Computer Graphics	Prof. Brian Curless
CSE 557	Winter 2003

Homework #1

Sampling theory, image processing, affine transformations

Assigned: Sunday, January 26, 2002
Due: Thursday, February 6, 2002

Directions: Please provide short written answers to the following questions. For problem #1, use and attach additional sheets of paper. You can use the space provided for the other problems, and attach additional sheets as needed. In general, a fair number of equations need to be written out, so hand-written answers are recommended, unless you are unusually proficient with a math formatting package (like latex). Feel free to talk over the problems in general terms with classmates, but please *answer the questions on your own*.

Name:____

1. Fourier transforms and signal reconstruction

In this problem, you will explore some of the fundamental theorems of Fourier analysis and apply them to practical signal and image processing problems. We start with reviewing some of the equations brought out in class, supplemented with a few known results from Fourier analysis.

The definitions of the 1D and 2D Fourier transforms are:

$$\Im_{\mathrm{1D}}\{f(x)\} = F(s) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi sx} dx \qquad \Im_{\mathrm{2D}}\{f(x,y)\} = F(s_x, s_y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y) e^{-i2\pi(s_x x + s_y y)} dx dy$$

1D continuous convolution can be written as:

$$f(x) * h(x) = \int_{-\infty}^{\infty} f(x') h(x - x') dx'$$

The 1D convolution theorem is:

$$\Im_{\mathrm{1D}}\{f(x) * h(x)\} = F(s) H(s)$$

Convolution is commutative, associative, and distributive over addition:

$$\begin{aligned} &a(x) * b(x) = b(x) * a(x) \\ &[a(x) * b(x)] * c(x) = a(x) * [b(x) * c(x)] \\ &a(x) * [b(x) + c(x)] = a(x) * b(x) + a(x) * c(x) \end{aligned}$$

A normalized 1D Gaussian of zero mean takes the form:

$$n_{\mathrm{1D}}(x;\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$$

A rotationally symmetric, normalized, 2D Gaussian of zero mean takes the form:

$$n_{\mathrm{2D}}(x,y;\sigma) = n_{\mathrm{1D}}(x;\sigma) n_{\mathrm{1D}}(y;\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{y^2}{2\sigma^2}} = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

The convolution of a 1D Gaussian with a 1D Gaussian is another 1D Gaussian:

$$n_{\mathrm{1D}}(x;\sigma_1) * n_{\mathrm{1D}}(x;\sigma_2) = n_{\mathrm{1D}}(x;\sqrt{\sigma_1^2 + \sigma_2^2})$$

The Fourier transform of a 1D Gaussian is a 1D Gaussian. In particular:

$$\Im_{\mathrm{1D}}\left\{e^{-\pi x^2}\right\} = e^{-\pi s^2} \quad \text{where } e^{-\pi x^2} = n_{\mathrm{1D}}\left(x; \frac{1}{\sqrt{2\pi}}\right)$$

The "sifting" property of delta functions is:

$$f(x)\delta(x-a) = f(a)\delta(x-a)$$

If we sift for a value at an integer location, we can think of this as collecting a discrete sample:

$$f[i] = f(x)\delta(x-i) = f(i)\delta(x-i)$$

The "shifting" property of delta functions is:

$$f(x) * \delta(x-a) = f(x-a)$$

Problem 1 (cont'd.)

The sampling process can be written as:

$$\{f[i]\} = f(x)\mathrm{III}(x) = \sum_{i=-\infty}^{i=\infty} f(i)\delta(x-i)$$

where the sampled result is equivalent to the set of discrete samples {f[i]}.

One-dimensional and two-dimensional discrete convolutions are defined as:

$$f[i]*h[i] = \sum_{i'} f[i']h[i-i'] \qquad f[i,j]*h[i,j] = \sum_{j'}\sum_{i'} f[i',j']h[i-i',j-j']$$

Armed with these equations, you can now set to solving the following problems:

a) Given a function $f(x)$ and its Fourier transform $F(s)$, what is the Fourier transform of $f(ax)$ for $a>0$? Your answer should be in terms of F.

b) Prove that, if $g(x) = f(x)*h(x)$, then $dg(x)/dx = f(x) * dh(x)/dx = df(x)/dx * h(x)$.

c) Let's say we have a function $f(x)$ that has been sampled with the Shah function $\mathrm{III}(x)$ (which gives us a set of discrete samples $\{f[i]\}$), and now we seek to filter the sampled function and still represent it as a sampled signal. We can think of this problem in four stages: sampling the original function with $\mathrm{III}(x)$, reconstructing the continuous function with a filter $r(x)$, filtering with a function $h(x)$, and resampling with $\mathrm{III}(x)$. Spelling this out:

$$\begin{aligned} \text{Original sampling} \quad & \{f[i]\} = f(x)\mathrm{III}(x) \\ \text{Reconstruction} \quad & f(x) = \left[f(x)\mathrm{III}(x)\right]*r(x) \\ \text{Filtering} \quad & g(x) = \left[f(x)\mathrm{III}(x)\right]*r(x)*h(x) \\ \text{Re-sampling} \quad & \{g[i]\} = \left\{\left[f(x)\mathrm{III}(x)\right]*r(x)*h(x)\right\}\mathrm{III}(x) \end{aligned}$$

