

Improving Tracking Using Structure *

S. B. López Mármol^{1,5}, N. M. Artner², M. Iglesias^{3,1}, W. Kropatsch¹, M. Clabian², and W. Burger⁴

¹ PRIP, Vienna University of Technology, Austria
{salva, krw, mabel}@prip.tuwien.ac.at

² smart systems, Vienna, Austria
{nicole.artner, markus.clabian}@arcs.ac.at

³ Advanced Technologies Applications Center, Havana, Cuba
miglesias@cenatav.co.cu

⁴ Upper Austria University of Applied Sciences, Hagenberg, Austria
wilhelm.burger@fh-hagenberg.at

⁵ Computational Topology and Applied Mathematics, University of Seville, Spain
sallopmar@alum.us.es

In this paper we focus on the problem of improving the performance of object tracking in motion sequences by exploiting the spatial and temporal structure of the scene. First we show typical examples where tracking methods not using structural information tend to fail. For that purpose we recorded video sequences and tried to track the parts of a human with Mean Shift, a tracking method where the structure of the object is not employed. We decided to track humans because much work is and has been done in this area and also because the human body exhibits an obvious structure. The aim of this paper is to identify some problematic cases for tracking methods not using structure and to propose solutions using a structural approach.

1 Introduction

Object tracking in video sequences is a very important field in computer vision. Its aim is to find correspondences between objects in consecutive frames.

In many applications it is not enough to know the position of the whole object, but to track the parts of which the object is composed of (articulated objects). By tracking each part separately, additional information about the object can be retrieved which could be used to analyze the motion of the object and to handle occlusions in a more robust way.

For example, tracking the parts of humans provides more information about the course of motions. This information could be used to analyze the person's behavior and to recognize actions, i.e. walking, running and shaking hands.

Besides the problems occurring while tracking the whole object (changes in illumination and/or appearance, partial and full occlusion, fast or abrupt motion and camera motion) it is necessary to cope with the problem of how to distinguish between the tracked parts.

Looking at the example of tracking people again it is obvious that the distinction of right and left leg, foot, arm and hand is difficult. If the tracking method uses only the color of the object, one additional problem arises that it is not possible to distinguish between all parts just by their color. It could happen that people in a video sequence are dressed all over in the same color. Even in the best case, where the people wear differently colored clothing on each part of the body, the problem that hands and the head will have nearly the same (skin) color still remains.

So what can be done to solve those problems and improve tracking? We need some kind of structural information describing the relationships between the parts. With such information we could define the spatial relationships between the parts, impose certain constraints on the parts and their connections (i.e. the head is not allowed to disconnect from the torso) and overcome the problems with occlusion. Different representations can be used for encoding the structural information. We use graphs like in [23].

1.1 Object tracking

In this section we briefly discuss common tracking methods and give an overview of the state of the art of tracking methods using structural information.

Kernel tracking uses a rectangular or elliptical kernel referring to the object shape and appearance. The objects are tracked by calculating the motion of the kernel in consecutive frames. In this paper we use tracking with Mean Shift, a kernel tracking method, to run the necessary tests. Some relevant papers on kernel tracking are [9, 1, 4, 14, 2].

Silhouette tracking stands for tracking methods providing an accurate shape description for the target objects. The silhouette tracker tries to find the target object region in every frame with the help of an object model generated using the previous frames. Tracking is done by either shape matching or dynamic snakes. Examples for silhouette tracking are [15, 33, 32, 13].

*Partially supported by the Austrian Science Fund under grants P18716-N13 and S9103-N13.

Particle filtering became well-known in the field of computer vision because of the work of Isard and Blake [13]. It usually uses contours, color features, or appearance models [17, 12, 11, 5].

Optical flow is a dense motion field with a vector for each pixel describing direction and velocity of the motion. Black et al. present in [3] an optical flow approach dealing with multiple motions and Sidenbladh model in [24] the motion of articulated 3D objects with the help of optical flow.

Point trackers represent target objects by points. Applications of point tracking are [21, 30, 22, 7].

1.1.1 State of the art of tracking methods using structural information: Tracking methods using structural information often employ graphs to describe the structure. Such graphs are built of nodes, spatial and temporal edges. In the nodes attributes like size, average color and position of the corresponding pixels (region) can be stored. The spatial edges are used to specify the spatial relationships (adjacency, border) between the nodes (regions). Temporal edges are applied to describe the correspondence between moving parts in consecutive frames.

An application of this representation can be found in [16] where a so-called Spatio Temporal Region Graph (STRG) is used to represent the content of video sequences.

