Questions may continue on the back. Please write clearly. What I cannot read, I will not grade.

1. A linear transformation of a point \( x \) on the plane to a point \( y \) on the plane is specified by the following equation

\[
y = Ax
\]

where \( A \) is a \( 2 \times 2 \) matrix.

For the questions (a), (b), (c) below, you need the following two items: The unit square \( S \) in the first quadrant is the polygon whose corners have coordinates \( x_1 = (0,0)^T \), \( x_2 = (1,0)^T \), \( x_3 = (1,1)^T \), \( x_4 = (0,1)^T \). This square is drawn in the figure below.

![Unit Square](image)

The second item is a real-valued function \( I(x) \) defined on the entire plane, \( x \in \mathbb{R}^2 \), but with support \( S \). This means that \( I(x) \) is nonzero exactly on the unit square defined above, and nowhere else:

\[
I(x) \neq 0 \iff x \in S.
\]

For each of the matrices in (a), (b), (c) below, draw (1) the quadrilateral \( Q \) that results from transforming \( S \) by a linear transformation with the given matrix; and (2) the support \( R \) of the function

\[
J(x) = I(Ax).
\]

Please note that questions (1) and (2) have different answers for each sub-problem below. You saw a similar phenomenon also for the standard Lucas-Kanade tracker: The transformation

\[
y = x + d
\]

with, say, \( d = (1,0)^T \) shifts point \( x \) to the right, but the function

\[
J(x) = I(x + d)
\]

is equal to \( I(x) \) shifted to the left.

Make sure that your drawings are clearly labeled with coordinates, and with which drawing is which (\( Q \) or \( R \)).

(a)

\[
A = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}
\]

(b)

\[
A = \begin{bmatrix} 1 & 1/2 \\ 0 & 1 \end{bmatrix}
\]

(c)

\[
A = \frac{1}{2} \begin{bmatrix} \sqrt{3} & 1 \\ -1 & \sqrt{3} \end{bmatrix}
\]

(d) Let us now generalize what you learned from these examples. Let \( T \) be any one-to-one transformation from \( \mathbb{R}^2 \) to \( \mathbb{R}^2 \), and let \( I \) and \( S \) be as above. If for a given set \( X \) we denote by

\[
T(X) = \{ y : y = T(x) \text{ for } x \in X \}
\]

the image of \( X \) through the transformation \( T \), and if \( S \) is, as stated earlier, the support of \( I \), then what is the support \( R \) of the function \( J(x) = I(T(x)) \)? Prove it.
(e) Prove that no linear transformation $A$ can transform the unit square into a trapezoid that is not a parallelogram. [Hint: see what happens to the lengths of the sides as a result of the transformation, and show that if the transformed quadrilateral is a trapezoid, then it is a parallelogram]

(f) Why is the fact you just proved relevant to tracking image windows from one frame to the next in a video sequence?

2. The Lucas-Kanade tracker suffers from the so-called aperture problem: If all gradients in a window point in the same direction, then only the component of motion in that direction can be determined uniquely. An example of a window that exposes this weakness is one with only vertical stripes in it. In that case, only the horizontal component of motion is determined uniquely. If the window translates vertically, its contents do not change, so that component cannot be determined.

The Shi-Tomasi tracker models motion with the affine model

$$y = Ax + d$$

rather than just with pure translation

$$y = x + d.$$ Since the former model subsumes the latter, the Shi-Tomasi tracker is still sensitive to the aperture problem for translation. In addition, estimation of the matrix $A$ introduces new opportunities for the aperture problem to arise. For each of the following three linear transformations, draw an example of a single image detail (in a square window) that has all the following three properties:

- Translation is detectable (i.e., there is no translation aperture problem), and
- The linear transformation in question is not detectable (i.e., the appearance of the detail is left unchanged by the transformation), and
- The other two linear transformations of the set are detectable.

$$A_1 = \begin{bmatrix} a & 0 \\ 0 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}, \quad A_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$ In these expressions, $a$ is a number different from 0 and 1, $\theta$ is some angle that is not a multiple of $\pi/4$. Assume that the origin of the reference system is at the center of the window, and that whatever pattern you draw extends indefinitely beyond the window. Indicate clearly which example is for which transformation.

3. The Shi-Tomasi tracker can be described as follows. A window of half-width $h \in \mathbb{N}$ centered at pixel position $x = (x_1, x_2)^T$ is a square of the form

$$W(x) = \{a = (a_1, a_2) : |a_i - x_i| \leq h \text{ for } i = 1, 2\}.$$ The affine transformation $x \rightarrow y$ of the plane defined by

$$y = Ax + d$$
can be rewritten as follows:

$$y = T(z, x) = x + Dx + d \quad \text{where} \quad D = A - I.$$ Here, $I$ is the $2 \times 2$ identity matrix and

$$z = \begin{bmatrix} D_{11} & D_{21} \\ D_{12} & D_{22} \end{bmatrix} [D_1, d_1, d_2]^T$$
is a vector that collects the six parameters that describe the transformation, that is, the six entries of $D$ and $d$.

