Project 5

EchoSure

Detecting Blood-Clots Post-Operatively In Blood Vessel Anastomoses

Seminar Presentation by Alessandro Asoni

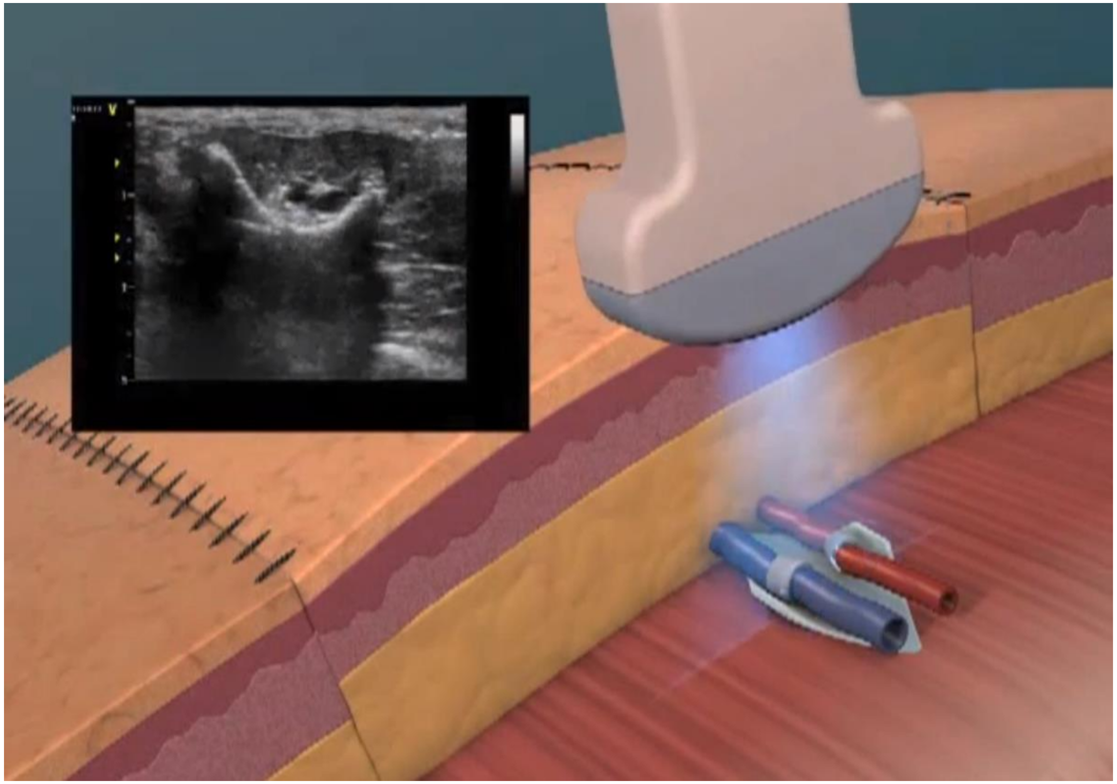
Students: Michael Ketcha
Alessandro Asoni

Mentors: Dr. Jerry Prince
Dr. Emad Boctor
Dr. Nathanael Kuo

Recap of Project

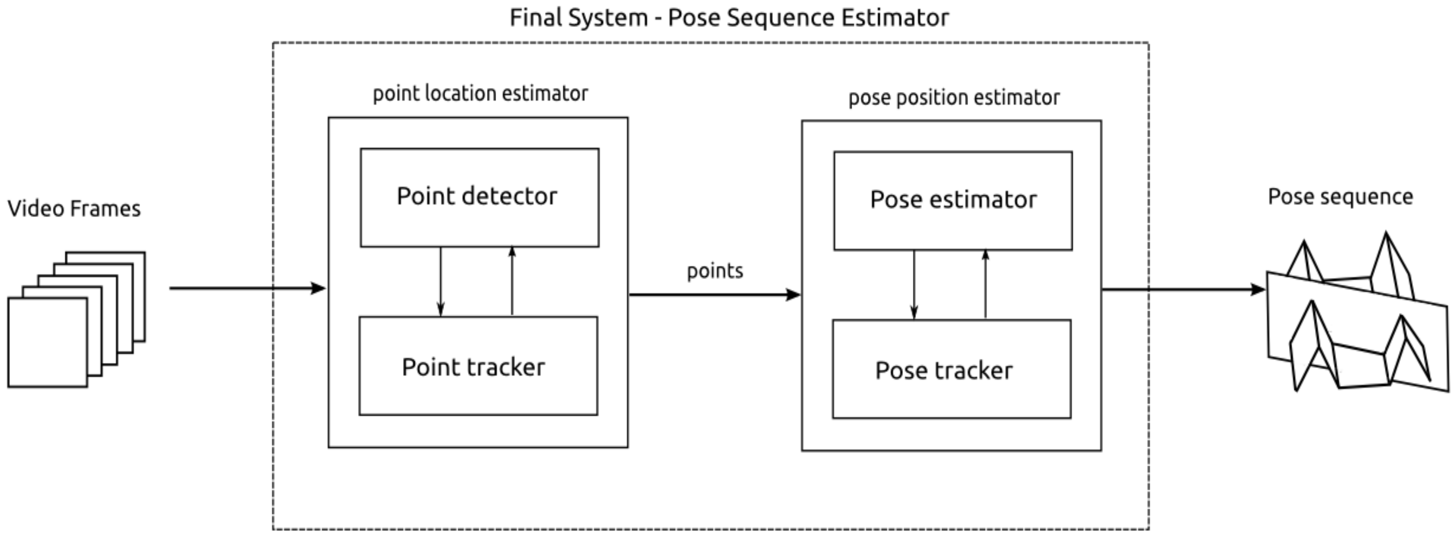
Ultrasound Doppler Imaging for Tracking Changes in Blood Flow Velocity

