Computer Vision (600.461/600.661)
Homework 5: Two-View Geometry and Optical Flow

Instructor: René Vidal

Due 10/02/2014, 11.59PM Eastern

1. **(20 points) Properties of $so(3)$ and $SO(3)$.** Given a nonzero vector $\boldsymbol{w} = (w_1, w_2, w_3) \in \mathbb{R}^3$, one can build a 3×3 skew-symmetric $\widehat{\boldsymbol{w}} = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix} \in so(3)$.

   (a) (4 points) Show that $\operatorname{rank}(\widehat{\boldsymbol{w}}) = 2$. What is the vector in the null space of $\widehat{\boldsymbol{w}}$? Give a geometric interpretation to your result.

   ANSWER: Recall from class that any essential matrix is of the form $E = \widehat{\boldsymbol{w}}R$, where $R \in SO(3)$, and must have an SVD of the form $U\Sigma V^\top$, where $U, V \in SO(3)$ and $\Sigma = \operatorname{diag}(\|\boldsymbol{w}\|, \|\boldsymbol{w}\|, 0)$. Since $\widehat{\boldsymbol{w}}$ is an essential matrix with $R = I$ and $\boldsymbol{w} \neq \boldsymbol{0}$, it follows that $\widehat{\boldsymbol{w}}$ has two nonzero singular values, hence it must be of rank two. Now, it is easy to see that $\widehat{\boldsymbol{w}}\boldsymbol{w} = 0$. Thus the vector in the nullspace is the vector $\boldsymbol{w}$. This makes sense geometrically because $\widehat{\boldsymbol{w}}$ is the matrix generating the cross product and so $\widehat{\boldsymbol{w}}\boldsymbol{w} = \boldsymbol{w} \times \boldsymbol{w} = \boldsymbol{0}$.

   (b) (4 points) Show that $\widehat{\boldsymbol{w}}^2 = \boldsymbol{w}\boldsymbol{w}^\top - \|\boldsymbol{w}\|^2 I$ and $\widehat{\boldsymbol{w}}^3 = -\|\boldsymbol{w}\|^2 \widehat{\boldsymbol{w}}$.

   ANSWER:

$$\begin{aligned}
\widehat{\boldsymbol{w}}^2 &= \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix} \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix} = \begin{bmatrix} -w_2^2 - w_3^2 & w_1 w_2 & w_1 w_3 \\ w_1 w_2 & -w_1^2 - w_3^2 & w_2 w_3 \\ w_1 w_3 & w_2 w_3 & -w_1^2 - w_2^2 \end{bmatrix} \\
&= \begin{bmatrix} w_1^2 & w_1 w_2 & w_1 w_3 \\ w_1 w_2 & w_2^2 & w_2 w_3 \\ w_1 w_3 & w_2 w_3 & w_3^2 \end{bmatrix} - \begin{bmatrix} w_1^2 + w_2^2 + w_3^2 & 0 & 0 \\ 0 & w_1^2 + w_2^2 + w_3^2 & 0 \\ 0 & 0 & w_1^2 + w_2^2 + w_3^2 \end{bmatrix} \\
&= \boldsymbol{w}\boldsymbol{w}^\top - \|\boldsymbol{w}\|^2 I \\
\widehat{\boldsymbol{w}}^3 &= (\boldsymbol{w}\boldsymbol{w}^\top - \|\boldsymbol{w}\|^2 I)\widehat{\boldsymbol{w}} = \boldsymbol{0} - \|\boldsymbol{w}\|^2 \boldsymbol{w}
\end{aligned}$$

   where $\boldsymbol{w}^\top \widehat{\boldsymbol{w}} = (\boldsymbol{w} \times \boldsymbol{w})^\top = \boldsymbol{0}^\top$.

