88

VAN – Homwork #2

Tom Avrech 039823893

Omri Asraf 305054488

Basic probability and Bayesian Inference

1)

$$x \sim N(\mu, \Sigma)$$

$$\Lambda = \Sigma^{-1};\ \eta = \Lambda\mu$$

$$\begin{aligned}
p(x) = \frac{1}{\sqrt{|2\pi\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \\
= \frac{1}{\sqrt{|2\pi\Lambda^{-1}|}} \exp\left(-\frac{1}{2}(x-\Lambda^{-1}\eta)^T\Lambda(x-\Lambda^{-1}\eta)\right) \\
= \frac{1}{\sqrt{|2\pi\Lambda^{-1}|}} \exp\left(-\frac{1}{2}(x-\Lambda^{-1}\eta)^T(\Lambda x-\eta)\right) \\
= \frac{1}{\sqrt{|2\pi\Lambda^{-1}|}} \exp\left(-\frac{1}{2}(\mathrm{x}^{\mathrm{T}}\Lambda x - x^T\eta - \eta^T x + \eta^T\Lambda^{-1}\eta)\right) \\
= \frac{1}{\sqrt{|2\pi\Lambda^{-1}|}} \exp\left(-\frac{1}{2}(\eta^T\Lambda^{-1}\eta)\right)\exp\left(-\frac{1}{2}\mathrm{x}^{\mathrm{T}}\Lambda x + \eta^T x\right) = N^{-1}(\eta, \Lambda)
\end{aligned}$$

2)

$$z = h(x) + v, \qquad v \sim N(0, \Sigma_{\mathrm{v}})$$

$$x \sim N(\hat{x}_0, \Sigma_0)$$

a.

$$p(x) = \frac{1}{\sqrt{|2\pi\Sigma_0|}} \exp\left(-\frac{1}{2}\|x-\hat{x}_0\|^2{}_{\Sigma_0}\right)$$

$$p(z|x) = \frac{1}{\sqrt{|2\pi\Sigma_{\mathrm{v}}|}} \exp\left(-\frac{1}{2}\|z-h(x)\|^2{}_{\Sigma_{\mathrm{v}}}\right)$$

b.

$$p(x|z_1) = \frac{p(z_1|x)p(x)}{p(z_1)} = \eta p(z_1|x)p(x)$$

you were asked to express the pdf
in terms of prior & observation models.....
[-1]

$$= \eta \frac{1}{\sqrt{|2\pi\Sigma_{\mathrm{v}}|}\sqrt{|2\pi\Sigma_0|}} \exp\left(-\frac{1}{2}\left(\|z_1-h(x)\|^2{}_{\Sigma_{\mathrm{v}}} + \|x-\hat{x}_0\|^2{}_{\Sigma_0}\right)\right)$$

c.

$$\begin{aligned}
x^* = arg\max_x p(x|z_1) = arg\min_x[-\log p(x|z_1)] \\
= arg\min_x[-\log p(z_1|x) - \log p(x)] = arg\min_x J(x)
\end{aligned}$$

$$x = \tilde{x} + \Delta x$$

First iteration we choose:

$$\tilde{x} = \hat{x}_0$$

$$z - h(\tilde{x} + \Delta x) = z - h(\tilde{x}) + H\Delta x, \qquad H = \left[\frac{dh}{dx}h\right]_{\tilde{x}}$$

$$\Delta x^* = arg\min_{\Delta x} J(\tilde{x} + \Delta x)$$

Where:

$$J(\tilde{x} + \Delta x) = \left\|\Sigma_0^{-\frac{1}{2}}(\Delta x + \tilde{x} - \hat{x}_0)\right\|^2 + \left\|\Sigma_v^{-\frac{1}{2}}(H\Delta x + h(\bar{x}) - z_1)\right\|^2$$

$$= \left\|\overbrace{\begin{pmatrix}\Sigma_0^{-\frac{1}{2}} \\ \Sigma_v^{-\frac{1}{2}}H\end{pmatrix}}^{A} x - \overbrace{\begin{pmatrix}\Sigma_0^{-\frac{1}{2}}(\hat{x}_0 - \tilde{x}) \\ \Sigma_0^{-\frac{1}{2}}(z_1 - h(\tilde{x}))\end{pmatrix}}^{b}\right\|^2 = \|Ax - b\|^2$$

$$= \left\|(A^TA)^{\frac{1}{2}}\left(x - (A^TA)^{-\frac{1}{2}}b\right)\right\|^2 = \left\|\left(x - (A^TA)^{-\frac{1}{2}}b\right)\right\|^2_{(A^TA)^{-1}}$$

$$\Delta x^* = arg\min_{\Delta x} J(\tilde{x} + \Delta x) = arg\min_{\Delta x}\|Ax - b\|^2$$

$$\Sigma_1 = (A^TA)^{-1}$$

$$A\Delta x^* - b = 0$$

$$\Delta x^* = (A^TA)^{-1}A^Tb$$

until convergence

$$\tilde{x} + \Delta x^* \longrightarrow \tilde{x}$$

After convergence

$$p(x|z_1) = N(\tilde{x}, (A^TA)^{-1})$$

d.

$$p(x|z_1, z_2) = \frac{p(z_2|x, z_1)p(x|z_1)}{p(z_2|z_1)} = \frac{p(z_2|x)p(x|z_1)}{p(z_2|z_1)} = \eta p(z_2|x)p(x|z_1)$$

$$J(\tilde{x} + \Delta x) = \left\|\Sigma_1^{-\frac{1}{2}}(\Delta x + \tilde{x} - \hat{x}_1)\right\|^2 + \left\|\Sigma_v^{-\frac{1}{2}}(H\Delta x + h(\tilde{x}) - z_2)\right\|^2$$

$$= \left\|\overbrace{\begin{pmatrix}\Sigma_1^{-\frac{1}{2}} \\ \Sigma_v^{-\frac{1}{2}}H\end{pmatrix}}^{A} x - \overbrace{\begin{pmatrix}\Sigma_0^{-\frac{1}{2}}(\hat{x}_1 - \tilde{x}) \\ \Sigma_0^{-\frac{1}{2}}(z_2 - h(\tilde{x}))\end{pmatrix}}^{b}\right\|^2$$

$$\Delta x^* = arg\min_{\Delta x} J(\tilde{x} + \Delta x) = arg\min_{\Delta x} \|Ax - b\|^2$$

$$\Sigma_1 = (A^T A)^{-1}$$

$$A\Delta x^* - b = 0$$

$$\Delta x^* = (A^T A)^{-1} A^T b$$

until convergence

$$\tilde{x} + \Delta x \longrightarrow \tilde{x}$$

After convergence

$$p(x|z_1, z_2) = N(\tilde{x}, (A^T A)^{-1})$$

3)

$$x_{k+1} = f(x_k, u_k) + w_k, \qquad w_k \sim N(0, \Sigma_\mathrm{w})$$

$$z = h(x_k) + v, \qquad v \sim N(0, \Sigma_\mathrm{v})$$

a.