Graciano et al. propose to describe objects by two different types of Attributed Relational Graphs (ARGs): the intra-frame ARG and the inter-frame ARG. These ARGs carry both local and relational information about the objects. Their aim is to recognize and track objects. The recognition part is done by inexact graph matching between an input video and a model image. The search for a suitable homomorphism between ARGs is achieved through a tree-search optimization algorithm and the minimization of a pre-defined cost function. Another example is [27] where Taj et al. solve the problem of multiple target tracking with an algorithm based on color change detection and multi-feature graph matching.

In [18] Ma et al. try to handle the multiple target tracking problem with a maximum a posteriori formulation. They use a graph to store the detected regions as well as their associations over time. A different approach to apply graph-like structures in tracking is proposed by Conte et al. in [10]. They are using graph pyramids to describe each frame in several levels of detail. With this graph pyramids they are able to label each pixel of a moving foreground region during partial occlusions. The label identifies to which object the corresponding pixel belongs.

Rehg and Kanade deal with self-occluding articulated objects in [20]. Unlike Conte et al. this approach uses a kinematic model to predict occlusions and applies a graph with just one level. In their experiments they track the fingers of a hand. They can distinguish between occluded cases by the order of the templates related to the fingers and their ordering relative to the camera. Regh and Kanade employed a directed occlusion graph to represent the occlusion relations for their multi-body system, the hand.

Both [6] and [19] present a contour-based algorithm for tracking moving objects. In [6] a bipartite graph is used whereas the two classes of nodes are called profile and ob-

ject nodes. The profile nodes are the nodes of the previous frame and the object nodes those from the actual frame. A bipartite matching algorithm is applied to find the best match between these two frames and resolve the identities of the nodes (to which object they belong). In [19] Ofer et al. use Region Adjacency Graphs (RAGs) to construct the tracked contour. Their algorithms depend on the RAGs of the input frames. Based on the RAGs the object's contour is divided into sub curves while junctions of the contour are derived.

Tang et al. model in [28] target objects with SIFT (scale-invariant feature transform) and represent their relationships with ARGs. They describe an interesting mechanism to update the object model by adding new stable features and deleting old inactive features.

There are some approaches that try to match the projection of a 3D model in 2D video sequences. Sminchisescu and Triggs present in [25] an approach that uses the structural constraints on human motion, together with a special search strategy for this purpose.

1.2 Organization of paper

The remaining parts of this paper are organized as follows: Section 2 summarizes tracking with Mean Shift. Subsequently section 3 presents the test cases and the results with Mean Shift. In Section 4 we propose how the cases can be solved with the help of structural information. Section 5 shows some general ideas on how structural information can be introduced in a tracking process.

2 Recall: Tracking with Mean Shift

The Mean Shift algorithm is a statistical and robust procedure which finds local maxima in any probability distribution. For that it uses a search window positioned over a section of the probability distribution. Within this search window the maximum can be determined by a simple average computation. Then the search window is moved to the position of the maximum and the calculation is repeated until the algorithm converges. The convergence criterion is that it is not possible to find a "better" position in the search window and this means that a local maximum has been found.

To apply the mean shift algorithm in a tracking procedure it is necessary to adjust the data of the video frames. For this purpose every pixel in a frame gets a probability value $P(u, v)$. P indicates how likely it is that the related pixel belongs to the target object. The implementation of the tracking with Mean Shift in this paper mainly follows the ideas in [9, 1]. Therefore the probability value depends on the color of the pixel.

In the first frame at least one target object is selected. With the help of this selection a target model \hat{q} is created in the form of a 3D histogram. Every dimension of the histogram corresponds to one channel of the HSV color space. The histogram is subdivided into bins $u = 1 \dots m$ to reduce the amount of data and to cluster similar colors. A histogram bin's probability value is calculated as [9]:

$$\hat{q}_u = C \sum_{i=1}^n k \left(\|x_i^*\|^2 \right) \cdot \delta(b(x_i) - u), \quad (1)$$

where C is a normalizing factor such that

$$\sum_{u=1}^m \hat{q}_u = 1. \quad (2)$$

k in Equation (1) stands for the Epanechnikov kernel [8] and is used to control the influence of the pixels in the user selection on the target model. The pixels are weighted depending on their distance to the center of the selection.

x_i are the pixel positions in the image and x_i^* are the normalized pixel positions. b is a function mapping a pixel in the 2D space to the 1D space of the histogram bin indices. Depending on the HSV value of a pixel the function b provides the index of the corresponding histogram bin. δ is the Kronecker delta function. For more details see [9, 1].

