Some image transform math

Owen Taylor

18 February 2003

1 Basics

The transform process is composed of three steps; first we reconstruct a continuous image from the source data $A_{i,j}$:

$$a(u,v) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} A_{i,j} F \begin{pmatrix} u - i \\ v - j \end{pmatrix}$$

Then we transform from destination coordinates to source coordinates:

$$b(x,y) = a \begin{pmatrix} u(x,y) \\ v(x,y) \end{pmatrix} = a \begin{pmatrix} t_{00}x + t_{01}y + t_{02} \\ t_{10}x + t_{11}y + t_{12} \end{pmatrix}$$

Finally, we resample using a sampling function G:

$$B_{x_0,y_0} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} b(x,y) G \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} dxdy$$

Putting all of these together:

$$B_{x_0,y_0} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} A_{i,j} F \begin{pmatrix} u(x,y) - i \\ v(x,y) - j \end{pmatrix} G \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} dxdy$$

We can reverse the order of the integrals and the sums:

$$B_{x_0,y_0} = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} A_{i,j} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F \begin{pmatrix} u(x,y) - i \\ v(x,y) - j \end{pmatrix} G \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} dxdy$$

Which shows that the destination pixel values are a linear combination of the source pixel values. But the coefficents depend on x_0 and y_0. To simplify this a bit, define:

$$i_0 = \lfloor u(x_0, y_0) \rfloor = \lfloor t_{00}x_0 + t_{01}y_0 + t_{02} \rfloor$$

$$j_0 = \lfloor v(x_0, y_0) \rfloor = \lfloor t_{10}x_0 + t_{11}y_0 + t_{12} \rfloor$$

$$\Delta_u = u(x_0, y_0) - i_0 = t_{00}x_0 + t_{01}y_0 + t_{02} - \lfloor t_{00}x_0 + t_{01}y_0 + t_{02} \rfloor$$

$$\Delta_v = v(x_0, y_0) - j_0 = t_{10}x_0 + t_{11}y_0 + t_{12} - \lfloor t_{10}x_0 + t_{11}y_0 + t_{12} \rfloor$$

Then making the transforms $x' = x - x_0$, $y' = y - x_0$, $i' = i - i_0$, $j' = j - x_0$

$$\begin{aligned}
F(u,v) &= F\begin{pmatrix} t_{00}x + t_{01}y + t_{02} - i \\ t_{10}x + t_{11}y + t_{12} - j \end{pmatrix} \\
&= F\begin{pmatrix} t_{00}(x' + x_0) + t_{01}(y' + y_0) + t_{02} - (i' + i_0) \\ t_{10}(x' + x_0) + t_{11}(y' + y_0) + t_{12} - (j' + j_0) \end{pmatrix} \\
&= F\begin{pmatrix} \Delta_u + t_{00}x' + t_{01}y' - i' \\ \Delta_v + t_{10}x' + t_{11}y' - j' \end{pmatrix}
\end{aligned}$$

Using that, we can then reparameterize the sums and integrals and define coefficients that depend only on (Δ_u, Δ_v), which we'll call the *phase* at the point (x_0, y_0):

$$B_{x_0,y_0} = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} A_{i_0+i,j_0+j} C_{i,j}(\Delta_u, \Delta_v)$$

$$C_{i,j}(\Delta_u, \Delta_v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F\begin{pmatrix} \Delta_u + t_{00}x + t_{01}y - i \\ \Delta_v + t_{10}x + t_{11}y - j \end{pmatrix} G\begin{pmatrix} x \\ y \end{pmatrix} dxdy$$

2 Separability

A frequent special case is when the reconstruction and sampling functions are of the form:

$$F(u,v) = f(u)f(v)$$

$$G(x,y) = g(x)g(y)$$

If we also have a transform that is purely a scale and translation; ($t_{10} = 0$, $t_{01} = 0$), then we can separate $C_{i,j}(\Delta_u, \Delta_v)$ into the product of a x portion and a y portion:

$$C_{i,j}(\Delta_u, \Delta_v) = c_i(\Delta_u)c_j(\Delta_v)$$

$$c_i(\Delta_u) = \int_{-\infty}^{\infty} f(\Delta_u + t_{00}x - i)g(x)dx$$

$$c_j(\Delta_v) = \int_{-\infty}^{\infty} f(\Delta_v + t_{11}y - j)g(y)dy$$

3 Some filters

gdk-pixbuf provides 4 standard filters for scaling, under the names "NEAREST", "TILES", "BILINEAR", and "HYPER". All of turn out to be separable as discussed in the previous section. For “NEAREST” filter, the reconstruction function is simple replication and the sampling function is a delta function[1]:

$$f(t) = \begin{cases} 1, & \text{if } 0 \le t \le 1; \\ 0, & \text{otherwise} \end{cases}$$

$$g(t) = \delta(t - 0.5)$$

For “TILES”, the reconstruction function is again replication, but we replace the delta-function for sampling with a box filter:

$$f(t) = \begin{cases} 1, & \text{if } 0 \le t \le 1; \\ 0, & \text{otherwise} \end{cases}$$

$$g(t) = \begin{cases} 1, & \text{if } 0 \le t \le 1; \\ 0, & \text{otherwise} \end{cases}$$

The “HYPER” filter (in practice, it was originally intended to be something else) uses bilinear interpolation for reconstruction and a box filter for sampling:

$$f(t) = \begin{cases} 1 - |t - 0.5|, & \text{if } -0.5 \le t \le 1.5; \\ 0, & \text{otherwise} \end{cases}$$

$$g(t) = \begin{cases} 1, & \text{if } 0 \le t \le 1; \\ 0, & \text{otherwise} \end{cases}$$

The “BILINEAR” filter is defined in a somewhat more complicated way; the definition depends on the scale factor in the transform (t_{00} or t_{01}]. In the x direction, for $t_{00} < 1$, it is the same as for “TILES”:

$$f_u(t) = \begin{cases} 1, & \text{if } 0 \le t \le 1; \\ 0, & \text{otherwise} \end{cases}$$

$$g_u(t) = \begin{cases} 1, & \text{if } 0 \le t \le 1; \\ 0, & \text{otherwise} \end{cases}$$

but for $t_{10} > 1$, we use bilinear reconstruction and delta-function sampling:

$$f_u(t) = \begin{cases} 1 - |t - 0.5|, & \text{if } -0.5 \le t \le 1.5; \\ 0, & \text{otherwise} \end{cases}$$

$$g_u(t) = \delta(t - 0.5)$$

The behavior in the y direction depends in the same way on t_{11}.

[1] A delta function is an infinitely narrow spike, such that:

$$\int_{-\infty}^{\infty} \delta(x) f(x) = f(0)$$