$$p(x_k|x_{k-1}, u_{k-1}) = \frac{1}{\sqrt{|2\pi\Sigma_\mathrm{w}|}} \exp\left(-\frac{1}{2}\|x_k - f(x_{k-1}, u_{k-1})\|^2{}_{\Sigma_\mathrm{w}}\right)$$

b.

incorrect & incomplete [-2]
incorrect since your are missing an integral over x0
since you addded it to the expression
incomplete since you were asked to express the pdf
in terms of prior motion & observation models.....

$$p(x_1|z_1, u_0) = \frac{p(z_1|x_1, u_0)p(x_1|u_0)}{p(z_1|u_0)}$$

$$= \frac{p(z_1|x_1)p(x_1|x_0, u_0)p(x_0)}{p(z_1|u_0)} \sim p(z_1|x_1)p(x_1|x_0, u_0)p(x_0)$$

c.

$$p(x_k|u_{0:k-1}, z_{1:k}) \overset{bayes\ rule}{\cong} \frac{p(z_k|x_k, u_{0:k-1}, z_{1:k-1})p(x_k|u_{0:k-1}, z_{1:k-1})}{p(z_k|u_{0:k-1}, z_{1:k-1})}$$

$$= \eta \overbrace{p(z_k|x_k)}^{Makrov\ assumption} p(x_k|u_{0:k-1}, z_{1:k-1})$$

$$= \eta p(z_k|x_k) \overbrace{\int p(x_k, x_{k-1}|u_{0:k-1}, z_{1:k-1})}^{Marginalization}$$

$$\overset{chain\ rule}{\cong} \eta p(z_k|x_k) \int p(x_k|x_{k-1}, u_{0:k-1}, z_{1:k-1})p(x_{k-1}|u_{0:k-1}, z_{1:k-1})$$

$$= \eta p(z_k|x_k) \int \overbrace{p(x_k|u_{0:k-1}, z_{1:k-1})}^{Makrov\ assumption} p(x_{k-1}|u_{0:k-1}, z_{1:k-1})$$

$$p(x_k|u_{0:k-1}, z_{1:k}) \sim \overbrace{p(x_{k-1}|u_{0:n-2}, z_{1:k-1})}^{posteriori\ previous} \overbrace{p(z_k|x_k)}^{observation} \overbrace{p(x_k|x_{k-1}, u_{k-1})}^{motion}$$

$$J(x) = \left\|\Sigma_{prv}^{-\frac{1}{2}}(x_k - \hat{x}_{prv})\right\|^2 + \left\|\Sigma_v^{-\frac{1}{2}}(h(x_k) - z_k)\right\|^2 + \left\|\Sigma_w^{-\frac{1}{2}}(x_k - f(x_{k-1}, u_{k-1}))\right\|^2$$

$$x_k = \tilde{x} + \Delta x$$

$$J(\bar{x} + \varDelta x) = \left\| \varSigma_{prv}^{-\frac{1}{2}}\left(\varDelta x + \tilde{x} - \hat{x}_{prv}\right) \right\|^2 + \left\| \varSigma_v^{-\frac{1}{2}}(H\varDelta x + h(\tilde{x}) - z_k) \right\|^2 + \left\| \varSigma_w^{-\frac{1}{2}}\left(\varDelta x + \tilde{x} - f(x_{k-1}, u_{k-1})\right) \right\|^2$$

$$= \left\| \overbrace{\begin{pmatrix} \varSigma_{prv}^{-\frac{1}{2}} \\ \varSigma_v^{-\frac{1}{2}} H \\ \varSigma_w^{-\frac{1}{2}} \end{pmatrix}}^{A} x - \overbrace{\begin{pmatrix} \varSigma_{prv}^{-\frac{1}{2}}(\tilde{x} - \hat{x}_{prv}) \\ \varSigma_v^{-\frac{1}{2}}(h(\tilde{x}) - z_k) \\ \varSigma_w^{-\frac{1}{2}}(\tilde{x} - f(x_{k-1}, u_{k-1})) \end{pmatrix}}^{b} \right\|^2$$

$$\Delta x^* = arg\max_{\Delta x} p(x_k | u_{0:k-1}, z_{1:k}) = arg\min_{\varDelta x} J(\bar{x} + \varDelta x) = arg\min_{\varDelta x} \|Ax - b\|^2$$

$$\Delta x^* = (A^T A)^{-1} A^T b$$

until convergence

$$\tilde{x} + \Delta x^* \longrightarrow \tilde{x}$$

After convergence

$$p(x_k | u_{0:k-1}, z_{1:k}) = N(\tilde{x}, (A^T A)^{-1})$$

d.

$$p(x_{0:1} | u_0, z_1) = N(\hat{x}_{0:1}, \Sigma_{0:1}) = N^{-1}(\hat{\eta}_{0:1}, \mathrm{I}_{0:1})$$

$$\Sigma_{0:1} = \begin{bmatrix} \Sigma_{00} & \Sigma_{01} \\ \Sigma_{01}^{\mathrm{T}} & \Sigma_{11} \end{bmatrix}$$

what about dimensions?
[-1]

$$\mathrm{I}_{0:1} = \begin{bmatrix} \mathrm{I}_{00} & \mathrm{I}_{01} \\ \mathrm{I}_{01}^{\mathrm{T}} & \mathrm{I}_{11} \end{bmatrix}$$

covariance and information from Marginalization:

$$p(x_1 | u_0, z_1) = N(\hat{x}_1, \Sigma_{11}) = N^{-1}\left(\eta_1 - I_{01}^T I_{00}^{-1} \eta_0, \mathrm{I}_{11} - \mathrm{I}_{01}^{\mathrm{T}} I_{00}^{-1} I_{01}\right)$$

Hands-on Exercises

1.