As proposed in [9] we calculate a candidate model \hat{p} in addition to the target model \hat{q} . The candidate model

$$\hat{p}_u(\mathbf{c}) = C \sum_{i=1}^n k\left(\|x_i^*\|^2\right) \cdot \delta(b(x_i) - u), \quad (3)$$

is created from the pixels in a so called candidate window $\mathbf{s}_c = w_s \times h_s$ at the actual position $\mathbf{c} = (c_x, c_y)$ of the target object. The candidate window \mathbf{s}_c is smaller than the search window \mathbf{s} and should prevent the tracking from considering too many background pixels in the calculation of the candidate model. In our implementation the candidate window size is 70% of the search window size which is calculated using the moment of zero order like in [4]. Equation (3) is a reformulation of Equation (1) for the position \mathbf{c} .

The candidate and the target model are used to acquire the position

$$\mathbf{c} = \frac{\sum_{i=1}^n x_i \cdot w_i}{\sum_{i=1}^n w_i} \quad (4)$$

of the target object within the search window with the Mean Shift algorithm. The pixels used to calculate position \mathbf{c} are weighted according to

$$w_i = \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\mathbf{c})}} \cdot \delta(b(x_i) - u), \quad (5)$$

whereas \hat{q}_u and $\hat{p}_u(\mathbf{c})$ are the target and the candidate model. This weight w_i denotes the probability value P already defined in the beginning of this section where we mentioned that it is necessary to adjust the data of the video frames.

Furthermore the candidate model could be applied in the adaptation of the target model. With the help of the Bhattacharyya coefficient

$$B = \sum_{u=1}^m \sqrt{\hat{p}_u(\mathbf{c}) \cdot \hat{q}_u}, \quad (6)$$

described in [9], the distance between the target and the candidate model could be calculated (now it is clear why the constant C is necessary). The Bhattacharyya coefficient lies in the interval $[0, 1]$, where 0 means that there is no correlation between the two models.

In this paper we use a very simple method to adapt the target model called parameterized neglecting. From the original target model \hat{q} for the first frame, the target model for the previous frame \hat{q}_{t-1} and the candidate model $\hat{p}(\mathbf{c})$ we calculate the actual target model

$$\hat{q}_t = \hat{q} \cdot \alpha_1 + \hat{q}_{t-1} \cdot \alpha_2 + \hat{p}(\mathbf{c}) \cdot \alpha_3, \quad (7)$$

where $0 \leq \alpha_1 \leq 1$, $0 \leq B \leq 1$, $\alpha_2 = ((1 - \alpha_1) \cdot B)$, $\alpha_3 = (1 - \alpha_1) - \alpha_2$ and $\alpha_1 + \alpha_2 + \alpha_3 = 1$. α_1 is a constant affecting the influence of the original target model on the actual target model.

Besides the position of the target object we determine width, height and orientation as in [4] out of the probability distribution within the search window. The probability distribution for this properties is calculated with the help of the actual target model. The width

$$w_{target} = 2 \cdot \left(\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2} \right)^{\frac{1}{2}} \quad (8)$$

and the height of the target

$$h_{target} = 2 \cdot \left(\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2} \right)^{\frac{1}{2}} \quad (9)$$

are calculated with the help of the statistical moments m of first, second and zero order:

$$\begin{aligned} a &= \frac{m_{20}}{m_{00}} - c_x^2, \\ b &= 2 \cdot \left(\frac{m_{11}}{m_{00}} - c_x \cdot c_y \right), \\ c &= \frac{m_{02}}{m_{00}} - c_y^2. \end{aligned}$$

Using the central moments μ of first and second order the orientation of the target object can be determined as follows:

$$\varphi_{target} = \frac{1}{2} \tan^{-1} \left(\frac{2 \cdot \mu_{11}}{\mu_{20} - \mu_{02}} \right). \quad (10)$$

3 Problems of tracking with Mean Shift

The tracking with Mean Shift described in Section 2 works reliably and robustly in videos with a rigid target object, no occlusions, no similar or even equal objects like the target object and more or less constant lighting. In this section we like to present cases where tracking with Mean Shift fails.

We decided to use people as target objects because they have an obvious structure. The human body can be divided into head, torso, arms, hands, legs and feet. As the tracking with Mean Shift in this paper uses color information to track objects we decided to track four parts independently: head, torso and arms together (as one part) and hands (see Figure 1).

The following sections show how Mean Shift reacts in different cases of occlusion and appearance change.

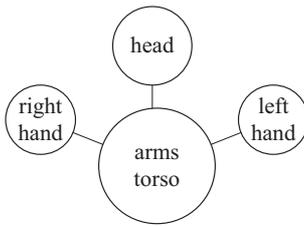


Figure 1: Decomposition of object for tracking with Mean Shift.

