

Model-Based Tracking of Self-Occluding Articulated Objects

James M. Rehg*

Takeo Kanade

Dept. of Electrical and Computer Eng.
Carnegie Mellon University
Pittsburgh, PA 15213
jimr@cs.cmu.edu

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
tk@cs.cmu.edu

Abstract

Computer sensing of hand and limb motion is an important problem for applications in human-computer interaction and computer graphics. We describe a framework for local tracking of self-occluding motion, in which one part of an object obstructs the visibility of another. Our approach uses a kinematic model to predict occlusions and windowed templates to track partially occluded objects. We present off-line 3D tracking results for hand motion with significant self-occlusion.

1 Introduction

Measurement of human hand and body motion is an important task for applications ranging from athletic performance analysis to advanced user interfaces. Human hands and limbs can be modeled as systems of rigid bodies connected together by joints with one or more degrees of freedom (DOFs). Thus human sensing can be formulated as the real-time visual tracking of articulated kinematic chains.

At a high image sampling rate (10 Hz or more), the local tracking problem consists of recovering incremental motions between successive frames. This paper describes a new approach to the local tracking of articulated objects based on a layered template representation of self-occlusions. A kinematic model is used in our approach to order the templates by their visibility to the camera. Window functions attached to each template capture the effects of occlusion, and partially-occluded templates are registered to the image sequence by minimizing an SSD residual

error. We present experimental results for 3D hand tracking under a significant amount of self-occlusion.

2 Tracking Self-Occluding Objects

Self-occlusion is an ubiquitous property of articulated object motion, that challenges standard template-based tracking algorithms. This section describes a local invariant of self-occluding motion that makes it possible to predict occlusions using a kinematic model. The three possible *occlusion relations* between two rigid bodies are illustrated in Fig. 1, for the first and second fingers of the hand. In this example, the hand rotates around the middle finger axis with the fingers held rigid. There is no occlusion in the disjoint case, shown in (b), and a template assigned to each finger can be registered with the image using standard techniques.

In Figs. 1 (a) and (c), the finger templates overlap due to occlusion, and pixels in the overlapping region of the image must be assigned to the correct template before registration can occur. The two occluded cases are distinguished by the order of the templates relative to the camera. A *visibility order* for a set of templates has the property that each template in the list will not be occluded by any of the templates that follow it. Templates can be ordered arbitrarily in the disjoint case. A set of ordered templates make up a layered representation for occluding motion [1].

Tracking requires the simultaneous solution of two problems: determining the visibility order for the templates that describe the object, and registering the overlapping templates to the input image. In bottom-up approaches to occlusion analysis, template order is estimated from image motion [2, 14] or contours [6]. This paper explores an alternative, top-

*Supported by the NASA George Marshall Space Flight Center, GSRP Grant NGT-50559.

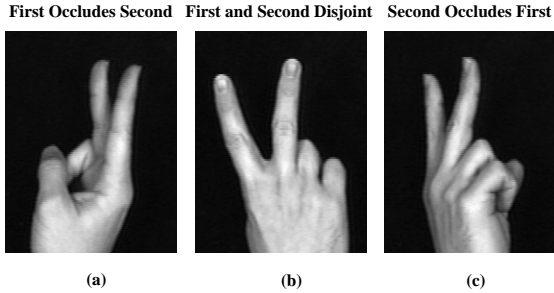


Figure 1: Three snapshots from a motion sequence, illustrating the possible occlusion relations between the first and second fingers of the hand.

down approach which uses the kinematic model in conjunction with a high image sampling rate to partition the state space into regions with a fixed visibility order. In this approach, the visibility order for the current frame is predicted from the previous state estimate and used to constrain image interpretation.

