Perception

Michael Rose

In which we connect the computer to the raw, unwashed world.

Perception provides agents with information about the world they inhabit by interpreting the response of sensors. The problem for a visual capable agent is this: Which aspects of the rich visual stimulus should be considered to help the agent make good action choices, and which aspects should be ignored? Vision and all other perception serves to further the agent's goals, not as an end to itself.

We can characterize three approaches to this problem:

- The **feature extraction** approach emphasizes simple computations applied directly to the sensor observations.
- The **recognition** approach has an agent draw distinctions among the objects it encounters based on visual and other information.
- The **reconstruction** approach has an agent build a geometric model of the world from an image or set of images

24.1 | Image Formation

Most surfaces reflect lights through the process of **diffuse reflection**. Diffuse reflection scatters light evenly across the directions leaving a surface, so the brightness of a diffuse surface doesn't depend on the viewing direction. Mirrors are not diffuse, because what you see depends on the angle at which you view the mirror. The behavior of a perfect mirror is known as **specular reflection**. A diffuse surface patch illuminated by a distant point light source will reflect some fraction of the light it collects; this is called the **diffuse albedo**.

Lambert's Cosine Law states that the brightness of a diffuse patch is given by

$$I = \rho I_0 \cos\theta$$

where ρ is the diffuse albedo, I_0 is the intensity of the light source, and θ is the angle between the light source direction and the surface normal.

Light arriving at the eye has different amounts of energy at different wavelengths; this can be represetned by a spectral energy density function. The **principle if trichromancy** states that for any spectral energy density, no matter how complicated, it is possible to construct another spectral energy density consisting of a mixture of three colors - red, green, and blue - such that a human can't tell the difference between the two.

24.2 | Early Image Processing Operations

In this section we look at three useful image processing operations: edge detection, texture analysis, and computation of optical flow.

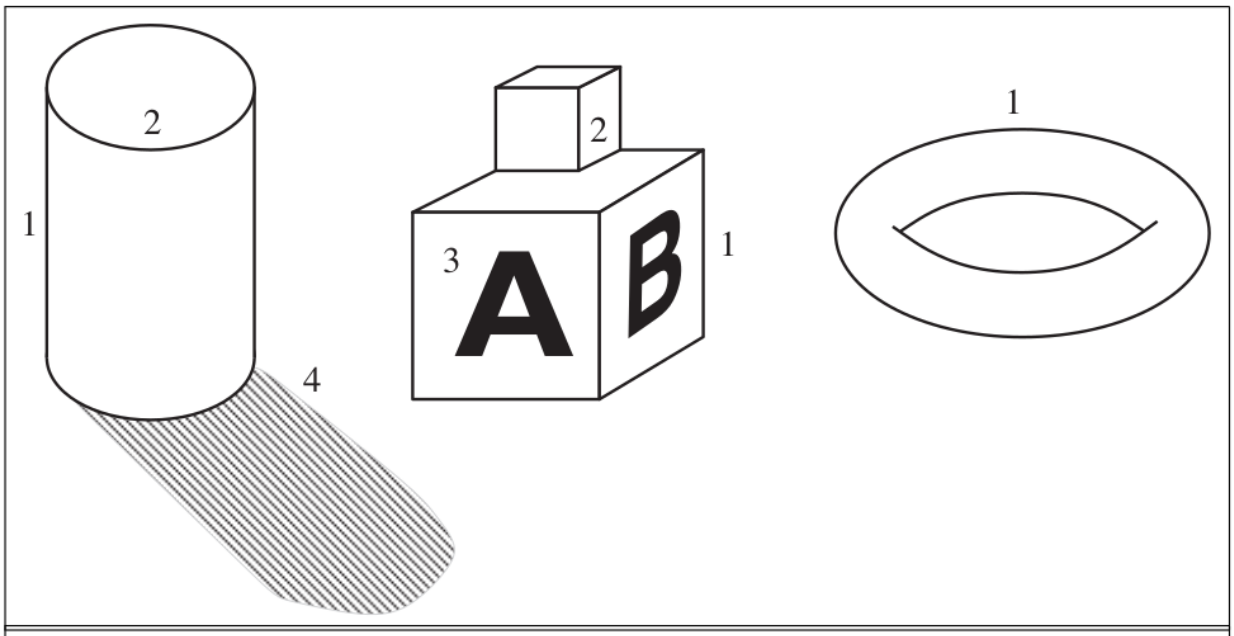

Figure 24.6 Different kinds of edges: (1) depth discontinuities; (2) surface orientation discontinuities; (3) reflectance discontinuities; (4) illumination discontinuities (shadows).

Figure 1:

24.2.1 | Edge Detection

Edges correspond to locations in images where the brightness undergoes a sharp change. A naive approach would be to differentiate the image and look for places where the derivative $I'(x)$ is large. This almost works, but there are peaks at other locations that arise due to the presence of noise in the image. If we smooth the image first, the spurious peaks are diminished. We can model noise in an image with a Gaussian probability distribution, with each pixel independent of the others. One good way to do this is with a weighted average that weights the nearest pixels the most, then gradually decreases the weight for more distance pixels. This is what the **Gaussian Filter** does, also known the the Gaussian blur operation in photoshop. The application of the Gaussian filter replaces the intensity $I(x_0, y_0)$ with the sum, over all (x, y) pixels, of $I(x, y)N_\sigma(d)$ where d is the distance from (x_0, y_0) to (x, y).