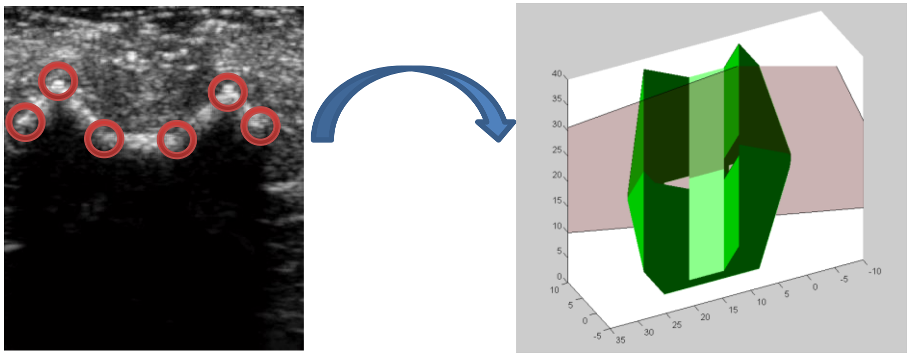
Biodegradable Plastic Fiducial for Supplying Reliable Pose



Animation by David A. Rini

Technical Approach





Today's Paper

Welch, G., and G. Bishop (1995), An introduction to the Kalman Filter. University of North Carolina, Department of Computer Science

The Kalman Filter

Brief Overview

- It was developed in 1960 by R.E. Kalman

- It's a recursive optimal solution to the so called 'Discrete-data linear filtering problem'

- What this means in practice:
  - It's an efficient set of mathematical equations to estimate the **optimal** state of a dynamic system governed by the following two equations:

The Kalman Filter Equations

Propagation equation: $$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$$

Measurement equation: $$z_k = Hx_k + v_k$$

$x_k \in \Re^n$ State vector

$u_k \in \Re^r$ input vector

$z_k \in \Re^m$ Measurement vector

$P(v_k) \sim N(0, R)$ Measurement noise

$P(w_k) \sim N(0, Q)$ Process noise

EchoSure

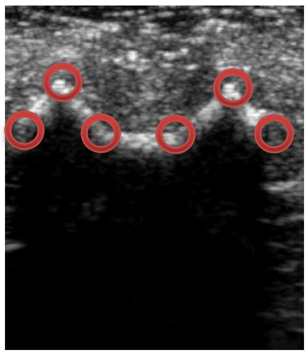
MUSiiC Research Laboratory

ciis

Laboratory for Computational Sensing + Robotics The Johns Hopkins University

The Kalman Filter Equations

Example: How does this apply to our project?



$$x_k = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ \vdots \\ y_{11} \\ x_{12} \\ y_{12} \\ \hline \dot{x}_1 \\ \dot{y}_1 \\ \vdots \\ \dot{y}_{12} \end{bmatrix}$$

positions

velocities

State vector

EchoSure

MUSiiC Research Laboratory

ciis

Laboratory for Computational Sensing + Robotics

The Johns Hopkins University

The Kalman Filter Equations

Measurement Equation:

We are detecting each point in every frame so:

$$z_k = Hx_k + v_k \longrightarrow H = \begin{array}{c} 24 \\ \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 0 & \ddots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & 0 & 1 & \cdots & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & \cdots & 0 \end{bmatrix} \end{array} 12$$

Select first 12 entries in state vector x_k

Zero out all the velocities

The Kalman Filter Equations

Propagation Equation:

NO INPUT

$$x_k = Ax_{k-1} + Bu_{k-1} + w_k \longrightarrow x_k = Ax_{k-1} + w_k$$

What should this equation be for our system?