If we define a new filter, $q(x) = r(x) * h(x)$, show that this entire process is equivalent to the discrete convolution: $g[i]=f[i]*q[i]$. You'll need to make use of one or more of the commutative, associative, and distributive properties of convolution along the way; please indicate where you use these.

d) Suppose I have a sampled function $f[i]$, and I want to reconstruct it with a Gaussian $n_{1D}(x; 1/2)$, smooth it with another Gaussian $n_{1D}(x; 3/2)$, and resample to get $g[i]$. What discrete filter, $q[i]$, would I use to perform this filtering? Write out the convolution mask, rounding values to two decimal places and truncating any values below 0.01. You do not need to normalize the result.

e) This question is similar to d). Suppose I have a sampled function $f[i]$, and I want to reconstruct it with a Gaussian $n_{1D}(x; 1/2)$, *take the derivative*, and resample to get $g[i]$. What discrete filter, $q[i]$, would I use to perform this filtering? Write out the convolution mask, rounding values to two decimal places and truncating any values below 0.01. You do not need to normalize the result.

f) What is the Fourier transform of the 2D Gassian $n_{2D}\left(x, y; 1/\sqrt{2\pi}\right) = e^{-\pi\left(x^2+y^2\right)}$? Hint: write out the Fourier integral and look for a pattern. You do not need to do any integration.

g) Suppose I convolve an n x n image with a discrete filter with m x m support. In general, using "big O" notation, what is the complexity of this filtering operation? Now assume that the discrete filter is, in particular, a 2D Gaussian, $q[i,j] = n_{2D}(i, j; \sigma)$ which is being represented over an m x m region of support. How might re-order computation to reduce the cost of filtering, and what is the complexity of this new method?

2. Affine Transformations

$$A=\begin{bmatrix}1&0&0&6\\0&1&0&3\\0&0&1&2\\0&0&0&1\end{bmatrix}\quad B=\begin{bmatrix}1&0&0&0\\0&0&1&0\\0&-1&0&0\\0&0&0&1\end{bmatrix}\quad C=\begin{bmatrix}0&-1&0&0\\1&0&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

$$D=\begin{bmatrix}1&0&0&0\\0&2&0&0\\0&0&3&0\\0&0&0&1\end{bmatrix}\quad E=\begin{bmatrix}6&0&0&0\\0&6&0&0\\0&0&6&0\\0&0&0&1\end{bmatrix}\quad F=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&-1&0\\0&0&0&1\end{bmatrix}$$

$$G=\begin{bmatrix}1&0&1&0\\0&1&1&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\quad H=\begin{bmatrix}1&0&-1&0\\0&1&0&0\\1&0&1&0\\0&0&0&1\end{bmatrix}\quad I=\begin{bmatrix}1&0&0&1\\0&.6&-.8&0\\0&.8&.6&0\\0&0&0&1\end{bmatrix}$$

a) As discussed in class, any three-dimensional affine transformation can be represented with a 4x4 matrix. Match the matrices above to the following transformations (not all blanks will be filled, some may be filled with more than one answer):

____ Differential (Non-Uniform) Scaling

____ Reflection

____ Rotation about the z-axis with non-uniform scaling

____ Rotation about the y-axis with non-uniform scaling

____ Translation

____ Rotation about the x-axis

____ Rotation about the y-axis

____ Rotation about the z-axis

____ Shearing with respect to the x-y plane

____ Shearing with respect to the y-z plane

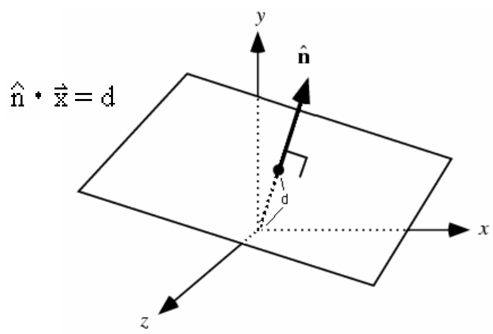
____ Rotation about the x-axis and translation

____ Uniform scaling

____ Reflection with uniform scaling

Problem 2 (cont'd.)

A plane can be described by the following equation and figure:



where $\hat{n}$ is the normal to the plane and d is the closest distance between the origin and the plane (which happens to be along the same direction as the normal).

Consider a plane with a normal vector of (cos θ, sin θ, 0). The equation for this plane can be simplified to:

$$x \cos \theta + y \sin \theta = d$$

Write out the product of 4x4 matrices that would perform a reflection across this plane. You must write out the contents of each matrix. Do not multiply the matrices out. You may write out the final product symbolically, after you have defined which symbol corresponds to which matrix. Show your work.