Spatio-Temporal Extraction of Articulated Models in a Graph Pyramid

Nicole M. Artner[1], Adrian Ion[1,2], and Walter G. Kropatsch[1]

[1] PRIP, Vienna University of Technology, Austria
{artner,ion,krw}@prip.tuwien.ac.at
[2] Institute of Science and Technology Austria (IST Austria)

Abstract. This paper presents a method to create a model of an articulated object using the planar motion in an initialization video. The model consists of rigid parts connected by points of articulation. The rigid parts are described by the positions of salient feature-points tracked throughout the video. Following a filtering step that identifies points that belong to different objects, rigid parts are found by a grouping process in a graph pyramid. Valid articulation points are selected by verifying multiple hypotheses for each pair of parts.

Keywords: articulated object, model extraction, graph pyramid.

1 Introduction

Tracking articulated objects is an important and active field of research in Computer Vision [10,15,1]. A model of the target (the object to be tracked) is used by tracking methods to detect and associate instances of the object of interest in consecutive frames. This model is at the minimum a rectangle-shaped close-up of the object (called a template) or a color histogram, but can be as sophisticated as an online-trained classifier [11], or a hierarchical description of the objects parts and their salient features [5].

The proposed method automatically builds such a model of an articulated object, from a set of trajectories of feature-points in an initialization video:

1. build a triangulated graph on the positions of the points in the first frame;
2. label each triangle as "relevant" (on an object) or "separating" (connecting objects/parts) based on the variation of its edge-lengths over the video;
3. group *relevant* triangles in a graph pyramid framework based on their orientation variation, and obtain the "rigid" parts (build on [4]);
4. localize and verify points of articulation by observing the articulated movement of the parts.

The presented approach is related to the work done in *video object segmentation* (VOS), where the task is to separate foreground from background with the help of a video sequence. VOS methods can be divided into two categories [7]: (1) Two-frame motion/object segmentation [3,8] and (2) Multi-frame spatio-temporal

X. Jiang, M. Ferrer, and A. Torsello (Eds.): GbRPR 2011, LNCS 6658, pp. 215–224, 2011.

segmentation/tracking [7,14]. With some exceptions (e.g. [8]), the output of VOS methods is a pixel-level assignment to foreground (FG) and background (BG). Less attention is given to modeling the FG into its constituent rigid parts and joints. Most VOS methods work on the pixel level.

Motion segmentation (MS) [13,16] works on the basis of trajectories of features. Here the main concern is with the segmentation of trajectories to objects and not with detecting their parts. An advantage is that many MS methods can deal with trajectories that do not span the whole video.

The work in factorization methods for *structure from motion* (SfM) [20,19,9,17] is probably the most related to ours. These methods use a factorization technique based on singular value decomposition to detect the linear subspaces in which trajectories of feature-points of rigid/non-rigid parts lie. Like in our case, articulation points/axis are computed in a step following part detection. Trajectories of feature-points that span the whole video are required.

Our approach analyzes the trajectories of features on a higher abstraction level – in a triangulation. The used triangulated graphs encode spatial relationships resulting out of spatial proximity between features, and are the basis for all processes and decisions. The advantage of using a triangulation is the additional information about the motion and behavior of features in this relationship. The motions of parts are not constrained to linear subspaces, however, in this work, parallel projection is assumed and only 2D motions are considered.

1.1 Paper Outline

This paper is organized as follows: Sec. 2 recalls irregular graph pyramids, which are later used for the grouping process. Sec. 3 describes the spatio-temporal filtering and the grouping process, where the "rigid" parts are identified. Sec. 4 explains how the points of articulation are determined. Sec. 5 presents the experiments and in Sec. 6 conclusions are given.

