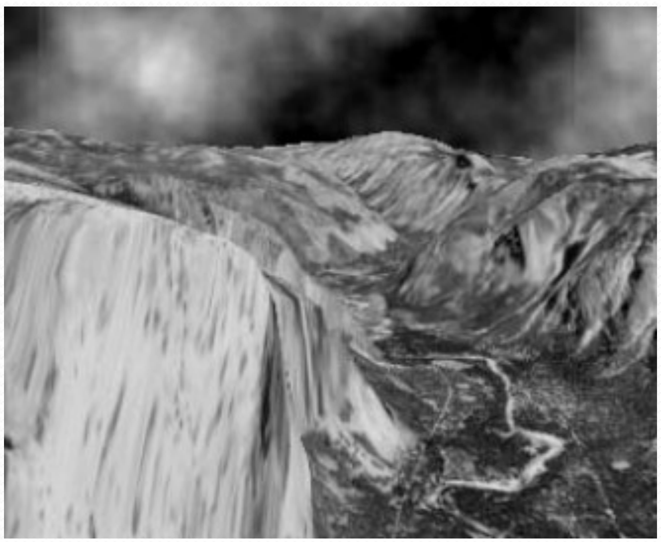
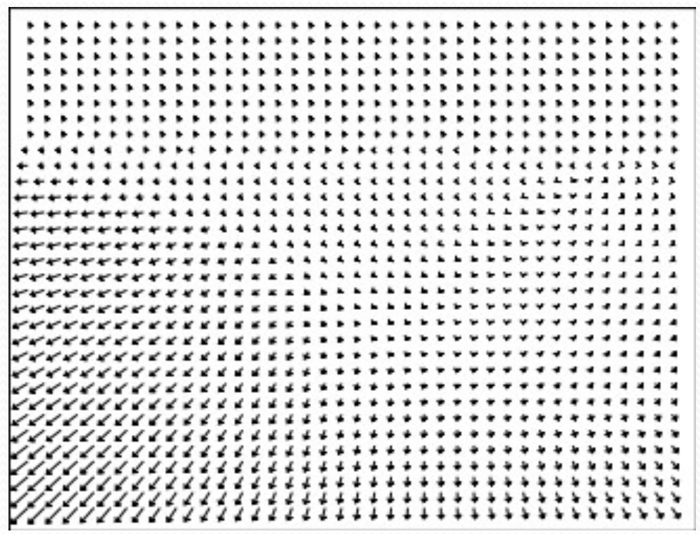
Dynamic Occlusion Analysis in Optical Flow Fields

CSE 668, Animate Vision Principles for 3D Image Sequences
CSE Department, SUNY Buffalo
luyuzhao@buffalo.edu

Optical Flow field:

Optical Flow Field: An optical flow field specifies the velocity of the image of points on a sensor plane due to the motion of the sensor and/or visible objects.





The picture is from Beauchemin and Barron, " The Computation of Optical Flow",N6A 5B7,1995

Discontinuity:

If a sensor is moving through a static scene, a discontinuity in optical flow occurs only if there is a discontinuity in the distance from the sensor to the visible surfaces on either side of the flow boundary.

The goal we want to achieve:

We want to use optical flow to locate dynamic occlusion boundaries in an image sequence.

Algorithm stated from Thompson, Kathleen and Berzins :

Main idea: Derive an edge detection sensitive to changes in flow fields likely to be associated with occlusion. In this algorithm, they developed an operator for finding occlusion boundaries in optical flow fields.

There are two main parts:

- Boundary detection
- Identifying occluding surface

Part 1:Boundary Detection

Two criteria:

●Sensitivity to rapid spatial change in one or both of the magnitude and direction of flow.

●Operation over a sufficiently large neighborhood to reduce sensitivity to noise in computed flow fields.

So they based on the zero-crossing detectors method of Marr and Hildreth which was put forward in the "Theory of edge detection", in Proc. Roy. Soc. London, vol. B-207, 1980,pp. 187-217.

Zero-crossing detectors:

Step1: Smooth the field using a symmetrical Gaussian kernel.

Step2: Compute the Laplacian of the smoothed function.