(a)

P = K [R t] =

$$\begin{pmatrix} ax & 0 & 0 & u0 \\ 0 & ay & 0 & v0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R1 & R2 & R3 & t1 \\ R4 & R5 & R6 & t2 \\ R7 & R8 & R9 & t3 \\ 0 & 0 & 0 & 1 \end{pmatrix} =$$

$$\begin{pmatrix} 480 & 0 & 0 & 320 \\ 0 & 480 & 0 & 270 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0.5363 & -0.8440 & 0 & -451.2459 \\ 0.8440 & 0.5363 & 0 & 257.0322 \\ 0 & 0 & 1 & 400 \\ 0 & 0 & 0 & 1 \end{pmatrix} =$$

$$1.0e{+}05 \begin{pmatrix} 0.0026 & -0.0041 & 0 & -2.1628 \\ 0.0041 & 0.0026 & 0 & 1.2365 \\ 0 & 0 & 0 & 0.0040 \end{pmatrix}$$

(b)

(x,y,z) T = PX G = K [R t] X G → (u,v) T = (x z , y z) T

```
function [u v] = project3DPointToPixels(K, R, t, X)
Rt = [R(1,1) R(1,2) R(1,3) t(1); R(2,1) R(2,2) R(2,3) t(2); R(3,1) R(3,2) R(3,3) t(3);];
P = (K * Rt);
projected = P * X;
u = (projected(1) / projected(3));
v = (projected(2) / projected(3));
end
——-
K = [480 0 320; 0 480 270; 0 0 1];
R = [0.5363 -0.8440 0; 0.8440 0.5363 0; 0 0 1];
t = [-451.2459 257.0322 400];
X = [350; -250; -35; 1];
[u v] = project3DPointToPixels(K, R, t, X)
```

and what about writing your results?
[-5]

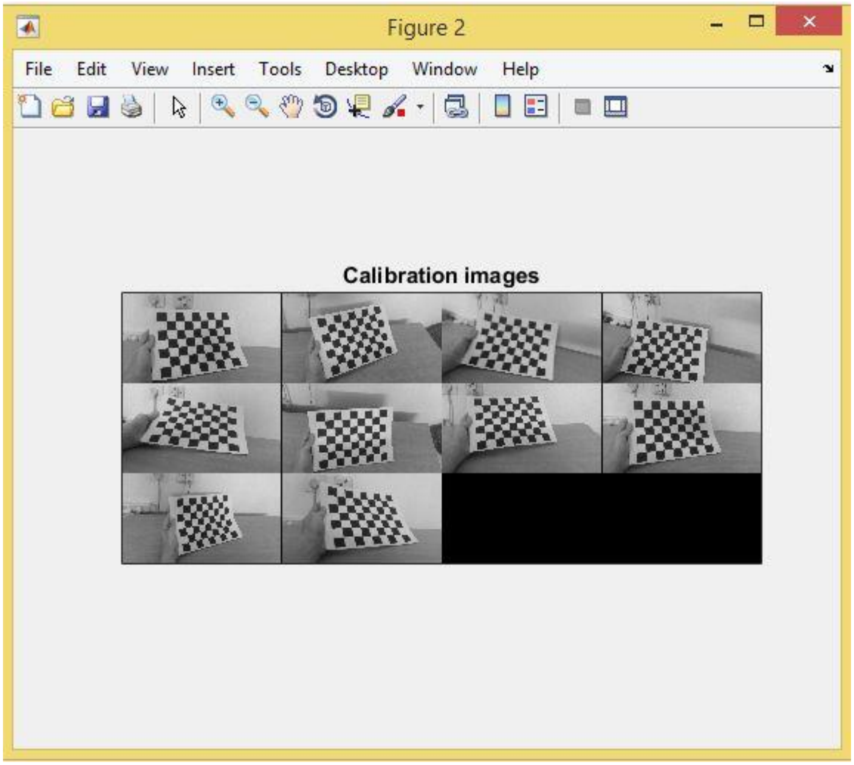
(c)

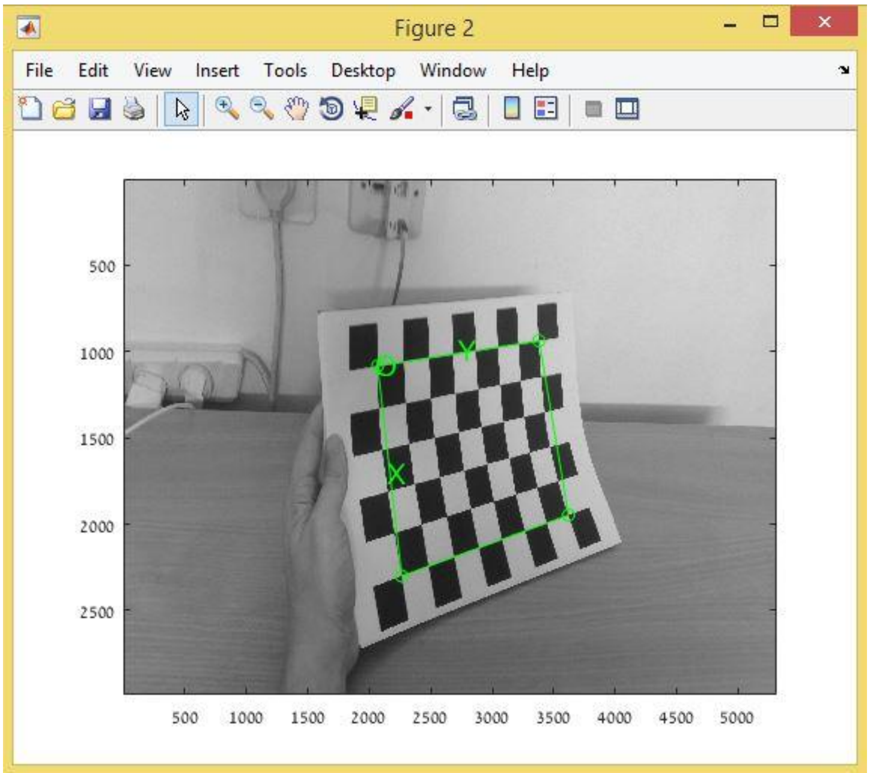
norm([u v] - [241.5 169]) = 651.2363

incorrect

[-3]

2)