2 Recall: Irregular Graph Pyramids

An irregular graph pyramid is a stack of successively reduced planar graphs $P = \{G_0, \ldots, G_n\}$. A pyramid is typically build in a bottom-up manner using only local operations. Each level $G_k, 0 < k \leq n$ is obtained by first *contracting* edges in G_{k-1}, if their vertices have the same label (regions should be merged), and then *removing* edges in the obtained intermediate graph to simplify the structure. In each G_{k-1} contracted edges form trees called *contraction kernels*. One vertex of each contraction kernel is called a *surviving vertex* and is considered to have been "survived" to G_k. The *receptive field* $F(v)$ of v is the (connected) set of vertices from level 0 that have been "merged" to v over levels $0 \ldots k$. Higher in the pyramid, the receptive fields cover more of the base level and decisions gradually change from local to global. Compared to regular pyramids, irregular graph pyramids have the advantage that their structure is not fixed, it adapts to the data. For more details about graph pyramids see for example [12].

3 Rigid Part Extraction

The input of the method consists of trajectories of feature-points from a training video. Two points can lie on: different objects, the same articulated object but on different rigid parts, or on the same rigid part of an object. To detect the feature points located on the same rigid part we proceed in two steps: (1) spatio-temporal filtering of edges connecting feature points and (2) grouping of triangles formed by the edges. In Sec. 4 we discuss the detection of points of articulation based on the rigid parts found by the method described in the following.

Detect points on different objects. A triangulated graph T is build by a Delaunay triangulation of the positions of the feature-points in the first frame. This step creates a neighborhood for the points and produces entities (edges, triangles), which have not just position, but also size and orientation. For a discussion on the robustness of graphs built on sets of points see [18].

In the graph T a high variation of the length of an edge over the video indicates that its end-points are very likely not located on the same rigid part. Based on this observation, the triangles (faces of T) are labeled as *relevant* if all three edges have the *maximum variation* $\Delta(e) = \max_{0 \leq t_1, t_2 < t_F} \{||e_{t_1}|| - ||e_{t_2}||\}$ below a defined threshold ϵ_r, and *separating* otherwise. Here $||e_{t_1}||$ is used to denote the length of edge e at time t_1 and F is the number of frames in the input sequence.

Connected components made of only *relevant* triangles of T identify detected objects (see Sec. 5, Fig. 5).

Detect points on the same rigid part. This step groups *relevant* triangles to identify the rigid parts. Only triangles that share an edge are grouped thus the obtained parts are guaranteed to be connected.

True rigid motion is rarely observed, e.g.: the skin of a human is elastic, and tracked feature positions are affected by noise. Thus a local decision (e.g. global threshold on $\Delta(e)$) cannot robustly determine which triangles belong to the same "rigid" part (see Fig. 4). Following the approach in [4], the grouping is done using a graph pyramid. Every vertex in the base level G_0 identifies a relevant triangle. Every vertex in the obtained top level G_n identifies a rigid part. Using a graph pyramid allows to make *local-to-global* decisions by using only local operations in a representation that is shift and rotation invariant.

The *orientation variation* $O_e(t)$ of an edge e over time is a 1D signal that encodes at each time t the accumulated orientation change relative to the orientation at frame 0. More formally, $O_e(t) = O_e(t-1) + \theta_e(t)$, where $\theta_e(t)$ is the relative change in orientation (signed angle) of the edge e between frames t and $t-1$. For example, turning around the x axis once will give a value of 360° degrees and turning twice in the same direction will give 720°, not 0°. The direction of rotation is encoded by the sign: counter clockwise (CCW) is positive, and clockwise (CW) is negative.

The *orientation variation* $O_r(t)$ of a triangle r, at time t is defined as the average of the orientation variations of its edges, at time t. The *maximal relative orientation change* of two triangles is the highest absolute difference of their

orientation changes over the input sequence $\Delta(r_1, r_2) = \max_{0 \leq t < t_F}\{|O_{r_1}(t) - O_{r_2}(t)|\}$. The value $\Delta(r_1, r_2)$ is used as a cue (grouping criterion) for the triangles r_1, r_2 belonging to the same part.