Then, the Shi-Tomasi tracker solves the problem

$$\hat{z}(x_0) = \arg \min_z d(z, x_0) \quad \text{where} \quad d(z, x_0) = \sum_{x \in W(x_0)} [J(x) - J(Ax + d)]^2$$
is the sum of squared differences between windows in image $I$ and image $J$. 

CPS274 — Duke — November 14, 2009
This problem is solved by the following algorithm:

```matlab
function z = track(I, J, x0)
    z = 0
    δz = [∞, ∞]T
    while norm(δz) > ε
        [B, b] = system(I, J, x0)
        δz = B \ b
        J(x) ← J(T(δz, x))
        z ← z + δz
    end
end
```

The function `system` builds the left- and right-hand sides of a 6 × 6 linear system whose solution δz (computed in the line after the call to `system` in the code above) is an estimate of the deformation parameters between a window around x₀ in I and a window around x₀ in J. The image J (or rather the relevant part of it) is transformed at each iteration by this estimated deformation through the change of variable J(x) ← J(T(δz, x)) in the code above. In Matlab, this transformation would actually be implemented with the `interp2` function, but we do not concern ourselves with low-level programming details in this exercise.

You should be able to recognize the general outlines of the algorithm for the Shi-Tomasi tracker in the code above. The `system` function works as follows:

```matlab
function [B, b] = system(I, J, x0)
    for all a ∈ W(x₀)
        g = ∇J(a) % Spatial gradient of J
        Jz(a) = [g₁x₁, g₂x₁, g₁x₂, g₂x₂, g₁, g₂]T
    end
    B = ∑a∈W(x₀) Jz(a)JzT(a)
    b = ∑a∈W(x₀) Jz(a) [I(a) − J(a)]T
end
```

Please compare this with the formulas given in the Shi-Tomasi paper. In that paper, the two components of x are called x and y and those of g are called gx and gy. In this exercise, we use subscripts 1 and 2 instead.

(a) Check that the function `system` above is consistent with the formulas in the Shi-Tomasi paper. Specifically, compute the 36 entries of the product

\[ J_z(a)J_z^T(a) \]

that appears in the formula for B in the code above. Since this matrix is symmetric, you only really need to compute the 21 entries in the upper triangle of the matrix (that is, on the diagonal and above). Show a 6 × 6 matrix with these entries. What does this matrix correspond to, in the paper?

(b) How do you modify the function `system` to implement the Lucas-Kanade tracker, rather than the Shi-Tomasi tracker? If you are comparing with the original Lucas-Kanade paper, you may want to let F = J and G = I. Only show the necessary change(s).

(c) Suppose that instead of affine motion you have a general transformation

\[ y = T(z, x) \]

linear or otherwise, that depends on a vector z of n parameters (same z for all x), and for which

\[ T(0, x) = x \].

Recall that T is a function from \( \mathbb{R}^2 \) to \( \mathbb{R}^2 \).

The sum of squared differences is then

\[ d(z, x₀) = \sum_{x ∈ W(x₀)} |I(x) − J(T(z, x))|^2 \],
and the function $J(T(z, x))$, which is nonlinear in $z$, can be linearized with the following Taylor series approximation:

$$J(T(z, x)) \approx J(T(0, x)) + z^T \sum_{i=1}^{2} \frac{\partial J}{\partial x_i} \frac{\partial T_i}{\partial z} \bigg|_{z=0}$$

where $J(T(0, x)) = J(x)$. The derivative of a scalar (such as $T_i$) with respect to a vector (such as $z$) is the vector of the derivatives of the scalar with respect to each of the components of the vector.

Verify by direct calculation that the expression

$$\sum_{i=1}^{2} \frac{\partial J}{\partial x_i} \frac{\partial T_i}{\partial z} \bigg|_{z=0}$$

is equal to

$$J_z(a) = \begin{bmatrix} g_1 x_1, & g_2 x_1, & g_1 x_2, & g_2 x_2, & g_1, & g_2 \end{bmatrix}^T$$

if motion is affine:

$$T(x) = x + Dx + d.$$ 

Show all your work. Hint: all you need to do here is to spell out all the equations and differentiate.

(d) Congratulations! You just developed the general algorithm for tracking windows under arbitrary deformations: Use the functions \textit{track} and \textit{system} as defined earlier, but with

$$J_z(a) = \sum_{i=1}^{2} g_i \frac{\partial T_i}{\partial z} \bigg|_{z=0}.$$ 

To try this out in a very simple case, suppose now that the motion (or rather deformation) model is

$$y = s x,$$

where $s$ is some positive number. In words, the image shrinks ($0 < s < 1$) or expands ($s > 1$) away from the origin. “Away from the origin” means that the origin (wherever that may be) stays fixed:

$$0 = s0$$

where $0$ is a vector of two zeros. What is $J_z(a)$ in this case?