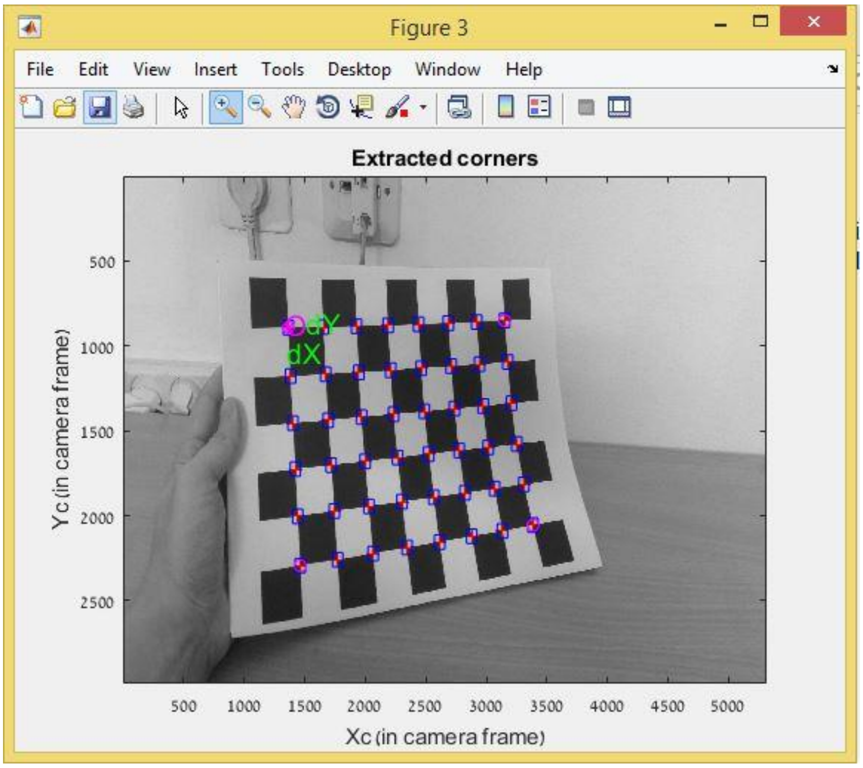


Figure 3
File Edit View Insert Tools Desktop Window Help
Extracted corners
dY
dX
Yc (in camera frame)
Xc (in camera frame)

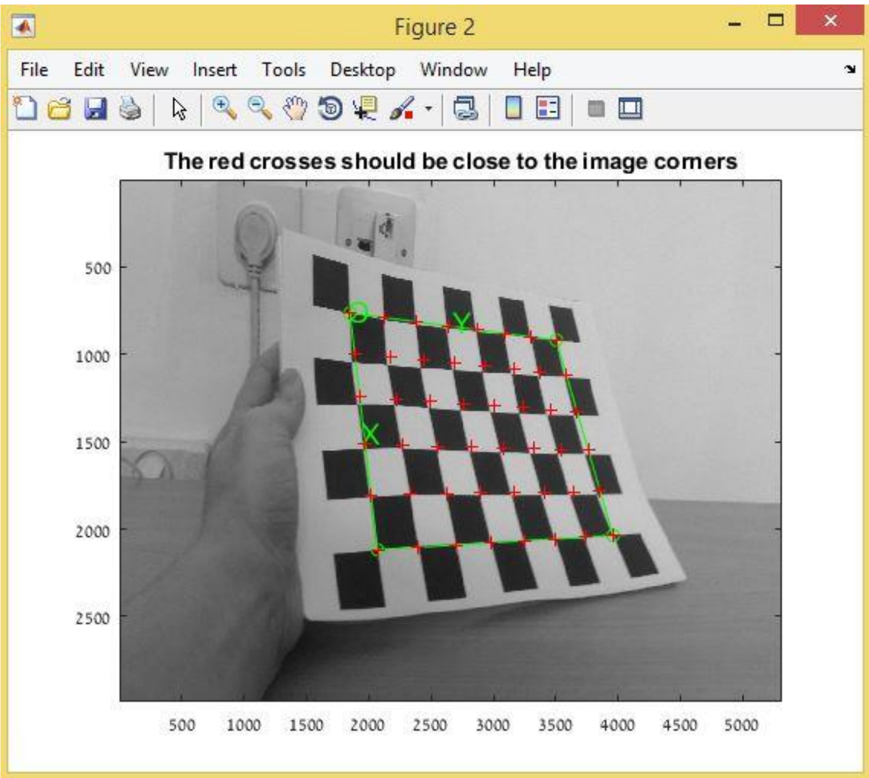
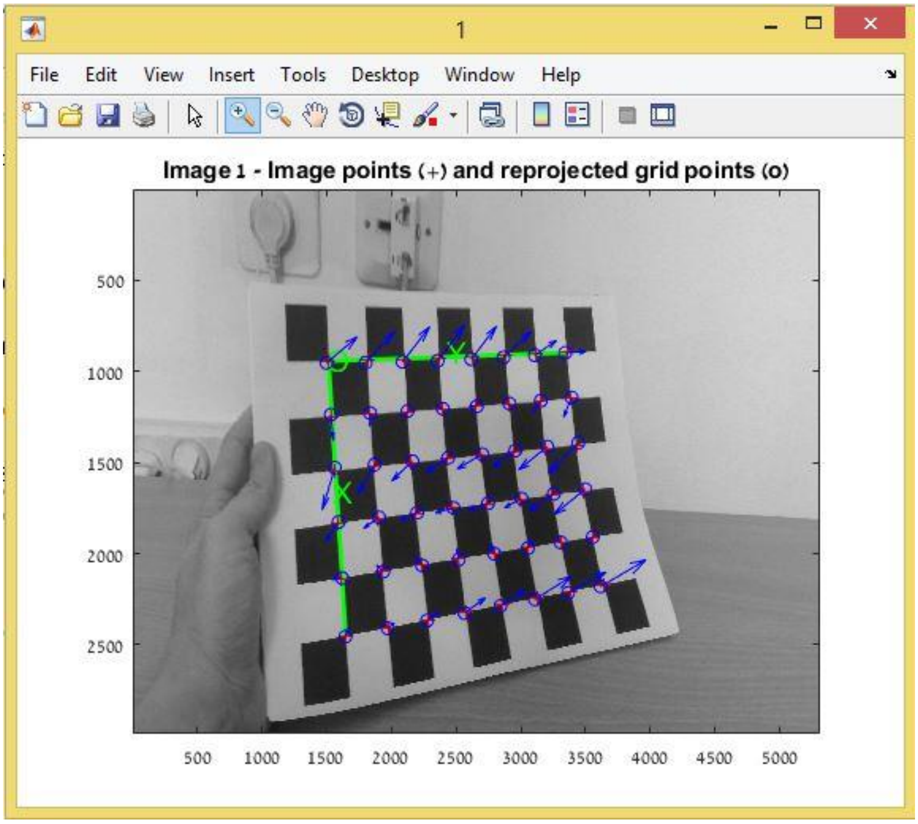
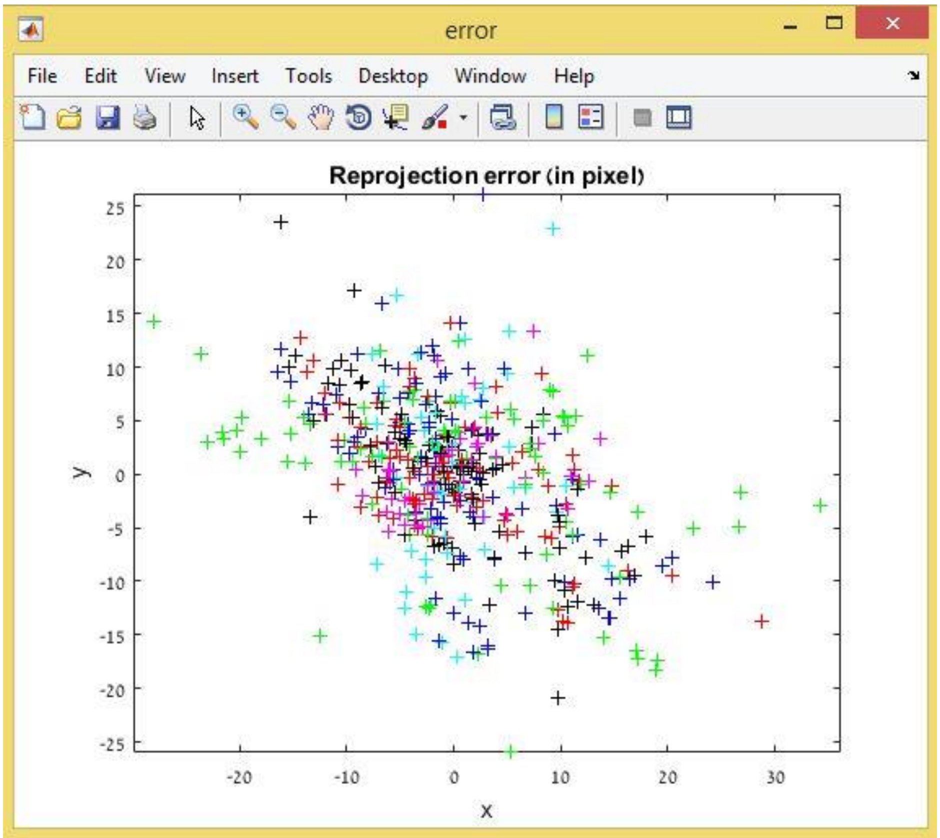