The graph pyramid groups *relevant* triangles into "rigid" parts such that:

- all the triangles inside a part have a similar O_r,
- the average O_r of triangles in two neighboring parts is different.

Vertices of G_k represent parts consisting of one or more triangles. An edge e in G_k encodes that there exist two triangles r_1, r_2, one belonging to each of the parts represented by the vertices connected by e, such that r_1 and r_2 share an edge in T. Notice the duality between T and G_0: the vertices of G_0 represent the triangles (faces) in T, the edges of G_0 encode adjacency of triangles in T, while the edges in T connect feature-points and make up (the boundaries of) the triangles.

This grouping is similar in spirit to the image segmentation task, where the results should be regions with homogeneous color/texture neighbored with regions that look very different. For more details on the grouping process see [4].

4 Determine Points of Articulation

Articulated motion is a piecewise rigid motion, where the rigid parts conform to the rigid motion constraints, but the overall motion is not rigid [2]. A *point of articulation* connects rigid parts. The parts can move independent of each other, but their distance to the point of articulation remains the same. This paper considers articulation in the image plane (1 degree of freedom). In the following we will call articulated parts, two parts that perform an articulated motion constrained by a point of articulation.

Having the rigid parts in the scene (vertices in the top level G_n) we proceed to discover the parts that move constrained by articulation and to find the corresponding point of articulation. Determining the points of articulation is done in two steps: (1) generate hypotheses for points of articulation (Section 4.1), and (2) verify hypotheses and select valid ones (Sec. 4.2).

4.1 Generation of Hypotheses for Points of Articulation

Given two time steps $0 \leq t_1 < t_2 < t_F$ the motion of the points of two articulated parts A and B, can be modeled by considering rotation and translation separately. Using matrices the point correspondence can be written as:

$$\mathbf{p}' = (R * (\mathbf{p} - \mathbf{c}) + \mathbf{c}) + \mathbf{o} \tag{1}$$

where $\mathbf{p}$ is the point at time t_1 and $\mathbf{p}'$ is the same point at time t_2. The point $\mathbf{p}'$ is obtained by first rotating $\mathbf{p}$ around $\mathbf{c}$ with angle θ and then translating it with offset $\mathbf{o}$. R is the 2D rotation matrix with angle θ given by:

$$R = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$

To compute the position of the point of articulation $\mathbf{c}$ at time t_1 it is sufficient to know at times t_1 and t_2 the positions of two points of each of the two rigid parts: $\mathbf{p}_i, \mathbf{p}'_i$, $0 < i \leqslant 4$. Indexes $i \in \{1, 2\}$ are used for the points of part A and $i \in \{3, 4\}$ for the ones of B. These points will be denoted by the term *reference points*. Taking Eq. 1 for the four points produces the following system of equations:

$$\left\{ \mathbf{p}'_i = (R_i * (\mathbf{p}_i - \mathbf{c}) + \mathbf{c}) + \mathbf{o} \qquad i = 1 \dots 4, \right. \tag{2}$$

where $R_i = R_A$ if $i \leq 2$ and $R_i = R_B$ otherwise. The matrices R_A, R_B are the 2D rotation matrices of the parts A, B with angles θ_A, θ_B, respectively. Solving the system gives the $2D$ coordinates of $\mathbf{c}, \mathbf{o}$ at time t_1 and the values of $\sin(\theta_A), \cos(\theta_A), \sin(\theta_B), \cos(\theta_B)$.

Points of articulation do not have to be visually salient i.e. easily trackable, and are thus not expected to belong to the tracked feature-points. We use Eq. 2 to generate hypotheses for points of articulation for every pair of parts and time stamps t_1, t_2 selected as described in the following. In addition to the detected rigid parts, the grouping step also computes for each part an orientation variation signal as the average of O_r over the contained triangles r. If no rotation exists between frames t_1, t_2 the system in Eq. 2 can have an infinite number of solutions for the point of articulation $\mathbf{c}$. Thus for any two parts A, B, time stamps t_1, t_2 are selected, where the orientation variation signals corresponding to both rigid parts differ between t_1 and t_2 with more than a value ϵ_a. In experiments we divide the sequence in 20 fixed length time windows, and verify the above condition for all of them. For more general purposes, the optimal time steps t_1, t_2 can be found in polynomial time.

