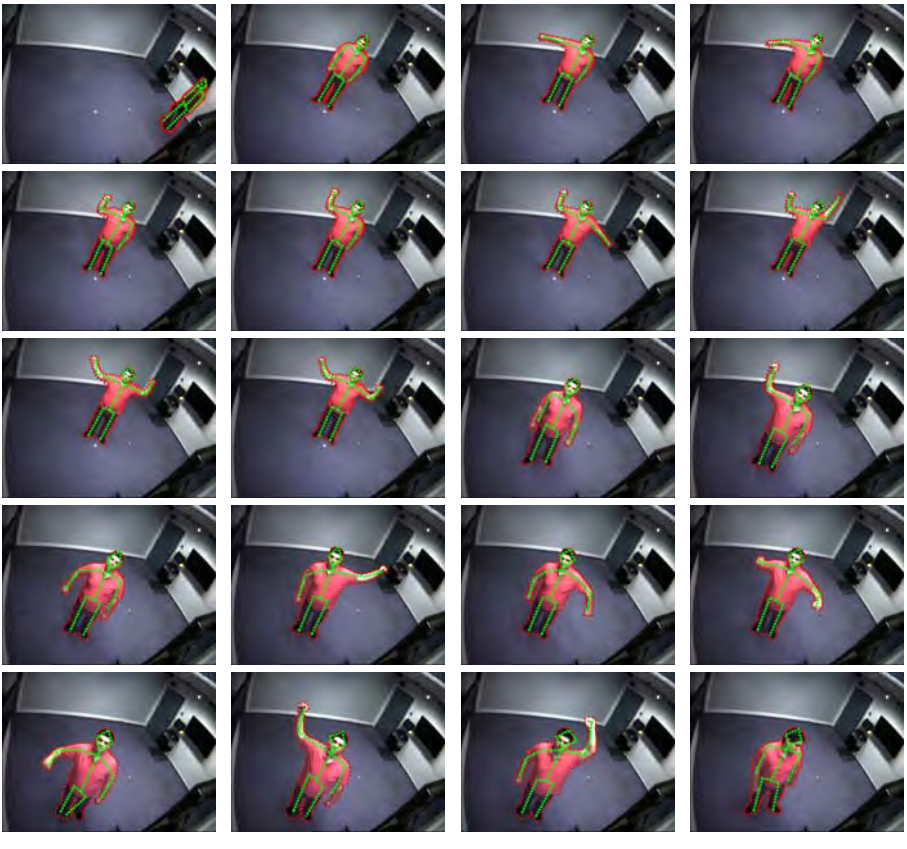


Fig. 4. Characteristic snapshots from the joints detection experiment

shows five such model figures for various postures of the right arm. A total of 25 models were created, depicting all possible combinations of articulations of the right (as shown in Fig.3) and the left arm.

In the reported experiments, the background subtraction method of [10] has been employed to detect foreground figures. Connected components of the resulting foreground mask image are then considered. If there exist more than one

connected components on the foreground image, only the one with the largest area is maintained for further processing. Its silhouette is then extracted and a fixed number n of roughly equidistant points are selected on it. This list of points constitutes the actual input to the proposed matching method. Each figure is compared to all model figures. The model with the lowest cost $\mathcal{D}$ is picked as corresponding to the input. The TPS transformation between the model and the input is subsequently used to warp the labeled points of interest on the input image.

Figure 4 shows characteristic snapshots from an extensive experiment where a human moves in a room in front of a camera while taking several different postures. The input image sequence contains approximately 1200 frames acquired at 20 fps. Having identified the joints, a skeleton model of each figure is obtained. Interestingly, the method performs well even under considerable scale and perspective distortions introduced because of the human motion that result in considerable differences between the actual foreground silhouettes and the considered prototypes.

The results presented in Fig.4 have been obtained without any exploitation of temporal continuity. This may improve results based on the fact that the estimation of the human configuration in the previous frame is a good starting point for the approximation in the current frame. To exploit this idea, at each moment in time, a synthetic figure like the ones shown in Fig.3 is custom rendered using the joint angles of the estimated skeleton. Thus, the result of the previous frame is used as a single model figure for estimating the human body configuration in the current frame. In case that the estimated distance between the synthetic model and the observed figure exceeds a specified threshold, the system is initialized by comparing the observed figure with the 25 prototype figures, as in the previous experiment. The exploitation of temporal continuity improves significantly the performance of the method.

4 Discussion

This paper proposed a rotation, translation and scale invariant method for matching 2D shapes that can be represented as single, closed contours. Affine transformations can be tolerated since the shape contexts are robust (but not strictly invariant) descriptors under this type of distortion. The performance of the method deteriorates gradually as the amount of noise increases. In this context, noise refers to either shape deformations due to errors in the observation process (e.g. foreground/background segmentation errors, sampling artifacts etc) or natural shape deformations (e.g articulations, perspective distortions, etc).

The time complexity of the method is $\mathcal{O}(n^2 \log(n))$ for n input points, an improvement over the respective performance of [1], which is $\mathcal{O}(n^3)$. In the application of Sec. 3.2, the employed unoptimized implementation performs 25 shape comparisons per second, including all computations except background subtraction. By exploiting temporal continuity, most of the time the method needs to compare the current shape with a single prototype, leading to real time

performance. Overall, the experimental results demonstrate qualitatively and quantitatively that the proposed method is competent in matching deformable shapes and that the exploitation of the order of contour points besides improving matching performance, also improves shape matching quality.

Acknowledgments

This work was partially supported by the IST-FP7-IP-215821 project GRASP. The contributions of Michel Damien and Thomas Sarmis (members of the CVRL laboratory of FORTH) to the implementation and testing of the proposed method are gratefully acknowledged.

References

1. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Transactions on PAMI 24, 509–522 (2002)
2. Felzenszwalb, P., Schwartz, J.: Hierarchical matching of deformable shapes. In: CVPR 2007, pp. 1–8 (2007)
3. Ebrahim, Y., Ahmed, M., Abdelsalam, W., Chau, S.C.: Shape representation and description using the hilbert curve. Pat. Rec. Let. 30, 348–358 (2009)
4. Ling, H., Jacobs, D.: Shape classification using the inner-distance. IEEE Transactions on PAMI 29, 286–299 (2007)
5. Schmidt, F., Farin, D., Cremers, D.: Fast matching of planar shapes in sub-cubic runtime. In: ICCV 2007, pp. 1–6 (2007)
6. Mokhtarian, F., Abbasi, S., Kittler, J.: Robust and efficient shape indexing through curvature scale space. In: BMVC 1996, pp. 53–62 (1996)
7. Petrakis, E.: Shape Datasets and Evaluation of Shape Matching Methods for Image Retrieval (2009), `http://www.intelligence.tuc.gr/petrakis/`
8. Jeannin, S., Bober, M.: Description of core experiments for mpeg-7 motion/shape (1999)
9. Yang, X., Bai, X., Latecki, L.J., Tu, Z.: Improving shape retrieval by learning graph transduction. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 788–801. Springer, Heidelberg (2008)
10. Zivkovic, Z.: Improved adaptive gaussian mixture model for background subtraction. In: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., vol. 2, pp. 28–31 (2004)

Scale invariant and deformation tolerant partial shape matching

Damien Michel[a], Iasonas Oikonomidis[b,a], Antonis Argyros[1b,a]

[a] *Institute of Computer Science, FORTH, Heraklion, Crete, Greece*
[b] *Computer Science Department, University of Crete, Greece*

Abstract

We present a novel approach to the problem of establishing the best match between an open contour and a part of a closed contour. At the heart of the proposed scheme lies a novel shape descriptor that also permits the quantification of local scale. Shape descriptors are computed along open or closed contours in a spatially non-uniform manner. The resulting ordered collections of shape descriptors constitute the global shape representation. A variant of an existing DTW matching technique is proposed to handle the matching of shape representations. Due to the properties of the employed shape descriptor, sampling scheme and matching procedure, the proposed approach performs partial shape matching that is invariant to Euclidean transformations, starting point as well as to considerable shape deformations. Additionally, the problem of matching closed-to-closed contours is naturally treated as a special case. Extensive experiments on benchmark datasets but also in the context of specific applications, demonstrate that the proposed scheme outperforms existing methods for the problem of partial shape matching and per-

[1]Corresponding author: Antonis Argyros, N. Plastira 100, Vassilika Vouton, GR-700-13, Heraklion, Crete, Greece, tel.: +30 2810 391704, FAX: +30 2810 391609, argyros@ics.forth.gr.

forms comparably to methods for full shape matching.

Key words: Partial shape matching, 2D shape descriptors, Dynamic Programming

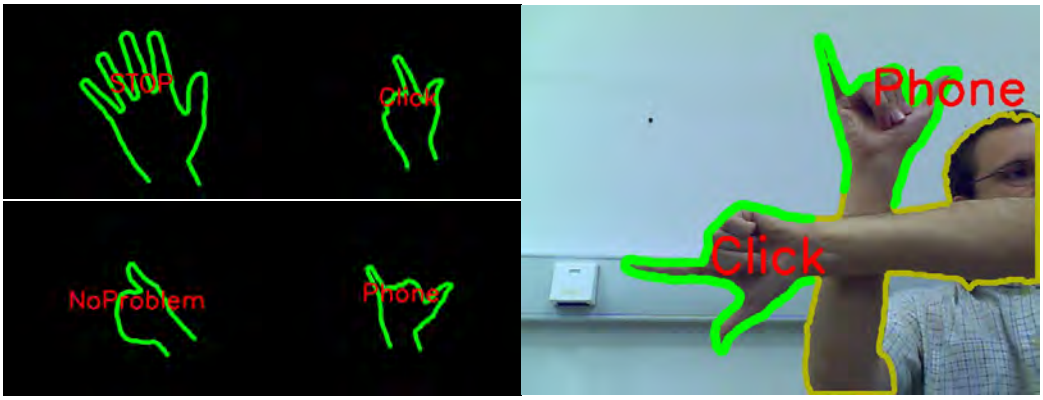
---

1. Introduction

Shape matching is a fundamental problem in computer vision and pattern recognition. It amounts to developing computational methods for comparing shapes that agree as much as possible with the human notion of shape similarity. The problem has significant theoretical interest and a wide range of applications, including, but not limited to object detection and recognition, content based retrieval of images and image registration.

To perform shape matching, most of the existing methods [12, 5, 8, 9, 3, 20, 7, 2] define shape representations and descriptors which are then compared through appropriately selected methods and metrics. The quality of the shape matching process depends on whether its final outcome agrees with human judgment. In this paper, we are interested in the particular case of 2D shapes that can be represented as binary images depicting foreground objects over their background.

Shape matching is a very challenging problem. Shapes to be matched are typically the result of some kind of segmentation process which, being imperfect, may introduce a considerable amount of noise that needs to be tolerated. In most of the cases, arbitrary differences in scale and orientation should not affect the matching process. Due to viewpoint dependencies and shape articulations and deformations, different 2D image projections of the shape of the same 3D object



(a) (b)

Figure 1: The four prototype silhouette parts in (a), need to be matched with the yellow, closed contour in (b). In (b), it is shown which of the four prototypes matched with parts of the closed contour and at which positions the best matches were achieved, based on the proposed partial shape matching method.

may differ considerably. Further complications are caused by occlusions which force shape matching to be based on partial evidence. In this particular case, the best matching of an open contour with part of a closed contour needs to be established [6, 11]. Last but not least, in realistic settings, all of the above complicating factors do not appear in isolation, but contribute collectively to increasing the complexity of the matching problem.

In the context of this work, we are interested in addressing the 2D shape matching problem by simultaneously considering all the above complicating factors. Consider, for example, Fig. 1(a) which shows four prototype silhouettes[2] (the green open contours) corresponding to parts of the outline of a human hand.

[2]The terms "silhouette" and "contour" are used interchangeably in this paper to denote the 2D outline of the shape of an object.

Given another, possibly scaled, rotated and deformed silhouette (the yellow closed contour in Fig. 1(b)) which might be the result of some segmentation process, we are interested in determining the best match between a part of it and the prototypes of Fig. 1(a). It can be verified that all the aforementioned difficulties may contribute to complicating this 2D partial shape matching problem. As stated in [11], none of the currently available, state of the art shape matching techniques provides solutions to all of these problems.

Towards the solution of this challenging problem, our contribution is threefold. First, we propose a novel descriptor as a means of local, 2D shape representation. The proposed descriptor is, by construction, scale and rotation invariant. Moreover, it tolerates substantial shape articulations and deformations. Second, we introduce a method for non-uniform sampling of a given 2D contour that decides *where* shape descriptors should be computed. The rationale behind this spatially non-uniform contour sampling method is to provide scale-dependent representations of a silhouette. Being scale dependent, the contour sampling method automatically produces the same number of shape descriptors[3] in scaled and rotated versions of the same contour. Third, we propose a variant of an existing dynamic-programming based matching technique [16] that accomplishes global 2D shape matching based on the computed shape descriptors. The key novelty in this variant is its ability to handle partial matching. Thus, matching of a source, open contour to the best matching part of another target, closed contour can be established. On

[3]Up to quantization errors.

top of a distance measure between shapes, the proposed method provides, as a byproduct, an alignment of the silhouette part to the complete silhouette (shown in Fig. 1(b)). Although primarily developed for matching open to closed contours, the proposed scheme treats the matching of two closed contours as a special case.

Experimental results have been obtained for contour matching (open to closed and closed to closed) in benchmark data sets but also in datasets that have been compiled in the context of this study. The results demonstrate that the proposed approach outperforms existing methods and is capable of dealing with the shape matching problem in challenging situations.

1.1. Related work

Before proceeding with a detailed description of our approach to partial shape matching, we briefly review existing approaches to the problem.

Shape matching is a problem that has been the focus of a lot of research. Loncaric in [13] adopts three different classifications proposed by Pavlidis in [14]. Shape matching methods can be either *boundary* or *global*, depending on whether they exploit only the silhouette or also the interior of the shapes. A second classification is based on whether the shape matching method computes a similarity measure between the compared shapes (*numeric* methods) or an alignment of the shapes (*non-numeric* methods). Shape matching methods can also be *information preserving* or not, depending on whether the used representations permit the recovery of the original shape.

A number of shape matching techniques are based on some kind of shape

skeletonization. Torres and Falcäo [7, 18] compute image skeletons at multiple scales and use them to detect salient points on the contour of the shape. Sebastian et al [17] present a technique that is based on the notion of shock graphs. Each shape can be considered as the resulting disturbance of a set of singularities (shocks) inside a fluid. Shapes that possess the same shock graph topology are considered equivalent. This is verified through a polynomial time, global optimization algorithm that performs graph comparison/matching.

Instead on relying on shape skeletal points, some other global methods are based on the representation and the properties of all interior points of a certain shape. Gorelick et al [10] propose the characterization of each interior point of the shape by the average distance that a random walker will travel before reaching it, assuming that it started at a point of the shape's silhouette. Ebrahim et al [8] present a method that transforms the raster of each shape to a one-dimensional signal according to the occurrence of shape points on a Hilbert curve. This signal is then smoothed by keeping the largest coefficients of a wavelet transform.

The category of methods most relevant to the proposed one are those that represent and match shapes based on their contour points. The general strategy is to extract information concerning the points of the shape's silhouette and then match the extracted descriptions. In [4], Basri et al propose a method to estimate shape similarity based on both part articulation and local deformation cost. Backes et al [3] use as descriptor the distribution of the distances between points on the boundary of the shape. They propose two different distributions as descriptors and use them for shape classification with the aid of Linear Discriminant Analysis

(LDA). Adamek and O'Connnor [1] propose a multiscale representation of shape silhouettes that is matched with the use of dynamic programming. Initially, they apply different levels of smoothing on the shape contours. They further process this result by applying a transformation that detects concave and convex parts of the contour. They then proceed to match such shape descriptions with the use of an appropriate comparison distance, and dynamic programming. Arica and Vural [2] propose a simple geometric transformation for the purpose of shape description. They compute the bearing angle of three consecutive contour points for variable offsets between these points. The values obtained for each point of the contour and for various offset sizes are considered as random variable measurements, the moments of which form the proposed descriptor. The matching of the descriptions is performed using dynamic programming.