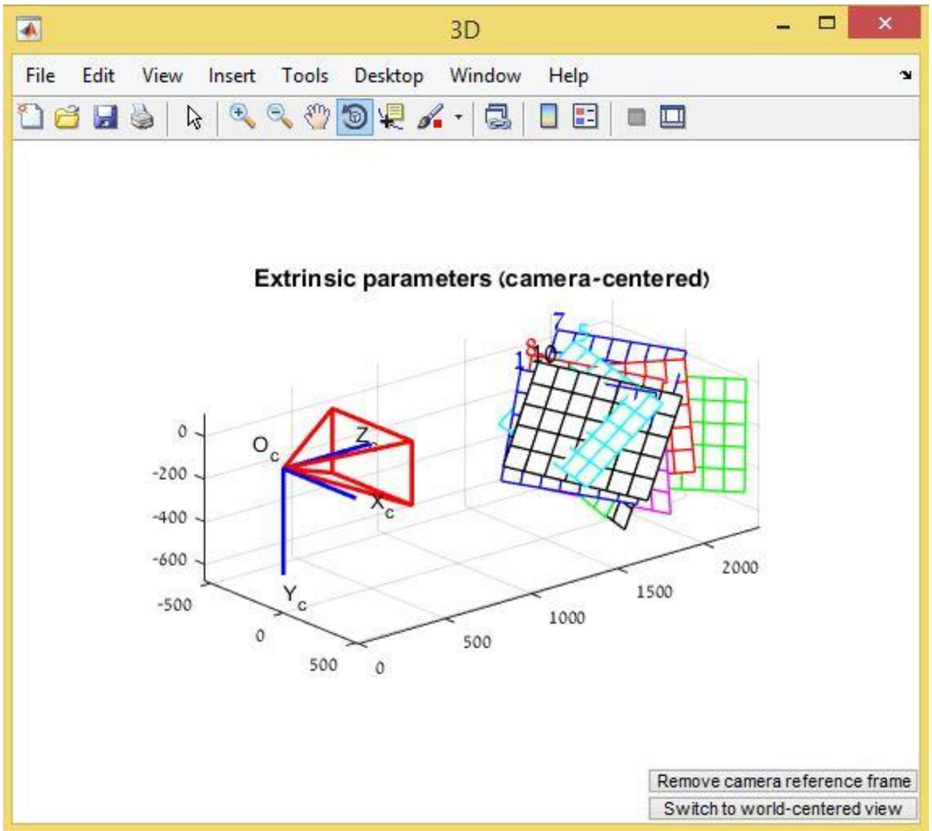
Figure 2
File Edit View Insert Tools Desktop Window Help
The red crosses should be close to the image corners
Y
X