   (c) (4 points) Assume that a camera moves with a rotation $R(t) \in SO(3)$ as a function of time $t \in \mathbb{R}$. Use the relationship $R(t)R(t)^\top = I$ to show that there exists a skew symmetric matrix $\widehat{\boldsymbol{w}}(t)$ such that $\dot{R}(t) = \widehat{\boldsymbol{w}}(t)R(t)$. Assume now that $\widehat{\boldsymbol{w}}(t) \equiv \widehat{\boldsymbol{w}}$ is constant, i.e., it doesn't change with time. Show that the solution to the differential equation $\dot{R}(t) = \widehat{\boldsymbol{w}}R(t)$ is given by

$$R(t) = \exp(\widehat{\boldsymbol{w}}t)R_0, \quad t \geq 0, \tag{1}$$

   where $R_0 \in SO(3)$ is the initial rotation at $t = 0$ and $\exp(A) = \sum_{n=0}^{\infty} \frac{A^n}{n!}$ is the matrix exponential.

   ANSWER: We have $\dot{R}(t)R(t)^\top + R(t)\dot{R}(t)^\top = 0$. Letting $S(t) = \dot{R}(t)R(t)^\top$, we see that $S(t) = -S(t)^\top$. Therefore, the exists $\boldsymbol{w}(t)$ such that $S(t) = \widehat{\boldsymbol{w}}(t)$. Now if $\boldsymbol{w}(t)$ is constant, then $\dot{R} = \widehat{\boldsymbol{w}}R$ is a linear differential equation with constant coefficients. To verify that (1) is the solution to this linear ODE, we need to verify that (1) it satisfies the initial condition, (2) it satisfies the ODE. It is easy to see that $\exp(\widehat{\boldsymbol{w}}0) = I$. Therefore, the initial condition is $R(0) = R_0$, which is the initial rotation. Next, notice that

$$\frac{d}{dt}\exp(\widehat{\boldsymbol{w}}t) = \frac{d}{dt}\sum_{n=0}^{\infty}\frac{\widehat{\boldsymbol{w}}^n}{n!}t^n = \sum_{n=1}^{\infty}\frac{\widehat{\boldsymbol{w}}^n}{(n-1)!}t^{n-1} = \sum_{n=0}^{\infty}\frac{\widehat{\boldsymbol{w}}^{n+1}}{n!}t^n = \widehat{\boldsymbol{w}}\exp(\widehat{\boldsymbol{w}}t). \tag{2}$$

   Therefore, the derivative of (1) is $\dot{R}(t) = \widehat{\boldsymbol{w}}\exp(\widehat{\boldsymbol{w}}t)R_0 = \widehat{\boldsymbol{w}}R(t)$.

(d) (4 points) Let $R = \exp(\widehat{\boldsymbol{w}}\theta)$ for some $\boldsymbol{w} \in \mathbb{R}^3$ such that $\|\boldsymbol{w}\| = 1$. Show that

$$R = I + \sin(\theta)\widehat{\boldsymbol{w}} + (1 - \cos(\theta))\widehat{\boldsymbol{w}}^2. \tag{3}$$

ANSWER: Since $\|\boldsymbol{w}\| = 1$, it follows from part b) that $\widehat{\boldsymbol{w}}^2 = \boldsymbol{w}\boldsymbol{w}^\top - I$ and $\widehat{\boldsymbol{w}}^3 = -\widehat{\boldsymbol{w}}$. Moreover, $\widehat{\boldsymbol{w}}^4 = -\widehat{\boldsymbol{w}}^2$, $\widehat{\boldsymbol{w}}^5 = -\widehat{\boldsymbol{w}}^3 = \widehat{\boldsymbol{w}}$, $\widehat{\boldsymbol{w}}^6 = \widehat{\boldsymbol{w}}^2$, and so on. Therefore,

$$\begin{aligned}
R = \exp(\widehat{\boldsymbol{w}}\theta) &= \sum_{i=1}^{\infty} \frac{\widehat{\boldsymbol{w}}^i\theta^i}{i!} = I + \Big(\frac{\widehat{\boldsymbol{w}}\theta}{1!} + \frac{\widehat{\boldsymbol{w}}^3\theta^3}{3!} + \frac{\widehat{\boldsymbol{w}}^5\theta^5}{5!} + \dots\Big) + \Big(\frac{\widehat{\boldsymbol{w}}^2\theta^2}{2!} + \frac{\widehat{\boldsymbol{w}}^4\theta^4}{4!} + \frac{\widehat{\boldsymbol{w}}^6\theta^6}{6!} + \dots\Big) \\
&= I + \widehat{\boldsymbol{w}}\Big(\frac{\theta}{1!} - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} + \dots\Big) + \widehat{\boldsymbol{w}}^2\Big(\frac{\theta^2}{2!} - \frac{\theta^4}{4!} + \frac{\theta^6}{6!} + \dots\Big) = I + \sin(\theta)\widehat{\boldsymbol{w}} + (1 - \cos(\theta))\widehat{\boldsymbol{w}}^2.
\end{aligned}$$

(e) (2 points) Show that $\text{trace}(R) = 1 + 2\cos(\theta)$.

