Exercise on Image Stiching

# 1 Question 1

Make a function Hest, which from two point pairs can estimate a homoraphy via the linear algorithm of section 2.91. in the lecture notes. Remember to normalize the points as described in ”Exercise in Estimating View Geometry”.

To test Hest make a test data set via: q1=rand(3,10);

Htrue=[10 0 -1;1 10 20;0.01 0 3];

q2=Htrue\*q1

\_ Demonstrate that Hest can estimate Htrue from q1 and q2.

\_ Do the same for Htrue=[10 0 -1;1 10 20;0.02 0 3];

Nomalization:

Normalizes 2D homogeneous points, T -The 3x3 transformation matrix, newpts = T\*pts’

% For the finite points ensure homogeneous coords have scale of 1

pts(1,finiteind) = pts(1,finiteind)./pts(3,finiteind);

pts(2,finiteind) = pts(2,finiteind)./pts(3,finiteind);

pts(3,finiteind) = 1;

c = mean(pts(1:2,finiteind)')'; % Centroid of finite points

newp(1,finiteind) = pts(1,finiteind)-c(1); % Shift origin to centroid.

newp(2,finiteind) = pts(2,finiteind)-c(2);

dist = sqrt(newp(1,finiteind).^2 + newp(2,finiteind).^2);

meandist = mean(dist(:)); % Ensure dist is a column vector

scale = sqrt(2)/meandist;

T = [scale 0 -scale\*c(1)

0 scale -scale\*c(2)

0 0 1 ];

q1=rand(3,10);

Htrue=[10 0 -1;1 10 20;0.01 0 3]

q2=Htrue\*q1;

Htrue = 10.0000 0 -1.0000

1.0000 10.0000 20.0000

0.0100 0 3.0000

H = 0.9962 -0.0000 -0.0996

0.0996 0.9962 1.9923

0.0010 0.0000 0.2989

function H = hest( q1,q2 )

% Finding H X1(2\*4)

s1=size(q1,2);

B=[];

for i=1:s1

b= kron(q1(:,i),vl\_hat(q2(:,i)));

B =[B; b'];

end

[~,~,V] = svd(B);

Q = V(:,end);

HL = reshape(Q,3,3);

S = svd(HL);

H = HL/S(2)

end

## 2 Question 2

Load the images ImL.jpg and ImR.jpg, and match them via the following code:

[fa, da] = vl sift(single(rgb2gray(ImL)));

[fb, db] = vl sift(single(rgb2gray(ImR)));

[matches, scores] = vl ubcmatch(da, db);

nMatch=size(matches,2);

After having initialized vl feat via vl setup, see ”Exercise in Geometry Constrained Feature

Matching”.

Devise away of illustrating the result and apply it.

The two original images have Matching points: 586.

## 3 Question 3

Use Ransac to fit a homography to the matches from Question 2, using the homography estimator

from Question 1. Use equation (2.45) from the lecture notes as a distance measure, with a distance

threshold corresponding to \_ = 3.1

Find the inlier matches and illustrate/document the result.

1Note: Using equation (2.41) in the lecture notes assumes that the the coordinates have been normalized.

4 points needed for H estimation.



function dist = Hdistance( H,p1,p2)

a=H\*p2;

b=inv(H)\*p1;

dist = (norm( (a(1:2,:)./ repmat(a(3,:),2,1)) - (p1(1:2,:)./ repmat(p1(3,:),2,1)))) +...

(norm( (b(1:2,:)./ repmat(b(3,:),2,1)) - (p2(1:2,:)./ repmat(p2(3,:),2,1))));

end

Ransac:

sigma=3

t = 5.99 \* sigma^2;%the smaller the t is, the more iter needed

iter=200;

for i = 1:iter

subset = vl\_colsubset(1:nMatch,4);

H1{i} = hest(p1(:,subset), p2(:,subset));

inlier=0;

for j= 1: size(p1,2)

d(j)= Hdistance(H1{i},p2(:,j),p1(:,j));

if d(j) <= t

inlier = inlier + 1;

tr1(:,j) = p1(:,j);

tr2(:,j) = p2(:,j);

end

inlierT (i) = inlier;

end

tt1{i} = mat2cell(tr1);

tt2{i} = mat2cell(tr2);