3.1 Occlusion by similar object

If the target object gets occluded by a similar object in the scene, it is not sure that Mean Shift follows the right object after the occlusion. The reason is that the target model of the tracked object also fits the other object very well.

In Figure 2 an example of that case can be found. The similarity of the color information of two different parts (head and hand) results in a failure in the tracking. Mean Shift finds a better match of the hand in the head than in the “real” hand.

3.2 Occlusion by equal object

The occlusion of the target object by an equal object is an even more problematic case than that of Section 3.1. In this case the models describing the objects are nearly the same.

In Figure 3 the occlusion happens between the two hands of the person. Obviously, non-structured methods cannot distinguish between the two hands after they get close enough.

It is possible that Mean Shift successfully identifies the hands after the occlusion. But there is no guarantee and it is likely that Mean Shift confuses the hands or even one hand will have a higher probability and hence attract both trackers. This outcome can be seen in Figure 3.

3.3 Complete Occlusion by different object

This case is quite different from the previous ones. Obviously, Mean Shift loses an object if it is completely occluded, and it cannot find the object when it becomes visible again without any help (see Figure 4).

Of course there are statistical methods to estimate the position and motion of the object during the occlusion. Very well known state estimators are the Kalman filter [31] and the Particle filter [17]. The big problem of this estimators is that they assume that the behavior of the object does not change during occlusion. That means that the object should not significantly change its moving direction and speed.

In the example of this case the information about the direction and the speed of the movement of the object would not help to solve the case. The reason is that the object changes its moving direction and moves up instead of down (what is expected by the human observer).

3.4 Change of objects appearance

Another problem of tracking with Mean Shift is a change in the target objects appearance. In this case we are not talking about changes in illumination, but significant color changes.

The only possibility to solve such a case successfully is to adapt the target model to the changes in the object. Ex-

amples for the adaption of the target model can be found in [29, 26]. This task is not that easy as it seems. The most difficult question is how to differentiate between changes in the object through appearance changes and changes through occlusions? As a matter of fact we cannot be totally sure about what is happening with the object.

Figure 5 shows a case of appearance change. In this example the tracked person is turning around and because of that the appearance of the head changes drastically. As the person is looking at the camera again, a simple workaround would be to prevent the search window from shrinking so that Mean Shift is able to catch the face again (see Figure 6). The advantage of this simple idea is that the target model is not changed unnecessarily. Unfortunately this solution is only reasonable in this particular case.

3.5 Different kinds of occlusions combined

This case combines different kinds of occlusions and should be a challenge for every tracking procedure. There are occlusions with equal and similar objects. As you can see in Figure 7 we tried to track the movements of two persons (head, torso and arms together and hands). Mean Shift fails as expected and loses track of objects from frame to frame. The reasons why tracking with Mean Shift is failing are already described above in Sections 3.1, 3.2 and 3.3.

4 Ideas for solving the cases

In this section we suggest for every case from Section 3 how to solve it using structural information. The following list explains the structural information we are using for our proposed solutions.

adjacency: parts are neighbors

connectivity: describes accurately how parts are connected
i.e. head is connected to the torso in a special way

ratio: ratio between the sizes of the parts can be used to improve results, we know that all parts of an object will scale with the same factor (scale invariance)

actual state of part: position, orientation, shape and size

constraints on parts: forbidden behavior of the parts (examples can be found in the following sections)

4.1 Occlusion by a similar object

The failure in the tracking described in Section 3.1 (see Figure 2) could be prevented if the adjacency information “hand is connected to arm” is taken into consideration. Additionally the constraints that hand and arm are not allowed to split or to merge help to solve the case.

4.2 Occlusion by equal object

In the case of Section 3.2 (see Figure 3) not only the information that each hand is connected to a arm is needed, but also how they are connected through joints (connectivity). For this connectivity analysis we need properties like relative position, orientation and shape. Of course again the both constraints that parts do not split and merge can be useful.

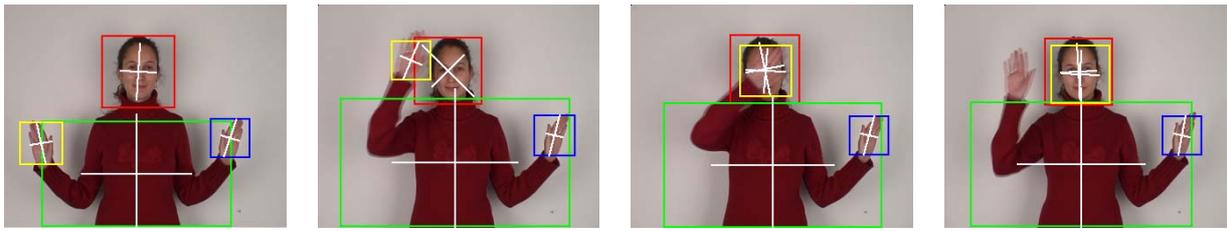


Figure 2: Occlusion by a similar object.