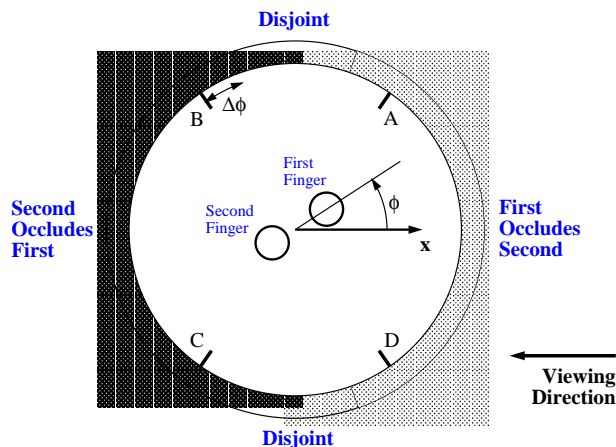


Figure 2: The partition of the rotation space (unit circle) into regions with invariant occlusion relations. This is a top view of the scene in Fig. 1, with the camera located on the right. ϕ gives the rotation of the hand relative to the camera.

The visibility order as a function of the hand state is illustrated, for the two finger example, in Fig. 2, which depicts the space of hand rotations as a unit circle. The angles marked A, B, C, D denote *occlusion events*, points at which the occlusion relations change. Passing through $\phi = A$, for example, causes a transition from (a) to (b). The amount of hand rotation between frames is limited by the sampling rate to a small angle, $\Delta\phi$. Therefore, in local tracking the state estimate for the current frame is restricted to a

motion interval of $\pm\Delta\phi$ around the previous estimate. Since the occlusion events are sparsely distributed, the occlusion relations will be constant from frame to frame across most of the image sequence. When the motion interval contains an occlusion event, the visibility order will change. However, the transition is always between an occluded and a disjoint case. As a result, the onset of occlusion can be anticipated by assigning the occluded order to the disjoint case near the event.

The preceding arguments demonstrate the existence of a *local occlusion invariant* for the object, which can be identified by constructing the state partition shown in Fig. 2 as dark and light grey bands. The partition is obtained by extending the occluded regions into the disjoint ones by the motion bound $\Delta\phi$. It has the following property: *The visibility order for the state at time k , as determined by membership in the partition, is fixed under all bounded motions at time $k + 1$.* As a result, it constitutes a prediction of the visibility order for the next frame. This paper presents a local tracking algorithm based on visibility order prediction.

3 Existence Conditions for Invariant Visibility Orders

A key step in our tracking algorithm is the use of the kinematic model to predict a visibility ordering between templates that holds for all bounded object motions between image frames. This section derives general existence conditions for such invariant visibility orders. Specific rules for the on-line generation of visibility orders for the hand are described in [11].

3.1 Binary Occlusion Relations

The simplest type of visibility order is a binary occlusion relation between two convex bodies undergoing bounded motion. In *binary occlusions*, the occluding object A is fully visible, while the occluded object B is obscured. The disjoint relation, $A \equiv B$, holds when the two bodies don't overlap in the image under the allowed motion. The occlusion relation, $A \succ B$, is true if A and B are not disjoint and A occludes B whenever their image plane projections overlap. A multi-body system has a local occlusion invariant if, for a given bounded motion, one of $A \equiv B$, $A \succ B$, or $B \succ A$ is true for each pair of bodies, A and B .

The first step in analyzing the existence of binary occlusion relations for an arbitrary pair of bodies is

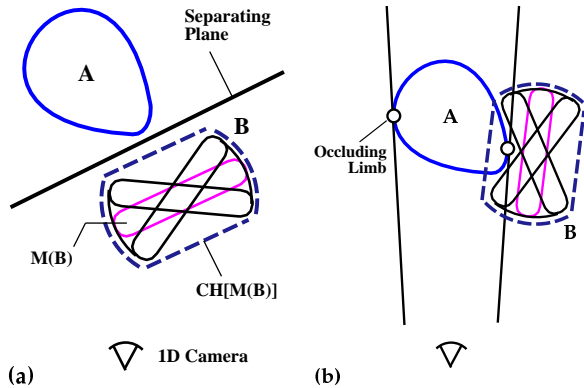


Figure 3: Occlusion relations for 2D objects viewed by a 1D camera. (a) Sufficient conditions for $B \succ A$, (b) geometric definition of occlusion ambiguity.

to model the bounded motion between them. We fix A and let $M(B)$ denote the union of all possible positions of B . Its convex hull, $CH[M(B)]$, can be partitioned from A by a separating plane if the occlusion is *unambiguous*. This is illustrated in Fig. 3 (a) for two 2D bodies viewed by a 1D camera. The partition creates two half-spaces. If the image plane projections of A and $CH[M(B)]$ don't overlap, $A \equiv B$. If they do overlap, the object in the half-space containing the camera will occlude the other object.