end

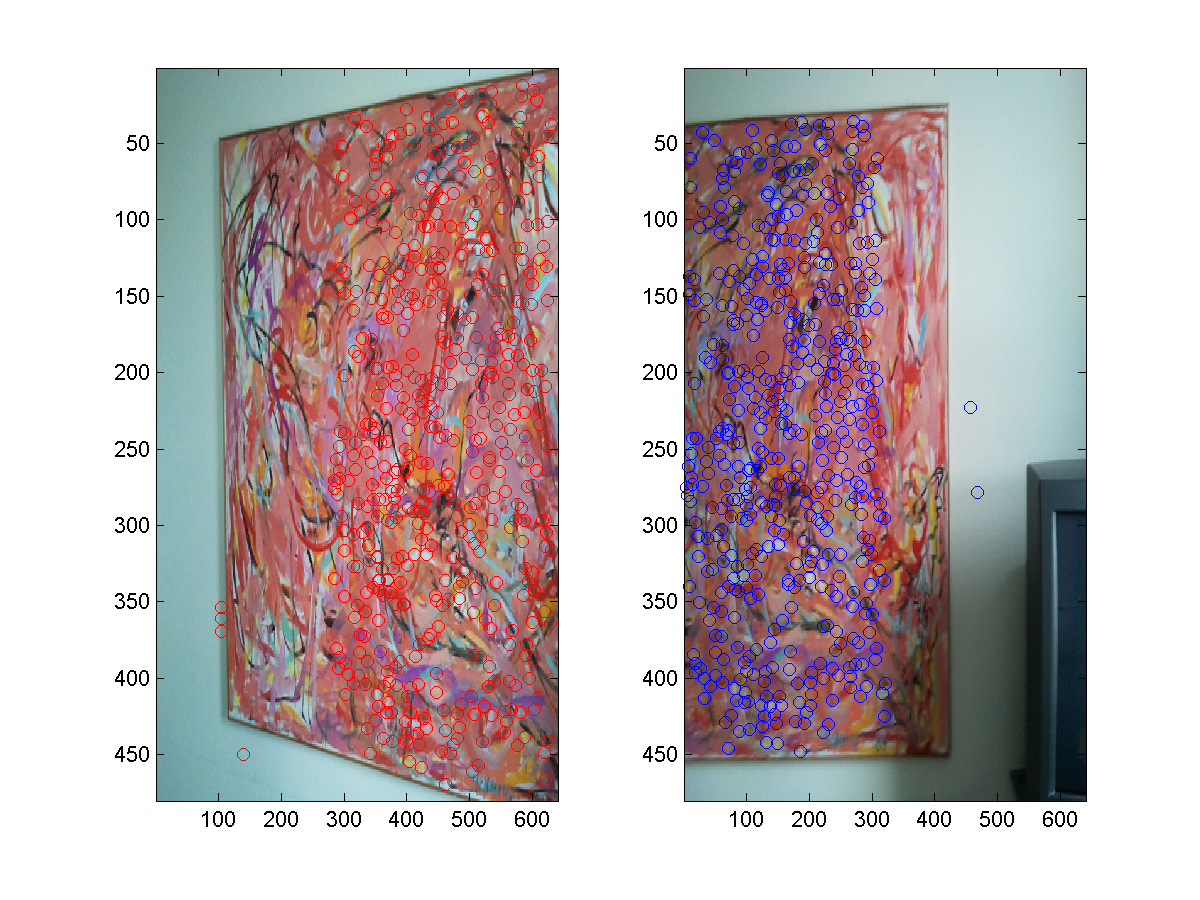
figure(2)

plot(p1,p2)

figure(3)

[val,I]=max(inlierT)

Highest inliers = 567



## Question 4

Improve the estimate by fitting a homography to all the inliers via the algorithm from Question 1.

Warp ImL vi this homography, e.g. using the code:

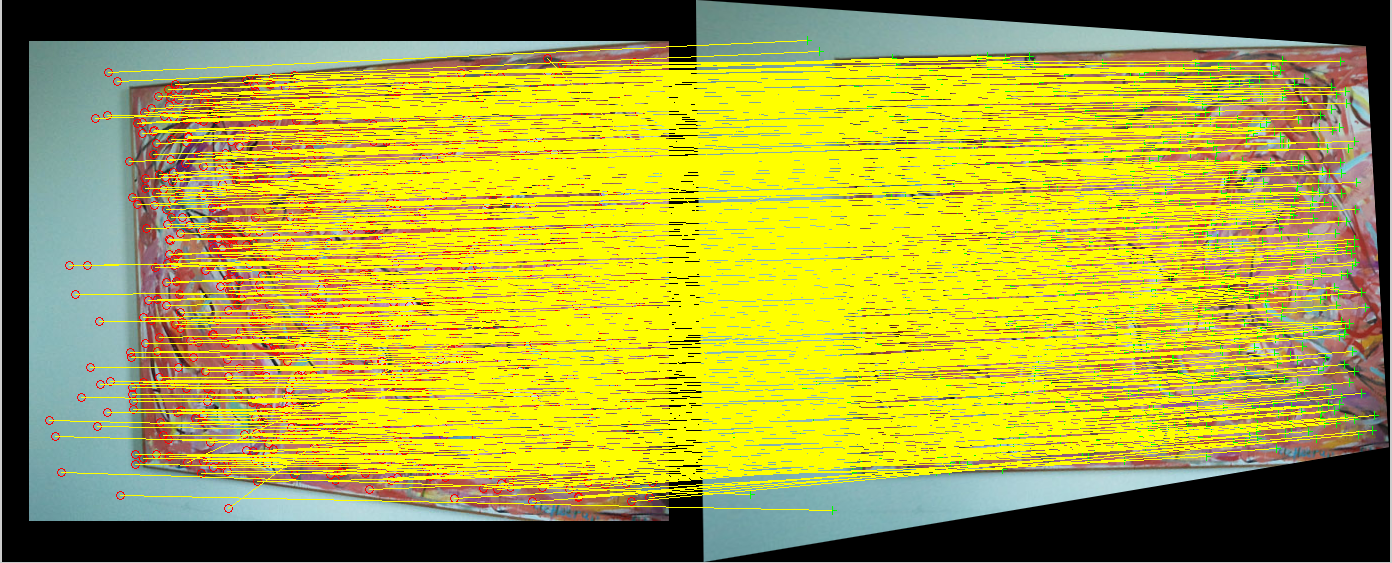
T = maketform(’projective’,H’);

ImH=imtransform(ImL,T);