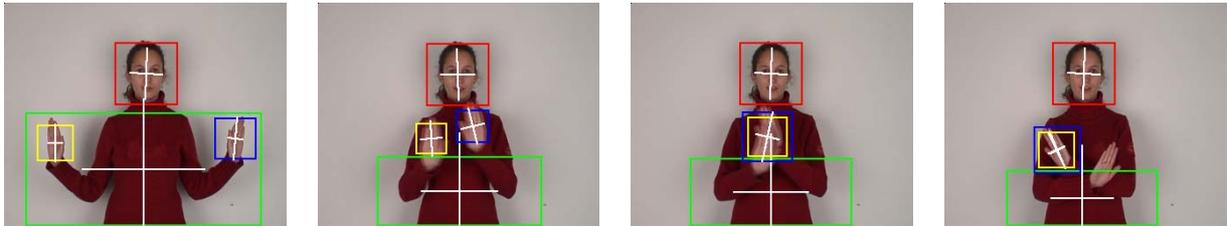


Figure 3: Occlusion by equal object.

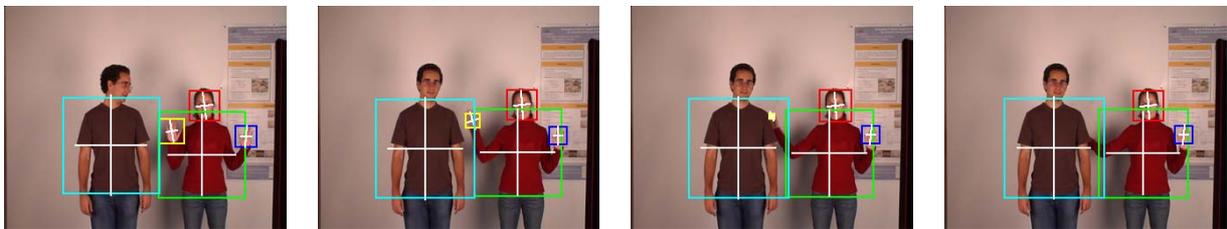


Figure 4: Complete Occlusion by different object.



Figure 5: Change of objects appearance.

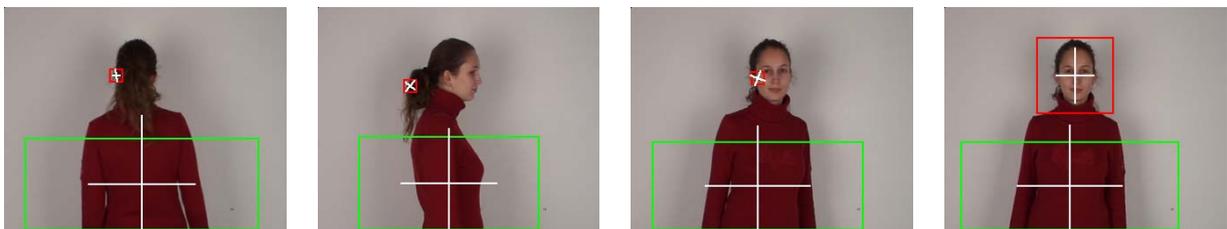


Figure 6: Change of objects appearance. Simple fix.

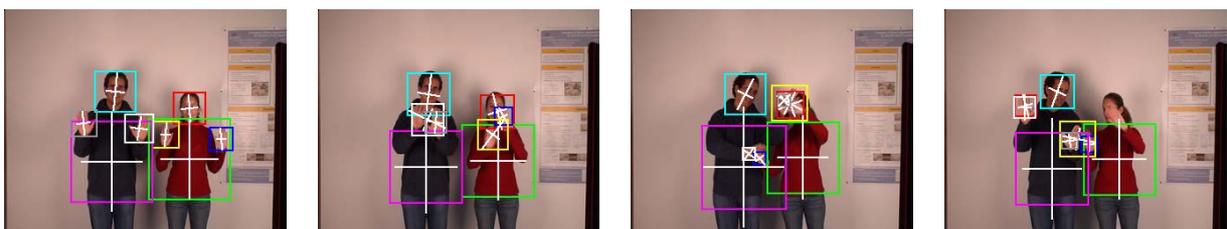


Figure 7: Different kinds of occlusions combined.

