Problem 1

Suppose we have a camera that is undergoing a purely translational motion, and that the camera, with a focal length of 100, is moving with velocity

(U,V,W) = (20, 30, 50).

a) Generally, what is the focus of expansion for a moving camera?
b) What is the location of the focus of expansion for this particular motion?
c) What is the pattern of the optical flow vectors for this type of motion?
d) What is the aperture problem in computing optical flow? How is it related to the linear constraint on optical flow ("brightness constraint") determined by the differential technique for computing flow?

Answer:

a) Refer to slides 7 and 8 of lecture 22.
For image flow caused by rigid motion of the scene or camera the motion field in the image can be written as

$$u = \frac{-U + xW}{Z} + \alpha xy - \beta\left(x^2+1\right) + \gamma y = \frac{u_{\text{tr}}}{Z} + u_{\text{rot}}$$

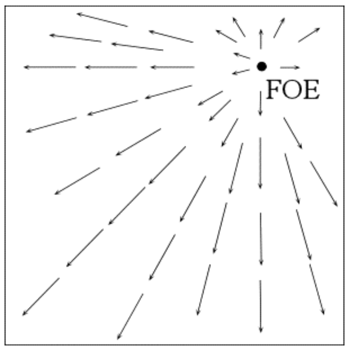
$$v = \frac{-V + yW}{Z} + \alpha\left(y^2+1\right) - \beta xy - \gamma x = \frac{v_{\text{tr}}}{Z} + v_{\text{rot}}$$

where the terms are as described in class
Observing the expression for the translational portion of the flow, we see that it can be written as
The focus of expansion or contraction is the point (x_0,y_0) from which the flow vectors due to the

$$\frac{\mathbf{u}_{\text{tr}}}{Z} = \left((x - x_0)\frac{W}{Z},\ (y - y_0)\frac{W}{Z} \right)$$

where $(x_0, y_0) = \left(\frac{U}{W}\cdot f,\ \frac{V}{W}\cdot f \right)$ is the focus of expansion (FOE) or focus of contraction (FOC).

translational component of a rigid motion appear to originate.



b) For this particular motion we have

$$(x_0, y_0) = \left(\frac{U}{W}\cdot f,\ \frac{V}{W}\cdot f \right) = \left(\frac{20}{50}.100, \frac{30}{50}.100 \right) = (40, 60)$$

c) The pattern of flow vectors for this motion is radially outwards from the focus of expansion.

d) The aperture constraint refers to the fact that when flow is computed for a point that lies along a linear feature, it is not possible to determine the exact location of the corresponding point in the second image. Thus, it is only possible to determine the flow that is normal to the linear feature.
The linear constraint on flow is derived by assuming that the brightness of a pixel is constant. This equation restricts the flow to lie along a line in velocity space. Again, it is a manifestation of the problem that it is possible to only determine the normal component of the flow.

Problem 2:

Contours of maximum rate of change in an image can be found by locating directional maxima in the gradient magnitude of the image or (almost) equivalently, by finding zero-crossings in the Laplacian of the image.

(a) Why is it considered important to convolve the image with a Gaussian before computing the Laplacian?

Images taken via cameras/CCDs often have noise. In addition, an image may have many features that would lead to edges. The operation of taking derivatives via finite differences leads to an increase in the noisiness of the image. Using a smoothing filter such as a Gaussian, blurs edges, thereby suppresses the local edges and retaining only the prominent ones.

(b) Because convolution and the Laplacian operator are both linear, convolving the image with a Gaussian and then taking the Laplacian, $(\nabla^2(G*I))$, is exactly equivalent to taking the Laplacian of the same Gaussian and then convolving that with the image, $(\nabla^2 G)*I$. Why does the second formulation lead to a more efficient implementation?

To perform the former operation we would have to apply the filter to the image via a discrete convolution, and then apply the discrete Laplacian operator to the image. However, we can precompute the action of the Laplacian on the Gaussian, and apply only one filter to the image, thereby improving the efficiency.

(c) The Laplacian of a Gaussian can be approximated by the difference of two Gaussian kernels with appropriately chosen standard deviations. As a result, show that the computation can be implemented as $(\nabla^2 G)*I \approx G_1*I - G_2*I$

To show that Laplacian of the Gaussian can be written as G_1*I-G_2*I, we can write

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1-\frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}}$$

This equation can be approximated by the difference of two Gaussians with appropriately chosen standard deviations

See http://www.dai.ed.ac.uk/CVonline/LOCAL_COPIES/OWENS/LECT6/node2.html for a discussion.