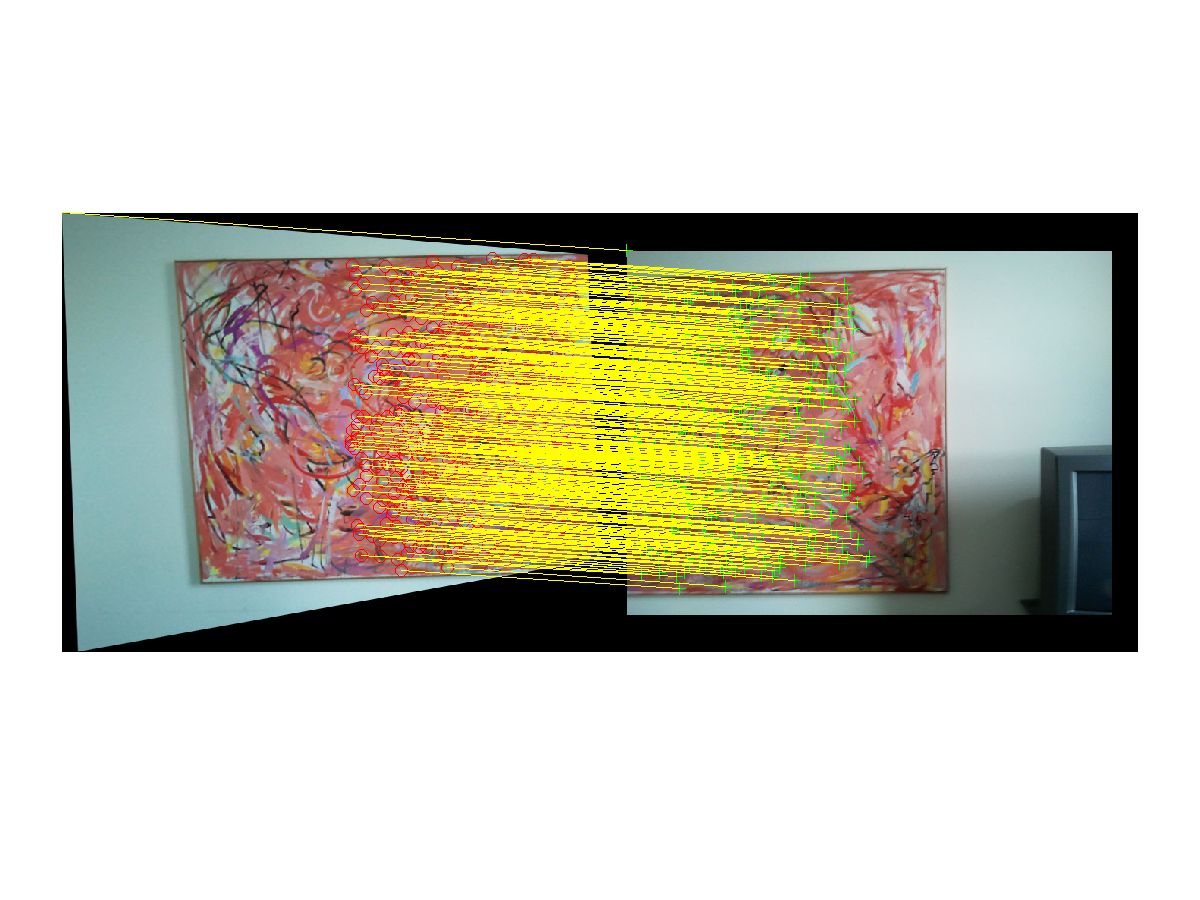
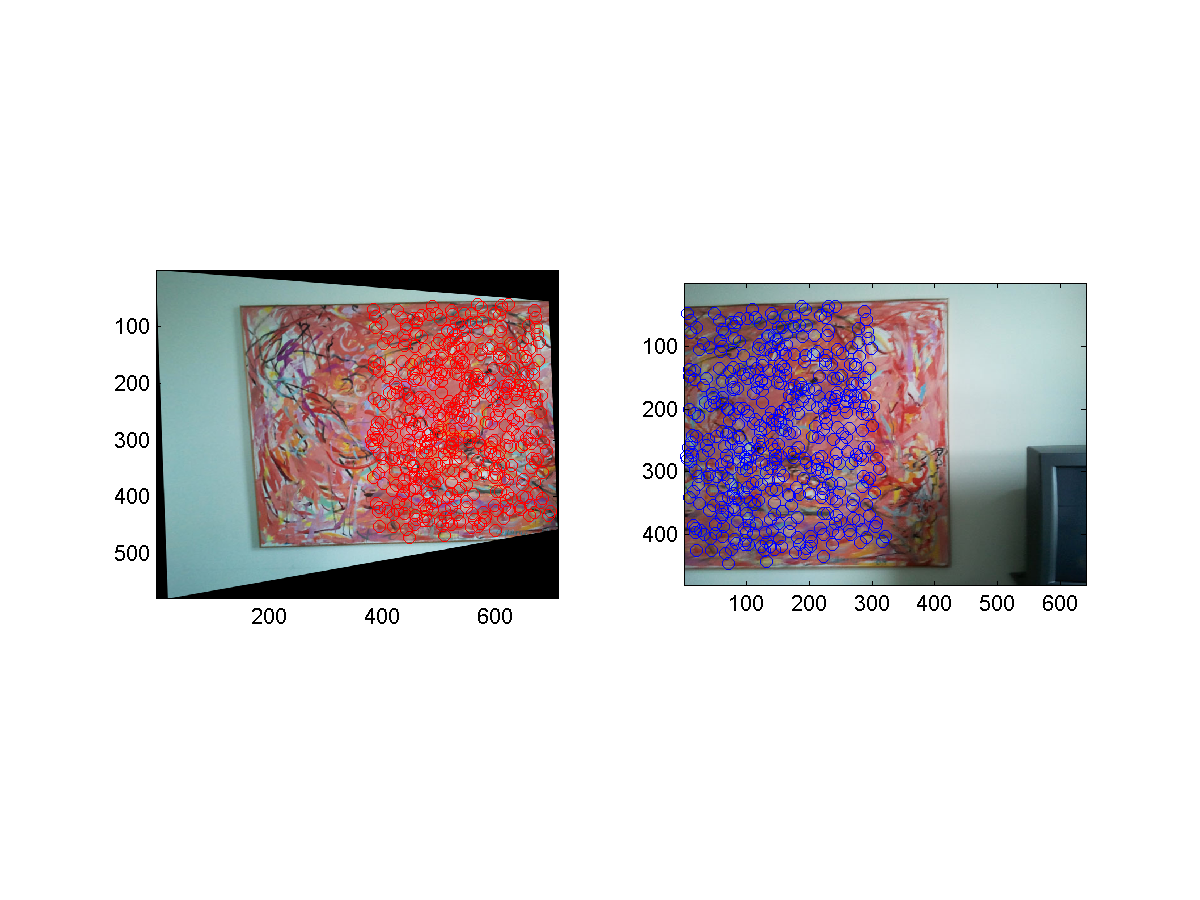
imagesc(ImH)

axis image

original and H transformed left images have matching point 546



Highest Inliers are 513



## 5 Question 5

Use the supplied function WarpNView(H,ImL,ImR), which warps image ImL via H and merges

the result with ImR. Comment on the result, in particular what this imples about the quality of your

estimate of the homography H. Note that the image input to WarpNView should be RGB images.

q1 = H q2

H = A2 R inv(A1)

The higher the H quality the better the two images merge (the 2d points are estimated).

H estimate 2d points from both sides, so that they could merge.