The case of occlusion ambiguity is illustrated in Fig. 3 (b), using the same two bodies. Here it is impossible to predict the occluder under the given motion bound. Ambiguity arises when $CH[M(B)]$ intersects the occluding limb of A , which is a point in 2D and a curve in 3D. In this case, B can both occlude A and be occluded by it for different motions. In general, the likelihood of an occlusion ambiguity and the consequences of a mistake decrease with the motion bound.

Ambiguous configurations are rare for systems of convex bodies, as they depend on a special combination of spatial proximity and viewing angle. An example of an ambiguous hand configuration is the “stop” gesture, with the hand held flat, fingers pressed together, and palm facing the camera. In this pose, rotation around the vertical axis changes the visibility order of the fingers. For a specific object like the hand, techniques like velocity-based prediction can be used to handle ambiguous configurations.

3.2 Occlusion Graphs

The occlusion relations for a multi-body system with no ambiguities can be represented by a directed

occlusion graph. The graph is a pair (V, E) , where the vertex set V contains all of the bodies. To construct the edge set, E , consider all pairs $x, y \in V$. Since there are no occlusion ambiguities, one of $x \equiv y$, $x \succ y$, or $y \succ x$ must be true. In the first case no edge is added, while the other two cases add the directed edges (x, y) and (y, x) respectively. Consider the collection of 2D rigid bodies viewed by a 1D camera which is illustrated in Fig. 4. Figure 5 (a) shows the occlusion graph for the system under bounded translations in the plane.

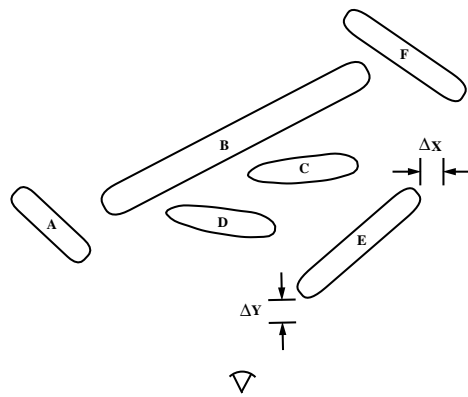


Figure 4: A collection of 2D rigid bodies under bounded translational motion relative to a 1D camera. Each body can translate by ΔX and ΔY , as shown for body E .

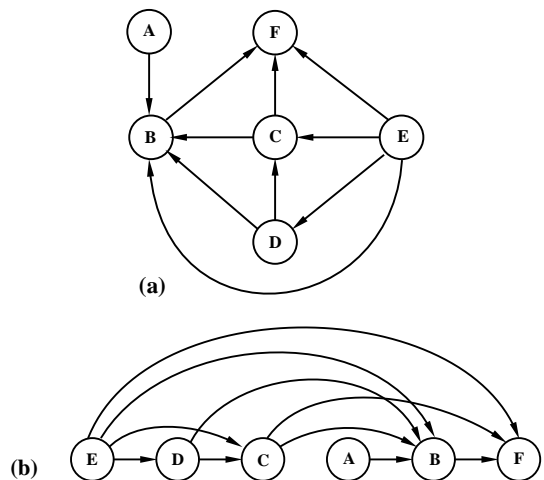


Figure 5: (a) Occlusion graph for the mechanism in Fig. 4, and (b) the visibility order produced by topologically sorting the graph.

When the object configuration admits a visibility ordering, it can be obtained by searching the occlu-

sion graph. In general, the occlusion graph must be *acyclic* to induce a natural order on the set of objects. When the occlusion graph is acyclic, it can be topologically sorted by depth-first search to produce a visibility ordering. Figure 5 (b) shows the ordering produced by sorting the sample occlusion graph. The sorted graph has the property that all edges are directed left to right. Taking the vertices in that order guarantees that no object will be occluded by an object that follows it in the list.