(d) Why is the difference of Gaussians implementation even more efficient than the $(\nabla^2 G)*I$ Laplacian of Gaussian implementation?

Gaussians are separable and can be implemented as two 1-D kernels … etc.

Problem 3.

(a) The general camera matrix is $\mathbf{P} = \mathbf{K}\,\mathbf{R}[\,\mathbf{I}\,|\,-\tilde{\mathbf{C}}]$ where $\tilde{\mathbf{C}}$ is the position of the camera center in scene coordinates. Here the scene coordinates are the camera coordinates for the first camera position. **T** is the translation of the camera center from its original position to its second position in scene coordinates. Therefore, **T = CC'** = $\tilde{\mathbf{C}}$.

For pure translation, the rotation **R** is the identity matrix. Therefore the camera matrix in this case is **P' = K [I | -T]**.

(b) Fundamental matrix as function of epipole.

One form of the fundamental matrix is $\mathbf{F} = [\mathbf{e}']_{\times}\,\mathbf{P}'\mathbf{P}^{+}$. For this problem, we have **P = K [I | 0]** and **P' = K [I | -T]**. First, we calculate

We have. **P = [K | 0]**, therefore $\mathbf{P}^{\mathbf{T}} = \begin{bmatrix} \mathbf{K}^{\mathbf{T}} \\ 0_3{}^{\mathbf{T}} \end{bmatrix}$ and $\mathbf{P}\,\mathbf{P}^{\mathbf{T}} = \mathbf{K}\mathbf{K}^{\mathbf{T}}$, so that $(\mathbf{P}\,\mathbf{P}^{\mathbf{T}})^{-1} = \mathbf{K}^{\text{-}\mathbf{T}}\mathbf{K}^{\text{-}\mathbf{1}}$

Therefore $\mathbf{P}^{+} = \begin{bmatrix} \mathbf{K}^{\mathbf{T}} \\ \mathbf{0}_3^{\mathbf{T}} \end{bmatrix} \mathbf{K}^{\text{-}\mathbf{T}}\mathbf{K}^{\text{-}\mathbf{1}} = \begin{bmatrix} \mathbf{K}^{\mathbf{T}}\mathbf{K}^{\text{-}\mathbf{T}}\mathbf{K}^{\text{-}\mathbf{1}} \\ \mathbf{0}_3^{\mathbf{T}} \end{bmatrix} = \begin{bmatrix} \mathbf{K}^{\text{-}\mathbf{1}} \\ \mathbf{0}_3^{\mathbf{T}} \end{bmatrix}$ and $\mathbf{P}'\,\mathbf{P}^{+} = [\mathbf{K}\,|\,\text{-}\,\mathbf{K}\,\mathbf{T}] \begin{bmatrix} \mathbf{K}^{\text{-}\mathbf{1}} \\ \mathbf{0}_3^{\mathbf{T}} \end{bmatrix} = \mathbf{I}$

Going back to the expression of **F**, this expression is reduced to $\mathbf{F} = [\mathbf{e}']_{\mathbf{X}}$

(c) The epipolar line of **x** is $\mathbf{l}' = [\mathbf{e}']_{\mathbf{X}}\,\mathbf{x}$. We want to show that **x** is on **l'**, i.e that $\mathbf{x}^{\mathbf{T}}$ **l'**=0. This is the case because $\mathbf{x}^{\mathbf{T}}\,[\mathbf{e}']_{\mathbf{X}}\,\mathbf{x} = \mathbf{0}$, a property resulting from the fact that $[\mathbf{e}']_{\mathbf{X}}$ is skew symmetric (Fundamental matrix, slide 10).

(d) If the camera translation is parallel to the x-axis, then **e'** = (1, 0, 0)$^{\mathbf{T}}$. Then $\mathbf{F} = [\mathbf{e}']_{\mathbf{X}}$ becomes

$$\mathbf{F} = \begin{bmatrix} \mathbf{0} & 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{-1} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \end{bmatrix}$$