4.3 Complete Occlusion by different object

Section 3.3 describes a case where a part of an object is occluded by another object in the scene (see Figure 4). The constraints “parts do not split” and “parts maintain connectivity”² provide the information that the hand, which is not visible anymore, is still there and just got occluded. By using the structural information that hand and arm are adjacent and analysing the connectivity of hand and arm (with the help of the properties of the parts), it is possible to estimate the position of the hand during occlusion and afterward. To determine a fitting size of the search window for the reappearing part the ratio between both hands can be used.

4.4 Change of objects appearance

To avoid the loss of the head in the case in Section 3.4 (see Figures 5 and 6) again adjacency and connectivity information is used, but in this case between head and torso. It is known that parts maintain the connectivity and are not allowed to split. To estimate where the head must be located during occlusion we would suggest to use the relative position, orientation and shape of the torso. Using the ratio between the size of the head and the torso the size of the head and its search window can be estimated.

4.5 Different kinds of occlusions combined

In Section 3.5 we describe a case where different kinds of occlusions are combined (see Figure 7). This case is the most challenging one. It is possible to solve it by combining all the structural information introduced in the beginning of this section. That means using adjacency, connectivity, the ratio between the parts, the actual state of the parts (their properties) and the constraints (do not merge and split, maintain connectivity of object).

Table 1 summarizes our solutions for the cases.

5 Tracking with Structure

This section gives some ideas on how to use structure in a tracking process. The problem can be divided in three:

- How to represent the structural information.
- How to analyse structural information of the objects in the scene.
- Find a method to apply structural information to improve tracking, especially in the situations where non structured methods can hardly work well.

These three points are explained in the next sections.

5.1 Structure of the object

The structural information of the objects in the scene is considered known beforehand in this approach. However, it is still necessary to have a representation of this knowledge in order to use it. The first work found on structural descriptions is in [23]. We divide the structural description of an object in two: description of the parts and description of their relationships. The division of the object in parts is usually driven by the articulation points.

²“maintain connectivity” means that the parts of the object are not allowed to disappear, the whole object with all its parts should be maintained.

structure	3.1	3.2	3.3	3.4	3.5
adjacency					
hand and arm	x	x	x		x
head and torso				x	x
connectivity					
hand and arm		x	x		x
head and torso				x	x
ratio					
between head and torso				x	x
between both hands			x		
actual state of parts					
position		x	x	x	x
orientation		x	x	x	x
shape		x	x	x	x
size				x	x
constraints: parts					
do not merge	x	x			x
do not split	x	x	x	x	x
maintain connectivity			x	x	x

Table 1: Solving the cases. The first table row lists the section numbers where the corresponding cases are described. The first column contains the structural information. The x marks which structural information could be used to solve the various cases.

The description of the parts is necessary to detect them in every frame. Depending on which method is used to track the parts different kinds of information will be needed. We could include features like color, size and shape of each part. It can be useful to know which other parts of the object are similar according to the criteria used by the tracking method. This could help to improve the tracking by checking if one part has been wrongly identified as one of its similar parts.

The description of the relationships between the parts includes the structural information that can be used to improve tracking. Using it, it is possible to tell if a configuration of the parts detected in the scene is correct or not according to structural constraints. It can also be used to estimate the position of a “lost” part by the knowledge of its relationships with the other parts. This description of the relationships includes adjacency information, possible relative positions and connectivity.

The typical representation for structural descriptions was introduced in [23]. It is mainly a graph where the nodes include the information of the parts and the edges contain a description of the relationships between parts.

5.2 Extracting structure from the scene

In this section we address the problem of matching the structural description with the objects in the scene. We approach this problem in two different ways:

First, without using any additional unstructured method (like Mean Shift) and just using structural information in every step of the tracking process. While in many computer vision applications the tasks segmentation, object detection and tracking are often solved independently, the underlying idea of the approach is to determine a structure within the observed scene that is tracked over time. The

structure is represented by a graph or a graph pyramid of the segmented image and correspondence can be found by graph matching. The advantage of using a graph pyramid is that it would allow grouping of structures and hence simplify graph matching. As graph matching is NP-complete, it is only fast enough on graphs with a few nodes. Higher levels of the pyramid, containing fewer nodes, can be efficiently matched. This matching can then be used to guide the matching of lower levels of the pyramid. The graph representation allows the detection and correction of over- and undersegmentation and therefore leads to a new representation of the scene structure. In this approach the steps of segmentation, detection and tracking are solved in a novel, more integrated way. The main drawback of this approach is the need of a segmentation of each frame and the corresponding graph matching process. Because of this, this approach can hardly lead to a real-time implementation.