ANSWER: Note that $\text{trace}(\widehat{\boldsymbol{w}}) = 0$ and $\text{trace}(\widehat{\boldsymbol{w}}^2) = \text{trace}(\boldsymbol{w}\boldsymbol{w}^\top - \|\boldsymbol{w}\|^2 I) = \|\boldsymbol{w}\|^2 - 3\|\boldsymbol{w}\|^2 = -2\|\boldsymbol{w}\|^2$. Then $\text{trace}(R) = \text{trace}(I + \sin(\theta)\widehat{\boldsymbol{w}} + (1 - \cos(\theta))\widehat{\boldsymbol{w}}^2) = 3 + 0 - 2(1 - \cos(\theta)) = 1 + 2\cos(\theta)$.

(f) (2 points) Show that $\boldsymbol{w}$ is an eigenvector of R and give a geometric interpretation to your result.

ANSWER: We have that $R\boldsymbol{w} = (I + \sin(\theta)\widehat{\boldsymbol{w}} + (1 - \cos(\theta))\widehat{\boldsymbol{w}}^2)\boldsymbol{w} = \boldsymbol{w} + \mathbf{0} + \mathbf{0}$. Thus $\boldsymbol{w}$ is an eigenvector of R with eigenvalue 1. This makes geometric sense because R is a rotation about $\boldsymbol{w}$ and the axis of rotation is not affected by the rotation.

2. **(16 points) Optical flow of a perspective camera.**

(a) (8 points) Let $\boldsymbol{x}(t) = (x(t), y(t), 1) \in \mathbb{P}^2$ be the projection of a fixed point $\boldsymbol{X} = (X, Y, Z) \in \mathbb{R}^3$ onto a perspective camera moving with rotation $R(t) \in SO(3)$ and translation $T(t) \in \mathbb{R}^3$ at time t, i.e., $\boldsymbol{x}(t) = \pi(R(t)\boldsymbol{X} + T(t))$, where $\pi(\boldsymbol{X}) = \boldsymbol{X}/Z$ denotes perspective projection. Assume that the initial pose of the camera is $R(0) = I$ and $T(0) = \mathbf{0}$. Let $\boldsymbol{w}(t) \in \mathbb{R}^3$ and $\boldsymbol{v}(t) \in \mathbb{R}^3$ be, respectively, the rotational and translational velocities of the camera at time t, and recall that $\dot{R}(t)R(t)^\top = \widehat{\boldsymbol{w}}(t)$ and $\dot{T}(t) = \boldsymbol{v}(t)$. Show that the optical flow induced by the camera at time $t = 0$ is given by

$$\begin{aligned}
\dot{\boldsymbol{x}} &= (-\widehat{\boldsymbol{x}} + \boldsymbol{x}e_3^T\widehat{\boldsymbol{x}})\boldsymbol{w} + \frac{1}{Z}(I - \boldsymbol{x}e_3^T)\boldsymbol{v} \\
\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} &= \begin{bmatrix}-xy & 1+x^2 & -y\\ -(1+y^2) & xy & x\end{bmatrix}\boldsymbol{w} + \frac{1}{Z}\begin{bmatrix}1 & 0 & -x\\ 0 & 1 & -y\end{bmatrix}\boldsymbol{v},
\end{aligned} \tag{4}$$

where $e_3 = (0, 0, 1) \in \mathbb{R}^3$, and $\boldsymbol{w}$ and $\boldsymbol{v}$ are the rotational and translational velocities at $t = 0$.