Step3: Look for directional zero crossings of the resulting function.

$$\nabla^2 G * I$$

Under a set of relatively weak assumptions, these zero-crossings can be shown to correspond to points of most rapid change in some direction in the original function!

The improving algorithm:

Main point:
An edge can be defined as a peak in the first directional derivation. At the edge, the second directional derivative has zero crossings in almost all directions, but the preferred direction is normal to the locus of the zero crossings, which is the same as the direction where the zero crossing is steepest for linearly varying fields. So we can get this conclusion:

For vector images such as optical flow fields, the directional derivatives are vector valued and we want the magnitude of the first directional derivative to have a peak!

Main Algorithm:

Let V be a twice continuously differentiable vector field, let N be an open neighborhood containing the origin such that ∂V/∂y is constant on N, let L be the intersection of N and the y axis, and let u be a unit vector.

Then the magnitude of the directional derivative in the u direction is

$$
\begin{aligned}
|\nabla V \cdot u|^2 &= (\nabla V_x \cdot u)^2 + (\nabla V_y \cdot u)^2 \\
&= \left[u_x \frac{\partial V_x}{\partial x} + u_y \frac{\partial V_x}{\partial y}\right]^2 \\
&\quad + \left[u_x \frac{\partial V_y}{\partial x} + u_y \frac{\partial V_y}{\partial y}\right]^2 \\
&= u_x^2 \left[\left[\frac{\partial V_x}{\partial x}\right]^2 + \left[\frac{\partial V_y}{\partial x}\right]^2\right] \\
&\quad + 2u_x u_y \left[\frac{\partial V_x}{\partial x}\frac{\partial V_x}{\partial y} + \frac{\partial V_y}{\partial x}\frac{\partial V_y}{\partial y}\right] \\
&\quad + u_y^2 \left[\left[\frac{\partial V_x}{\partial y}\right]^2 + \left[\frac{\partial V_y}{\partial y}\right]^2\right] \\
&= u_x^2 \left|\frac{\partial V}{\partial x}\right|^2 + 2u_x u_y \frac{\partial V}{\partial x} \cdot \frac{\partial V}{\partial y} + u_y^2 \left|\frac{\partial V}{\partial y}\right|^2
\end{aligned}
$$

The partial derivative of this quantity can be simplified as follows:

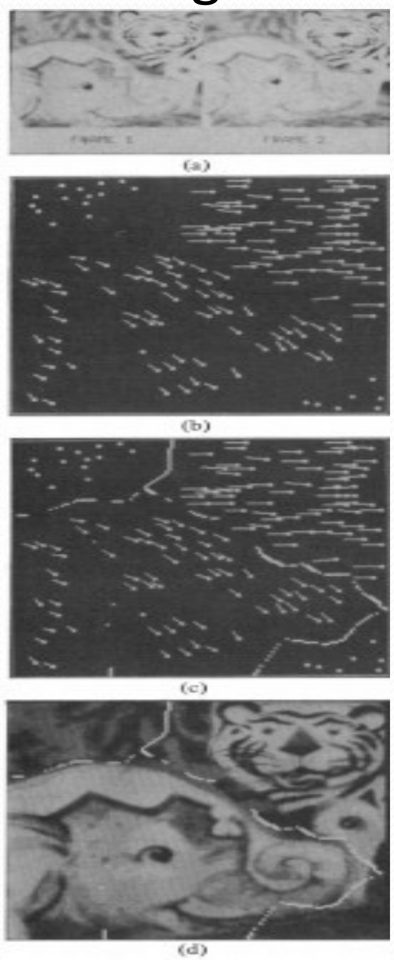
$$\frac{\partial}{\partial x}|\nabla V \cdot u|^2 = 2u_x^2\left[\frac{\partial V}{\partial x}\cdot\frac{\partial^2 V}{\partial x^2}\right] + 2u_x u_y\left[\frac{\partial V}{\partial y}\cdot\frac{\partial^2 V}{\partial x^2}\right]$$