Second, by tracking the parts of the objects separately using an unstructured approach like Mean Shift and then studying the relationships between them. This approach has a great advantage: the method used for tracking can label the found parts, making the graph matching process easier (even not necessary in the ideal case in which every part is found and identified correctly). However, tracking methods do not provide 100% accurate results, the point of this approach is to check whether these results are correct or not according to structural information. By doing this, we keep the advantages of non-structured methods, while adding more robustness with the addition of further structural checks. The main drawback of this approach is that usually methods that do not use structure do not provide enough information for a complete structural analysis.

The combination of the first and the second approach derives a third approach, where segmentation and graph matching could be used only for ambiguous cases where the areas of the image and the parts of the structural description of the objects are missing (or represent an inconsistent state according to structural description). This analysis can be helped by the information provided by the tracker (i.e. mean-shift) of the parts to get fast and robust results. In that way it is possible to get full understanding of the structure in the scene when needed, while maintaining good performance in the cases where the parts are tracked correctly by the tracking method (without structural information).

5.3 Improving tracking with Mean Shift

In this section we describe how to introduce structural information in a non structured tracking process, concretely tracking with Mean Shift.

As a first step to improve tracking, it is necessary to detect when the tracking process is failing. This are some clues that can indicate a difficult or structurally inconsistent situation for the tracker:

- Some tracking algorithms provide values of how exact the matching of an object in a frame is. These values can help to decide when to do further structural checks.
- Inconsistent relative positions of the search windows. For example, two parts overlapping or getting too far away

can indicate a difficult situation for the tracker (occlusion) or an structurally inconsistent configuration.

- Big variations in the size ratio of the different parts.

When any of these cases is detected a further structural analysis is needed to correct the tracking or prevent errors. Unfortunately, the information that Mean Shift provides about the parts (only size and position of their search windows) is not enough to correct some of the cases. Even so, part of the structural information described in Section 4 can be extracted using the data that Mean Shift provides:

adjacency: If two parts are adjacent then their search windows must be overlapping or nearby. This can help when searching for a “lost” part if it is adjacent to a part that is successfully tracked.

connectivity: A further analysis of connectivity is not possible.

ratio: The size of the parts is known because it can be calculated using the zero moment (see Section 2). This allows an analysis of the ratio and therefore improve the tracking (see Sections 4.3 and 4.4).

actual state of parts: Mean Shift provides information about position, orientation and size. However, the values delivered for orientation are not accurate enough. It is not possible to extract useful knowledge of the shape of a part from the results of Mean Shift.

constraints on parts: These constraints can be introduced in the Mean Shift algorithm so it can identify structurally inconsistent states that indicate a failure in the tracking process.

In order to be able to extract concrete connectivity and shape information an additional, more detailed analysis of the scene (i.e. segmentation) is necessary. This analysis can be driven by the results of Mean Shift to get more accurate results and reduce computational costs.

6 Conclusion and future work

This paper presents the most common cases where methods not using structural information are failing. Tests have been done using tracking with Mean Shift. Results got analysed and hints for a solution with structural information have been given. Aspects of structure like how to describe it, identify it in the scene and use it in the tracking process are addressed. In future we intend to concentrate on our structural approach, formulate it in more detail and repeat our tests with the help of structural information.

References

- [1] J. G. Allen, R. Y. D. Xu, and J. S. Jin. Object tracking using camshift algorithm and multiple quantized feature spaces. In *Proceedings of the Pan-Sydney area workshop on visual information processing*, pages 3–7, Darlinghurst, Australia, 2004. Australian Computer Society, Inc.