<u>Problem 4:</u>

(a) As stated above, the general camera matrix is $\mathbf{P} = \mathbf{K}\,\mathbf{R}[\,\mathbf{I}\,|\,\text{-}\,\tilde{\mathbf{C}}]$ where $\tilde{\mathbf{C}}$ is the position of the camera center in scene coordinates. Here the scene coordinates are the camera coordinates for the first camera position. **T** is the translation of the camera center from its original position to its second position in scene coordinates. Therefore, **T = CC'** = $\tilde{\mathbf{C}}$. Using **T** from now on, and the fact that here **K = I**, we have for the second camera position $\mathbf{P}' = \mathbf{R}[\,\mathbf{I}\,|\,\text{-}\,\mathbf{T}] = [\mathbf{R}\,|\,\text{-}\,\mathbf{RT}]$. The column **–RT** gives the components of the initial camera center **C** in the new camera coordinate system. The rotation matrix is $R = \begin{bmatrix} c & 0 & s \\ 0 & 1 & 0 \\ -s & 0 & c \end{bmatrix}$, therefore the coordinates of the initial camera center **C** in the new camera coordinate system is $\begin{bmatrix} -cT_x - sT_z \\ 0 \\ +sT_x - cT_z \end{bmatrix}$. The camera matrix **P'** is

$$\mathbf{P}' = \begin{bmatrix} c & 0 & s & -cT_x - sT_z \\ 0 & 1 & 0 & 0 \\ -s & 0 & c & sT_x - cT_z \end{bmatrix}$$

(b) We write $\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$ for the first camera, and

$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} c & 0 & s & -cT_x - sT_z \\ 0 & 1 & 0 & 0 \\ -s & 0 & c & sT_x - cT_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$ for the second camera. The conditions for zero disparity are *x* = *x'*, and *y* = *y'*, i.e. *u/w* = *u'/w'* and *v/w* = *v'/w'*.

The first condition can be written $\frac{X}{Z} = \frac{cX + sZ - cT_x - sT_z}{-sX + cZ + sT_x - cT_z}$ and the second condition is

$\frac{Y}{Z} = \frac{Y}{-sX + cZ + sT_x - cT_z}$. The first condition leads to

$X^2 + Z^2 - X(T_x - t'T_z) - Z(t'T_x + T_z) = 0$, where *t'*=*c/s* (*t'* is the cotangent of the angle of rotation).

The second condition leads to $Y(-sX + (c-1)Z + sT_x - cT_z) = 0$, another quadratic equation

(c) The first condition is $X^2 + Z^2 - X(T_x - t'T_z) - Z(t'T_x + T_z) = 0$. It is a 3D surface. Its intersection with any horizontal plane *Y* = *k* is a circle, because the equation of the intersection is of the form $(X\text{-}a)^2 + (Y\text{-}b)^2 = R^2$, where *a* and *b* are the coordinates of the center of the circle and *R* is its radius. Therefore the surface is a cylinder of revolution.

The second condition leads to $Y(-sX + (c-1)Z + sT_x - cT_z) = 0$, i.e. either Y=0, or $-sX + cZ + sT_x - cT_z - Z = 0$, which are the equations of two planes, the horizontal plane and a vertical plane. Therefore, the locus of points with zero disparity is defined by two surface intersections: the intersection of the cylinder by the horizontal plane, and the intersection of the cylinder by the vertical plane. The first intersection defines a circle in the horizontal plane of the camera centers. The second intersection defines two vertical lines. Therefore, the locus of points with zero disparity is defined by the intersection of the cylinder and the horizontal plane, or by the intersection of the cylinder and a vertical plane.

The equation of the circle is $X^2 + Z^2 - X(T_x - t'T_z) - Z(t'T_x + T_z) = 0$. The coordinates of the center of the circle are

$(T_x - t'\,T_z)/2$ and $(t'\,T_x + T_z)/2$. The circle passes through a specific point if the coordinates of that point satisfy its equation. One verifies that the circle passes through the camera center **C** (of coordinates (0,0)), the camera center **C'** (of coordinates $(T_x, 0, T_z)$), and also the point of intersection of the optical axes, $(0, t'\,T_x + T_z)$. It also passes through the intersection of the x-axes of the camera coordinate systems

$(T_x - t'\,T_z, 0)$.

The vertical plane $-sX + (c-1)Z + sT_x - cT_z = 0$ can be verified to also pass through the intersection of the x-axes of the camera coordinate systems $(T_x - t'\,T_z, 0)$.

Therefore, that point belongs to the vertical line that is the intersection of the vertical plane with the cylinder. However, that line is not visible, because it is behind both image planes (it is the intersection of the planes defined by the x and y axes of each camera).