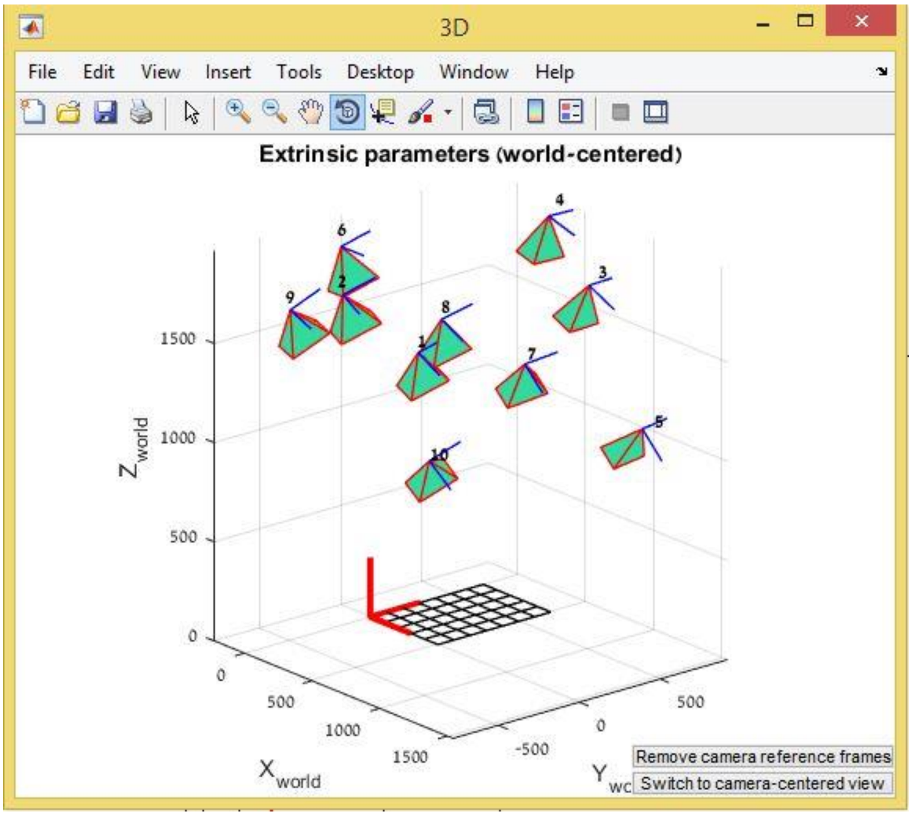
Command Window

```
Aspect ratio optimized (est_aspect_ratio = 1) -> both components of fc are estimated (DEFAULT).
Principal point optimized (center_optim=1) - (DEFAULT). To reject principal point, set center_optim=0
Skew not optimized (est_alpha=0) - (DEFAULT)
Distortion not fully estimated (defined by the variable est_dist):
     Sixth order distortion not estimated (est_dist(5)=0) - (DEFAULT) .
Initialization of the principal point at the center of the image.
Initialization of the intrinsic parameters using the vanishing points of planar patterns.

Initialization of the intrinsic parameters - Number of images: 10


Calibration parameters after initialization:

Focal Length:          fc = [ 4599.58346   4599.58346 ]
Principal point:       cc = [ 2655.50000   1493.50000 ]
Skew:             alpha_c = [ 0.00000 ]   => angle of pixel = 90.00000 degrees
Distortion:            kc = [ 0.00000   0.00000   0.00000   0.00000   0.00000 ]

Main calibration optimization procedure - Number of images: 10
Gradient descent iterations: 1...2...3...4...5...6...7...8...9...10...11...12...13...14...15...16...17...18...19...20...21...22...23...24...25...26...27...28...29...30...done
Estimation of uncertainties...done


Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 4851.53764   4956.86251 ] +/- [ 326.43008   392.43450 ]
Principal point:       cc = [ 2512.87444   922.34861 ] +/- [ 191.16057   294.40907 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ 0.28781   0.05202   -0.08281   -0.01321  0.00000 ] +/- [ 0.17487   0.79740   0.03016   0.01769  0.00000 ]
Pixel error:          err = [ 8.45466   7.17735 ]

Note: The numerical errors are approximately three times the standard deviations (for reference).


Recommendation: Some distortion coefficients are found equal to zero (within their uncertainties).
                To reject them from the optimization set est_dist=[1;0;1;1;0] and run Calibration
```




error
File Edit View Insert Tools Desktop Window Help
Reprojection error (in pixel)
y
x


3D
File Edit View Insert Tools Desktop Window Help
Extrinsic parameters (camera-centered)
Remove camera reference frame
Switch to world-centered view




```
Calibration results (with uncertainties):

Focal Length:          fc = [ 4851.53764   4956.86251 ] ± [ 326.43008   392.43450 ]
Principal point:       cc = [ 2512.87444   922.34861 ] ± [ 191.16057   294.40907 ]
Skew:             alpha_c = [ 0.00000 ] ± [ 0.00000  ]   => angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion:            kc = [ 0.28781   0.05202   -0.08281   -0.01321  0.00000 ] ± [ 0.17487   0.79740   0.03016   0.01769  0.00000 ]
Pixel error:          err = [ 8.45466   7.17735 ]
```

Calibration Matrix:

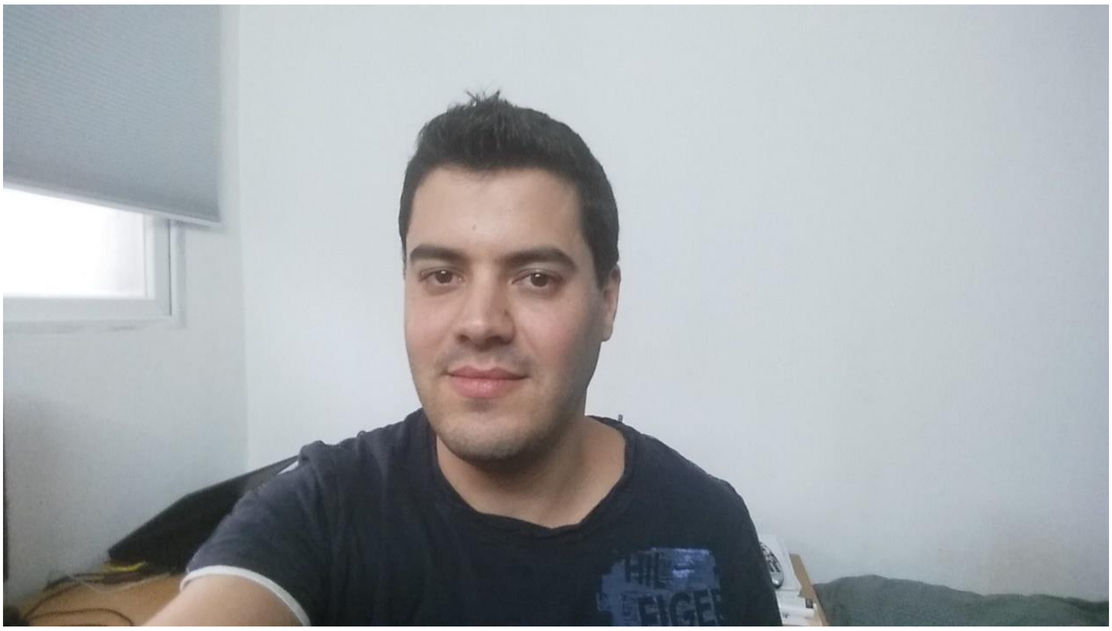
$$\begin{pmatrix} 4851.53 & 0 & 2512.87 \\ 0 & 4956.86 & 922.34 \\ 0 & 0 & 1 \end{pmatrix}$$