ANSWER: Let $\lambda(t)$ denote the depth of point $\boldsymbol{X}$ at time t. We have that $\lambda(t)\boldsymbol{x}(t) = R(t)\boldsymbol{X} + T(t)$. Therefore $\dot{\lambda}\boldsymbol{x} + \lambda\dot{\boldsymbol{x}} = \dot{R}\boldsymbol{X} + \dot{T}$. Moreover, $\lambda(t) = e_3^\top(R(t)\boldsymbol{X} + T(t))$, and so $\dot{\lambda} = e_3^\top(\dot{R}\boldsymbol{X} + \dot{T})$. We then have

$$\lambda\dot{\boldsymbol{x}} = \dot{R}\boldsymbol{X} + \dot{T} - \dot{\lambda}\boldsymbol{x} = \dot{R}\boldsymbol{X} + \dot{T} - \boldsymbol{x}e_3^\top(\dot{R}\boldsymbol{X} + \dot{T}) = (I - \boldsymbol{x}e_3^\top)(\dot{R}\boldsymbol{X} + \dot{T})$$

Since at $t = 0$ we also have $\dot{R} = \widehat{\boldsymbol{w}}$, $\dot{T} = \boldsymbol{v}$, $\lambda\boldsymbol{x} = \boldsymbol{X}$ and $\lambda = Z$, and also $\widehat{\boldsymbol{w}}\boldsymbol{x} = -\widehat{\boldsymbol{x}}\boldsymbol{w}$, we obtain

$$\dot{\boldsymbol{x}} = \frac{1}{\lambda}(I - \boldsymbol{x}e_3^\top)(\widehat{\boldsymbol{w}}\lambda\boldsymbol{x} + \boldsymbol{v}) = -(I - \boldsymbol{x}e_3^\top)\widehat{\boldsymbol{x}}\boldsymbol{w} + \frac{1}{\lambda}(I - \boldsymbol{x}e_3^\top)\boldsymbol{v}.$$

The rest follows by direct substitution as

$$\begin{aligned}
\dot{\boldsymbol{x}} &= -\begin{bmatrix}1 & 0 & -x\\ 0 & 1 & -y\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}0 & -1 & y\\ 1 & 0 & -x\\ -y & x & 0\end{bmatrix}\boldsymbol{w} + \frac{1}{Z}\begin{bmatrix}1 & 0 & -x\\ 0 & 1 & -y\\ 0 & 0 & 0\end{bmatrix}\boldsymbol{v} \\
&= \begin{bmatrix}-xy & 1+x^2 & -y\\ -(1+y^2) & xy & x\\ 0 & 0 & 0\end{bmatrix}\boldsymbol{w} + \frac{1}{Z}\begin{bmatrix}1 & 0 & -x\\ 0 & 1 & -y\\ 0 & 0 & 0\end{bmatrix}.
\end{aligned}$$

(b) (7 points) Assume that a perspective camera observes a 3-D plane $\boldsymbol{N}^T\mathbf{X} = d$, where $\boldsymbol{N} = (n_1, n_2, n_3)$ is the normal to the plane and d is the distance from the plane to the camera center. Show that the optical flow observed by a camera moving with angular and linear velocity $\boldsymbol{w}$, $\boldsymbol{v}$ is a polynomial of degree 2 in (x, y) with 8 different coefficients. Find explicit formulae for each one of the coefficients as a function of $\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{N}, d$. Under what conditions on $\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{N}, d$ can the optical flow be approximated by an affine flow model $\dot{\boldsymbol{x}} = A\boldsymbol{x} + \boldsymbol{b}$? Find expressions for A and $\boldsymbol{b}$ as a function of $\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{N}, d$ under those conditions.