- [2] C. Beleznai, B. Frühstück, and H. Bischof. Human tracking by fast mean shift mode seeking. *Journal of Multimedia*, 1(1):73–76, April 2006.
- [3] Michael J. Black and P. Anandan. The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996.
- [4] G. R. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, 2, April–June 1998.
- [5] L. Changjiang Yang; Duraiswami, R.; Davis. Fast multiple object tracking via a hierarchical particle filter. *ICCV*, 1:212–219 Vol. 1, 17-21 Oct. 2005.
- [6] H.-T. Chen, H.-H. Lin, and T.-L. Liu. Multi-object tracking using dynamical graph matching. *CVPR*, 2:210–217, 2001.
- [7] D. Chetverikov, D. Svirko, and D. Stepanov. The trimmed iterative closest point algorithm. In *ICPR*, volume 3, pages 545–548, Quebec, Canada, August 2002. IEEE Computer Society.
- [8] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 24(5):603–619, 2002.
- [9] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *PAMI*, 25(5):564–575, 2003.
- [10] D. Conte, P. Foggia, J.-M. Jolion, and M. Vento. *Graph-Based Representations in Pattern Recognition*, chapter A Graph-Based, Multi-Resolution Algorithm for Tracking, pages 193–202. Springer Berlin / Heidelberg, 2005.
- [11] Jacek Czyz, Branko Ristic, and Benoit M. Macq. A particle filter for joint detection and tracking of color objects. *Image and Vision Computing*, 25(8):1271–1281, 2007.
- [12] C. Hue, J.-P. Le Cadre, and P. Perez. Tracking multiple objects with particle filtering. *Aerospace and Electronic Systems, IEEE Transactions on*, 38(3):791–812, July 2002.
- [13] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, 1998.
- [14] M. Isard and J. MacCormick. Bramble: a bayesian multiple-blob tracker. In *ICCV*, volume 2, pages 34–41, Vancouver, Canada, July 2001. IEEE Computer Society.
- [15] J. Kang, I. Cohen, and G. Medioni. Object reacquisition using invariant appearance model. In *ICPR*, volume 4, pages 759–762, Cambridge, UK, August 2004. IEEE Computer Society.
- [16] J. K. Lee, J. H. Oh, and S. Hwang. Clustering of video objects by graph matching. *IEEE International Conference on Multimedia and Expo*, pages 394–397, July 2005.
- [17] P. Li and H. Wang. *Computer Vision/Computer Graphics Collaboration Techniques*, chapter Object Tracking with Particle Filter Using Color, pages 534–541. Springer Berlin / Heidelberg, June 2007.
- [18] Y. Ma, Q. Yu, and I. Cohen. *Advances in Visual Computing*, chapter Multiple Hypothesis Target Tracking Using Merge and Split of Graph’s Nodes, pages 783–792. Springer Berlin / Heidelberg, 2006.
- [19] O. Miller, E. Navon, and A. Averbuch. Tracking of moving objects based on graph edges similarity. *International Conference on Multimedia and Expo*, 3:73–76, July 2003.
- [20] J. M. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. In *ICCV*, pages 612–617, Boston, USA, 1995. IEEE Computer Society.
- [21] V. Salari and I. K. Sethi. Feature point correspondence in the presence of occlusion. *PAMI*, 12(1):87–91, January 1990.
- [22] K. Shafique and M. Shah. A non-iterative greedy algorithm for multi-frame point correspondence. In *ICCV*, volume 1, pages 110–115, Nice, France, 2003. IEEE Computer Society.
- [23] L. G. Shapiro and R. M. Haralick. Structural descriptions and inexact matching. *PAMI*, 3(5):504–519, Sept. 1981.
- [24] Hedvig Sidenbladh, Michael J. Black, and David J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *ECCV*, pages 702–718, Dublin, Ireland, 2000. Springer-Verlag Berlin Heidelberg.
- [25] Cristian Sminchisescu and Bill Triggs. Covariance scaled sampling for monocular 3D body tracking. In *CVPR*, pages 447–454. IEEE Computer Society, 2001.
- [26] H. Stern and B. Efron. Adaptive color space switching for face tracking in multi-colored lighting environments. In *International Conference on Automatic Face and Gesture Recognition*, pages 236–241, Washington, DC, USA, Mai 2002. IEEE Computer Society.
- [27] M. Taj, E. Maggio, and A. Cavallaro. *Multimodal Technologies for Perception of Humans*, chapter Multi-feature Graph-Based Object Tracking, pages 190–199. Springer Berlin / Heidelberg, 2007.
- [28] Feng Tang and Hai Tao. Object tracking with dynamic feature graph. In *ICCCN '05: Proceedings of the 14th International Conference on Computer Communications and Networks*, pages 25–32, Washington, DC, USA, 2005. IEEE Computer Society.
- [29] H. Uemura, J. K. Tan, and S. Ishikawa. A color tracker employing a two-dimensional color histogram under changeable illumination. In *Annual Conference on IEEE Industrial Electronics*, pages 3273–3278, Paris, Frankreich, November 2006. IEEE Computer Society.
- [30] C. J. Veenman, M. J. T. Reinders, and E. Backer. Resolving motion correspondence for densely moving points. *PAMI*, 23(1):54–72, January 2001.
- [31] G. Welch and G. Bishop. An introduction to the kalman filter. Technical report, University of North Carolina, Chapel Hill, NC, USA, 1995.
- [32] A. Yilmaz, L. Xin, and M. Shah. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *PAMI*, 26(11):1531–1536, November 2004.
- [33] C. Yunqiang, R. Yong, and T. S. Huang. JPDAF based HMM for real-time contour tracking. In *CVPR*, volume 1, pages 543–550, Kauai, HI, USA, December 2001. IEEE Computer Society.