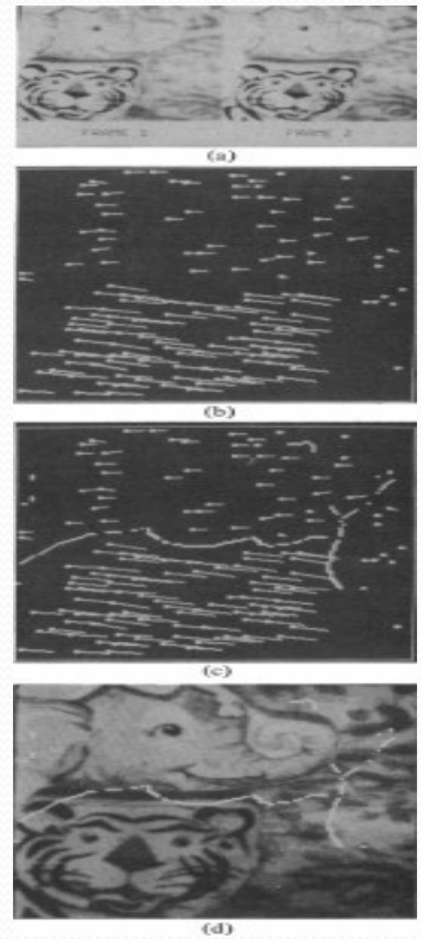
$$= 2u_x\left[u_x\frac{\partial V}{\partial x} + u_y\frac{\partial V}{\partial y}\right]\cdot\frac{\partial^2 V}{\partial x^2}$$

$$= 2u_x(u \cdot \nabla V)\cdot\frac{\partial^2 V}{\partial x^2}$$

since ∂V/∂y is constant on N. For the same reason, $\partial^2 V/\partial^2 = 0$ and $\partial^2 V/\partial^2 = \nabla^2 V$. Therefore, $\partial/\partial x\,|\nabla V \bullet u|^2$ has a zero crossing whenever $u_x(u \cdot \nabla V)\cdot\nabla^2 V$ does. But $|\nabla V \bullet u|^2$ has an extremum in the x direction whenever $\partial/\partial x\,|\nabla V \bullet u|^2$ has a zero crossing.

These are two examples of this technique applied to real images are shown below.





The picture is from Thompson, Kathleen, Berzins, “Dynamic Occlusion Analysis in Optical Flow Fields”.

Part 2: Identifying Occluding Surfaces

Prerequisite:

At a flow boundary, the side having the larger magnitude of flow will be closed, and thus will be occluding the father surface. Surfaces corresponding to regions on opposite sides if a boundary may move in arbitrary and unrelated ways.

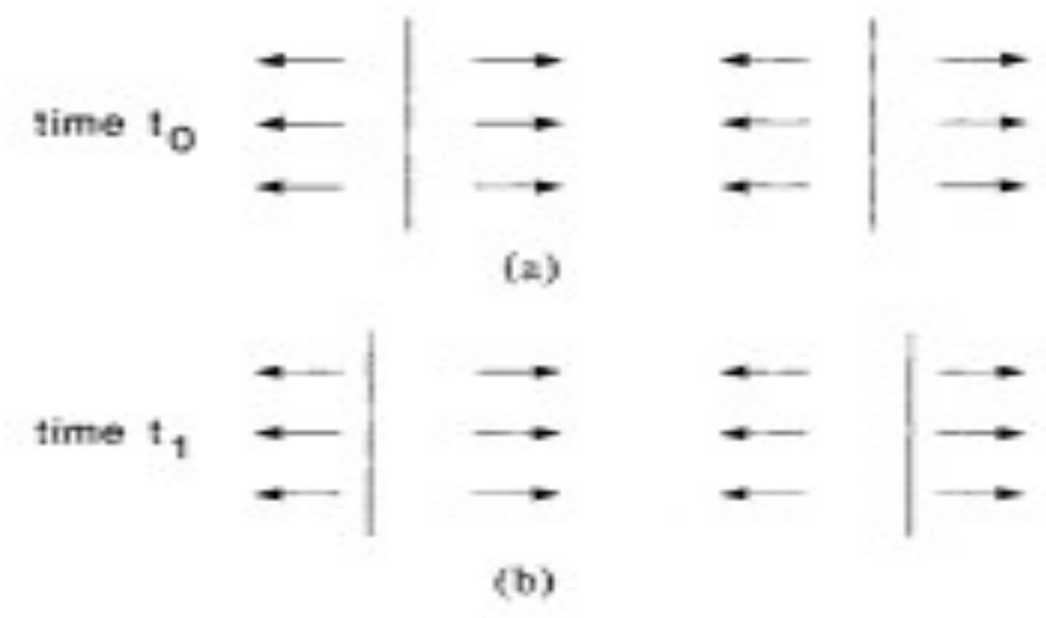
Conclusion:

By considering the flow values on either side of the boundary and the manner in which the boundary itself changes over time, it is usually possible to find which side of the boundary corresponds to the occluding surface, although the depth to the surfaces on either side cannot be determined.

