Layered Representations for Vision and Video

Edward H. Adelson
Dept. Brain and Cognitive Sciences
and
Media Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract

Human vision, machine vision, and image coding, each demand representations that are useful and efficient. The best-established techniques today are based on low-level processing. Future systems for image analysis and image coding will increasingly use image representations that involve such concepts as surfaces, lighting, transparency, etc. These representations fall in the domain of "mid-level" vision, and there is accumulating evidence of their importance in human vision. By representing images with these more sophisticated vocabularies we can increase the flexibility and efficiency of our vision and image coding systems. We are developing systems that decompose image sequences into overlapping layers, rather like the "cels" used by a traditional animator. These layers are ordered in depth, sliding over one another and being combined according to the rules of transparency and occlusion. Using the layered representation we can achieve greatly improved motion analysis and image segmentation. By applying layers to image coding we can achieve data compression far better than MPEG, and achieve frame-rate independence as a side benefit. Moreover, the image sequence is decomposed in a meaningful way, which allows flexible image editing and access.

Introduction

Vision systems and image coding systems have related tasks. A vision system typically processes image data to achieve some decision or action, whereas an image coding system typically seeks to store and retrieve image data with as few bits as possible. In both cases the representations must be both useful and efficient. Future image coding systems will increasingly use techniques from computer vision.

In each case, the central issues hinge on representation. Efficient image descriptions, which are obviously important for image coders, are also important for vision systems, since one wishes to do as much processing as possible using as few bits, CPU cycles, or neurons, as possible. Useful and flexible image descriptions are obviously important to vision systems, but are also important to image coders, since one wishes not only to store images but to access them and manipulate them intelligently, making good use of image properties.

It is convenient to divide image representations into three levels: low-level, mid-level, and high-level. Low-level representations include pixels, subbands, DCT's, and the outputs of local operations such as edge detection and motion detection. High-level representations include full descriptions of objects and actions. Mid-level representations occupy the ill-defined territory in between, perhaps containing information about texture, global motion, surfaces, lighting, and so on [3, 10]. Mid-level processing is conducted in a set of hidden languages that we do not yet understand. Therefore the specification of mid-level vision remains a matter of research rather than definition.

The languages of low-level processing are fairly well worked out. Biological vision systems begin by establishing a multiscale representation that captures local aspects of orientation, color, motion, etc. Most machine vision systems that have ambitions of general utility begin with similar representations. And most image data compression systems use the related low-level vocabularies of DCT's, pyramids, wavelets, etc., along with local measurements of motion and color.

The languages of high-level vision must ultimately include the familiar language of everyday life, by which we describe the objects and actions in the world. The difficulty is that no one knows how to get there, and no one will until many hard problems are solved. High-level vision and high-level image coding have been

0-8186-7122-X/95 $4.00 © 1995 IEEE

demonstrated only in highly constrained domains. For example, in coding head-and-shoulder shots for teleconferencing, the world model consists of faces modeled with a high-level description, plus everything-but-faces, handled by falling back to a low-level description.

Perhaps the most fruitful area for progress today is in mid-level vision and image coding. The argument is as follows: low-level is too easy; high-level is too hard; mid-level is just right. A good mid-level representation should be able to handle arbitrary inputs and should be able to derive a useful and efficient description of natural scenes.

Layers

The main representation that we have explored in our laboratory is "layers." We think of an image or image sequence as being built up as a set of overlapping 2-D sheets that have varying color, intensity, and transparency at each point. This offers a vocabulary similar to that used by a traditional cel animator (e.g., Disney), in which scenes are built from overlying transparent sheets with painted characters on them. This vocabulary is also used in computer graphics, where the transparency of a layer is stored point-by-point in an alpha channel. Related representations have been explored by others in the context of both biological vision and machine vision [2, 11, 13, 20]. Layered coders may regarded as a form of analysis-synthesis coder [12].

A layered representation, as previously defined [1], consists of a set of overlapping 2-D layers locally ordered in depth, where each layer contains a set of registered 2-D maps. The intensity map defines the color and luminance of each point in the layer; the alpha map defines transparency; the velocity map describes the motion field by which the layer is warped over time. There may also be a depth map encoding the z-coordinate, a bump map encoding the surface normal, a delta map encoding the rate of change of image intensity, and other maps as may be convenient. A schematic example is illustrated in figure 1(a), which shows a moving hand.