Principal point: (2512.87, 922.34)

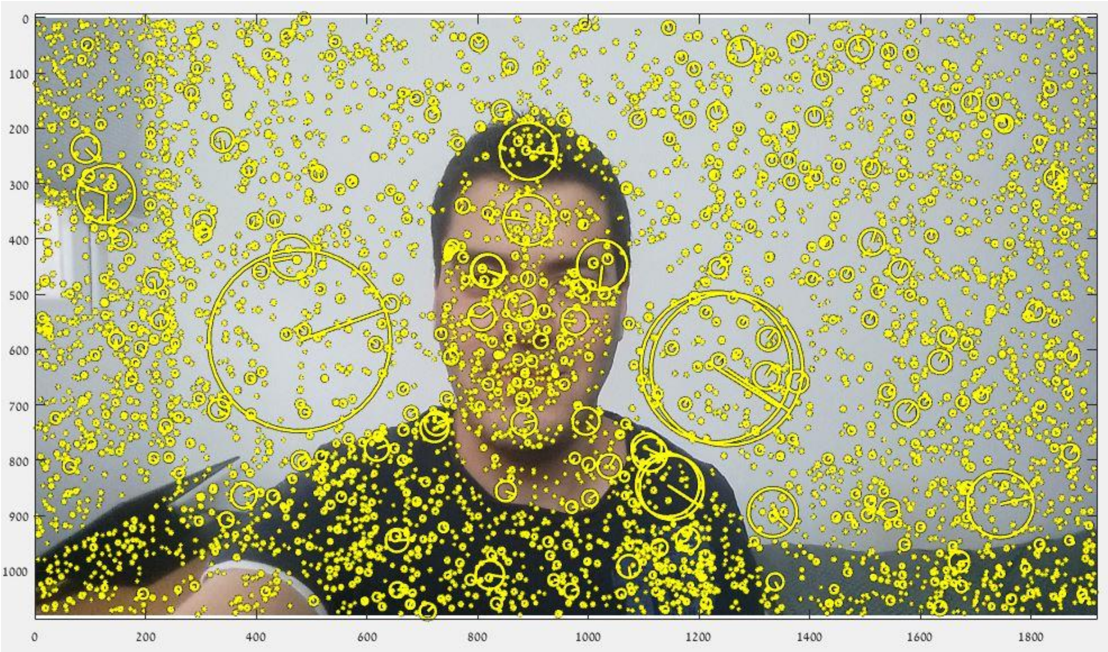
Focal length: (x: 4851.53, y: 4956.86)

3)

(a)

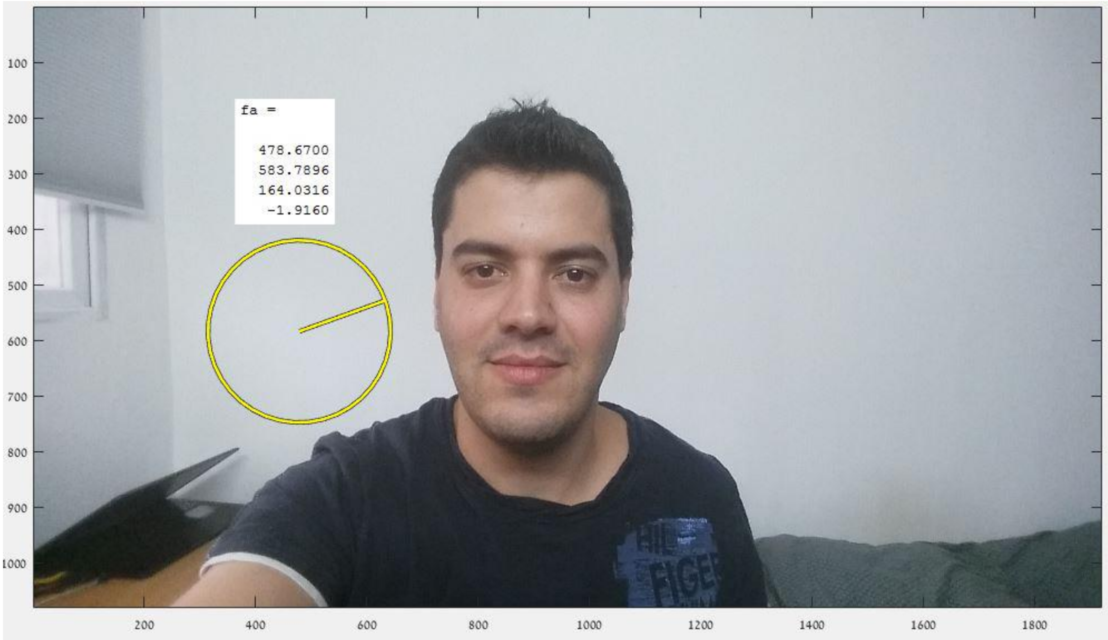


(b)