The reference points for each rigid part, at time t are the centroid of the part and a point $\mathbf{q}$. The point $\mathbf{q}$ is obtained by translating the centroid over a fixed distance $d > 0$, in the direction given by the corresponding orientation variation of the part, at time t. (This strategy gives a more stable estimate than selecting two of the possibly noisy feature-point positions.) For each rigid part a local coordinate system is derived based on the two reference points. The coordinate system has the origin at the centroid of the part, and the directions of the two axis defined by the motion of the part. Having the coordinates of $\mathbf{c}$ in each of the two local coordinate systems at time t_1, and having the reference points at time t, it is possible to calculate the expected position of $\mathbf{c}$ w.r.t. each adjacent part, at time t. Note that d only acts as a scaling factor for the local coordinate system, and its exact value does not affect the final result. Fig. 1 illustrates the explained concepts.

The output of this step are multiple hypotheses of points of articulation for each pair of rigid parts. Each hypothesis for a point of articulation is described by its position in the local coordinate system of each incident part.

4.2 Verification and Selection of Hypotheses

By definition, rigid parts performing articulated motion keep a constant distance to the point of articulation. This is equivalent to saying that the positions of the

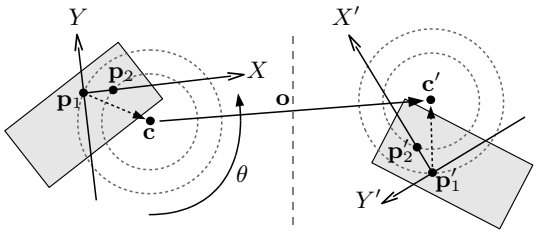

Fig. 1. Determining and encoding the point of articulation in the local coordinate system, during two time steps: left t_1, right t_2

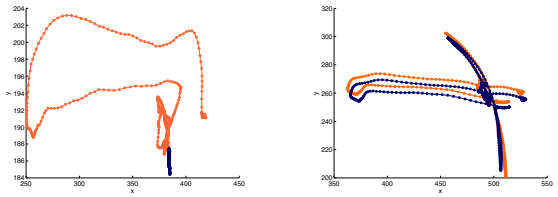

Fig. 2. Verification of hypotheses for points of articulation. Left: invalid hypothesis, right: valid hypothesis. Each curve shows the positions of the hypothesized point of articulation relative to one part. Positions corresponding to consecutive frames have been connected.

points of articulation calculated with the local coordinate systems of the two connected object parts coincide. We use this property to verify the previously generated hypotheses and select the valid ones.

Given a hypothesis for a point of articulation, two positions are computed for each frame of the video – one using the local coordinate system of each connected part. The inaccuracy μ of a hypothesis is the maximum of the distances between the two computed positions over the whole video. If μ is small the hypothesis is considered valid (in practice we take a threshold ϵ_v). If for a pair of rigid parts multiple valid hypotheses exist, the one with the smallest inaccuracy μ is taken. If no hypotheses with a small μ exists, the parts are not considered connected through a point of articulation. Fig. 2 shows positions for hypotheses of points of articulation generated during the verification step.

5 Experiments

In our experiments the Kanade-Lucas-Tomasi tracker [6] is used to track feature points (in this case corners) and supply the necessary trajectories. Only the points which could be tracked successfully over the whole sequence are used.

Sequence 1 is a video with a human, sequence 2 with a finger, and sequence 3 is a synthetic video, all undergoing a globally articulated motion with locally arbitrary deformations (see Fig. 5).