The intensity of a rendered image, $I(x,y)$, is generated by compositing layers according to the equation:

$$I(x,y) = E_0(x,y)(1-\alpha_1(x,y)) + E_1(x,y)\alpha_1(x,y)$$

where E_1 is the additive component of layer 1, α_1 is the alpha channel of layer 1, and E_0 is additive component of the background layer, layer 0. Any number of stages can be cascaded, allowing for any number of layers.

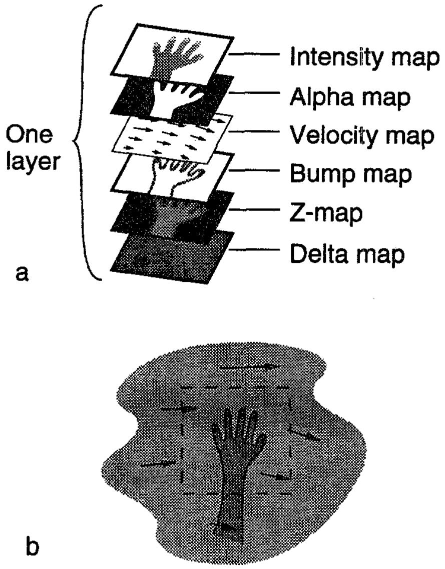

Figure 1: **(a) A layer consists of a set of registered maps. (b) A layer can extend beyond the viewing frame.**

In principle, the layer may extend indefinitely beyond the viewing frame as indicated in figure 1(b). As the layer moves, new information may become visible or hidden at the edges of the viewing frame. (The frame can itself be conceptualized as a layer that is opaque outside the clear viewing region). The information in the layer itself is stable as it moves in or out of visibility. The velocity field is applied uniformly to the whole layer. In the hand example, air space between the fingers moves smoothly with the fingers, although it is unseen in the rendered image.

The layered vocabulary allows a segmentation that is fundamentally different from the standard one. For example, consider figure 2(a), which humans perceive as a black square occluded by a translucent gray circle. A standard intensity-based segmentation algorithm would break it into the four pieces shown in figure 2(b-e). The standard vocabulary does not allow for overlap, so the scene must be broken into non-overlapping jig-saw puzzle pieces. The resulting decomposition does describe the image data but it fails to capture many facts that are apparent to a human observer, e.g., that the circle and square are separate objects, that the circle is in front of the square, that the circle is translucent, and that the gray piece in the middle (figure 2(b)) results from the transparent overlap of the circle and the square.

One possible layered decomposition of the scene is

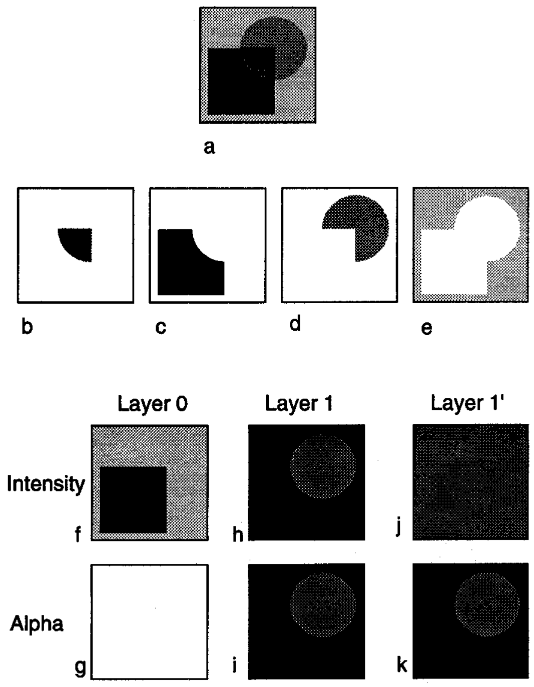

Figure 2: **(a) An image showing transparent overlap. (b)-(e) an intensity-based segmentation of the image into jig-saw pieces. (f)-(i) a layered decomposition of the image into two layers. (j), (k) an alternate version of layer 1.**