Several methods try to align silhouettes by exploring a space of geometric transformations. For example, Wu et al [20] employ genetic algorithms to search over the space of affine transformations. They describe representation and re-sampling schemas suitable for the specific application, and propose variations to improve the time and accuracy of shape matching. Felzenszwalb et al [9] represent each silhouette as a tree, with each level representing a different description level. The root of the tree represents a properly selected cut on the curve while the left and right children represent cuts on the occurring sub-curves. They propose an iterative matching scheme that can be efficiently solved using dynamic programming. They proceed with the formulation of an algorithm that can locate query shapes in real-world color images. A very interesting shape descriptor is the

so called shape context, introduced by Belongie et al in [5]. The main idea is that the local distribution of points for the purpose of local description is well captured using a log-polar histogram. In its original form, selected points from the contour of an image were used as centers, and the distribution of the other contour points around each center was used as a descriptor vector. Shape contexts capture the fact that local features play a more important role than more distant ones for the purpose of local matching. Effectively, the employed logarithmic function weighs more the proximate features, and less the more distant ones.

An interesting variant is presented in [12] where the goal is to improve shape matching by exploiting the articulated nature of many common shapes. The authors suggest that the distances and angles between contour points should be measured only inside the closed contour of a figure. This means that articulations are handled quite well by this type of description. The key idea is that the inner distance, in contrast to the classic Euclidean distance, is invariant to articulation.

Cui et al [6] propose a method to efficiently match whole-to-part and part-to-part shapes. They choose the integral of absolute curvature as shape descriptor, and use the normalized cross correlation for matching parts of the occurring curves. The method is rotation, scale and translation invariant and tolerates moderate amounts of noise. Latecki et al in [11] propose a method for shape matching based on dynamic programming. A particularly interesting aspect of this method is that it addresses the partial shape matching problem. More specifically, the method is able to establish the best match between an open silhouette and a part of a closed silhouette. The method combines the strengths of Dynamic

Time Warping [15] and the Longest Common Subsequence technique [19] in another dynamic programming based technique coined Minimum Variance Matching (MVM). Local tangents to silhouettes are used for the purpose of shape description.

2. Proposed approach

The proposed matching method employs only boundary points for shape description. A similarity measure between shapes is computed, together with shapes alignment. In the following sections, we describe the proposed shape representation and the procedure used to compare and match shapes.

2.1. Shape representation

The proposed descriptor is defined on shape silhouettes, i.e., the external contour of each input shape [2, 12]. At a first step, a given silhouette is uniformly sampled and one descriptor is computed on each point sample. The descriptor consists of the distances of the particular point from the closest silhouette points, along equiangular directions defined in the inner part of the shape. The mean value of these distances provides an estimate of local scale. This gives the possibility to resample the silhouette so that the smaller the scale, the denser the resampling of the silhouette becomes. Additionally, as it will become more clear in Sec. 2.1.2, this non-uniform sampling that automatically adapts to local scale, makes shape description independent of global scale changes. Shape descriptors are then recomputed at the newly estimated silhouette samples. The set of all

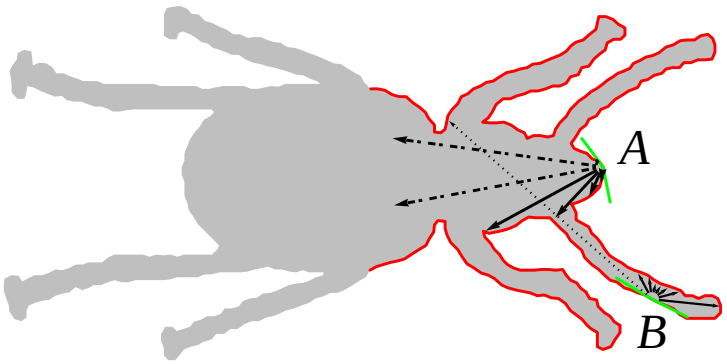


Figure 2: Example descriptors computed on points of a silhouette.

these descriptors constitutes the global representation of a shape. The following sections describe these ideas in more detail.

2.1.1. The proposed local shape descriptor

The fundamental idea behind the proposed descriptor lies on measuring the distance of a certain silhouette point from the closest points of the same silhouette, along properly defined directions. Let s_i be a point on a silhouette s for which a local shape descriptor should be computed. We define k rays starting at s_i. The directions $\theta_1(s_i)$ and $\theta_k(s_i)$ of the first and the last ray, coincide with the directions of the lines (s_i, s_{i-1}) and (s_i, s_{i+1}), respectively. Indexing with i is modulo the number of silhouette points. We ensure that $0 \leq \theta_k(s_i) - \theta_1(s_i) < 2\pi$ by adding integer multiples of 2π to the values $\theta_1(s_i)$ and $\theta_k(s_i)$ so that the intermediate values represent directions pointing towards the inner part of the shape. The angular separation of two consecutive directions is then defined as $(\theta_k(s_i) - \theta_1(s_i))/(k-1)$. Figure 2 visualizes this process for a few points on the silhouette s.

Starting at s_i, we extend a straight line along each of the k directions spec-

ified so far, until it meets the silhouette s for the first time. The length l of this line segment is then recorded. Care must be taken so that quantization errors are avoided. The above described process results in a k-dimensional vector $d(s_i) = \{l_{i1}, l_{i2}, ..., l_{ik}\}$ of distances, as visualized in Fig. 2. In the case of closed contours, it is guaranteed that each of the defined rays will intersect the silhouette and, thus, the l_{ij} will be finite numbers. For open contours, it is possible that an intersection does not exist. Such rays are marked with a special label l_U (e.g., the dashed vectors of the descriptor at point A in Fig. 2).

The values $l_{ij} \neq l_U$ in $d(s_i)$, $1 \leq j \leq k$, are further filtered for outlying values. More specifically, the median value m of such l_{ij}s, is computed. For $\beta > 0$, each $l_{ij} > \beta m$ is flagged as an outlier, by assigning it the label l_∞ (e.g., the dotted vector of the descriptor at point B in Fig. 2). An empirical choice of $\beta = 15$ was made, ensuring that only very large distances are discarded. Vector $d(s_i)$ effectively constitutes the proposed descriptor for local shape appearance at point s_i.

An important issue is related to how the "inner" part of a shape is defined for an open contour. In practice, this is handled by defining open contours as parts of some closed contour, for which the inner part is unambiguously defined. Additionally, it is worth noting that, by construction, the employed descriptor treats unevenly the exterior and the interior of a shape. If the entire open contour is concave, then all coordinates of the corresponding descriptors have l_U values which are uninformative and useless for matching. The case of strongly concave parts can be handled by "reverting" the descriptor so as to take into account the exterior

part of the shape, as opposed to the interior part that is now considered. Thus, uninformative descriptors will only result in the case of low curvature, almost straight contours. Fortunately, this is a rather uninteresting case because, also due to scale invariance, such structures could fit anywhere on a closed contour.

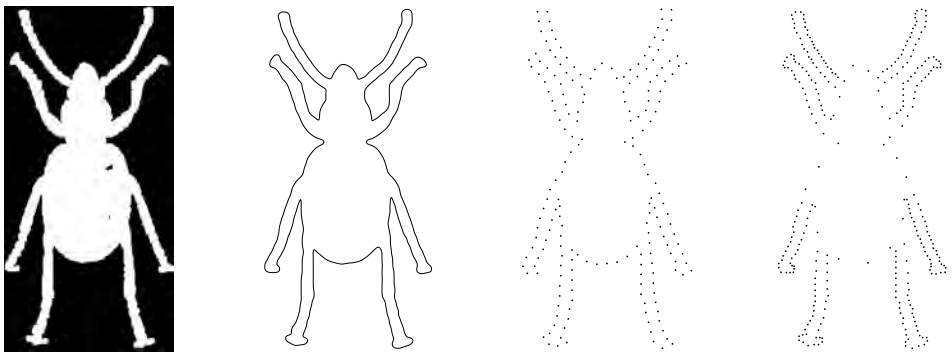
The proposed descriptor shares some similarity with the one proposed in [12]. Both of them essentially measure distances in the interior of the shapes. However, while [12] considers all possible paths in the interior of the shape, we consider only straight, unobstructed paths. The approach in [12] reflects a more global choice for the description of the shape, while the one proposed here is better suited to local shape description. Because of this fundamental difference, the descriptor proposed in [12] cannot be used for local scale estimation that drives non-uniform contour sampling, or for partial shape matching.

2.1.2. Local scale estimation

An estimate of the local scale $\mathcal{S}(s_i)$ of a silhouette point s_i can be computed as the mean of the finite distances of a descriptor:

$$\mathcal{S}(s_i) = \frac{1}{|F_i|} \sum_{j \in F_i} l_{ij}, \tag{1}$$

where $F_i = \{l_{ij} \in d(s_i) : l_{ij} \neq l_\infty \wedge l_{ij} \neq l_U\}$ and $|.|$ denotes set cardinality. The intuition behind the particular representation of local scale is that $\mathcal{S}(s_i)$ is indeed proportional to the level of detail of local shape. As an example, $\mathcal{S}(A)$ is expected to be much larger than $\mathcal{S}(B)$ in Fig. 2. This estimate of local scale is used in Sec. 2.1.4 to guide the non-uniform sampling of a particular silhouette.



(a) (b) (c) (d)

Figure 3: Shape preprocessing steps. An example input binary image is shown in (a) and the smoothed contour in (b). The fixed-rate and non-uniformly subsampled silhouettes are shown in (c) and (d), respectively.

2.1.3. Contour extraction and preprocessing

The input to the proposed method is a binary image containing a foreground object (e.g., Fig. 3(a)). The silhouette of this object is extracted (Fig. 3(b)) and traversed in some predefined order. Both shape description and matching require consistency with respect to this order. Therefore, lists of contour points are reversed, depending on the sign of the area covered by a silhouette and the convention that this must be positive. We proceed by performing a fixed subsampling of the silhouette by retaining one out of r pixels. In order to eliminate small amounts of quantization noise, prior to subsampling, Gaussian smoothing of the silhouette is performed. Figure 3(c) shows the fixed rate subsampling of a smoothed version of the contour shown in Fig. 3(b).

Let s be the sequence of smoothed silhouette points and $\overline{s}$ the sequence of points resulting from the fixed interval subsampling of s. Shape description is

performed on an image raster, so s is rasterized at the same resolution as the input image. The descriptor presented in Sec. 2.1.1 is computed at all points of $\overline{s}$. A byproduct of this process is the local scale estimates $\mathcal{S}(\overline{s}_i)$ for all points of $\overline{s}$.

To be able to perform a scale-dependent subsampling of the contour (see Sec. 2.1.4) we need to have an estimate of local scale for all points in s. We achieve this by interpolating scale values already computed for points in $\overline{s}$. Interpolation has been selected for reasons of computational efficiency. Alternatively, we could have computed descriptors and scale estimates for all points in s. Experiments have demonstrated that the increased accuracy thus obtained, is not worth the associated extra computational overhead.

2.1.4. Scale adjusted sampling and shape representation

Given the local scale estimates $\mathcal{S}(s_i)$ for each point s_i in s, we can sample s in a local, scale-dependent way. More specifically, a local sampling offset o is chosen proportional to the local scale, or, equivalently, inversely proportional to local detail. The resampling process is iterative. We start at an arbitrary point on s, adding it as the first point of the final contour sampling $\hat{s}$. We then compute the offset $o(s_i) = a\,\mathcal{S}(s_i)$, which is the distance (in pixels) on s to the next point to be appended to $\hat{s}$. The constant a is empirically estimated and controls the total number of samples per silhouette. We iterate this process until the whole contour is sampled (see Fig. 3(d)).

Once contour resampling has been performed, the descriptors defined in Sec. 2.1.1 are computed again on the points of $\hat{s}$. For each new descriptor, the computed dis-

tances are normalized by the mean of the finite distances, i.e., the local scale $\mathcal{S}(\hat{s}_i)$, as defined in Eq.(1). The resulting k-dimensional vector $d(\hat{s}_i)$ is the description for the point $\hat{s}_i$ and the concatenation of such vectors for all i, $0 \leq i \leq N_{\hat{s}}$, constitutes the global representation of a contour.

2.2. Shape matching

The goal of the matching step is to estimate the similarity of two given contours based on the descriptors already computed on them. This is achieved by establishing correspondences between contour points. We treat contours as strings of descriptors computed as described in Sec. 2.1. Closed contours correspond to cyclic strings. Correspondences between symbols are established through string alignment with a method that is based on Dynamic Time Warping (DTW) [15].

More specifically, we consider a source silhouette s represented as an ordered set of N_s descriptors $d(s_i)$ that is to be matched with a target silhouette t represented as an ordered set of N_t descriptors $d(t_j)$. s might be an open or closed contour, while t is always a closed one. According to the established terminology, matching s with t amounts to identifying a set of elementary operations (i.e., symbol replacements, insertions and deletions) that are required to transform s to t. Each such elementary operation is associated with a cost that depends on the pair of symbols to which it is applied. The set of operations that results in the minimum sum of individual costs represents the best possible alignment between the two strings. Additionally, the minimum value of this objective function can be used as an estimate of the dissimilarity of the compared strings.

Intuitively, the replacement of a descriptor $d(s_i)$ of s with the descriptor $d(t_j)$ of t is associated with a replacement cost $R(s_i, t_j)$[4] that reflects the cost of matching $d(s_i)$ with $d(t_j)$. Insertion can be interpreted as the expansion of a point in s so that it corresponds to more than one points in t and is associated with a cost $E(s_i, t_j)$. Symmetrically, deletion can be interpreted as the contraction of several points in s that need to be aligned with a single point on t and is associated with a cost $C(s_i, t_j)$. Essential to the definition of the replacement, insertion and deletion costs is the definition of a distance measure $\mathcal{D}(x, y)$ between two shape descriptors x and y.

2.2.1. Comparing shape descriptors

Let $d(s_x) = \{l_{x1}, l_{x2}, ..., l_{xk}\}$, $d(t_y) = \{l_{y1}, l_{y2}, ..., l_{yk}\}$ be two shape descriptors at points s_x and t_y, respectively. The goal is to establish a distance measure $\mathcal{D}(s_x, t_y)$ between the descriptors $d(s_x)$ and $d(t_y)$. $\mathcal{D}(s_x, t_y)$ is defined based on the pairwise comparison of the descriptors' coordinates, according to:

$$\mathcal{D}(s_x, t_y) = \frac{1}{k} \sum_{i=1}^{k} \Delta(l_{xi}, l_{yi}), \tag{2}$$

where $\Delta(.,.)$ is a function that compares its arguments and returns a value in the range $[0..1]$. As it has been described in Sec. 2.1.1, each of the k dimensions of the descriptors may contain arithmetic, but also categorical values (the labels l_∞ and l_U). Thus, the definition of $\Delta(p, q)$ entails a number of cases depending on

[4]Formally, this should have been written as $R(d(s_i), d(t_j))$. We choose to drop the descriptor indicator $d(.)$ for the sake of notational brevity.

the type of dimensions p and q compared in Eq. (2):

$$\Delta(p,q) = \begin{cases} \frac{|p-q|}{\max\{|p|,|q|\}} & \text{if } p,q \notin \{l_\infty, l_U\} \\ 1 & \text{if } (p = l_\infty \wedge q \notin \{l_\infty, l_U\}) \vee \\ & \quad (q = l_\infty \wedge p \notin \{l_\infty, l_U\}) \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The first branch of Eq.(3) states that if p and q are finite distances, $\Delta(p,q)$ will vary in the range $[0..1]$ depending on their relative difference. The second branch states that there is a total mismatch between finite distances and outlying ones. Finally, the third branch states that in all other cases (distances are either outlying or undefined due to open contours), $\Delta(l_{xi}, l_{yi}) = 0$, thus signifying a perfect match.