To find the second intersection between the circle and the line $-sX+(c-1)Z+sT_x-cT_z=0$ in the plane *Y*=0, we do a change of coordinates to place the origin of the scene at the first intersection in order to simplify the equation of the line. The old coordinates (*X*, *Z*) in relation to the new ones (*X'*, *Z'*) are
X=*X'*+ T_x - *t'* T_z , Z=Z'. The equation of the circle becomes $X'^2+Z'^2+X'(T_x-t'T_z)-Z'(t'T_x+T_z)=0$ and the equation of the line becomes $sX'=(c-1)Z'$. Plugging this value of X' into the equation of the circle, we find

$X=\frac{sT_x-(c+1)T_z}{2s}$ and $Z=\frac{(c+1)T_x+sT_z}{2s}$

Looking at simple configurations of cameras, we infer that this point also happens to be the intersection between a line perpendicular to the translation vector at its midpoint and the circle (this line also goes through the center of the circle). To verify this, we write the equation of such a line: $T_xX+T_zZ-(T_x^2+T_z^2)/2=0$ and we find that indeed our second intersection belongs to this line. Since we know it belongs to the circle it is at the intersection of this line and the circle. To check that this intersection is visible from both cameras, we would have to check that for each camera it is on the side of the image plane that does not contain the image center.

Problem 6:

Recall the definition of an eigenvalue and an eigenvector **Ax**=λ**x**. We can define

$$f(\mathbf{x})=\frac{\mathbf{x}^t\mathbf{Ax}}{\mathbf{x}^t\mathbf{x}}.$$

Given a matrix **A**, we can view f(**x**) as a scalar valued function of **x**.(This function is called the Rayleigh quotient in the literature.)

a) Show that stationary points of the f (**x**) with respect to **x** are eigenvectors of **A**, and that the corresponding values of f (**x**) are eigenvalues.

To determine the stationary points we need to find locations where . $\frac{\partial f}{\partial \mathbf{x}}=0$
This is most conveniently done using the summation convention. This yields

$$\frac{\partial f}{\partial \mathbf{x}}=\frac{\partial}{\partial x_k}\left(\frac{x_iA_{ij}x_j}{x_lx_l}\right)=\frac{(x_lx_l)(\delta_{ki}A_{ij}x_j+x_iA_{ij}\delta_{kj})-(x_iA_{ij}x_j)(2x_l\delta_{kl})}{(x_lx_l)^2}$$

$$=\frac{(x_lx_l)(A_{kj}x_j+x_iA_{ik})-(x_iA_{ij}x_j)(2x_k)}{(x_lx_l)^2}$$

We are looking for nontrivial solutions. So we want

$$(x_lx_l)(A_{kj}x_j+x_iA_{ik})=(x_iA_{ij}x_j)(2x_k)$$

$$\frac{1}{2}(A_{kj}x_j+x_iA_{ik})=\frac{x_iA_{ij}x_j}{x_lx_l}x_k$$

Putting this in matrix notation

$$\frac{1}{2}(\mathbf{Ax}+\mathbf{x}^t\mathbf{A})=\left(\frac{\mathbf{x}^t\mathbf{Ax}}{\mathbf{x}^t\mathbf{x}}\right)\mathbf{x}$$

Using the fact that **A** is symmetric

$$\mathbf{Ax} = \left(\frac{\mathbf{x}^t\mathbf{Ax}}{\mathbf{x}^t\mathbf{x}} \right)\mathbf{x}$$

Comparing with eigenvalue relation we see that at the stationary point of *f*(**x**) we satisfy the eigenvalue relation with the eigenvalues given by the value of *f*(**x**) and **x** the eigenvector.

b) Write a function that returns the Rayleigh quotient of a 2 by 2 matrix **A**, given a vector **x**. Test your code using the square matrix $\mathbf{A} = \begin{bmatrix} 3 & \sqrt{3} \\ \sqrt{3} & 3 \end{bmatrix}$.Use the Matlab **mesh** function to plot the negative of the Rayleigh quotient, –f(**x**), where **x** is the 2D vector $[x_1, x_2]'$ with component values in the range $-1.5 \le x_1, x_2 \le 1.5$ (You may want to avoid the region around (0,0) by choosing your points judiciously).