These results give sufficient conditions for the existence of a visibility ordering for an arbitrary object. Existence hinges primarily on the absence of occlusion ambiguities, which is determined by the relative motion and the temporal sampling rate. These results can be used to identify the most likely configurations for occlusion ambiguities in a known object.

Looking beyond model-based tracking, there is increasing interest in layered representations for computer vision, because of their potential to simplify the 3D description of the scene [1, 2, 6]. The results in this report provide general conditions under which a layered representation could be expected to exist, for a given type of moving object.

4 Registering Layered Templates

Given an invariant visibility order, local tracking consists of registering overlapping templates with an image sequence through gradient-based minimization of an SSD residual. This registration problem has two main components: *window functions* that block the contributions of occluded templates, and *deformation functions* that position the templates in the image as a function of the state.

Each template in the layered representation has a unit window function which evaluates to 1 for pixels inside the template and 0 otherwise. The combination of templates and windows form a composite image which is compared to the input image through the SSD residual. Figure 6 gives a 1D example of forming a composite image from two templates whose visibility order is $1 > 2$. The composite image can be written

$$I_c(x) = M_1(x - x_1)I_1(x - x_1) + [1 - M_1(x - x_1)]I_2(x - x_2) \quad (1)$$

where $I_{1,2}(\cdot)$ are the templates and M_1 is the window function for template 1.

Each rigid link in an object like the hand is modeled by a template, which is painted on a plane in the

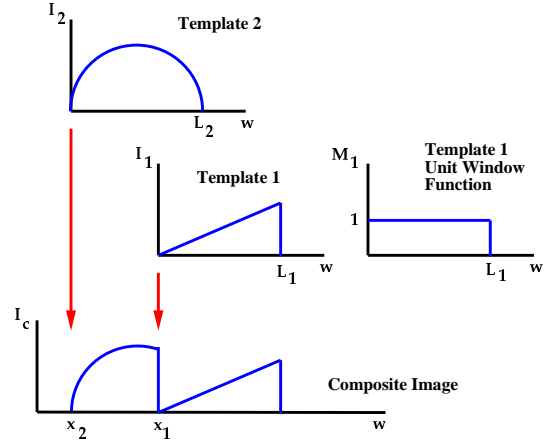


Figure 6: Image composition example for two 1D templates and a unit window.

link coordinate frame. Projecting the template plane through the camera model captures the effects of rotation and foreshortening in the image. The kinematic model gives the spatial position of each link frame as a function of the state vector \mathbf{q} , which contains the pose of the palm and joint angles of the fingers and thumb. The combination of kinematic and camera transforms make up a *deformation function* [12], $\mathbf{f}(\mathbf{q}, \mathbf{s})$, which maps template coordinates, $\mathbf{s} = [u \ v]$, to image coordinates, $\mathbf{w} = [x \ y]$, as a function of \mathbf{q} . This mapping is illustrated in Fig. 7, and more details can be found in [11].

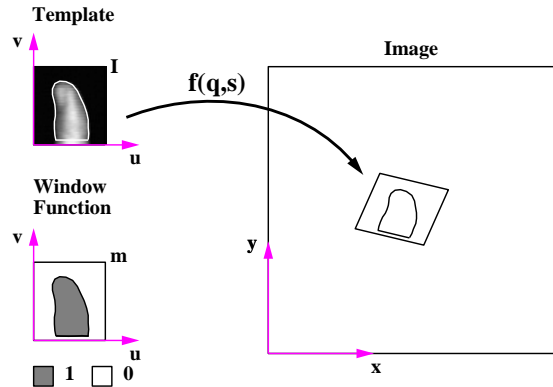


Figure 7: A finger tip template and its unit window function. The boundary contour (in white) encloses the template pixels, and the deformation function maps it into the image.

The SSD error function measures the difference between the input image and the composite image formed by the collection of deformed, windowed hand

templates. Generalizing Eq. 2 to 2D templates with deformation functions gives the SSD error:

$$\begin{aligned}
 E(\mathbf{q}) &= \frac{1}{2} \int_I \{I_c(\mathbf{q}, \mathbf{w}) - I(\mathbf{w})\}^2 d\mathbf{w} \\
 &= \frac{1}{2} \int_I \{M_1(\mathbf{q})I_1(\mathbf{f}_1^{-1}(\mathbf{q})) + \\
 &\quad [1 - M_1(\mathbf{q})]I_2(\mathbf{f}_2^{-1}(\mathbf{q})) - I\}^2 d\mathbf{w}
 \end{aligned}
 \tag{2}$$

where $\mathbf{f}_{1,2}^{-1}$ are inverse deformation functions for the two templates that map from image to template coordinates, and the argument \mathbf{w} is omitted in the last equation for simplicity. $M_1(\mathbf{q})$ is the unit window function for template 1, positioned in the image.