2.2.2. Defining DTW costs

We proceed with defining the replacement, insertion and deletion costs used for shape matching. All DTW costs are defined based on $\mathcal{D}(s_i, t_j)$. Because of the non-uniform sampling, a point that represents a large portion of the contour must be weighted more compared to another point which represents a smaller part. To achieve this, the offset $o(x_i)$ (see Sec. 2.1.4) divided by the total length N_x of a contour x is used as a weighting factor, i.e., $w(x_i) = o(x_i)/N_x$. For the closed contour case, the deletion cost is defined as $C(s_i, t_j) = w(s_i)\mathcal{D}(s_i, t_j)$, and the insertion cost is defined as $E(s_i, t_j) = w(t_j)\mathcal{D}(s_i, t_j)$. The replacement cost $R(s_i, t_j)$ is then defined as $R(s_i, t_j) = \max\{C(s_i, t_j), E(s_i, t_j)\}$.

The definition of costs for the open contour case has an extra complication. The open contour has a known length, but we do not know where on the closed contour the open contour will be matched. This means that we cannot account correctly for the closed contour scale. We proceed with weighting the costs only with the known length of the closed contour. Thus, both insertion and deletion costs are set in this case to the value $E(s_i, t_j) = C(s_i, t_j) = w(t_j)\mathcal{D}(s_i, t_j)$. The replacement cost is set to half this value.

2.2.3. Matching algorithm

As detailed in Sec. 1.1, the efficient computation of elastic matching through dynamic programming techniques has been frequently applied to shape matching. Techniques based on Dynamic Time Warping (DTW) match points on one silhouette to points on another by finding the shortest path through a graph, the nodes of which encode the similarity of respective point pairs. Assuming that both silhouettes consist of n points, the fastest available algorithm [16] for solving this problem has a complexity of $O(n^2 \log n)$. Essentially, the technique presented in [16] provides a mechanism to avoid the exhaustive consideration of the alignments of two cyclic strings for every possible initial match. We employ this algorithm to match closed to closed contours. More precisely, we implemented and employed a generalization of [16] that makes it suitable for matching strings of unequal length. The matching costs were defined as described in Sec. 2.2.2.

Matching open contours against parts of closed contours is similarly treated, but requires additional attention. Assume an open contour s that needs to be

matched with part of a closed contour t. We capitalize on the observation that this can be computed as the best match between s and any substring of length less than N_t from $t \oplus t$, i.e. the concatenation of t with itself. The duplication of the target string ensures that the source string s can be matched with the target string without having to wrap around at string ends. Then, the problem is transformed into one of searching for minimum cost paths in a directed acyclic graph and employ dynamic programming techniques that prune the search space by exploiting previously computed paths. Care is taken so as to enforce the constraint that the source, open string s cannot match a substring of $t \oplus t$ that has length greater than N_t; This would mean that s wraps around the cyclic string t.

3. Experimental results

The proposed approach for 2D shape matching has been validated by several experiments. The experiments can be grouped into two categories, one that assesses the performance of the proposed method in matching open to closed contours and another one that concerns the matching of closed contours.

3.1. Matching open to closed contours

The experiments for matching open with closed contours have been performed based on the MPEG7 Core Experiment CE-Shape-1 dataset [11] as well as in the context of its application to human upper body detection and hand posture recognition.

3.1.1. Experiments on the MPEG7 dataset

The full MPEG7 dataset consists of 70 different classes of objects, each containing 20 class representatives, resulting in 1400 different shapes. In the experiment reported in [11], 5 shapes out of each shape class have been used, for a total of 350 shapes. Then, 10 query open contours are selected and matched with each shape in the database yielding similarity ranks. The important difference in our experimental setup is that we did not restrict the shape database as in [11] to 350 shapes, but instead, we used the full set of 1400 shapes. We reproduced the 10 open contour queries (top 10 rows of Figs. 4, 5). To investigate a richer set of possible types of open contour queries, we selected 8 more (bottom 8 rows of Figs. 4 and 5), including different parts of the same silhouettes. The shapes retrieved using the proposed method for the total of 18 queries are shown in Fig. 4. The first column in Figs. 4 indicates the queries superimposed on the shape used to define it. The rest 10 columns depict retrieved shapes (with the matched part highlighted) in the order of decreasing similarity. For the purposes of quantitative and comparative evaluation, we implemented the MVM method presented in [11]. We converted each silhouette of the database to a sequence of 100 tangent values using the Discrete Curve Evolution method, as indicated in [11]. The actual queries were obtained from these sequences by taking the cyclic subsequence that best corresponded to the depicted contour part. Similarly to Fig. 4, the 10 best shapes retrieved using the MVM method for the set of 18 queries are shown in Fig. 5. As can be verified, the proposed method retrieves shapes that are perceptually more relevant to the queries.

	Queries of [11]	Our queries	Aggregate
MVM	24.0%	23.7%	23.9%
Our approach	52.5%	65.0%	58.1%

Table 1: Comparison between the MVM and the proposed method for partial shape matching.

In order to quantitatively compare the two methods, we counted the number of retrieved images that belong to the class of the image used to define the query contour, in the top 40 matches. The percentage of correct retrievals is the so called bulls-eye score. We performed this test for the defined 18 queries. Table 1 summarizes the obtained results and demonstrates that the proposed method performs substantially better compared to the MVM method.

3.1.2. Body parts matching

The proposed method was also applied to human upper body detection and hand posture recognition. More specifically, the 10 open contours representing parts of human postures corresponding to the upper human body (see Fig. 6) have been manually defined and then matched with full contours resulting from background subtraction. Snapshots of the results obtained are shown in Fig. 7. Having annotated the prototype, partial contours with human joins location information, it becomes possible to localize them in the current frame. The skeleton of the upper body (appearing in red in Fig.7) can then be computed based on the locations of these points. Interestingly, the method succeeds to compensate for the large deformations of human body and the noise that is inevitably introduced because

of the color-based background subtraction. Additionally, since the prototype contours are *parts* of silhouettes, the recognition of the upper-body posture becomes invariant to the configuration of legs. Thus, only 10 prototypes suffice to encapsulate the frontal variability of the upper body. Additionally, the method accounts for inter-person variability since, as shown in Fig.7, it succeeds to match prototypes in the silhouettes of different persons.

Similarly, 9 open contours corresponding to 9 hand postures have been manually defined (see Fig.8), and then matched against performing hands. Snapshots of the results obtained are shown in Fig. 9. By associating hand postures with semantic information, we can robustly recognize them in videos, with only a single prototype per hand posture and despite considerable rigid transformations and non rigid deformations. The bottom-right image shows two different input prototypes matching simultaneously with different parts of the closed contour, permitting the interpretation of bimanual postures performed by occluding hands.

Representative videos of the results obtained in both experiments, are available online[5].

3.2. Matching closed to closed contours

Despite that the proposed solution has been developed for the problem of partial shape matching, it also proves itself very competent in the problem of matching closed contours. We performed an exhaustive classification test that employed each of the 1400 images of the full MPEG7 dataset as a query object. Table 2

[5]http://www.ics.forth.gr/∼argyros/research/partialshapematching.html

Bull's eye score	Proposed	[12]	[8]	[9]
No GT [21]	83.4%	85.4%	88.3%	87.7%
With GT [21]	89.9%	91.0%	-	-

Table 2: Performance of existing methods for closed shapes matching, with and without graph transduction [21].

presents the bull's eye score of several existing methods on this problem. It can be verified that the proposed method has a comparable performance to methods specifically designed for closed shape matching. Table 2 also includes the performance of the proposed method and IDSC [12], after graph transduction [21].

Figure 11 presents quantitative results on the performance of the proposed method on the bulls eye test over the MPEG7 dataset. The performance of another state of the art method [12] is also provided for comparison. For this exhaustive shape classification experiment, we also computed the confusion matrix which is shown in Fig. 12. Similarity scores have been obtained before the application of graph transduction, so that the merit of the proposed method can be assessed without the improvement introduced by it. The block diagonal structure of this image illustrates the accuracy of the proposed method on shape classification.

Representative qualitative results from this experiment are shown in Fig. 10. The complete view of the results is not included in this paper due to size considerations, but is available online[6].

[6]http://www.ics.forth.gr/∼argyros/research/partialshapematching.html

3.3. Implementation notes

The parameters for the presented experiments were kept constant for all data sets. The initial subsampling interval was set to $r = 20$. The descriptor's dimension was set to $k = 16$. Slightly better results were obtained using $k = 32$ but with a disproportionate increase in computation time. The parameter a for the non-uniform contour sampling was set to $a = 0.3$. The proposed approach does not inherently handle mirroring. Thus, both original and mirrored shapes were matched and then the lowest of the two scores was kept.

The whole matching process runs on commodity hardware at a frame rate of 1 to 20 fps, depending on parameters such as image resolution, sampling rates, number of descriptor rays and number of prototypes. For comparison, a similar computational performance was achieved in our implementation of MVM [11].

4. Summary

This article presented a novel solution to the problem of partial shape matching. Partial and full shape matching are treated in a unified way that proves very competent compared to existing methods. The key ideas and main contributions of this work lie in the proposed shape descriptor, the scale dependent sampling, and the cost assignment for descriptor matching. The shape descriptor is robust under significant deformations due to articulation, efficient to compute, and captures sufficient information to enable high performance. The proposed contour sampling method makes silhouette descriptions independent of scale. More importantly, it allows uneven scaling of different parts of a silhouette (as for example

in an affine transformation) to be treated in a consistent way. From a qualitative point of view, the proposed cost assignment and shape matching, in most cases, provide results that are intuitive. Finally, extensive quantitative and comparative experiments demonstrated the effectiveness of the proposed method compared to existing ones.

Acknowledgements

This work was partially supported by the IST-FP7-IP-215821 project GRASP.

References

[1] T. Adamek and N.E. O'Connor. A multiscale representation method for nonrigid shapes with a single closed contour. *IEEE Trans. on Circuits and Systems for Video Technology*, 14(5):742–753, May 2004.

[2] N. Arica and F.T.Y. Vural. A perceptual shape descriptor. *16th ICPR*, 3:375–378, 2002.

[3] Andr Ricardo Backes, Dalcimar Casanova, and Odemir Martinez Bruno. A complex network-based approach for boundary shape analysis. *Pattern Recognition*, 42(1):54 – 67, 2009.

[4] R. Basri, L. Costa, D. Geiger, and D. Jacobs. Determining the similarity of deformable shapes. *Vision Research*, 38:135–143, 1998.

[5] S. Belongie, G. Mori, and J. Malik. Matching with shape contexts. In *IEEE Workshop on Content-based access of Image and Video-Libraries*, page 20, 2000.

[6] M. Cui, J. Femiani, J. Hu, P. Wonka, and A. Razdan. Curve matching for open 2d curves. *Patt. Recogn. Letters*, 30(1):1 – 10, 2009.

[7] R. da S. Torres and A.X. Falco. Contour salience descriptors for effective image retrieval and analysis. *IVC*, 25(1):3 – 13, 2007.

[8] Y. Ebrahim, M. Ahmed, W. Abdelsalam, and S.C. Chau. Shape representation and description using the hilbert curve. *Patt. Recogn. Lett.*, 30(4):348–358, 2009.

[9] P.F. Felzenszwalb and J.D. Schwartz. Hierarchical matching of deformable shapes. *In CVPR*, pages 1–8, June 2007.

[10] L. Gorelick, M. Galun, E. Sharon, R. Basri, and A. Brandt. Shape representation and classification using the poisson equation. *IEEE Trans. on PAMI*, 28(12):1991–2005, 2006.

[11] L.J. Latecki, V. Megalooikonomou, Q.A. Wang, and D. Yu. An elastic partial shape matching technique. *Patt. Recogn.*, 40(11):3069–3080, 2007.

[12] H. Ling and W.W. Jacobs. Shape classification using the inner-distance. *Trans. on PAMI*, 29(2):286–299, Feb. 2007.

[13] S. Loncaric. A survey of shape analysis techniques. *Patt. Recogn.*, 31:983–1001, 1998.

[14] Theodosios Pavlidis. A review of algorithms for shape analysis. *Comp. Graphics and Im. Proc.*, 7(2):243–258, 1978.

[15] H. Sakoe and S. Chiba. A dynamic programming approach to continuous speech recognition. In *Proc. of the 7th Int'l. Congress on Acoustics, Budapest*, 1971.

[16] F.R. Schmidt, D. Farin, and D. Cremers. Fast matching of planar shapes in sub-cubic runtime. In *ICCV*, pages 1–6. IEEE, 2007.

[17] T.B. Sebastian, P.N. Klein, and B.B. Kimia. Recognition of shapes by editing shock graphs. In *ICCV*, pages 755–762, 2001.

[18] R.S. Torres, A.X. Falcao, and L.F. Costa. Shape description by image foresting transform. *14th Int'l Conf. on Digital Signal Processing*, 2:1089–1092 vol.2, 2002.

[19] Michail Vlachos, Marios Hadjieleftheriou, Dimitrios Gunopulos, and Eamonn Keogh. Indexing multi-dimensional time-series with support for multiple distance measures. In *9th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, pages 216–225, New York, NY, USA, 2003. ACM.

[20] A. Wu, P.W.M. Tsang, T.Y.F. Yuen, and L.F. Yeung. Affine invariant object shape matching using genetic algorithm with multi-parent orthogonal

recombination and migrant principle. *Applied Soft Computing*, 9(1):282 – 289, 2009.

[21] Xingwei Yang, Xiang Bai, Longin Jan Latecki, and Zhuowen Tu. Improving shape retrieval by learning graph transduction. In *ECCV (4)*, pages 788–801, 2008.

Figure 4: Examples of partial contour matching in the MPEG7 dataset with the proposed method.

Figure 5: Examples of partial contour matching in the MPEG7 dataset with the method proposed in [11].

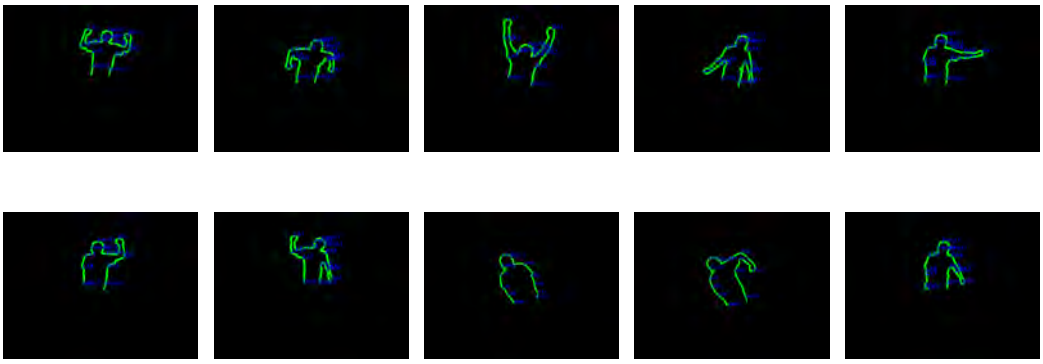


Figure 6: Prototypes used for upper human body detection.