shown in figure 2(f-i). There are two layers: a background layer, denoted layer 0 and a foreground layer, denoted layer 1. The background consists of a light gray field with a black square painted on it, as shown by the intensity map, figure 2(f). The background is opaque everywhere, as shown by its alpha map, figure 2(g). The foreground consists of a gray circle, shown in figure 2(h), which is semi-transparent, as shown by the alpha map. Outside the circle region the alpha map is perfectly transparent, as shown. Thus within the circle the foreground color is added to the image and the background color is attenuated. Outside the circle, the background is seen unattenuated and the foreground adds nothing. Note that an alternate version of layer 1, denoted layer 1′, figure 2(j) and (k), would accomplish the same thing. Since the alpha map clips everything outside the circle, the intensity map in the outer region is of no consequence.

The layered vocabulary is overcomplete, and there are many layered descriptions of the same scene. For example, the black square could have a layer of its own, separate from the background, leading to three layers. Or the entire scene could be relegated to a single opaque layer with an intensity map identical to the observed image. Or the old jigsaw puzzle decomposition could be replicated within the layered vocabulary. The choice of a particular layered decomposition can be based on the task at hand.

Layers are especially useful in dealing with moving images, because they help solve some problems that have been classically difficult. The most popular traditional model for motion (optic flow) has been the rubber sheet. An image is assumed to be painted on this sheet at time 1, and then the sheet is smoothly distorted to produce a new image at time 2. This model fails in many ways, most notably at occlusion boundaries, where background information is appearing or disappearing, and in motion transparency, such as occurs with shadows, specularities, motion blur, focal blur, etc. These problems can never be solved in the context of traditional optic flow, and attempts to merely patch up the old algorithms are doomed to failure.

The layered approach offers a few key extensions to the old vocabulary. With layers there can be several rubber sheets rather than just one, and they can overlap each other rather than abutting like jig-saw pieces. In addition the sheets can be transparent or opaque in varying degrees as defined by the alpha map. The extensions sound modest enough, but they make a great difference in motion analysis.

Figure 3 shows schematically how motion estimation may be treated by different approaches. Figure 3(a) depicts a set of local velocity estimates across a raster line of a scene, which might be observed when a hand moves in front of a background. The faster velocities belong to the fingers, two of which are shown. The slower velocities belong to the background. Stable velocity estimation always demands some sort of smoothing across space, either implicitly or explicitly. The imposition of a smoothness constraint leads to a velocity field as shown by the curve in figure 3(a). Unfortunately, at the object boundaries the motions from the background and foreground are mixed, giving spurious velocities that don't properly correspond to any moving object. To fix this problem, some algorithms allow piecewise smoothing, as shown in figure 3(b). This improves matters but is still unsatisfactory. It does not capture the fact that the finger velocities all arise from one global motion and the background velocities all arise from another. In addition it implies that the velocity field is full of sharp breaks, which is at odds with the physical fact that everything in the world is moving smoothly.

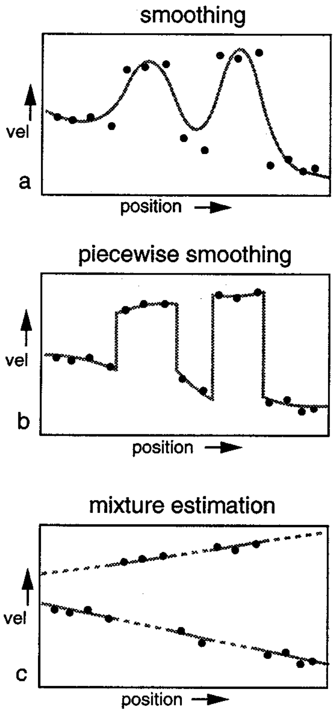

Figure 3: **(a) smoothed velocity field. (b) velocity field with piecewise smoothing. (c) overlapping velocity fields with mixture estimation.**

A better solution is shown in figure 3(c). Using mixture estimation [6, 7, 17] one can discover that the finger velocities are part of one smooth motion while the background motions are part of a second smooth motion. In this case the two motions have been fit with straight lines. The discontinuities have been banished from the motion fields and have instead been taken care of in the alpha channels, which are not shown, but which serve to assign support across space. Thus the representation captures important properties of the physical world. The hand's motion is smooth, and the empty space between the fingers moves with the hand. The background's motion is smooth, and it continues smoothly behind the fingers. The discontinuities are explained by the opacity of the foreground object, as they should be.