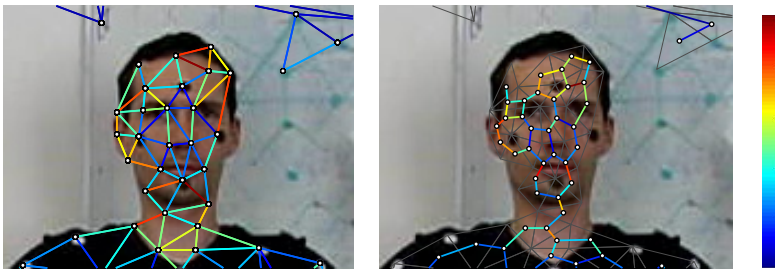

Fig. 3. Left: deformation of the edges over time. Right: dual graph of motion of triangles over time. Color map on the right encodes the degree of deformation and dissimilarity of motion of triangles, where red is high and blue is low.

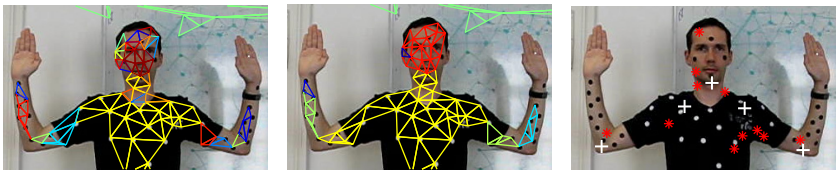

Fig. 4. Left, middle: Grouping result with global thresholds 0.25 and 0.6, respectively (different color means different part). Right: Comparison of identified points of articulation with baseline approach (red stars) and proposed approach (white crosses).

Fig. 3 provides an insight into the motion of the triangles in sequence 1. It visualizes the difference in deformation of edges and motion of triangles over time, which points out the challenge for the grouping process.

In Fig. 4 the results of two baselines are shown: (1) grouping of triangles with a global threshold (criterion: similar orientation of triangles over time) and (2) identification of points of articulation depending on the number of "rigid" edges in a triangle (criterion: two "rigid" edges and one highly deformed over time). Notice that a correct grouping into the whole hand, torso, upper and lower arms is not possible. The number and positions of the articulation points detected with the baseline is incorrect (see Fig. 5 for comparison with the proposed approach).

Fig. 5 collects the results with the proposed approach for sequences 1, 2 and 3 using the parameter values in Table 1. For sequence 1, the torso is connected with the base of the chin, because the features at the base of the chin slide when the head is tilted and remain in the same position in the image, creating a "rigid" triangle. In all three sequences the found points of articulation correctly connect the rigid parts. The threshold ϵ_v is sufficient to separate valid hypotheses from invalid ones, where there is no point of articulation in reality (e.g. head with background in sequence 1).

As in [20], a kinematic model (tree) of the objects can be defined by the found parts and their connections through joints.