ANSWER: Suppose P is a 3-D plane $\boldsymbol{N}^\top \boldsymbol{X} = d$ with normal $N = (n_x, n_y, n_z)$ observed by a camera moving with linear and angular velocity $\boldsymbol{w}, \boldsymbol{v}$. Since $\boldsymbol{x} = \boldsymbol{X}/Z$, we have $\boldsymbol{N}^\top \boldsymbol{x} = d/Z$ and so

$$Z = \frac{d}{\boldsymbol{N}^\top \boldsymbol{x}} = \frac{d}{n_x x + n_y y + n_z}.$$

Substituting X to the general motion field equation (4) we obtain

$$\begin{aligned}\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} &= \begin{bmatrix}-xy & 1+x^2 & -y\\ -(1+y^2) & xy & x\end{bmatrix}\boldsymbol{w} + \frac{n_x x + n_y y + n_z}{d}\begin{bmatrix}1 & 0 & -x\\ 0 & 1 & -y\end{bmatrix}\boldsymbol{v},\\ &= \begin{bmatrix}a_1x^2 + a_2xy + a_3x + a_4y + a_5\\ a_4xy + a_5y^2 + a_6y + a_7x + a_8\end{bmatrix}\end{aligned} \tag{5}$$

where the coefficients have the following form

$$\begin{array}{lllllllll} a_1 &=& w_y + \frac{v_z n_x}{d} & a_2 &=& -w_x - \frac{v_z n_y}{d} & a_3 &=& \frac{1}{d}(n_x v_x - n_z v_z)\\ a_4 &=& -w_z + \frac{v_x n_y}{d} & a_5 &=& w_y + \frac{v_x n_z}{d} & a_6 &=& \frac{1}{d}(n_y v_y - n_z v_z)\\ a_7 &=& w_z + \frac{v_y n_x}{d} & a_8 &=& -w_x + \frac{v_y n_z}{d}. & & & \end{array}$$

In order to approximate this model by an affine motion models we can neglect the second order terms assuming $a_1 = a_2 = a_4 = a_5 = 0$ and obtain a 6 parameter affine motion model

$$\begin{array}{lll}\dot{x} &=& a_3x + a_4y + a_5\\ \dot{y} &=& a_6y + a_7x + a_8\end{array} \implies \dot{\boldsymbol{x}} = A\boldsymbol{x} + d.$$

3. **(15 points) 8-point algorithm.** Implement a function `[R,T,X] = twoviewSFM(x1,x2)` that reconstructs camera motion `(R,T)` $\in SE(3)$ and 3D structure $X \in \mathbb{R}^{3\times P}$ from a set of point correspondences `x1,x2` $\in \mathbb{P}^{2\times P}$ using the 8-point algorithm (Alg. 5.1 of book-MASKS). Test the algorithm on the dataset $(\boldsymbol{x}_1, \boldsymbol{x}_2)$ available here. The given point correspondences are noise free, and so your algorithm should be able to recover the given R_{true}, T_{true} and X_{true}. To evaluate the robustness of your algorithm with noisy data, add Gaussian noise with standard deviation $\sigma \in \{0, 10^{-4}, 2\times 10^{-4}, \ldots, 10^{-3}\}$. Be careful to add noise only to the two pixel coordinates. Repeat the experiment for $N = 100$ trials. For each trial, compute the the error in rotation, $\theta_R = |\operatorname{acos}(\operatorname{trace}(R_{est}^\top R_{true}) - 1)/2|$, the error in translation $\theta_T = |\operatorname{acos}(T_{est}^\top T_{true})|$, and the error in 3D reconstruction $\theta_{\boldsymbol{X}} = |\operatorname{acos}(\operatorname{trace}(X_{est}^\top X_{true}))|$ (measured in degrees) between the estimated and the true quantity. Make sure that $\|T\|_2 = 1$ and $\|X\|_F = 1$ before you compute errors.