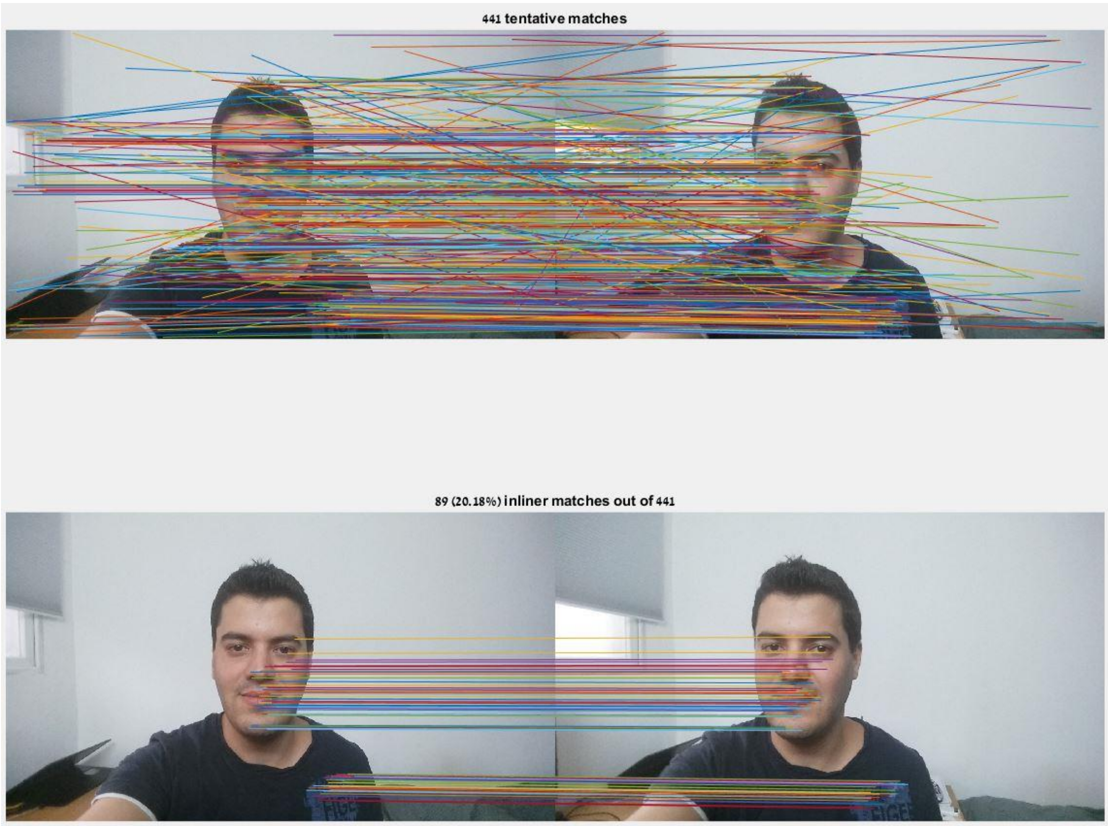


0
100
200
300
400
500
600
700
800
900
1000
0
200
400
600
800
1000
1200
1400
1600
1800

0
100
200
300
400
500
600
700
800
900
1000
200
400
600
800
1000
1200
1400
1600
1800



(c)



Outliner:

5255: 1.0e+03 * (1.3778, 0.6584, 0.0192, -0.0023) –

5260: 1.0e+03 * (1.5481, 0.6929, 0.0198, -0.0023)

Inliner:

5252: 1.0e+03 * (1.1800, 0.9437, 0.0211, 0.0012) –

5268: 1.0e+03 * (1.1887, 0.9723, 0.0241, 0.0010)

Code:

```
function [matches, scores] = sift()
clc;
Ia = imread('1.jpg') ;
Ib = imread('2.jpg') ;

[fa,da] = vl_sift(im2single(rgb2gray(Ia))) ;
[fb,db] = vl_sift(im2single(rgb2gray(Ib))) ;

% image(Ia);
% axis image
% perm = randperm(size(fa,2)) ;
% sel = perm(1:50) ;
% h1 = vl_plotframe(fa(:,:)) ;
% h2 = vl_plotframe(fa(:,:)) ;
% set(h1,'color','k','linewidth',3) ;
% set(h2,'color','y','linewidth',2) ;

% image(Ib);
% axis image
% perm = randperm(size(fb,2)) ;
% sel = perm(1:50) ;
% h1 = vl_plotframe(fb(:,:)) ;
% h2 = vl_plotframe(fb(:,:)) ;
% set(h1,'color','k','linewidth',3) ;
% set(h2,'color','y','linewidth',2) ;

[matches, scores] = vl_ubcmatch(da,db) ;

% ------

numMatches = size(matches,2) ;

X1 = fa(1:2,matches(1,:)) ; X1(3,:) = 1 ;
X2 = fb(1:2,matches(2,:)) ; X2(3,:) = 1 ;
clear H score ok ;
for t = 1:100
  % estimate homograpyh
  subset = vl_colsubset(1:numMatches, 4) ;
  A = [] ;
  for i = subset
    A = cat(1, A, kron(X1(:,i)', vl_hat(X2(:,i)))) ;
  end
  [U,S,V] = svd(A) ;
  H{t} = reshape(V(:,9),3,3) ;

  % score homography
  X2_ = H{t} * X1 ;
  du = X2_(1,:)./X2_(3,:) - X2(1,:)./X2(3,:) ;
  dv = X2_(2,:)./X2_(3,:) - X2(2,:)./X2(3,:) ;
  ok{t} = (du.*du + dv.*dv) < 6*6 ;
  score(t) = sum(ok{t}) ;
end
```

```
[score, best] = max(score) ;
H = H{best} ;
ok = ok{best} ;


function err = residual(H)
 u = H(1) * X1(1,ok) + H(4) * X1(2,ok) + H(7) ;
 v = H(2) * X1(1,ok) + H(5) * X1(2,ok) + H(8) ;
 d = H(3) * X1(1,ok) + H(6) * X1(2,ok) + 1 ;
 du = X2(1,ok) - u ./ d ;
 dv = X2(2,ok) - v ./ d ;
 err = sum(du.*du + dv.*dv) ;
end

if exist('fminsearch') == 2
  H = H / H(3,3) ;
  opts = optimset('Display', 'none', 'TolFun', 1e-8, 'TolX', 1e-8) ;
  H(1:8) = fminsearch(@residual, H(1:8)', opts) ;
else
  warning('Refinement disabled as fminsearch was not found.') ;
end


dh1 = max(size(Ib,1)-size(Ia,1),0) ;
dh2 = max(size(Ia,1)-size(Ib,1),0) ;

figure(1) ; clf ;
subplot(2,1,1) ;
imagesc([padarray(Ia,dh1,'post') padarray(Ib,dh2,'post')]) ;
o = size(Ia,2) ;
line([fa(1,matches(1,:));fb(1,matches(2,:))+o], ...
     [fa(2,matches(1,:));fb(2,matches(2,:))]) ;
title(sprintf('%d tentative matches', numMatches)) ;
axis image off ;

subplot(2,1,2) ;
imagesc([padarray(Ia,dh1,'post') padarray(Ib,dh2,'post')]) ;
o = size(Ia,2) ;
line([fa(1,matches(1,ok));fb(1,matches(2,ok))+o], ...
     [fa(2,matches(1,ok));fb(2,matches(2,ok))]) ;
title(sprintf('%d (%.2f%%) inliner matches out of %d', ...
              sum(ok), ...
              100*sum(ok)/numMatches, ...
              numMatches)) ;
axis image off ;

drawnow ;
end
```