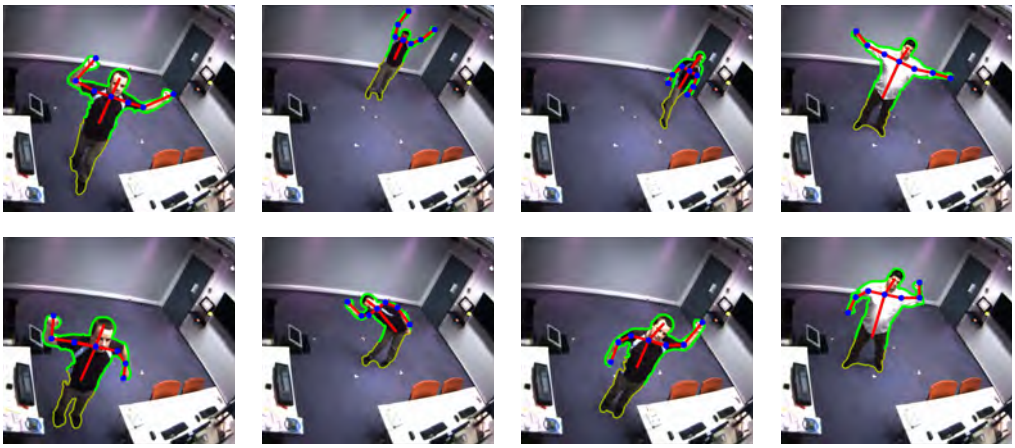


Figure 7: Sample results from the application of the proposed method for human upper body detection.

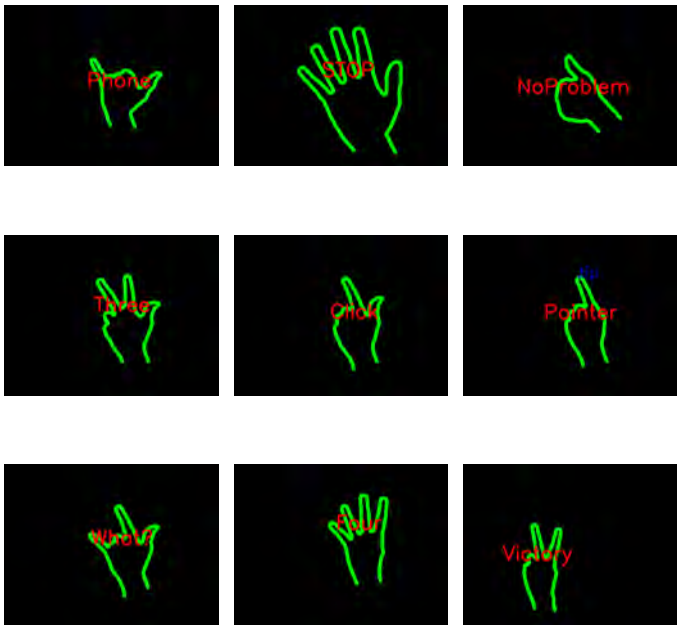


Figure 8: Prototypes used for hand posture recognition.

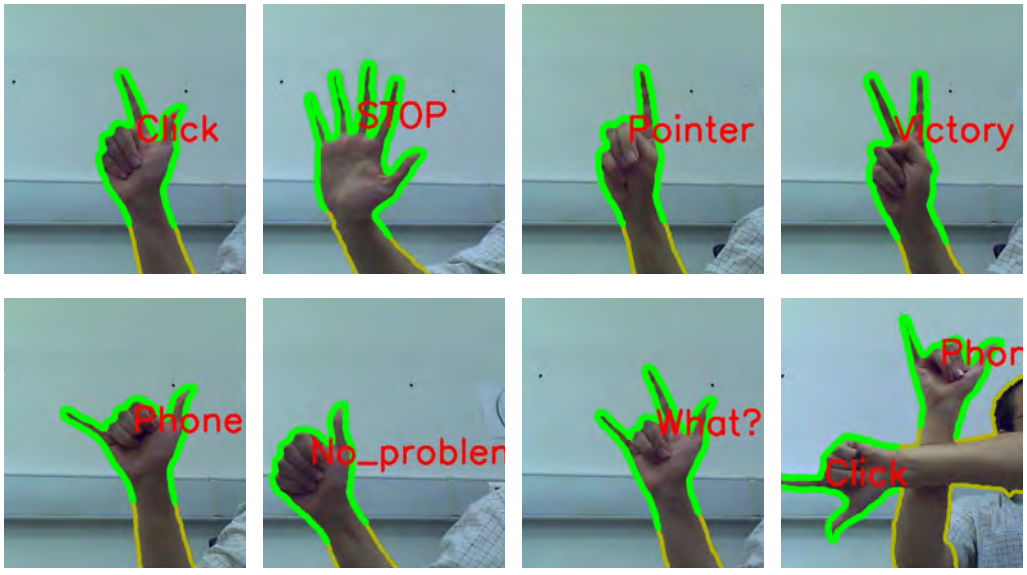


Figure 9: Sample results from the application of the proposed method for hand postures recognition.

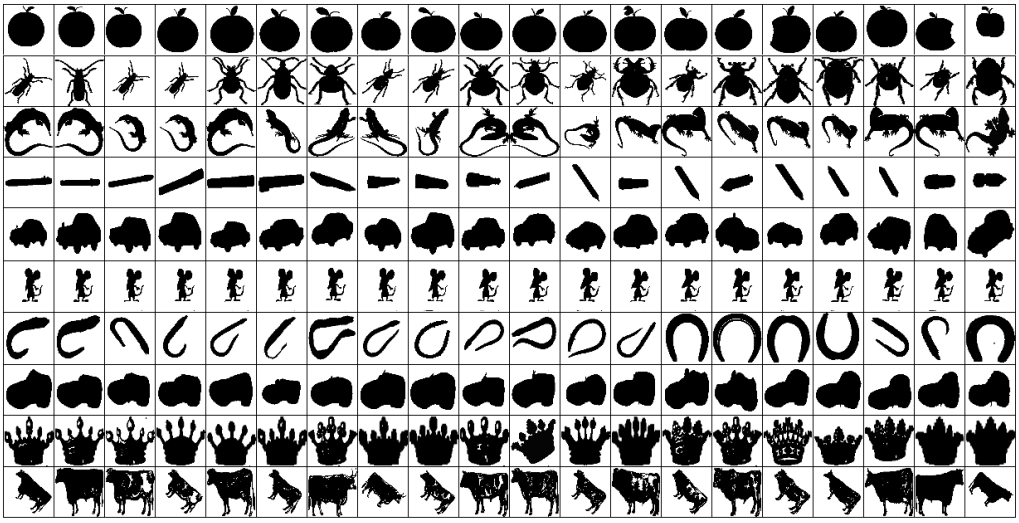


Figure 10: Characteristic results for the MPEG7 sequence. The first column shows query images. The rest of each row includes retrieved shapes in the order of decreasing similarity.

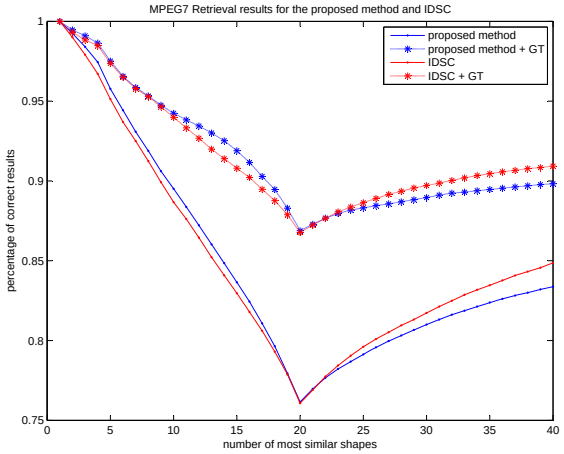


Figure 11: Performance comparison of the proposed method and IDSC [12] in the Bull's Eye test on the MPEG7 dataset. The results of each method are improved using the graph transduction technique (GT) proposed in [21] for different values of the window (W) parameter.

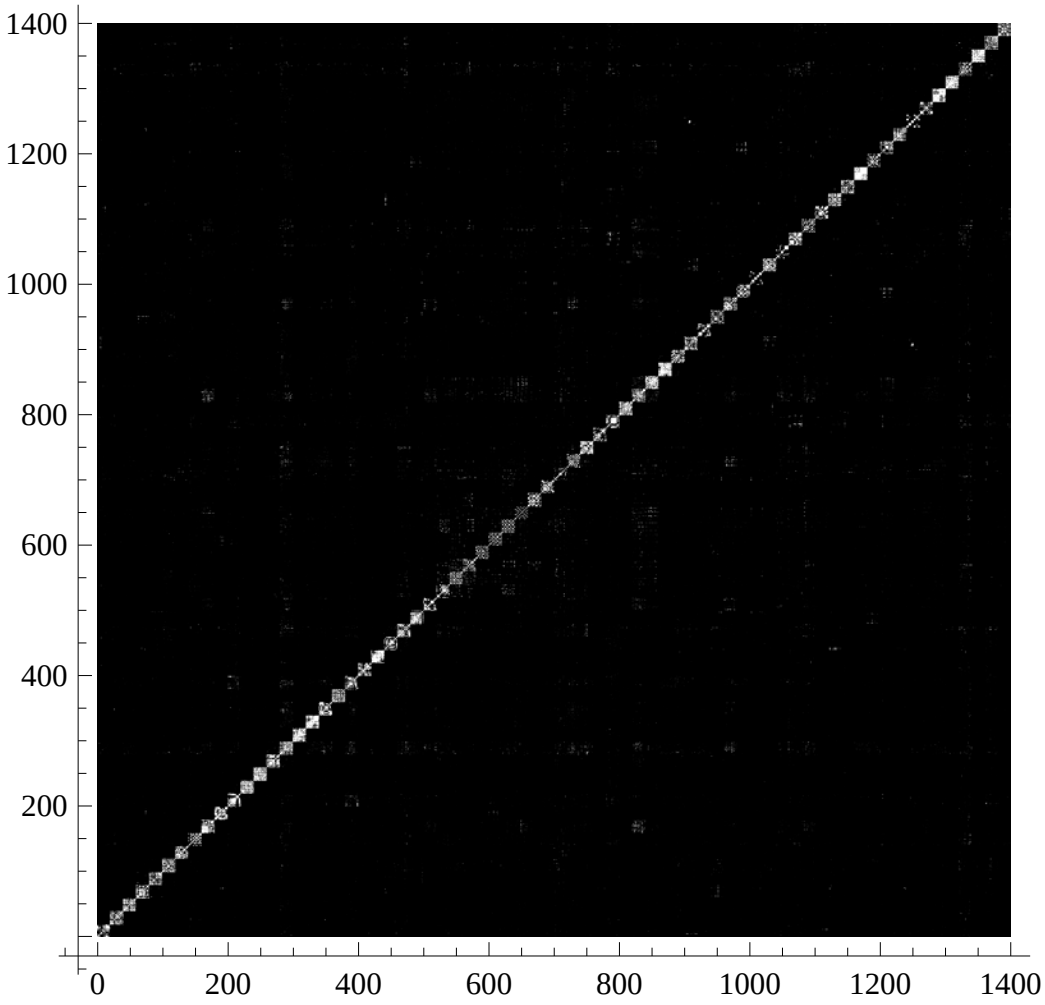


Figure 12: The confusion matrix for the exhaustive MPEG7 classification experiment.

Multiple objects tracking in the presence of long-term occlusions

Vasilis Papadourakis[a], Antonis Argyros[1b,a]

[a] *Institute of Computer Science, FORTH, Heraklion, Crete, Greece*
[b] *Computer Science Department, University of Crete, Greece*

Abstract

We present a robust object tracking algorithm that handles spatially extended and temporally long object occlusions. The proposed approach is based on the concept of "object permanence" which suggests that a totally occluded object will re-emerge near its occluder. The proposed method does not require prior training to account for differences in the shape, size, color or motion of the objects to be tracked. Instead, the method automatically and dynamically builds appropriate object representations that enable robust and effective tracking and occlusion reasoning. The proposed approach has been evaluated on several image sequences showing either complex object manipulation tasks or human activity in the context of surveillance applications. Experimental results demonstrate that the developed tracker is capable of handling several challenging situations, where the labels of objects are correctly identified and maintained over time, despite the complex interactions among the tracked objects that lead to several layers of occlusions.

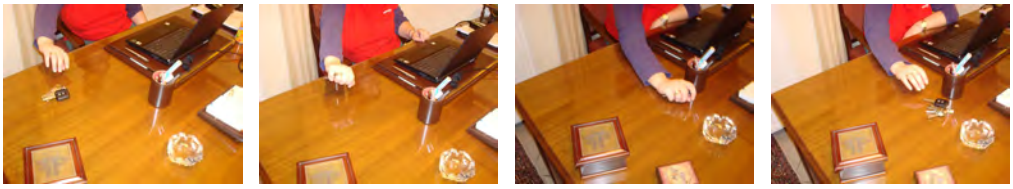
Key words: Multiple objects tracking, Occlusion, Object permanence

[1]Corresponding author: Antonis Argyros, N. Plastira 100, Vassilika Vouton, GR-700-13, Heraklion, Crete, Greece, tel.: +30 2810 391704, FAX: +30 2810 391609, argyros@ics.forth.gr.

1. Introduction

Visual tracking of multiple objects is an important problem with instances appearing in several application domains. Despite the huge amount of excellent research in the field, the effective and robust solution to the problem remains challenging in many realistic scenarios and settings. Part of the difficulty of the problem stems from the fact that even simple object interactions may result in full occlusions that last for quite long time periods. An object may totally disappear behind another object and reappear after considerable time, close to it, at a different location. As an example, consider the situation illustrated in Fig. 1 where a person grasps his keys to place them somewhere else. Once the keys are firmly grasped, they totally disappear behind the hand. When the transfer is complete, the same keys reappear. Reasoning about the activities in this scene requires the capability to associate the same label to the object seen before and after manipulation. Clearly, the problem may become much more complicated, for example in scenarios involving bimanual interaction with several objects that may (or may not) differ in shape, size, appearance, etc. Similar kinds of problems can be encountered in other applications, involving, for example, tracking individual persons in crowded scenes. In this work, we present our approach to solving this kind of tracking problems.

A lot of approaches have already been proposed for object tracking in the presence of occlusions. Huang and Essa [6], provide a very informative overview of existing approaches. According to their categorization, several of the exist-



(a) (b) (c) (d)

Figure 1: The need for handling long-term occlusions in the context of tracking. From left to right, a human hand moves towards the keys, grasps and transfers them to a different position. We are interested in a tracking framework which, without a priori information about the tracked objects, will be able to infer that the object disappearing in the second frame, is the same to the one reappearing in the fourth frame.

ing methods handle occlusions implicitly. In the work of Khan and Shah [10] for people tracking, a person is segmented into classes of similar color using the Expectation Maximization (EM) algorithm. Then, the maximization of the a posteriori probability of these classes drives frame-to-frame tracking. McKenna [13] and Marques [12] employ appearance models of tracked regions to identify people after the occurrence of occlusions but their approach provides limited support of complex object interactions. In [7], Isard introduces a Bayesian filter for tracking a potentially varying number of objects. A particle filter is used to perform joint inference on both the number of objects present and their configurations. Occlusion handling is achieved by incorporating the number of interacting persons into the observation model and inferring it using a Bayes network. Jepson et al. [8] proposes a framework for learning appearance models to be used for motion-based tracking of natural objects. The appearance model involves a mixture of stable image structure, learned over long time courses, along with two-frame motion information and an outlier process. This model is used in a motion-based tracking algorithm to provide robustness in the presence of outliers, such as those

caused by occlusions.