Principle:

The image of the occluding contour moves with the image of the occluding surface.

Shown on the figure are the optical flow of points on each surface and the flow of points on the image of the boundary.



Now we can state the basic principle more precisely:
Choose a coordinate system in the image plane with the origin at a particular boundary point and the x axis oriented normal to the boundary contour, with x > 0 for the occluding surface. The camera points in the z direction, and the image plane is at z = 0. Let $f_x(x, y)$ be the x component of optical flow at the point (x,y). Let f_b be the x component of the flow of the boundary itself at the origin. Then, for rigid objects,

$$f_b = \lim_{x \to 0^+} f_x(x,0) = f_x(0,0)$$

This equation specifies a purely local constraint and, as the limit is taken from only one side of the boundary, is dependent on flow values on a single surface.

When developing an algorithm for actually identifying the occluding surface at a detected boundary, we can first start by assuming only translational motion is occurring, and ignoring the rotation. According to $f_b = \lim_{x \to 0^+} f_x(x,0) = f_x(0,0)$, we need only look at the flow at the edge point and immediately to either side to determine which side corresponds to the occluding surface.

But, is it practical precise?

A simple binary decision test:

We will use a coordinate system with its origin at the location of some particular boundary point at a time t_0, we consider only $flow_x$, the projection of flow onto the x axis. In this new coordinate system, positive velocity values will correspond to motion to the right.

We have two cases here:

Case1:

The two possible velocity functions are convolved with a Gaussian blurring kernel.

Given the step function:

$$s(x) = \begin{cases} 1, & x < 0 \\ -1, & x > 0 \end{cases}$$

convolve $s(x)$ with a Gaussian blurring function $g(x)$.

$$h(x) = g * s.$$

Let $s(x) = -2u(x) + 1$ where

$$u(x) = \begin{cases} 0, & x < 0 \\ 1, & x > 0. \end{cases}$$

Then

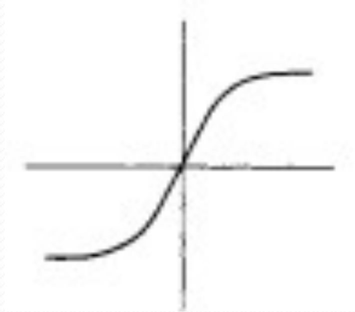
$$h(x) = 1 - 2\int_{-\infty}^{x} \frac{1}{\sigma\sqrt{2\pi}} e^{-\lambda^2/2\sigma^2}\, d\lambda$$

$$h''(x) = \frac{2x}{\sigma^3\sqrt{2\pi}} e^{-x^2/2\sigma^2}.$$

Therefore,

$h''(x) < 0$ when $x < 0$

$h''(x) > 0$ when $x > 0$.



Case2:

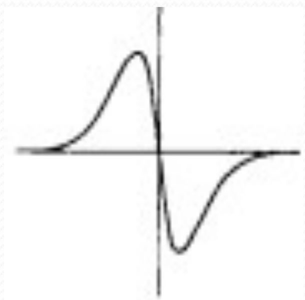
The Laplacian of these functions in the direction perpendicular to the edge is equal to the second derivative.
The step function for case 2 is –s(x), where s(x) and u(x) are defined above

$$h''(x) = \frac{-2x}{\sigma^3 \sqrt{2\pi}} e^{-x^2/2\sigma^2}.$$

Therefore,

$$h''(x) > 0 \quad \text{when} \quad x < 0$$

$$h''(x) < 0 \quad \text{when} \quad x > 0.$$



At some time t_1, $h''(x)$ will have shifted right or left, depending upon whether the edge moves with the surface moving to the right or left.