In the tracking experiment in this paper, the SSD error is minimized at each frame by a simple gradient-descent algorithm, which takes the previous frame’s state estimate as its starting point. We are investigating more sophisticated Gauss-Newton algorithms. The Jacobian evaluation and the computational requirements of minimization are discussed in [11].

5 Experimental Results

Figure 8 shows the performance of our tracking algorithm on a two finger motion sequence with significant self-occlusion. In the sequence, the first author’s index finger curls into his palm while the hand and remaining fingers are held still. The camera was positioned at approximately 45 degrees to the table top and fully calibrated using the procedure of [13]. An 80 frame sequence was digitized from videotape with an average finger tip displacement of about three pixels per frame. It was tracked using a 9 DOF kinematic model of the index and middle fingers (3 planar joints per finger) of the hand with full translation. Both the input images and templates were convolved with a 13x13 LOG filter to emphasize edges. The gradient descent algorithm was iterated twice for each frame. In this example, the visibility order did not change throughout the sequence. More details are given in [11], along with the estimated state trajectories.

From a classical feature detection perspective, the images in the sequence are quite difficult. All of the phalanges of the middle finger are partially occluded during some portion of the motion sequence, and the index finger is silhouetted against the fingers and palm for most of its motion. A significant advantage of the window-based approach is that it can tolerate any amount of occlusion and continue to extract useful information from the pixels that are visible.

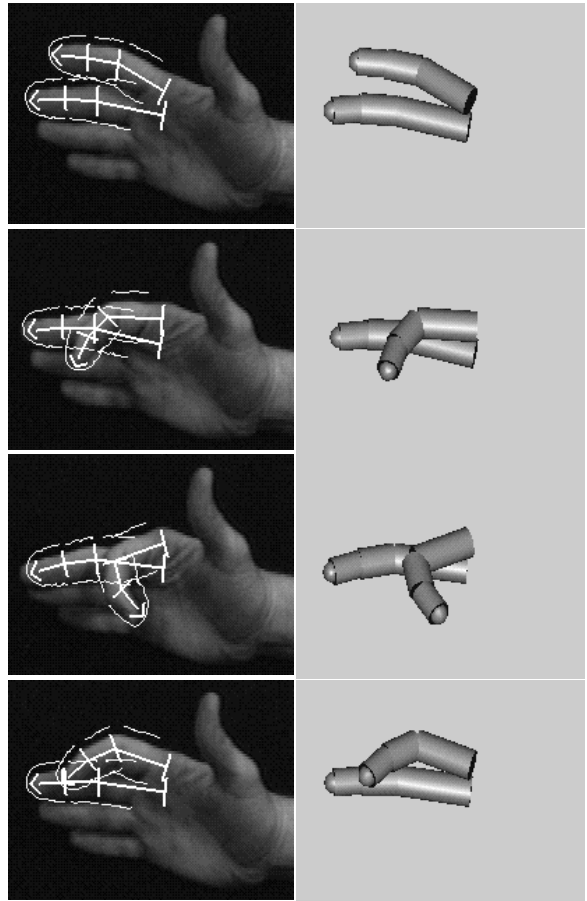


Figure 8: Sample images from frames 0, 13, 30, and 75 of the motion sequence. The overlays show the template boundaries and projection of cylinder center axes. The two finger model is rendered with respect to the calibrated camera model using the estimated state.

6 Previous Work

This paper extends our earlier work on the *DigitEyes* system for model-based articulated object tracking [10, 9]. Other previous work on tracking general articulated objects includes [5, 15, 8, 4, 7]. In [15], Yamamoto and Koshikawa propose the use of kinematic models for body tracking, and present 2D tracking results for an arm and torso. In [3], Dorner describes a system for interpreting American Sign Language from image sequences of a single hand. Two of the earliest systems were developed by Hogg [4] and O’Rourke and Badler [7]. None of these previous works have demonstrated 3D tracking in the presence of significant occlusions using natural images.