Several other methods have been proposed that treat explicitly the problem of tracking in the presence of occlusions. Rehg [15] describe a framework for local tracking of self-occluding motion, in which one part of an object obstructs the visibility of another. His approach uses a kinematic model to predict occlusions and windowed templates to track partially occluded objects. Brostow [3] present a method to decompose video sequences into layers that represent the relative depths of complex scenes. Activity in a scene is used to extract temporal occlusion events, which are, in turn, used to classify objects on the basis of whether they are occluded by or occlude other objects. Jojic [9] proposes a technique for automatically learning probabilistic 2D appearance maps and masks of moving occluders. The model explains each input image as a layered composition of "flexible sprites". A variational expectation maximization algorithm is employed to learn a mixture of sprites from a video sequence. Tao [18] decomposes video frames into coherent 2D motion layers and introduces a complete dynamic motion layer representation in which spatial and temporal constraints on shape, motion and appearance are estimated using the EM algorithm. His method has been applied in an airborne vehicle tracking system and examples of tracking vehicles in complex interactions are demonstrated. Zhou [22] introduces the concept of background occluding layers and explicitly infer depth ordering of foreground layers. A MAP estimation framework is proposed to simultaneously update the motion layer parameters, the ordering parameters, and the background occluding layers. Wu [19] proposes a dynamic Bayesian network which accommodates

an extra hidden process for occlusion. The statistical inference of such a hidden process reveals the occlusion relations among different targets. Yu [20] proposes a framework for treating the general multiple target tracking problem, which is formulated in terms of finding the best spatial and temporal association of observations that maximizes the consistency of both motion and appearance of object trajectories. Leibe [11] considers multi-object tracking as a search for the globally optimal set of space-time trajectories which provides the best explanation for the current image and for all evidence collected so far, while satisfying the constraints that no two objects may occupy the same physical space, nor explain the same image pixels at any point in time. In a recent work, Zhang [21] proposed a network flow based optimization method for data association in multiple objects tracking. The maximum-a-posteriori (MAP) data association problem is mapped into a cost-flow network with a non-overlap constraint on trajectories. The optimal data association is found by a min-cost flow algorithm in the network that is augmented with an explicit occlusion model (EOM) to track long-term occlusions.

The majority of the above methods assume that even partial observations of the occluded objects are possible. As such, they fail to handle total occlusions, especially when they last for considerable amounts of time. The method proposed in this paper is able to handle occlusions that are challenging because of both their spatial extend and duration. The proposed method uses two types of information regarding the scene. The first is the result of scene background subtraction which produces a map showing "where" action takes place in the scene. The second comes from the estimation of several (one per tracked object) Gaussian Mixture

Models (GMMs) of color that represent "what" is the appearance of moving objects. The proposed method does not need training to account for the variability in the number of tracked objects, their shape, appearance, or motion characteristics. On the contrary, such information is automatically derived and appropriately updated over time through the use of simple, generic models.

Much of the success of the method depends on a mechanism inspired by the work in [1], that properly associates foreground pixels to different objects. Thus, models of object appearance can be properly maintained and tracked. Occlusion handling is treated through a method founded on the principle of *object permanence* [14, 2], which refers to the ability of children to realize that an object exists even when it cannot be seen. Recent studies [2], indicate that infants can reach the object permanence stage at the age of five months, showing the fundamental role of the concept in visual perception.

The proposed algorithm exploits the powerful data association mechanism that has been proposed in Argyros et al. [1], where a method is proposed for tracking multiple skin-colored objects in images acquired by a possibly moving camera. The proposed method encompasses a collection of techniques that enable the detection and modeling of skin-colored objects as well as their temporal association in image sequences. Although not explicitly stated, this tracking algorithm handles occlusions between objects sharing the same color model (skin color). Nevertheless, the method requires prior training to the color model of the objects to be tracked. The approach presented in this paper may handle objects of completely different appearances for which no a priori information is assumed to be known.

In addition to the more complete appearance models, the exploitation of the concept of "object permanence" makes the proposed method much more competent in handling long-term occlusions. Huang et al. [6] also used the concept of "object permanence" to successfully handle long-term occlusions of a varying number of objects over extended image sequences. Their approach incorporates (i) a region-level association process and (ii) a object-level localization process to track objects through long periods of occlusions. Region association is approached as a constrained optimization problem and solved using a genetic algorithm. Objects are localized using adaptive appearance models, spatial distributions and occlusion relationships. The approach in [6] does not explicitly handle interacting objects of similar appearance and is, therefore, expected to fail in tracking them. On the contrary, the proposed method succeeds in treating such cases.

The rest of the paper is organized as follows. Section 2 presents the adopted object representation model. Section 3 describes in detail the proposed tracker and occlusion reasoning. In Sec. 4, we present results from the application of the proposed methodology in several video sequences that demonstrate important aspects of the performance of the proposed method. Among other things, the method is shown to successfully handle dynamic updating of the object's appearance models, long-term occlusions, layered object occlusions and occlusions among objects of similar appearance. Finally, Sec. 5 provides the main conclusions of this work as well as extensions that are under investigation.

2. Object modeling

The proposed method is able to detect and track an arbitrary and potentially time varying number of objects. No a priori knowledge regarding the object's 2D or 3D shape, appearance or motion is assumed. To achieve tracking, simple, generic object models are automatically built and maintained.

In the following, we represent an image point as $p = (x, y, c)$ under the convention that it is located at (x, y) and has color c. Each object is represented with a parametric model that takes into account both its spatial layout and its photometric appearance. More specifically, the object model $o \equiv (e, g)$ consists of an ellipse e that accounts for the position and spatial distribution of an object and a Gaussian Mixture Model (GMM) g that represents its color distribution.

The ellipse $e = (c_x, c_y, \alpha, \beta, \theta)$ represents the spatial extend of an object o that is located at (c_x, c_y), has an orientation θ with respect to a local 2D image coordinate frame, and the lengths of its major and minor axes are α and β, respectively. Given a set of image points $\mathcal{P}(o)$ comprising the image of an object, the parameters of e can be computed from the covariance matrix of the locations of pixels in $\mathcal{P}(o)$.

We define the spatial distance $D(p, e)$ of an image point p from ellipse e as in [1]. Intuitively, the ellipse is transformed to a circle of radius equal to one and the same affine transformation is applied to the coordinates of the point p. The distance $D(p, e)$ of p from e is the Euclidean distance of p from the center of ellipse e in this normalized frame. The set $I(e)$ of points p that are interior to the

ellipse e can be defined based on $D(p, e)$:

$$I(e) = \{p | D(p, e) \leq 1\}. \quad (1)$$

The appearance g of an object o is modeled as a Gaussian Mixture Model (GMM) $g = g(w_k, \mu_k, \Sigma_k)$, $1 \leq k \leq K$, representing the color (UV components of YUV color space) distribution of the object's pixels. Each of the K triplets (w_k, μ_k, Σ_k) represents the weight, the mean and the covariance matrix of the kth Gaussian component of the mixture. The Expectation Maximization algorithm [4] is employed to determine the parameters of the GMM g for each object o based on the set $\mathcal{P}(o)$ of points that comprise it. We also define the probability that the pixel's color c was drawn from a GMM g as

$$P_A(p, g) = \sum_{k=1}^{K} w_k P(c | \mu_k, \Sigma_k). \quad (2)$$

$P_A(p, g)$ is a measure of the compatibility of p's color with g.

3. Proposed method

Figure 2 illustrates the information flow of the proposed tracking algorithm. Each frame of the input image sequence is first background subtracted [23] to detect foreground pixels and to form distinct blobs, i.e regions of connected foreground pixels. Assuming a still camera, background subtraction gives rise to a change mask that can be attributed to the moving objects. A set of objects that must be correctly associated to the pixels of the detected foreground blobs is also

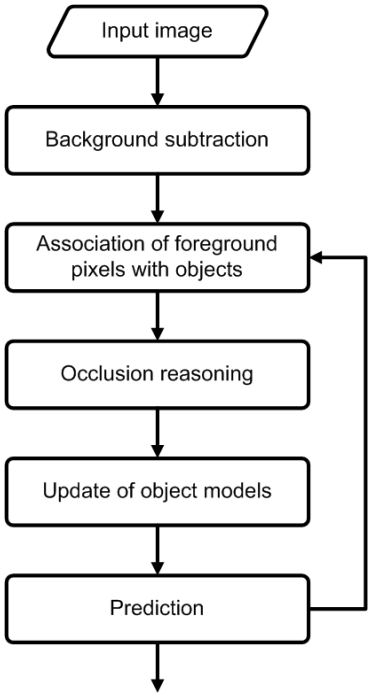


Figure 2: The flow diagram of the proposed method for tracking multiple objects in the presence of long-term occlusions.

maintained. Clearly, even in the simple case of partial occlusions, there is no one-to-one mapping between objects and blobs. Therefore, the goal of the proposed method is to exploit spatial and photometric object information in order to (a) associate foreground blob pixels with objects, (b) investigate occlusion relationships between objects, (c) update the object models and (d) use all extracted information to enable tracking. The rest of this section provides further details on these algorithmic steps.

3.1. Associating foreground blob pixels with objects

The aim of this part of the proposed method is to define the set $\mathcal{P}(o)$ of pixels belonging to an object o. It is assumed that at a given moment in time, M foreground blobs b_j, $1 \leq j \leq M$ have been detected and that N objects o_i, $1 \leq i \leq N$ are already being tracked. A single connected object can give rise to at most one connected blob.[2] However, due to occlusions, two or more different objects may appear as a single connected blob. Thus, it holds that $M \leq N$. As a direct consequence, each blob may correspond to one or more objects.

To resolve the data association problem, the method takes into account both the spatial proximity and the appearance similarity between a blob and an object. This is performed in two steps. In the first step, an object is associated with a certain blob. The validity of this algorithmic step stems from the reasonable assumption that a single connected object can give rise to at most one connected blob. In the second step, each object takes its share from the pixels of the blob it is associated with. Figure 3, graphically illustrates four objects (o_1, o_2, o_3 and o_4, visually represented as the associated ellipses) and three blobs (b_1, b_2 and b_3, shown as colored image regions).

[2]The implicit assumption at this point is that change detection through background subtraction cannot give rise to multiple blobs for a single object. This is safeguarded through morphological filtering applied to the result of background subtraction.

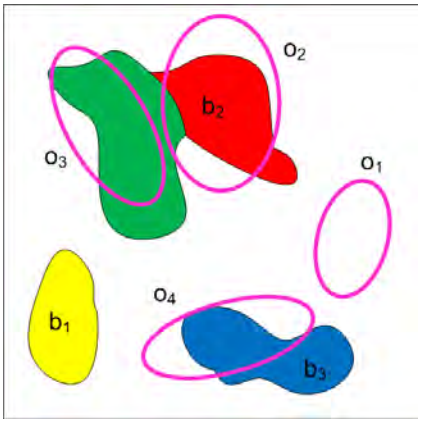


Figure 3: Possible relations between objects and blobs. For illustration purposes, each object hypothesis is shown as an ellipse and each blob as a monochrome or bi-color image region.

3.1.1. Associating objects with blobs

For an object $o_i = (e_i, g_i)$ and a blob b_j, the degree $C(b_j, o_i)$ of their association is defined as:

$$C(b_j, o_i) = \sum_{p \in (b_j \cap I(e_i))} P_A(p, g_i). \tag{3}$$

Intuitively, all image points in the intersection of blob b_j with object's ellipse e_i are tested for compatibility with the object's appearance model. Each object is associated with the blob that gives rise to the highest degree of association. More specifically, the blob $\mathcal{B}(o)$ with which object o is associated is defined as:

$$\mathcal{B}(o) = \arg\max_{b_j} C(b_j, o). \tag{4}$$

Thus, an object is associated with only one blob, whereas a blob may be associated with many objects.

3.1.2. Identifying object support regions

After associating objects with blobs, four interesting cases may arise: (a) a blob might be associated with no object, (b) an object might be associated with no blob, (c) an object might be associated with exactly one blob, and (d) a blob might be associated to multiple objects. In the following, we investigate these cases in more detail.

Blobs not associated to objects: Consider a blob b such that

$$\forall o_i, b \cap I(e_i) = \emptyset \Rightarrow \forall o_i, C(b, o_i) = 0. \tag{5}$$

Equation (5) implies that none of the existing object hypotheses explains the existence of this blob. Thus, this has to be a new object, an object that has just appeared in the scene for the first time. In the example of Fig. 3, b_1 is such a blob. In this case, a new object is generated and its set $\mathcal{P}(o)$ becomes equal to b.

Objects not associated to blobs: Consider the case of an object o such that

$$\left(\cup_{j=1}^{M} b_j\right) \cap I(e) = \emptyset \Rightarrow \forall b_j, C(b_j, o) = 0. \tag{6}$$

In this case, the hypothesis for an object o is not supported by any foreground blob pixel observations. Thus, o has disappeared and must be removed from further consideration. In the example of Fig. 3, object o_1 satisfies the criterion of Eq.(6).

Blobs in one-to-one correspondence with objects: In case that a single object o is

associated to a single blob b, the set $\mathcal{P}(o)$ becomes equal to b. This is the case with object o_4 and blob b_3 in Fig. 3.

Blobs associated to multiple objects: As discussed earlier, the correspondence between blobs and objects is not necessarily one-to-one. Two objects in an occlusion relationship will give rise to a single image blob. Consider, for example, the relevant situation in Fig. 3. Objects o_2 and o_3 must "compete" for the pixels of blob b_2. Having already associated an object o with the blob $\mathcal{B}(o)$ (see Eq.(4)), we search for the set $\mathcal{P}(o)$ of pixels to be associated with object o only within blob $\mathcal{B}(o)$. Equivalently, each pixel p of such a blob is associated with the object o^* defined as:

$$o^* = \arg\max_o \frac{P_A(p,g)}{D(p,e)}. \tag{7}$$

Intuitively, Eq.(7) assigns blob pixels p to the object o^* that minimizes spatial distance and maximizes appearance compatibility.

The approach described so far assigns image points to objects with distinct appearance models. Still, in several tracking tasks, interacting objects of similar appearance are frequently encountered. For such cases, an approach similar in spirit to that of [1] has been adopted. As a first step, it is required to quantitatively characterize the appearance similarity of two objects. Having represented an object's appearance with a GMM, this boils down to employing a criterion measuring the similarity between two GMMs. For this purpose, a Bhattacharyya-based distance has been employed. More specifically, the distance $\Delta(g, g')$ between two

mixtures of Gaussians g and g', is given by [16]:

$$\Delta(g, g') = \sum_{i=1}^{K} \sum_{j=1}^{K} w_i w'_j B(g_i, {g_j}'). \tag{8}$$

In Eq. (8), both g and g' are composed of K kernels, g_i and g'_j denote the corresponding kernel parameters and w_i, w'_j the mixing weights. $B(\cdot\,,\cdot\,)$ denotes the Bhattacharyya distance between two Gaussian kernels, defined as [5]:

$$B(g, g') = \frac{1}{8}(\mu - \mu')^T \frac{(\Sigma + \Sigma')}{2}^{-1} (\mu - \mu') + \frac{1}{2} \ln \frac{|\frac{\Sigma+\Sigma'}{2}|}{\sqrt{|\Sigma||\Sigma'|}}. \tag{9}$$

Let a number of objects of the same appearance compete for the pixels of the same blob. Having already associated an object o to the blob $\mathcal{B}(o)$ (see Eq.(4)), pixels $\mathcal{P}(o)$ defining object o will be searched only within blob $\mathcal{B}(o)$. Let also $\mathcal{P}(o)$ be initialized to the empty set, that is, $\forall o \mathcal{P}(o) = \emptyset$. Then, the rules governing the assignment of blob points to objects are the following:

- If for a pixel $p \in \mathcal{B}(o)$ of the blob associated to an object $o = (e, g)$ it holds that $p \in I(e)$, then $\mathcal{P}(o) := \mathcal{P}(o) \cup \{p\}$. Note that this way, p may be assigned to several different objects having the same appearance models.