ANSWER:

```
function [R,T,X] = twoviewSFM(x1,x2)
assert(size(x1,1)==size(x2,1), '# of points do not agree.');
n=size(x1,2);

% form the Xi matrix
Xi=zeros(n,9);
for i = 1:n
    Xi(i,:) = kron(x1(:,i),x2(:,i))';
end

[¬,¬,V] = svd(Xi);
E = reshape(V(:,9),3,3);
%SVD of E
```

```
[U,¬,V] = svd(E);
%Rz(pi/2)
RzT = [0 1 0; -1 0 0; 0 0 1];
R(:,:,1) = U*RzT*V';
skew_T = U*RzT'*diag([1 1 0])*U';
T(:,1) = [skew_T(3,2);skew_T(1,3);skew_T(2,1)];
%Rz(-pi/2)
R(:,:,2) = U*RzT'*V';
skew_T = U*RzT*diag([1 1 0])*U';
T(:,2) = [skew_T(3,2);skew_T(1,3);skew_T(2,1)];
%SVD of -E
[U,¬,V] = svd(-E);
%Rz(pi/2)
RzT = [0 1 0; -1 0 0; 0 0 1];
R(:,:,3) = U*RzT*V';
skew_T = U*RzT'*diag([1 1 0])*U';
T(:,3) = [skew_T(3,2);skew_T(1,3);skew_T(2,1)];
%Rz(-pi/2)
R(:,:,4) = U*RzT'*V';
skew_T = U*RzT*diag([1 1 0])*U';
T(:,4) = [skew_T(3,2);skew_T(1,3);skew_T(2,1)];

% used to store the valid transofrmation
R_valid=zeros(3,3);
T_valid=zeros(3,1);
% find correct transformation
valid_count = 0;
for i=1:4
    [valid, lambda1,lambda2] = check(R(:,:,i),T(:,i),x1,x2);
    if valid == 1
        R_valid = R(:,:,i);
        T_valid = T(:,i);
        fprintf('%d th tranformation is valid\n',i);
        valid_count = valid_count + 1;
    end
end
if valid_count > 1
    fprintf('something wrong\n');
end
R=R_valid;
T=T_valid;
X = x1.*repmat(lambda1(:,1)',3,1);
end

%to check whether the R T is valid and return the depth of each point for
%camera 1
function [valid,depth1,depth2] = check(R,T,x1,x2)
depth = computeDepth(R,T,x1,x2);
isLessThanZero = depth < 0;
numOfLess = sum(double(isLessThanZero(:)));
if numOfLess == 0 && det(R)>0
    valid = 1;
else
    valid = 0;
end
depth1 = depth(:,1);
depth2 = depth(:,2);
end

%compute the depth of each point for camera 1
function depth = computeDepth(R,T,x1,x2)
P1 = [R T];
P2 = [eye(3) zeros(3,1)];
for i=1:size(x1,2)
    p1 = x1(:,i);
    p2 = x2(:,i);
    A = [p1(1,1).*P1(3,:) - P1(1,:); p1(2,1).*P1(3,:) - P1(2,:);p2(1,1).*P2(3,:) - ...
        P2(1,:);p2(2,1).*P2(3,:) - P2(2,:)];
```

```
    [¬,¬,V] = svd(A);
    X(:,i) = V(:,4) ./ repmat(V(4,4),4,1);
end
depth = X(3,:)';
depth(:,2) = 1;

end
```

```
clear all
close all
load('data-8-point-algorithm.mat');
[Rest,Test,Xest] = twoviewSFM(x1,x2);
theta_R = acos(trace(Rest'*R)-1)/2;
theta_T = acos(Test'*T);
```

Submission instructions. Send email to vision14jhu@gmail.com with subject 600.461/600.661:HW5 and attachment `firstname-lastname-hw5-vision14.zip` or `firstname-lastname-hw5-vision14.tar.gz`. The attachment should have the following content:

1. A file called `hw5.pdf` containing your answers to each one of the analytical questions. If at all possible, you should generate this file using the latex template hw1-vision14.tex. If not possible, you may use another editor, or scan your handwritten solutions. But note that you must submit a single PDF file with all your answers.

2. For coding questions, submit a file called `README`, which contains instructions on how to run your code. Use separate directories for each coding problem. Each directory should contain all the functions and scripts you are asked to write in separate files. For example, for HW2 the structure of what you should submit could look like

   (a) `README`

   (b) `hw2.pdf`

   (c) `hw2q3`: `hw2q3c.m`, `hw2q3e.m`

   (d) `hw2q4`: `hw2q4b.m`, `hw2q4c.m`

   The TA will run your scripts to generate the results. Thus, your script should include all needed plotting commands so that figures pop up automatically. Please make sure that the figure numbers match those you describe in `hw2.pdf`. You do not need to submit input or output images. The output images should be automatically generated by your scripts so that the TA can see the results by just running the scripts. In writing your code, you should assume that the TA will place the input images in the directory that is relevant to the question solved by your script. Also, make sure to comment your code properly.