Our representation of self-occlusion is related to other work in tracking and motion coding. Layered representations based on clustering optical flow are presented in [2, 14]. These works address the automatic generation of a layered, velocity-based representation of a motion sequence for use in coding applications. A layered representation based on the occluding contours of a single image is described in [6]. These works are complementary to our approach, which is concerned with making the best use of available models. In addition, our representation of self-occlusions is a generalization of layered representations based on depth ordering in the scene, since it is designed to exploit orderings within configuration space.

7 Conclusion

Self-occlusion is an intrinsic visual property of articulated object motion. We have presented a novel representation of self-occlusion in configuration space, that generalizes current research in layered representations for motion analysis. We have developed a local tracking algorithm based on our representation, and tested it on a natural hand image sequence. We present the first experimental 3D tracking results for nontrivial self-occlusion.

In future work, we plan a real-time implementation of our occlusion-handling algorithm and experimental evaluation of its 3D tracking accuracy with more complicated models. We are also interested in the application of this technology to novel user-interfaces.

We would like to thank Luc Robert for making his camera calibration code available, and Fabio Cozman and Heung-Yeung Shum for their careful reading of this report and many useful comments.

References

- [1] E. H. Adelson. Layered representation for image coding. Technical Report 181, MIT Media Lab, 1991.
- [2] T. Darrell and A. Pentland. Robust estimation of a multi-layered motion representation. In *Proc. of IEEE Workshop on Visual Motion*, pages 173–178, Princeton, NJ, 1991.
- [3] B. Dorner. Hand shape identification and tracking for sign language interpretation. In *Looking at People Workshop, IJCAI*, Chambéry, France, 1993.
- [4] D. Hogg. Model-based vision: a program to see a walking person. *Image and Vision Computing*, 1(1):5–20, 1983.
- [5] D. Metaxas and D. Terzopoulos. Shape and nonrigid motion estimation through physics-based synthesis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(6):580–591, 1993.
- [6] M. Nitzberg and D. Mumford. The 2.1-d sketch. In *Proc. Third Int. Conf. on Comp. Vision*, Osaka, Japan, 1990.
- [7] J. O'Rourke and N. Badler. Model-based image analysis of human motion using constraint propagation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2(6):522–536, 1980.
- [8] A. Pentland and B. Horowitz. Recovery of nonrigid motion and structure. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(7):730–742, 1991.
- [9] J. Rehg and T. Kanade. Digiteyes: Vision-based hand tracking for human-computer interaction. In J. Aggarwal and T. Huang, editors, *Proc. of Workshop on Motion of Non-Rigid and Articulated Objects*, pages 16–22, Austin, Texas, 1994. IEEE Computer Society Press.
- [10] J. Rehg and T. Kanade. Visual tracking of high dof articulated structures: an application to human hand tracking. In J. Eklundh, editor, *Proc. of Third European Conf. on Computer Vision*, volume 2, pages 35–46, Stockholm, Sweden, 1994. Springer-Verlag.
- [11] J. Rehg and T. Kanade. Visual tracking of self-occluding articulated objects. Technical Report CMU-CS-TR-94-224, Carnegie Mellon Univ. School of Comp. Sci., 1994.
- [12] J. Rehg and A. Witkin. Visual tracking with deformation models. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, pages 844–850, Sacramento, CA, April 1991.
- [13] L. Robert. Camera calibration without feature extraction. *Computer Vision Graphics and Image Processing*, 1995. Accepted for Publication.
- [14] J. Wang and E. Adelson. Layered representation for motion analysis. In *Proc. IEEE Conf. Comput. Vis. and Pattern Rec.*, pages 361–366, 1993.
- [15] M. Yamamoto and K. Koshikawa. Human motion analysis based on a robot arm model. In *IEEE Conf. Comput. Vis. and Pattern Rec.*, pages 664–665, 1991. Also see Electrotechnical Laboratory Report 90-46.