- If a pixel $p \in \mathcal{B}(o)$ does not belong to any of the ellipses of the competing object models, then $\mathcal{P}(o^*) := \mathcal{P}(o^*) \cup \{p\}$ where o^* is the object defined as:

$$o^* = \arg\min_{o} D(p, e). \tag{10}$$

Intuitively, Eq.(10) assigns blob pixels p outside any object ellipse, to an object that minimizes the spatial distance to it.

3.2. Object models update

Once each and every blob pixel has been assigned to some object, point sets $\mathcal{P}(o_i)$ have been computed for all objects o_i. Then, an update of the objects $o_i = (e_i, g_i)$ can be performed based on the sets $\mathcal{P}(o_i)$. As stated earlier, e_i can be computed from the spatial distribution of points in $\mathcal{P}(o_i)$. Additionally, each object's area is defined as $A_i = |\mathcal{P}(o_i)|$. The appearance model g_i is computed through the application of Expectation Maximization algorithm [4] over the colors of the image points in $\mathcal{P}(o_i)$.

The appearance model of an object is updated only for objects that are in one-to-one correspondence with an image blob. In fact, and as it will become more clear in Sec. 3.3, this is equivalent to updating an object's appearance model when it is observed in isolation, without any occlusions occurring. Having two objects competing for the pixels of a single blob signals occlusion. In that case, the appearance models of the corresponding objects are stopped from being updated. We also denote with A'_i the area of object o_i at the last frame in which this object appeared in isolation.

3.3. Object visibility and occlusion handling

Occlusion reasoning is based on both the spatial and the appearance components on an object's model. As an example, consider the situation graphically

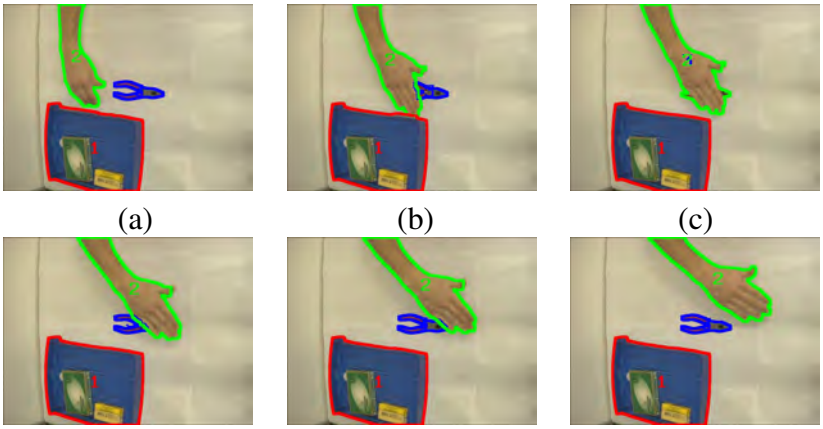


Figure 4: The size of an object being occluded decreases considerably as occlusion progresses.

illustrated in Fig. 4. Figure 4(a) shows two objects (a hand and a pincer) prior to occlusion.

At this time, each of the objects is associated with its own blob. As long as occlusion occurs (Figs. 4(b)-(e)), the two objects compete for the points of a single blob. The blob pixels that are compatible to the appearance of the occluder (hand) will be assigned to it, so no significant changes in its area will be observed. The occluded object (pincer), will appear to shrink, since fewer and fewer image points will be assigned to it (Figs. 4(b), (c)). Thus, for the occluded object a significant decrease of its area will be observed as soon as occlusion starts. Therefore, the occlusion ratio R_i of an object o_i is defined as:

$$R_i = \frac{A_i}{A'_i}. \tag{11}$$

The occlusion ratio R_i is measured for objects o_i sharing the points of a blob

with other objects. A small R_i indicates that its currently observed size is small compared to the area of the same object before occlusion started. Thus, R_i can be used to quantitatively characterize a certain occlusion. In fact, in case that

$$R_i \leq T, \tag{12}$$

object o_i is declared as disappeared because of a full occlusion (e.g., Fig. 4(c)).

Occlusion reasoning does not only require understanding whether an object is occluded or not but also requires the identification of the occluder. In case that only two objects compete for the points of a blob, the situation is straightforward. In case that more than two objects compete for the pixels of the same blob, the definition of the occluder needs more attention. The occluder should be an object that lies in the close proximity of the occluded object o_B and has recently occupied a portion of the occluded object's image. Formally, for each possible occluder o_i, the number of pixels p in $\mathcal{P}(o_i) \cap I(e_B)$ is calculated. The object o_i that produces the largest such number of pixels is defined as the object occluding o_B.

Objects reported as fully occluded according to the definition of Eq.(12) are treated as suggested by the object permanence principle. This means that, until the object appears again (i.e., $R_i > T$), it is assumed to be behind its occluder and to move with it. The object is excluded from the association of objects to blobs (Sec. 3.1.1). Instead, it inherits the associations of its occluder. In the pixel assignment part (Sec. 3.1.2), the occluded object is assumed to share the same ellipse with its occluder. This allows the occluded object to continuously claim

pixels that are compatible to its color appearance model and lie in the proximity of its occluder.

When a previously occluded object reappears ($R_i > T$) in the proximity of its occluder, the two objects are dis-associated and the image points assigned to the occluded object are used to construct a new spatial model. As the object emerges (see, for example, Fig. 4(e)), the spatial model grows smoothly through frames and accurately encapsulates the object's shape. As discussed earlier, the appearance model of the object will be updated only when the occluded object appears isolated (Fig. 4(f)), that is, in a one-to-one correspondence with a blob.

3.4. Layered occlusions

The term layered occlusions is used to describe situations where multiple objects participate in an occlusion relationship. The proposed method forms and maintains dependencies between occluded objects and their occluders. For a set of objects in a layered occlusions relation, there will always be the foremost occluder and a number of occluded ones behind it. All occluded objects declare all other objects as potential occluders. The reappearance of one of these has the following implications:

- The remaining occluded objects will be searched not only in the proximity of the original occluder, but also in the proximity of the newly reappeared object.

- The label of the reappeared object will be removed from the list of all of its potential occluders.

As an example, consider object X which occludes object Y. Let object Z be occluded by the constellation of X and Y. Then, if Y appears, Z has to be searched around both X and Y. Simultaneously, Y should stop from being searched around X. This could lead to a fast grow of the number of alternative hypotheses that need to be monitored and maintained. On the other hand, for all practical purposes, the adopted convention performs well in realistic depths of layered occlusions and number of objects involved.

3.5. Linear prediction and object model propagation

In the process described so far, data association is based on the relations of an object's spatial distribution (represented as the ellipse e in the object's model $o = (e, g)$) with the detected blobs. However, instead of using the ellipses as those were computed in the previous frame, we may use a prediction about the position of an object's ellipse based on its recent motion. Assuming that the immediate past is a good prediction for the immediate future, a simple linear scheme is used to predict the object's ellipse position in the current frame. Blob and blob pixel associations with objects (described in Sec. 3.1.1 and 3.1.2, respectively) is then based on these predicted ellipse positions.

4. Experimental results

The proposed method has been tested and evaluated in a series of image sequences demonstrating challenging tracking scenarios. Results from several representative input video sequences are presented in this paper. Videos demonstrat-

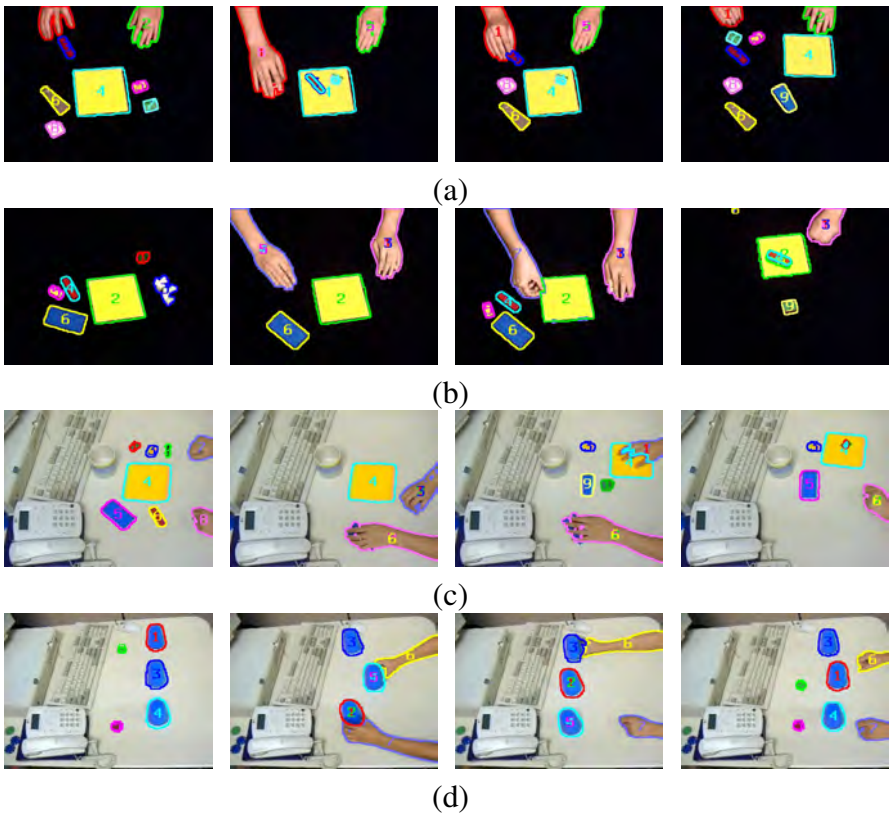


Figure 5: Characteristic snapshots from the tracking experiments with the sequences tested in [6]. Rows (a), (b), (c) and (d) correspond to datasets "lego 1", "lego 2", "lego 3" and "shellgame", respectively.

ing tracking results are available online.[3] In all experiments, input sequences are composed of images of VGA resolution (480×640).

The first set of experiments has been carried out to assess the performance of the proposed method in the image sequences[4] employed in [6]. The four sequences ("lego 1", "lego 2", "lego 3" and "shellgame") consist of 309, 398, 412

[3] Supplementary video material showing tracking results can be retrieved at the following web address: http://www.ics.forth.gr/∼argyros/research/occlusions.html

[4] Available at http://www.cc.gatech.edu/cpl/projects/occlusion/

and 460 frames, respectively. No background model for these sequences is available. Sequences "lego 1" and "lego 2" are pre-segmented, i.e., foreground colors appear on a black background. For the rest two sequences, a background model is built based on frames in which no foreground object appears. Each row of images in Fig. 5, provides characteristic frames from object tracking in each of these sequences. Individual objects are identified through the use of different colors for their contours and through arithmetic labels located on object centroids. Thus, an object is successfully tracked if it maintains the same color and label in all of its occurrences. Overall, the proposed method managed to successfully track all objects in all of these videos.

To evaluate the proposed method in even more challenging situations, several other videos have been recorded and used for testing. In all reported experiments, input sequences consisted of standard VGA resolution images (640×480) acquired at 20 Hz. Background subtraction has been performed with Zivkovic's improvements [23] of the Stauffer and Grimson's method [17]. The U, V components of the YUV color space has been used for building the GMMs g of object appearances. For each GMM, the EM algorithm had to estimate the parameters of $K = 10$ components. The threshold T on the occlusion ratio (Eq.(12)) signaling full object occlusion was set to 35%. The selection of this threshold value is related to the robust handling of the re-emergence and the subsequent tracking of previously occluded objects. Setting the threshold value T close to 0%, would mean that a minor color misclassification would suffice to falsely signal the reappearance of an occluded object. Additionally, the detection of very small visible

parts of partially occluded objects necessitates their subsequent tracking. This can be error-prone if these parts are very small.

The first such image sequence ("objects" sequence) consists of 1280 frames and shows a person manipulating several objects on a tabletop. Characteristic snapshots demonstrating tracking results are shown in Fig. 6. The sequence scenario is as follows. Initially, a hand brings into the scene a basket containing several objects. Then, he empties the basket, interacts with the objects, fills the basket again and finally empties it once more. At the beginning of the experiment, the system has no a priori knowledge about the type, size, color, shape or motion of the objects to be observed. At the end of the experiment the proposed method has been able to track individual objects and has built a model of their color appearance.

More specifically, Fig. 6(a) shows the empty desktop on which the experiment is performed and of which a background model has been built. In Fig. 6(b), the human hand has already brought into the scene a box containing a few objects. Having no a priori knowledge about the scene other than a background model of it, the system identifies the constellation of the hand, the blue box and the rest of the objects as a single multicolor object, for which it builds a single object model. As soon as the hand leaves the box on the table (Fig. 6(c)), the originally connected set of pixels becomes disconnected. The original object hypothesis (red contour) is assigned to the blue box because this is more similar to the previous box/hand constellation. Another object (hand, green contour) is automatically generated. For the next frames, the hand color appearance model is updated.

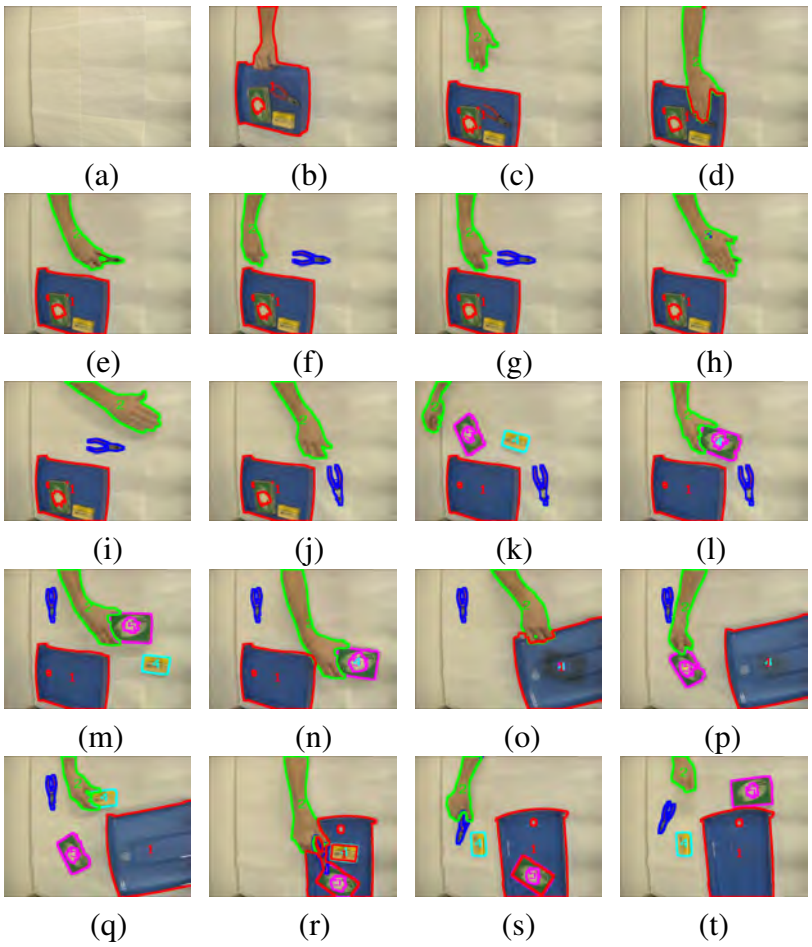


Figure 6: Characteristic snapshots from the tracking experiment on the "objects" image sequence.