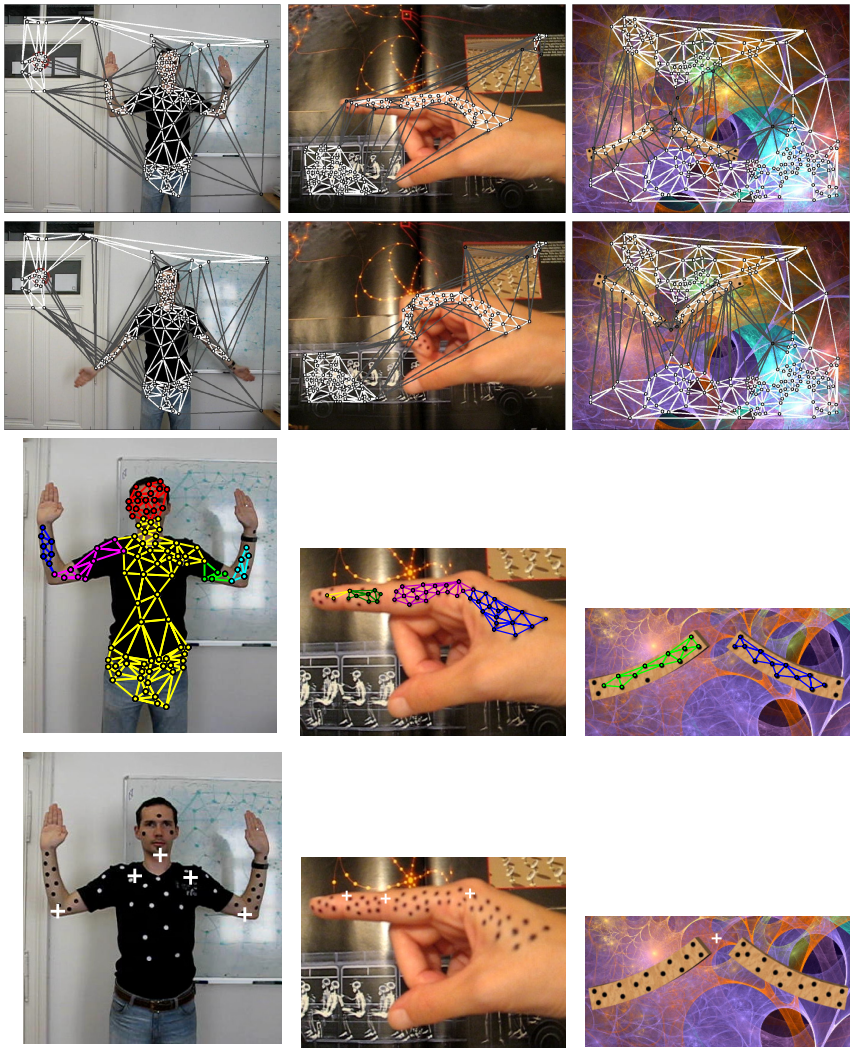

Fig. 5. Results obtained by the proposed method. First and second row: two frames of sequence 1,2, and 3 with spatio-temporal filtering result (white edges: *relevant* triangle and gray edges: *separating* triangle). Third row: Results of grouping process. Forth row: Identified points of articulation.

Discussion: The robustness of the tracker, the presence of salient points on the object(s), and the quality of the video should be sufficient to create the required observations (trajectories of feature-points).

The spatio temporal filtering (Sec. 3) will correctly identify triangles with all vertices on the same object if the motion of the objects relative to each other (distance variation) are larger than the local distance variation between neighboring feature-points of the same object.

Table 1. Values of the used parameters for sequences 1, 2 and 3

Sequence	ϵ_r (*relevant* triangles)	ϵ_a (generate hyp.)	ϵ_v (verify hyp.)
1 (human)	20	0.4	20
2 (finger)	10	0.4	10
3 (synthetic)	15	0.4	20

The grouping into rigid parts gives a correct result if the relative orientation change between two parts is larger than the local differences due to non-rigid deformation (e.g skin) or to imprecisions of the computed feature-point positions.

Points of articulation can be produced between any two pairs of detected rigid parts. To avoid detecting points of articulation between object parts and the background or between unrelated object parts, the unrelated parts (background) should translate with respect to each other.

In the presented approach no prior knowledge is used and it can be applied to videos with any arbitrary articulated or rigid object (i.e.: human, finger, animal, basket ball ...). The approach can only detect points of articulation, when there is articulated motion in the video. If the video contains a rigid foreground object moving in front of a "static" background, the result of the approach is a separation between foreground and background.

6 Conclusion

This paper presented a graph-based approach to identify the rigid parts of articulated objects and find their points of articulation. Trajectories of feature-points are used to describe the motion of objects in the scene. In the first frame a triangulation is built with the positions of the feature-points. A spatio-temporal filtering labels the triangles as *relevant* (object) or *separating* depending on the deformation of the edge lengths over time. A graph pyramid is used to group the *relevant* triangles into rigid parts depending on their orientation change over time. In a following step points of articulation connecting the rigid parts are identified. Experiments on natural and synthetic videos with articulation between quasi-rigid parts (skin, cloth) are used to verify the approach. In future work we plan to deal with input data containing incomplete trajectories and out of the plane motion.