Once an image is represented in layers, resynthesizing the image is easy. Indeed real-time hardware is increasingly powerful and cheap due to the economics of the video-game industry, where warping and compositing are the bread and butter of image synthesis. The hard part, of course, is doing the analysis – the vision problem. When we embark on the task of finding a layered representation of an image sequence, we can be comforted by the knowledge that there is always one such representation, namely the one that describes each frame as paint on a single layer, and accommodates frame differences with delta maps. We hope, of course, that we can do better, but we always have this as a fall-back.

Analyzing motion sequences into layers is usually easier than analyzing single frames, since motion offers information about segmentation, occlusion, and so on. Recent work in motion and stereo analysis offers an array of tools [4, 5, 7, 9, 16, 19] that are quite helpful in this task.

Video coding experiments

We have used layers to analyze a one-second video clip of the MPEG flower garden test sequence [17, 18] (see figure 4(a)). This sequence was chosen for two reasons: (1) it is difficult to code using standard MPEG coders because of the motion and texture; (2) it should be easy to encode using layers.

The procedure involves two main steps: (1) Motion segmentation using mixture estimation with affine models; (2) layer accumulation using inverse warps and temporal accumulation. At present we assume binary alpha maps, i.e. the layers are fully opaque or transparent at each point.

In the first step, we look for regions that are moving coherently, where coherent is defined as belonging to a single affine flow. We begin by estimating local flow using a standard least-squares algorithm, and then perform patchwise affine flow estimation by linear regression, i.e. we fit planes to the velocity fields. The results are then clustered in the affine parameter space using k-means to produce an initial segmentation of the image. The process is iterated until the segmentation is stable.

In the next stage we accumulate the pixels that belong in a single layer. We inverse warp each frame in the sequence to bring pixels from a single layer into alignment, on the assumption that each motion segment successfully captures the motion of a single layer. Having aligned all the frames in the sequence we perform median temporal filtering in a straight line in time. Stable pixels are assumed to belong to the desired layer and they are given an alpha value of 1; other pixels are given an alpha of 0. The process is repeated to retrieve each layer in turn. The resulting layers contain information that may be hidden in individual images, such as the parts of the flower bed

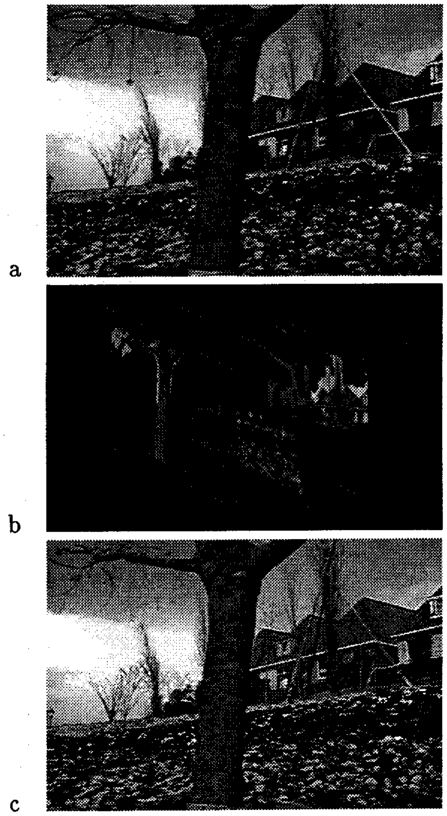

Figure 4: **(a) An original image from the MPEG flower garden sequence. (b) Three layers derived from the sequence. (c) Image resynthesized from the layers.**

that were hidden behind the tree, or the parts of the image that were hidden behind the viewing frame at various moments.

Figure 4(a) shows a single frame from the original flower garden sequence; figure 4(b) shows the layered decomposition; and figure 4(c) shows an image resynthesized from the layers. The synthesized image sequence looks quite good.

In this research have used binary alpha maps, which assume that an object is perfectly opaque or perfectly transparent at every point. We do not yet know how to deal with alpha maps that take on intermediate values between 0 and 1. Such an understanding will be needed to handle transparency, shadows, motion blur, etc., and we see it as an important future challenge.