The same happens also to the appearance model of the blue box, in which the components corresponding to the previously joined hand, now vanish. The hand interacts with the box again (Fig. 6(d)). Now, the color models built assist the method in correctly assigning the pixels of the single connected blob to the two object hypotheses (hand, box). In Fig. 6(e), the hand has taken the pincer off the blue box and moves it to another position on the table. For the moment, the

method interprets this as a change in the appearance of the hand and, at that stage, the pincer appears as part of the hand object. This is because the pincer has never been observed in isolation but only as part of another object (box). As soon as the hand leaves the pincer on the table, the pincer is understood as an individual object (Fig. 6(f), blue contour). The identity of the pincer object is not lost even when the hand passes several times over it, grasps it and moves it to another place on the table (Figs. 6(g)-(j)). In a similar manner, the hand empties the basket. As shown in Fig. 6(k), the hand, the box and the pincer maintain their original identity, while the two other objects have acquired their own object identities. In Fig. 6(l), the hand has grasped the object with the purple contour and has used it to completely occlude the one with the cyan contour. The full occlusion has been signaled and both object hypotheses are maintained and tracked together with the observed region of the occluder. Both objects are transfered to a new position, the hand removes the occluding object (Fig. 6(m)) and the correct identity for the occluded object is still maintained. The purple object is again brought on top of the cyan one, fully occluding it once more. This time, the big box is also brought on top of the purple object creating a layered occlusion (Fig. 6(o)). When the hand brings the purple object again in sight dragging it under the big box, the purple object still maintains its original identity (Fig. 6(p)). The same happens to the cyan object (Fig. 6(q)). The manipulation of objects continues; the hand brings all objects again into the blue basket and starts moving the latter around (Fig. 6(r)). The experiment ends with the hand emptying the basket once more (Figs. 6(s), (t)). Correct object identities are still maintained.

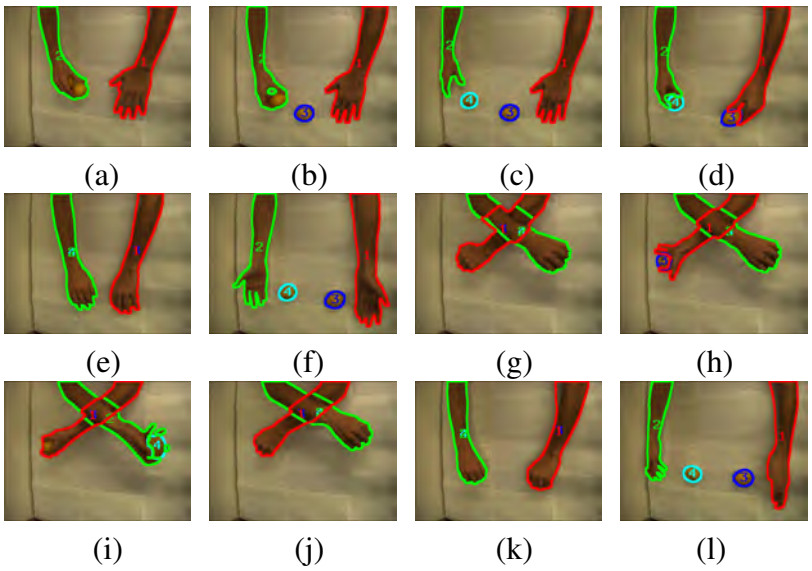


Figure 7: Characteristic snapshots from the tracking experiment on the "lemons" image sequence.

Another experiment was performed on the "lemons" sequence (350 frames in total, presented in Fig. 7), demonstrating that the method succeeds in handling occlusions when tracking objects of similar appearance. In a scene setting that is similar to the previous one, two hands appear in front of a camera (Fig. 7(a)) and are assigned two different object identities. The hand appearing at the left (green contour) holds two lemons. As soon as lemons appear in isolation (Figs. 7(b), (c)) they get their own object labels. Then, each hand grasps a lemon (Fig. 7(d)), fully occludes it (Fig. 7(e)) and then reveals it (Fig. 7(f)). Lemon identities have been maintained. The two hands grasp the two lemons totally occluding them and then cross (Fig. 7(g)). Hands reveal what they carry (Fig. 7(h), (i)), showing that despite the complex interaction of two objects of similar color appearance (arms) with two other objects of similar color appearance (lemons) and the simul-

taneous presence of two full occlusions, the identities of the lemons are correctly tracked. The experiment ends after the hands leave the objects they hold on table (Figs. 7(j)-(l)).

In all the experiments reported so far, a person manipulates certain objects in front of a visually simple background. Although that background model building and maintenance is not the main focus of this paper, it is interesting to verify the performance of the proposed approach in cases where background modeling and foreground detection is performed in more realistic conditions. Towards this goal, two image sequences have been recorded in a room that is monitored by several cameras. The proposed tracking approach has been employed to monitor the activity of humans interacting in this room.

Figure 8 shows characteristic snapshots of the first such sequence ("bag" sequence), which consists of 287 frames. Figure 8(a) shows the appearance of the space where the experiments have been conducted; the background model has been built for this environment appearance. Figure 8(b) shows the first person that has been detected and tracked (red contour, person 1). This human is successfully tracked as he approaches the camera (Fig. 8(c)) and rotates around his vertical axis (Fig. 8(d)). In the meantime (Fig. 8(c)) a new person (green contour, person 2) enters the room holding a bag. The bag is identified as a new object as soon as the person holding it leaves it on the floor (Fig. 8(e)). After this, both humans move around the bag occluding it as well as occluding each other (Fig. 8(f)-(g)). At a certain point (Fig. 8(h)), person 2 has grasped again the bag and hands it to person 1. Despite the complex humans/object interaction, all objects maintain

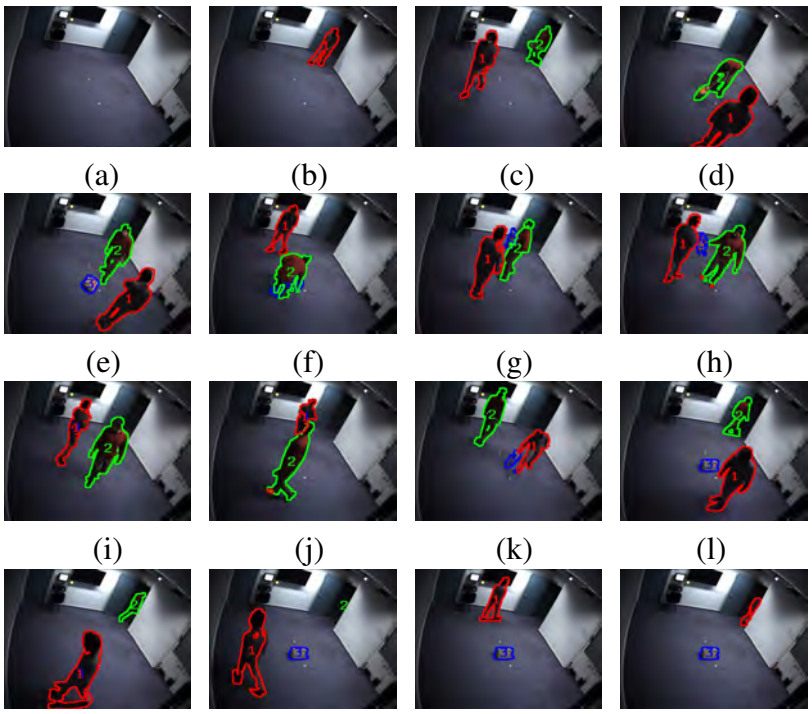


Figure 8: Characteristic snapshots from the tracking experiment on the "bag" image sequence.

correct identities. In Fig. 8(i), person 1 holds the bag, although totally occluding it. Both persons continue to move occluding each other (Fig. 8(j)) until the one holding the bag leaves it again on the floor (Fig. 8(k)). Finally, both persons are successfully tracked until they exit the room at Figs. 8(m) and 8(p), respectively. Overall, the proposed tracking method was able to detect and track correctly all the individual objects moving in the scene. It should be stressed that this has been achieved without any kind of a priori known object models. The achievement of object tracking in such complex situations is difficult even if somebody takes into account additional context dependent knowledge such as the fact that there are

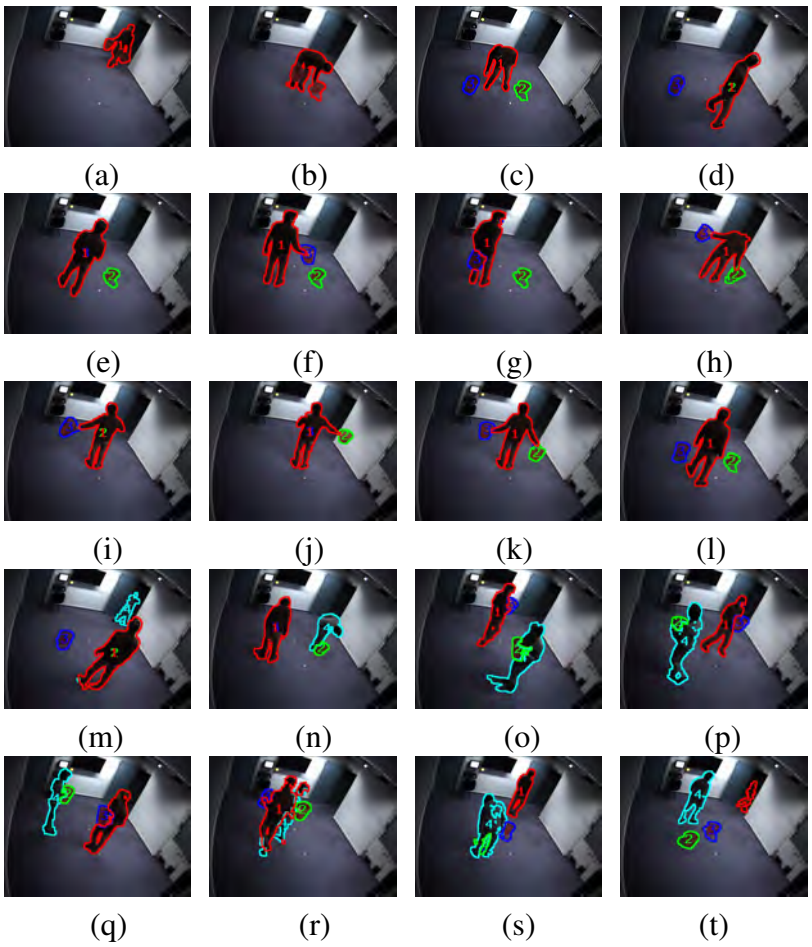


Figure 9: Characteristic snapshots from the tracking experiment on the "buckets" image sequence.

persons walking on a ground floor, etc. Clearly, accommodating such important additional cues and knowledge can only improve tracking.

In another experiment ("bucket" sequence, 398 frames), an even more challenging situation is encountered. This image sequence involves two persons that interact with two almost identical looking objects. Snapshots from this sequence are provided in Fig. 9. In Fig. 9(a), a person enters a room holding, in each of

his hands, a red bucket. The person leaves the two buckets on the floor (Fig. 9(b), (c)) and starts moving around them (Fig. 9(d), (e)). Then, he stands in front of one of the buckets occluding it, grasping it with his right hand and then passing it to his left hand behind his back (Fig. 9(f), (g)). He then grasps the second bucket with his right hand and then starts hiding each of the buckets from the camera (Fig. 9(i)-(k)). At some point in time, he leaves both objects on the floor again (Fig. 9(l)). Right after, another person appears (Fig. 9(m)). While various types of occlusions continue to occur, each person grasps a bucket (Fig. 9(n)) and start moving around in the room. Between frames corresponding to Figs. 9(p) and 9(r), the two persons move around each other holding the buckets, thus creating several layered occlusions. Finally, the two persons leave the buckets again on the floor and exit the room (Figs. 9(s), (t)). Throughout the whole sequence, object identities are correctly assigned and propagated in time.

In the above sequences, foreground detection was adequately accurate and tracking was not affected by errors in background subtraction. To investigate the influence of background subtraction on tracking, we performed a number of tests where we forced background subtraction to produce poor results. More specifically, we varied a parameter of the employed background subtraction method [23] that affects the true positives and false negatives of the method. As a test case, we considered the "buckets" sequence (Fig. 9). For a broad range of parameter values, tracking produced identical results. In extreme cases, the amount of false positives (or false negatives) produced by background subtraction affected the correctness of the performance of the method.

When background subtraction produces many false positives, several background pixels are labeled as foreground. Lenient background subtraction may produce hypotheses for non-existent objects. Additionally, foreground blobs are larger than the real objects. This results in the generation of wrong object hypotheses. Different objects appear connected in the foreground masks and object hypotheses are built for constellations of objects rather than for individual objects. The built appearance models might also be inaccurate because they are affected by the colors of the background pixels falsely identified as foreground, leading to inaccuracies and false similarities between objects.

When background subtraction produces many false negatives, several foreground pixels are labeled as background. Clearly, an object will not be tracked if it cannot account for a sufficiently large blob in the foreground mask. More often, a single object will give rise to multiple separate blobs. This violates the basic assumption that a single object gives rise to a single blob. As a result, multiple object hypotheses will be generated for a single object.

Figure 10 provides evidence for the behavior of the tracker with respect to the performance of background modeling and subtraction. Figures 10(a) and 10(d) shows the results of the tracker for two frames of the "buckets" sequence (Fig. 9). In this particular experiment, background subtraction gives fairly accurate results. As a consequence, when the person enters the room (Fig. 10(a)) the person and the buckets he holds are identified as a single connected blob and object. Later on (Fig. 10(d)), the person leaves the two buckets on the floor, which results on their identification as two new objects. Figures 10(b) and 10(e) shows the same frames

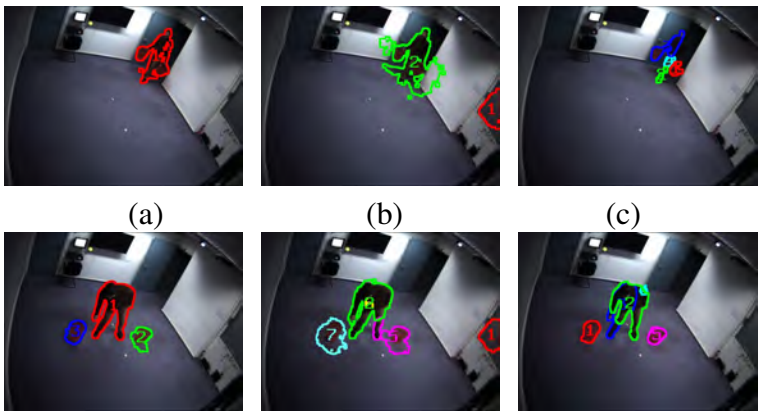


Figure 10: Influence of background subtraction on the results of tracking for the case of the "buckets" sequence. The three columns show indicative tracking results for the cases of accurate, lenient and strict background subtraction.

in the case of lenient background subtraction. In Fig. 10(b), background subtraction has already contaminated the object model of Fig. 10(a) with background pixels. Additionally, a wrong object hypothesis (object 1) has been created. When the person leaves the buckets on the floor (Fig. 10(e)), one of them is correctly identified, but the second one, although clearly separated from the person's body, has been assigned a considerable part of the background and of the person's foot. Figures 10(c) and 10(f) shows the same frames in the case of strict background subtraction. When the person first appears in the scene, several individual objects are identified as a result of the disconnectedness of the foreground mask. When the person leaves the buckets on the floor, the buckets are correctly identified, but the originally detected and propagated object hypotheses still live on the region of the person's body.

5. Discussion

We presented a method for tracking multiple objects in the presence of occlusions with long temporal duration and large spatial extends. The proposed method can cope successfully with multiple objects dynamically entering and exiting the field of view of a camera and interacting in complex patterns. Towards this end, simple models of object shape, appearance and motion are dynamically built and used for supporting tracking and occlusion reasoning. Tracking is performed by systematically assigning pixels of foreground blobs to simple geometrical models of objects, taking into account object's appearance. Occlusion reasoning is based on the concept of "object permanence".