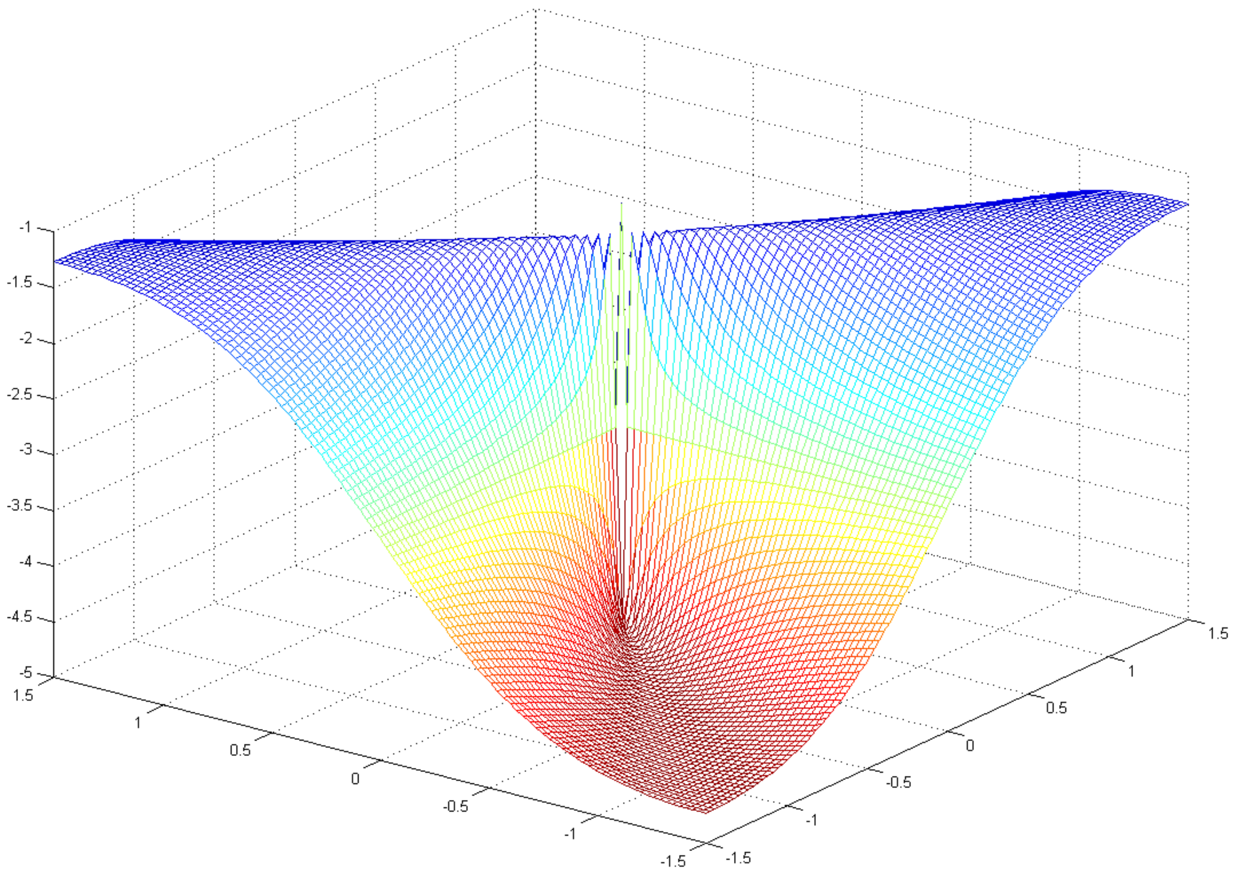
A possible implementation that uses a global variable to pass the matrix is shown below.

```
function lam=lambda(x)
global aglobal
A=aglobal;
x1=A*x;
lam=x'*x1;
if(lam>=1.e-15)
   lam=lam/(x'*x);
else
   lam=1.;
end
lam=-lam;
```

This function can be called with a script such as `global aglobal`

```
x1=-1.5:0.03:1.5;
imax=max(size(x1));
x2=-1.5:0.03:1.5;
b=sqrt(3)
a=[3 b; b 3];
aglobal=a;
lam=zeros(imax);
for i=1:imax
   x=x1(i);
   for j=1:imax
      y=x2(j);
      v=[x,y]';
      lam(i,j)=lambda(v);
   end
end
mesh(x1,x2,lam);
```

(note: This surface is actually the negative of what I asked but it is acceptable)



c) Find the **maximum** of this function using the Matlab function **fminsearch**. Compare the values you obtain for the eigenvector and eigenvalue using the Matlab function **eig**. Are the values different? Why?

Adding the line

```
fminsearch('lambda',[0.4 0.1]')
```

to the script above does the job. The result I got is

```
Ans =

    0.1553
    0.1553
```

Obtaining the result via the Matlab built in function eig, I obtain.

```
[v,d]=eig(a)

v =
    0.7071    0.7071
   -0.7071    0.7071
d =

    1.2679         0
         0    4.7321
```

The eigenvalue obtained is the same, but the eigenvector is different. In general, since eigenvectors are not unique. This is because, if **x** is an eigenvector, so is *s***x**, for *s* any nonzero scalar. (you can observe this in the figure in *6b* – both the max and the min lie on a ridge).