The data rate for the 30 frame clip of the flower garden can be as low as 600 kbit/sec, when straightforward data compression is applied to the layers [8]. An MPEG coder would require some 5 times as much data for similar image quality. Thus the layered representation has the potential of offering major improvements in compression. Of course, one should keep in mind that this sequence was chosen because it is well-suited to layered representation, and the same level of compression may not apply to other image sequences.

A complex video sequence – for example, a basketball game – will not be as easy to analyze as the flower garden. Indeed our current algorithms fall apart when confronted by such inputs. To extend the layered approach to more complex motions we are developing more flexible analysis tools. Consider, for example, walking figures. A person walking across the image plane can be modeled as a set of flexible layers, as shown in figure 5(a). The basic layered vocabulary is the same, but the domain of smooth motions is different: now we must allow motions more complex than affine, and we must fit them to the complex spatio-temporal pattern of the walker. We have achieved such an analysis by fitting snakes in XT and deformable surfaces in XYT [14, 15], and using the resulting fits to control the warping of the flexible layers corresponding to body parts. More recently, work in our lab by David Askey indicates that these flexible layers, combined with the constraints of articulated figures, can lead to good analysis and synthesis of walkers. Figure 5(b) shows a single reconstructed frame of a sequence containing a walker. As with the flower garden sequence, layer images are accumulated by iteratively estimating motion fields and layer ownership. A one second video sequence of this walker can be coded with only slightly more data than is required for a single still image. While this sequence is simpler than a basketball game, it suggests that the layered vocabulary can be extended to handle a variety of situations.

In addition to being efficient, the layered representation allows useful image manipulation. Fig. 5(c) shows a single frame from the flower garden sequence in which the walker has been added. In the animation he is seen to "tip-toe through the tulips." Having been assigned the appropriate depth order, the walker occludes the flower bed and is occluded by the tree. It is also possible to resynthesize this sequence without the tree. Because the layered representation breaks the image data into meaningful chunks, layering can be a powerful tool for image editing.

The layered representation has the additional advantage of being frame-rate independent. Note that a normal video sequence is sampled at 30/60 Hz (NTSC) or 25/50 Hz (PAL), and the sampling rate normally places strong constraints on how the sequence is pro-

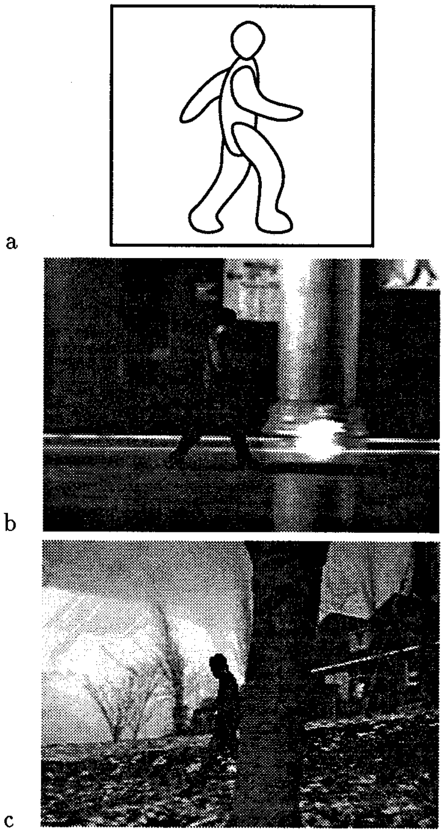

Figure 5: **(a) A walker can be composed of a set of flexible layers. (b) A frame from a reconstructed sequence with a walker. (c) The walker strides through the flower garden sequence.**

cessed and stored. By contrast, with a layered representation the notion of sampling rate almost disappears. Each layer is deformed smoothly over time, and one can choose to deform it by an intermediate amount, thereby synthesizing the frame at any point in time. Thus frame-rate conversion or slow motion becomes trivial.

Layered representations will also provide useful capabilities for image database access. When the flower garden sequence is decomposed into its layers, each layer is fairly uniform in properties such as texture, color, motion, and depth. Thus a search for a flower-bed texture or a tree-bark texture would be made much easier with a layered representation, in which this segregation has already occurred. Similarly the motion parameters needed to represent a walking person automatically offer compact and informative motion descriptors that can be used for database search. One could look for walking motions, or even look for a particular style of walking, characteristic of the individual.