Acknowledgments

This work has been partially supported by the Austrian Science Fund under grants S9103-N13 and P18716-N13.

References

1. Aggarwal, J.K., Cai, Q.: Human motion analysis: A review. Computer Vision and Image Understanding 73(3), 428–440 (1999)

2. Aggarwal, J.K., Cai, Q., Liao, W., Sabata, B.: Articulated and elastic non-rigid motion: A review. In: IEEE Workshop on Motion of Non-Rigid and Articulated Objects, pp. 2–14 (1994)
3. Altunbasak, Y., Eren, P.E., Tekalp, A.M.: Region-based parametric motion segmentation using color information. Graphical Models and Image Processing 60(1), 13–23 (1998)
4. Artner, N.M., Ion, A., Kropatsch, W.G.: Rigid part decomposition in a graph pyramid. In: The 14th Iberoamerican Congress on Pattern Recognition, pp. 758–765. Springer, Heidelberg (2009)
5. Artner, N.M., Ion, A., Kropatsch, W.G.: Multi-scale 2d tracking of articulated objects using hierarchical spring systems. Pattern Recognition 44(4), 800–810 (2010)
6. Birchfeld, S.: Klt: An implementation of the kanade-lucas-tomasi feature tracker (March 2008), `http://www.ces.clemson.edu/~stb/klt/`
7. Celasun, I., Tekalp, A.M., Gokcetekin, M.H., Harmanci, D.M.: 2-d mesh-based video object segmentation and tracking with occlusion resolution. Signal Processing: Image Communication 16(10), 949–962 (2001)
8. Chen, H.T., Liu, T.L., Fuh, C.S.: Segmenting highly articulated video objects with weak-prior random forests. In: European Conference on Computer Vision, pp. 373–385. Springer, Graz (2006)
9. Drouin, S., Hébert, P., Parizeau, M.: Incremental discovery of object parts in video sequences. Computer Vision and Image Understanding 110, 60–74 (2008)
10. Gavrila, D.M.: The visual analysis of human movement: A survey. Computer Vision and Image Understanding 73(1), 82–980 (1999)
11. Godec, M., Leistner, C., Saffari, A., Bischof, H.: On-line random naive bayes for tracking. In: ICPR, pp. 3545–3548 (2010)
12. Kropatsch, W.G., Haxhimusa, Y., Pizlo, Z., Langs, G.: Vision pyramids that do not grow too high. Pattern Recognition Letters 26(3), 319–337 (2005)
13. Lauer, F., Schnrr, C.: Spectral clustering of linear subspaces for motion segmentation. In: ICCV, pp. 678–685. IEEE, Los Alamitos (2010)
14. Li, H., Lin, W., Tye, B., Ong, E., Ko, C.: Object segmentation with affine motion similarity measure. In: Multimedia and Expo., pp. 841–844 (2001)
15. Moeslund, T.B., Hilton, A., Krüger, V.: A survey of advances in vision-based human motion capture and analysis. Computer Vision and Image Understanding 104(2-3), 90–126 (2006)
16. Nordberg, K., Zografos, V.: Multibody motion segmentation using the geometry of 6 points in 2d images. In: ICPR, pp. 1783–1787. IEEE, Istanbul (2010)
17. Ross, D.A., Tarlow, D., Zemel, R.S.: Learning articulated structure and motion. International Journal of Computer Vision 88(2), 214–237 (2010)
18. Tuceryan, M., Chorzempa, T.: Relative sensitivity of a family of closest-point graphs in computer vision applications. Pattern Recognition 24(5), 361–373 (1991)
19. Walther, T., Würtz, R.P.: Unsupervised learning of human body parts from video footage. In: 2nd Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment, pp. 336–343 (2009)
20. Yan, J., Pollefeys, M.: A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video. IEEE Trans. Pattern Anal. Mach. Intell. 30(5), 865–877 (2008)