Specific Algorithm:

■First: Optical flow fields are obtained and approximation to the Laplacians of Gaussian blurred versions of these flow fields are calculated.
■Second: Edge points are found in the first flow fields by searching for vector reversals in the Laplacian of the field.
■Third: Find the appropriate offset to add to the edge location to find P.
■Fourth: The direction perpendicular to the edge point is estimated.

There is an example of this technique applied to an image sequence shown below.

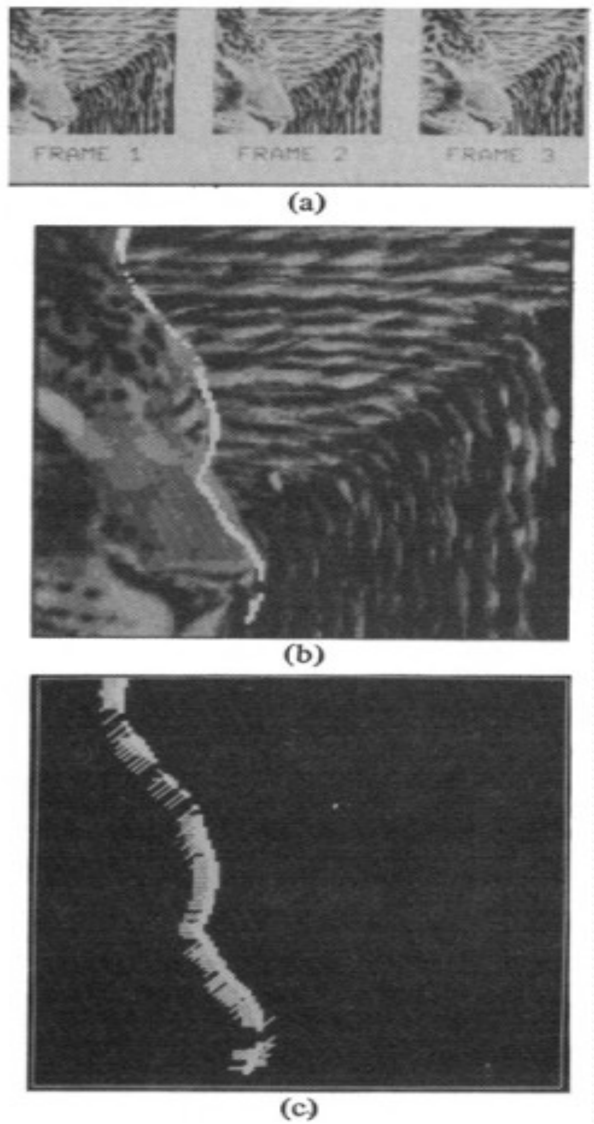


Fig. 6. (a) Image sequence. (b) Dectected boundary overlaid onto first frame of sequence. (c) Identification of occluding surface. Each edge point has a line segment projecting from it toward the occluding surface.

The picture is from Thompson, Kathleen, Berzins, “Dynamic Occlusion Analysis in Optical Flow Fields”.

The shortcoming of dynamic occlusion analysis:

Motion-based boundary detection is sensitive only to depth discontinuities and/or object boundaries.

Significant edges will not be detected unless there is perceived motion between the surfaces in either side.

The idea of Improving:

In the first part of the dynamic occlusion analysis algorithm, we can also use the intensity-based edge detection which is proposed by Witkin to detect the edge points. Then all the detected edge points are of direct significance to the interpretation of object shape. We can compare the two methods then we choose the more obvious and has better effect one to use it into the part2, identifying occluding surfaces.

Intensity-based edge detection:

Using basic continuity and independence properties of scenes and images, signatures were deduced for each of several edge types expressed in terms of correlational properties of the image intensities in the neighborhood of the edge. This procedure's ability to discriminate occluding contours from cast shadow boundaries was demonstrated for cases where line junction cues are absent from the image.

Goal:

◆The minimum goal: Use matlab to test my idea and verify if it can get high accuracy of occlusion detection.

◆If there still has time, I will devote myself to deal with the other problem of the algorithm.


Thank you!
http://www.wps.cn/index.php?mod=template&app=wpp&id=256&type=browser&O=g&page=2