Conclusions

Still and moving images will become an increasingly important data types in the future. Images are more than mere pixels, and vision techniques can be used to build image coders that can store, edit, and access images in useful and efficient ways. Full scale high-level machine vision remains out of reach, but mid-level techniques offer a great deal of promise.

Layers offer a powerful representation for mid-level vision. The layer vocabulary extends the standard vocabulary of motion analysis to allow multiple overlapping motions with transparency. Many of the classically difficult problems in motion analysis become tractable within this framework.

Various labs are developing tools that are helpful in decomposing image sequences into layers. By applying these tools we can perform robust motion segmentation on sequences involving multiple motions and occlusions. The layered representation offers frame rate independence, and may also allow major improvements in video data compression. Applications such as image editing, special effects, and image database access will also benefit from mid-level representations such as layers.

References

[1] E. H. Adelson. Layered representation for image coding. Technical Report 181, The MIT Media Lab, 1991.

[2] E. H. Adelson and P. Anandan. Ordinal characteristics of transparency. In *AAAI Workshop on Qualitative Vision*, pages 77–81, Boston, Massachusetts, 1990.

[3] H. G. Barrow and J. M. Tenenbaum. Recovering intrinsic scene characteristics from images. In A. Hanson and E. M. Riseman, editors, *Computer Vision Systems*, pages 3–26. Academic Press, New York, 1978.

[4] P.N. Belhumeur. Global priors for binocular stereopsis. In *Proc. 1st Int'l Conf. Image Proc.*, volume 2, pages 730–734, Austin, Texas, November 1994.

[5] J. Bergen, P. Anandan, K. Hana, and R. Hingorini al. Hierarchial model-based motion estimation. In *Proc. Second European Conf. on Comput. Vision*, pages 237–252, Santa Margherita Ligure, Italy, May 1992.

[6] M. J. Black and A. Jepson. Estimating multiple independent motions in segmented images using parametric models with local deformations. In *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, November 1994. (in press).

[7] T. Darrell and A. Pentland. Robust estimation of a multi-layered motion representation. In *Proc. IEEE Workshop on Visual Motion*, pages 173–178, Princeton, New Jersey, October 1991.

[8] U. Y. Desai. Coding of segmented image sequences. Thesis for master of engineering, Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, MA, June 1994.

[9] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In *Proc. Second European Conf. on Comput. Vision*, pages 282–287, Santa Margherita Ligure, Italy, May 1992.

[10] D. Marr. Representing visual information. In A. Hanson and E. M. Riseman, editors, *Computer Vision Systems*, pages 61–80. Academic Press, New York, 1978.

[11] F. Metelli. The perception of transparency. *Scientific American*, 230(4):91–98, 1974.

[12] H. G. Musmann, M. Hotter, and J. Ostermann. Object-oriented analysis-synthesis coding of moving images. *Signal Processing:* Image Communication *1*, pages 117–138, 1989.

[13] M. Nitzberg and D. Mumford. The 2.1-d sketch. In *Proc. Third Int'l Conf. Comput. Vision*, pages 138–144, Osaka, Japan, December 1990.

[14] S. A. Niyogi and E. H. Adelson. Analyzing and recognizing walking figures in xyt. In *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, pages 760–761, Seattle, Washington, June 1994.

[15] S. A. Niyogi and E. H. Adelson. Analyzing gait with spatiotemporal surfaces. In *IEEE Workshop on Nonrigid and Articulated Motion*, pages 64–69, Austin, TX, November 1994.

[16] H. Sawhney. 3d geometry from planar parallax. In *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, pages 760–761, Seattle, Washington, June 1994.

[17] J. Y. A. Wang and E. H. Adelson. Layered representation for motion analysis. In *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, pages 361–366, New York, June 1993.

[18] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing Special Issue: Image Sequence Compression*, 3(5):625–638, September 1994.

[19] Y. Weiss and E. H. Adelson. Motin estimation and segmentation with a recurrent mixture-of-experts architecture. In *IEEE Neural Network for Sig. Proc.*, 1995.

[20] L. R. Williams. Perceptual organization of occluding contours. In *Proc. Third Int'l Conf. Comput. Vision*, pages 133–137, Osaka, Japan, December 1990.