This kind of operation (weighted sum) is called a **convolution**. We say that the function h is the **convolution** of two functions f and g (denoted $f * g$) if we have

$$h(x) = (f * g)(x) = \sum_{u=-\infty}^{+\infty} f(u)g(x-u) \text{ in one dimension, or}$$
$$h(x,y) = (f * g)(x,y) = \sum_{u=-\infty}^{+\infty} \sum_{v=-\infty}^{+\infty} f(u,v)g(x-u, y-v) \text{ in two dimensions}$$

Our smoothing function is achieved by convolving the image with a Gaussian, $I * N_\sigma$. A σ of 1 pixel is enough to smooth over a small amount of noise, whereas 2 pixels will smooth a large amount, but at a loss of some detail. Since the Gaussian's influence fades quickly, we can replace the $\pm\infty$ in the sums with $\pm 3\sigma$.

We can optimize the computation by combining smoothing and edge finding into a single operation. There is a theorem that states: For any functions f and gm the derivative of the convolution, $(f * g)'$ is equal to the convolution with the derivative $f * g'$. Rather than smoothing the image and then differentiating, we can

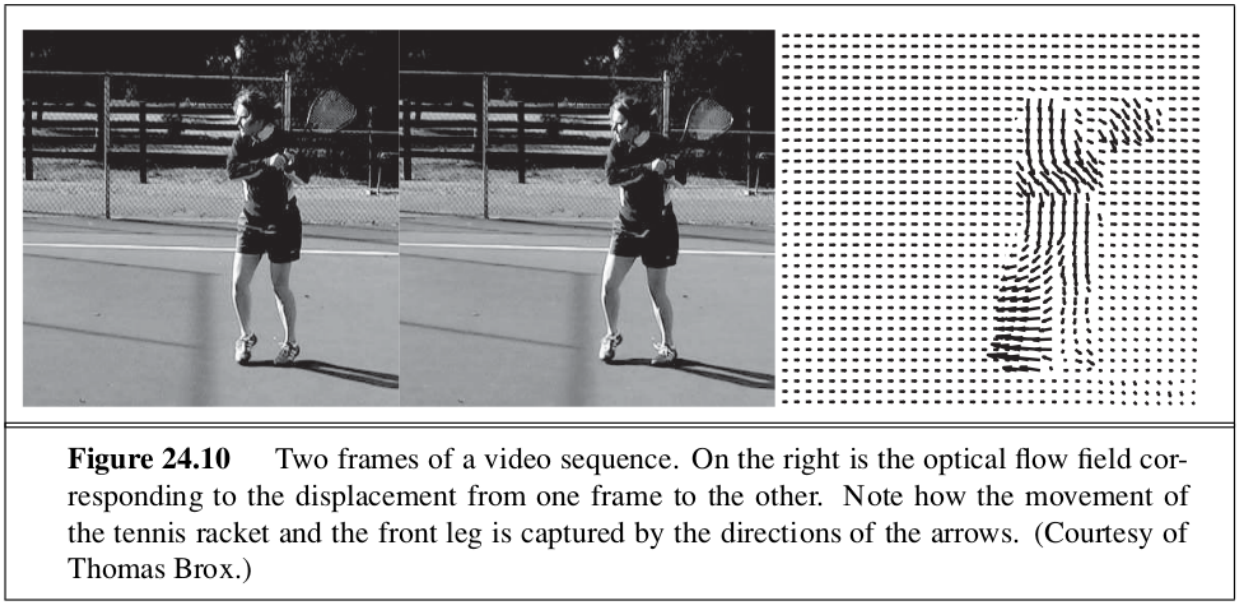

Figure 24.10 Two frames of a video sequence. On the right is the optical flow field corresponding to the displacement from one frame to the other. Note how the movement of the tennis racket and the front leg is captured by the directions of the arrows. (Courtesy of Thomas Brox.)

Figure 2:

just convolve the image with the derivative of the smoothing function, N'_σ. We then mark as edges those peaks in the response that are above a given threshold.

24.2.3 | Optical Flow

When an object in a video is moving, or when the camera is moving relative to an object, the resulting apparent motion in the image is called the **optical flow**. It describes the direction and speed of motion of features in the image.

The optical flow vector field can be represented at any point (x, y) by its components $v_x(x, y)$ in the x direction and $v_y(x, y)$ in the y direction. To measure optical flow we need to find corresponding points between one time frame and the next. A simple technique is based on the fact that imaage patches around corresponding points have similar intensity patterns. Consider a block of pixels centered at pixel $p, (x_0, y_0)$ at time t_0. This block of pixels is to be compared with pixel blocks centered at various candidate pixels at $(x_0 + D_x, y_0 + D_y)$ at time $t_0 + D_t$. One possible measure of similarity is the sum of squared differences:

$$SSD(D_x, D_y) = \sum_{(x,y)} (I(x, y, y) - I(x + D_x, y + D_y, t + D_t))^2$$

where (x, y) ranges over pixels in the block centered at (x_0, y_0). We then find the (D_x, D_y) that minimizes the SSD.

24.2.4 | Segmentation of Images

Segmentation is the process of breaking an image into regions of similar pixels.