Our method is based on the approach proposed in [1]. As already stated in Sec. 1, the tracker proposed in [1] successfully tracks multiple skin color objects in images acquired by a possibly moving camera and can handle partial occlusions and short term full occlusions. Prior training is required to obtain the color model of the objects to be tracked. In this paper, we use background subtraction to find the image regions that are occupied by moving objects and thus we are able to use color information to track objects of different colors and to explicitly reason about occlusions. Therefore, our method assumes a steady camera for image acquisition but can handle many more cases of object appearance and interactions than [1].

Inspired by [6], our approach reasons about occlusions by relying on the concept of "object permanence". The authors in [6] use background subtraction, color, shape and spatial distribution to track objects in the presence of occlusions. Despite the fact that the two approaches share some methodological aspects, some

key features allow our method to cope with a broader spectrum of situations. First of all, the work in [6] employs distinct region and object level tracking mechanisms. The evaluation of the region level correspondences is based solely on the shape and the displacement of the candidate regions (blobs). In our method, we omit this level by assigning blobs (regions) to object hypotheses in a direct manner that makes use of predicted object displacement, shape and color. By taking into account richer information about objects, errors in blob association are avoided. The major difference between the two methods is the treatment of objects of similar appearance. If two similar objects share the same blob, the method in [6] is forced to assign each pixel to a single object by using information about color and distance. This hard decision is bound to misclassify pixels and eventually distort the object models. The longer similar objects share the same blob the harder it gets to obtain the correct object shape and to acquire the correct object to region association when the region splits again. On the contrary, our method detects objects of similar appearance and uses the data association mechanism of [1]. Thus, depending on the spatial and appearance proximity of pixels to object models, pixels may be assigned to more than one object hypotheses.

The proposed method was successfully tested in the complete data set of [6]. Given the fact that the aforementioned data set does not contain even small interactions between objects of similar appearance, we tested our method on additional image sequences showing complex interaction between such objects ("lemons" and "buckets" sequences). The obtained experimental results demonstrate that the developed tracking methodology can successfully handle occlusions in chal-

lenging situations. The tracker incorporates and maintains very simple models of object shape, appearance and motion. This makes the tracker simple, fast and generic in the sense that no strong assumptions are imposed on the characteristics of the tracked objects. Our approach is expected to fail when objects to be tracked have too complex shapes and appearance or move with irregular motion patterns. Moreover, in our approach, successful background subtraction is an important factor that affects tracking. This is because background subtraction determines where in the scene action takes place and, therefore, what needs to be represented, modeled and associated between consecutive frames. If background subtraction has many false negatives, a single object may appear as a set of disconnected foreground blobs. This violates the main assumption, that a single object can give rise to a single blob. As a result, more than one object hypotheses will be generated for a single object. On the other hand, if background subtraction results in too many false positives, objects will be mixed with the background and their appearance models may drift and fail to accurately represent them. Towards removing these drawbacks, future research will consider the use of more elaborate spatial and appearance models that will provide more accurate object representations. Additionally, tracking results might be improved by a soft assignment of foreground pixels to object hypotheses as opposed to the current approach which bases this assignment on the strict notion of blob connectedness.

Acknowledgements

This work was partially supported by the IST-FP7-IP-215821 project GRASP.

References

[1] A.A. Argyros and M.I.A. Lourakis. Real-time tracking of multiple skin-colored objects with a possibly moving camera. In *European Conference on Computer Vision (ECCV)*, pages 368–379, 2004.

[2] B. Baillargeon, E.S. Spelke, and S. Wasserman. Object permanence in five-month-old infants. *Cognition*, 20(3):191–208, 1985.

[3] G.J. Brostow and I. Essa. Motion based decompositing of video. In *International Conference on Computer Vision (ICCV)*, pages 8–13, 1999.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[5] K. Fukunaga. *Introduction to statistical pattern recognition (2nd ed.)*. Academic Press Inc., San Diego, CA, USA, 1990.

[6] Y. Huang and I. Essa. Tracking multiple objects through occlusions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1051–1058, 2005.

[7] M. Isard and J. Maccormick. Bramble: a bayesian multiple-blob tracker. In *International Conference on Computer Vision (ICCV)*, volume 2, pages 34–41, 2001.

[8] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi. Robust online appearance models for visual tracking. *IEEE Trans. on PAMI*, 25(10):1296–1311, 2003.

[9] N. Jojic and B.J. Frey. Learning flexible sprites in video layers. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 199, 2001.

[10] S. Khan and M. Shah. Tracking people in presence of occlusion. In *Asian Conference on Computer Vision (ACCV)*, pages 1132–1137, 2000.

[11] B. Leibe, K. Schindler, and L. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *International Conference on Computer Vision (ICCV)*, pages 1–8, 2007.

[12] J.S. Marques, P.M. Jorge, A.J. Abrantes, and J.M. Lemos. Tracking groups of pedestrians in video sequences. In *IEEE International Conference on Pattern Recognition (CVPR)*, page 101, 2003.

[13] S.J. Mckenna, S. Jabri, Z. Duric, H. Wechsler, and A. Rosenfeld. Tracking groups of people. *Computer Vision and Image Understanding*, 80:42–56, 2000.

[14] J. Piaget. *The construction of reality in the child*. New York: Basic books, San Diego, CA, USA, 1937/1954.

[15] J.M. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. In *International Conference on Computer Vision (ICCV)*, pages 612–617, 1995.

[16] G. Sfikas, C. Constantinopoulos, A. Likas, and N.P. Galatsanos. An analytic distance metric for gaussian mixture models with application in image retrieval. In *ICANN (2)*, volume 3697 of *Lecture Notes in Computer Science*, pages 835–840. Springer, 2005.

[17] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEC Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 2246, 1999.

[18] Hai Tao, H. S. Sawhney, and R. Kumar. Object tracking with bayesian estimation of dynamic layer representations. *IEEE Trans. on PAMI*, 24(1):75–89, 2002.

[19] Y. Wu, T. Yu, and G. Hua. Tracking appearances with occlusions. In *IEEE Conference on COmputer Vision and Pattern Recognition (CVPR)*, volume 1, page 789, 2003.

[20] Q. Yu, G. Medioni, and I. Cohen. Multiple target tracking using spatio-temporal markov chain monte carlo data association. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[21] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[22] Y. Zhou and H. Tao. A background layer model for object tracking through

occlusion. In *International Conference on Computer Vision (ICCV)*, page 1079, 2003.

[23] Z.Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2004.

Submission (please tick): ☐ **Poster** ☒ **Paper**

Generation of Human-like Motion for Humanoid Robots Based on Marker-based Motion Capture Data

Stefan Gärtner[1], Martin Do[1], Christian Simonidis[2], Tamim Asfour[1], Wolfgang Seemann[2] and Rüdiger Dillmann[1]
[1] Karlsruhe Institute of Technology (KIT), Institute for Anthropomatics - IAIM, 76131, Karlsruhe, {stefan.gaertner | martin.do | tamim.asfour | ruediger.dillmann}@kit.edu
[2] Karlsruhe Institute of Technology (KIT), Institute of Engineering Mechanics, 76131, Karlsruhe, {christian.simonidis | wolfgang.seemann}@kit.edu

To increase acceptance of humanoids as part of our everyday lives, it is essential that motions of humanoids become more realistic and human-like. A proper approach to achieve this requirement will be introduced within the scope of this paper by adopting marker-based human motion capture. For this purpose, constraining and mapping of prerecorded motions will be applied since robots may have different degrees of freedom (DoFs) as well as a different kinematic structure than a human. Regarding this challenge, the motion must be adapted to a given robot while preserving important human-like characteristics of the recorded motion.

In order to efficiently reuse captured movements on various robots, an intermediate model is needed decoupling representation of a motion, which can be stored in a motion repository, from its execution on an actual robot. On the contrary, there exist numerous human motion capture systems that produce output in terms of different models stored in different formats. To overcome this problem, the Master Motor Map (MMM), firstly introduced in [1], presents an appropriate interface based on a unified model. An overview of the proposed system is illustrated in Figure 1. In this paper, we will further propose an extension of this model by adding certain anthropomorphic properties, such as mass distribution, segment length, moment of inertia, etc. Such an anthropomorphic model of the segmented body is of use in terms of determining forward and inverse dynamics as well as motion synthesis and retargeting.

Over the last decades, a lot of attempts have been made to develop sufficient dynamic models for simulating and analyzing complex motions of the human. Various biomechanical models are thoroughly reviewed in [2]. In order to calculate forward and inverse dynamics, knowledge of body segment properties reported in [3, 4, 5, 6], are required. Since the effort is very high to create model for each subject individually, a unified whole-body model is used instead that can be scaled in terms of body weight and height. Linear scaling equations are therefore commonly used due to their expediency.

The MMM is defined as a three-dimensional reference kinematic model enriched with proper body segment properties. The strategy with respect to the kinematic model is to define the maximum number of DoFs that might be used by any applied module. The kinematic model of the MMM including DoFs and the Euler angle conventions is shown in [1]. The linear equations published in [6] are applied to our model as they represent the most complete and practical series of predictive equations providing all frontal, sagittal, and horizontal moments of inertia. The body segment properties are adjusted with respect to the kinematics of the MMM and listed in Table 1.

Our approach of adapting movements consists of two constrained large-scale non-linear optimizations covering different requirements as illustrated in Figure 1. The used objective functions should maintain desirable properties of the motion, such as characteristic oscillations or particular configurations, and should refuse undesirable artefacts leading to unnaturalness. In general, constraints are associated with anatomic, mechanical, and motor task limitations. These are required to be able to determine a unique configuration that fits best with the given motion data and meets predefined requirements corresponding to the observed environment. To solve the mentioned optimization problems, sequential quadratic programming (SQP) is applied.

Our first optimization adapts a motion, represented through three-dimensional marker trajectories that can be captured with sophisticated marker-based system such as Vicon [7], to the articulated MMM model. The applied marker set is shown in Figure 2. Several approaches [8, 9, 10] have been proposed in order to compute feasible joint angle trajectories applying non-linear optimization. The construction of a sufficient objective function based on minimization of the sum of the squared distance between precaptured and virtual markers will be shown in this paper. Within this scope, virtual markers are defined as fixed points on the surface of the voluminous anthropomorphic model which have to be set up in advance.

To finally execute movements on the robot ARMAR-III [11], we will show the required transformation from MMM to ARMAR-III including another constrained non-linear optimization, as firstly proposed in [12]. The method has been further enhanced by adding appropriate spacetime constraints, introduced in [13], and additional constraints covering dynamic requirements. Spacetime constraints are required in order to satisfy certain task-related constraints on a motion while minimizing the changes of the captured motion. We will adapt various pick-and-place, passing over, and pouring movements, captured with a Vicon human motion capture system, to our robot ARMAR-III.

Literature

[1] P. Azad, T. Asfour, and R. Dillmann, *Toward an unified representation for imitation of human motion on humanoids*, Robotics and Automation, 2007 IEEE International Conference on, 2007, pp. 2558 - 2563.

[2] A. I. King, *A review of biomechanical models*, Journal of biomechanical engineering, vol. 106, 1984, pp. 97–104.

[3] J. T. Barter, *Estimation of the mass of body segments*, Wright Air Development Center, Wright-Patterson Air Force Base, Ohio, WADC Technical Report 57-260, 1957.

[4] R. Chandler, C. Clauser, J. Mc Conville, H. Renolds, and J. Young, *Investigation of the inertial properties of the human body*, National Technical Information Service, Virginia, Technical Report, 1975.

[5] H. Hatze, *A mathematical model for the computational determination of parameter values of anthropomorphic segments*, Journal of biomechanical engineering, vol. 13, 1980, pp. 833–843.

[6] P. de Leva, *Adjustments to Zatsiorsky-Seluyanov's segment inertia parameters*, Journal of Biomechanics, vol. 29, no. 9, 1996, pp. 1223–1230.

[7] Vicon Motion Systems. The Vicon Manual, 2002.

[8] M. Riley, A. Ude, and C. Atkeson, *Methods for motion generation and interaction with a humanoid robot: Case studies of dancing and catching*, AAAI and CMU Workshop on Interactive Robotics and Entertainment, 2000.

[9] A. Safonova, N. S. Pollard, and J. K. Hodgins, *Optimizing human motion for the control of a humanoid robot*, Adaptive Motion of Animals and Machines, 2nd International Symposium on, 2003.

[10] T.-W. Lu and J.J. O'Connor, *Bone position estimation from skin marker co-ordinates using global optimisation with joint constraints*, Journal of Biomechanics, vol. 32(2), 1999, pp 129-134.

[11] T. Asfour, K. Regenstein, P. Azad, J. Schroeder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, *Armar-III: An integrated humanoid platform for sensory-motor control*, IEEE-RAS International Conference on Humanoid Robots, 2006.

[12] M. Do, P. Azad, T. Asfour, and R. Dillmann, *Imitation of human motion on a humanoid robot using nonlinear optimization*, In International Conference on Humanoid Robots, 2008.

[13] M. Gleicher, *Retargetting motion to new characters*, International Conference on Computer Graphics and Interactive Techniques, 1998, pp. 33- 42.

Table 1: Adjusted body segment properties for the MMM model. Segment masses are relative to body masses; segment lengths are relative to body heights. Both segment center of mass and radii of gyration are relative to the respective segment lengths.

Segment	Segment Length/ Total Body Height	Segment Weight/ Total Body Weight	Center of Mass/ Segment Length [x,y,z]	Radius of Gyration/ Segment Length [rxx, ryy, rzz]
Hip	0.26	0.11	[0 4 0]	[38 36,5 34]
Spine	0.10	0.10	[4 46 0]	[32 26 28,6]
Chest	0.18	0.17	[0 46 0]	[35 28,5 31,3]
Neck	0.05	0.024	[0 20 0]	[31,6 22 31,6]
Head	0.13	0.07	[12 13 0]	[31 26 30]
Shoulder R/L	0.10	0.021	[66 0 0]	[12 26 16]
Upper Arm R/L	0.16	0.027	[0 -57,3 0]	[26,8 15,7 28,4]
Lower Arm R/L	0.13	0.016	[0 -53,3 0]	[31 14 32]
Hand R/L	0.11	0.006	[0 -36 0]	[23,5 18 29]
Thigh R/L	0.25	0.14	[0 -33 0]	[25 11,4 25]
Shank R/L	0.23	0.04	[0 -44 0]	[25,4 10,5 26,4]
Foot R/L	0.15	0.013	[0 -6- 39]	[21 19,5 12]

Figure 1: Overview of the proposed system.

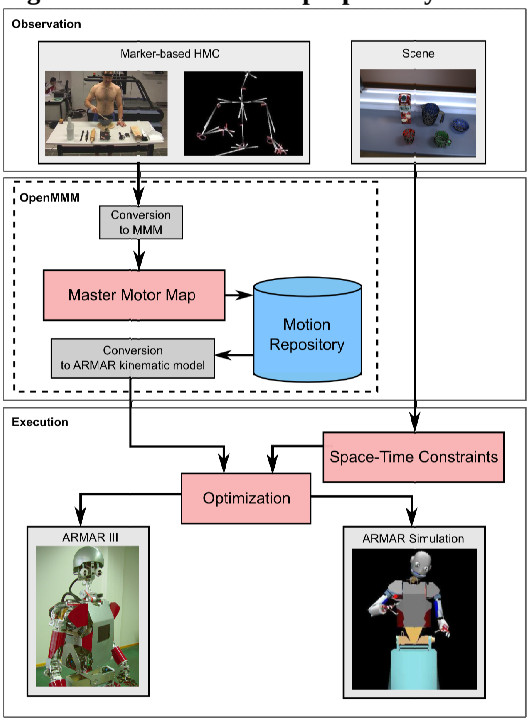


Figure 2: Applied marker set for capturing whole-body motions of a human.