Now working with constant velocity:

$$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$ (For a two dimensional state vector)

NOTE: We are looking into a better approach
(shown at the end of this presentation - time permitting)

The Kalman Filter – Derivation

Two step process:

- ***A priori*** estimate by propagating from previous state:

$$\hat{x}_k^- = Ax_{k-1} + Bu_{k-1}$$

  - With ***a priori*** error:

$$e_k^- = x_k - \hat{x}_k^-$$

  - And ***a priori*** Covariance:

$$P_k^- = E[(e_k^-)(e_k^-)^\top]$$

The Kalman Filter – Derivation

Two step process:

- ***A posteriori*** estimate as linear blending:

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-)$$

  - With ***a posteriori*** error:

$$e_k = x_k - \hat{x}_k$$

  - And ***a posteriori*** Covariance:

$$P_k = E[\,(e_k)(e_k)^\top]$$

The Kalman Filter – Derivation

A posteriori from ***a priori*** estimate and measurements:

$$\hat{x}_k = \hat{x}_k^- + \underbrace{K}_{\text{Kalman gain}}\underbrace{(z_k - H\hat{x}_k^-)}_{\text{Residual}}$$

NOTE: The rigorous justification for why this equation is used is a bit tricky. If you are interested read Appendix II of my summary on our website

The Kalman Filter – Derivation

Objective: minimize the trace of the ***a posteriori*** error covariance:

$$P_k = E[\,(e_k)(e_k)^\top]$$

$$= E[\,(x_k - \hat{x}_k)(x_k - \hat{x}_k)^\top]$$

$$= E[\,(x_k - \hat{x}_k^- + K(z_k - H\hat{x}_k^-))(x_k - \hat{x}_k^- + K(z_k - H\hat{x}_k^-))^\top]$$

Doing the expectation and taking the derivative of the trace with respect to K gives:

$$K = P_k^- H^\top (HP_k^- H^\top + R)^{-1}$$

$$P_k = (I - KH)P_k^-$$

NOTE: for a more detailed derivation see Appendix I of my summary

The Kalman Filter – Derivation

Behavior in the limit:

$$\lim_{R \to 0} K = H^{-1}$$

R : measurement error covariance

$$x_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-) = H^{-1}z_k$$

We trust the measurement more

$$\lim_{P_k^- \to 0} K = 0$$

P_k^-: state error covariance

$$x_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-) = \hat{x}_k^-$$

We trust the *a priori* estimate more

The Kalman Filter in Action

Time Update ("Predict")

(1) Project the state ahead

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$$

(2) Project the error covariance ahead

$$P_k^- = AP_{k-1}A^T + Q$$

Measurement Update ("Correct")

(1) Compute the Kalman gain

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$$

(2) Update estimate with measurement z_k

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$$

(3) Update the error covariance

$$P_k = (I - K_k H)P_k^-$$

Initial estimates for $\hat{x}_{k-1}$ and P_{k-1}

Source: Welch & Bishop, An Introduction to the Kalman Filter

The Extended Kalman Filter

Propagation equation: $$x_k = f(x_{k-1}, u_{k-1}, w_{k-1})$$

Measurement equation: $$z_k = h(x_k, v_k)$$

Very Similar Conceptually:
Use **Jacobian** to linearize the system and then do the same as for standard Kalman Filtering

A : Jacobian of f with respect to x

W : Jacobian of f with respect to w

H : Jacobian of h with respect to x

V : Jacobian of h with respect to v

The EKF in Action

Time Update ("Predict")

(1) Project the state ahead

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}, 0)$$

(2) Project the error covariance ahead

$$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T$$

Measurement Update ("Correct")

(1) Compute the Kalman gain

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1}$$

(2) Update estimate with measurement z_k

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-, 0))$$

(3) Update the error covariance

$$P_k = (I - K_k H_k) P_k^-$$

Initial estimates for $\hat{x}_{k-1}$ and P_{k-1}

Source: Welch & Bishop, An Introduction to the Kalman Filter

Back To Out Project

Propagation Equation:

$$x_k = Ax_{k-1} + w_k$$

What should this equation be for our system?

Constant velocity is a bad assumption

Constant acceleration might be a bad assumption as well

Principal Component Analysis

IDEA: Generate a big set of acceptable poses - **simulated**:

$$\begin{bmatrix} x_1^1 & x_1^2 & x_1^3 & x_1^4 & & x_1^k \\ y_1^1 & y_1^2 & y_1^3 & y_1^4 & & y_1^k \\ x_2^1 & x_2^2 & x_2^3 & x_2^4 & & x_2^k \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ y_{11}^1 & y_{11}^2 & y_{11}^3 & y_{11}^4 & & y_{11}^k \\ x_{12}^1 & x_{12}^2 & x_{12}^3 & x_{12}^4 & & x_{12}^k \\ y_{12}^1 & y_{12}^2 & y_{12}^3 & y_{12}^4 & & y_{12}^k \end{bmatrix} \xrightarrow{\text{PCA}} \begin{array}{ll} v_1, v_2, v_3 \dots & \text{principal components} \\ \alpha_1, \alpha_2, \alpha_3, \dots & \text{coefficients} \end{array}$$

NOTE: This analysis was done by Nathanael Kuo who provided us with the PCs and the respective coefficients

PCA + State space

At the heart of the Kalman filter is a linear difference equation:

$$x_k = Ax_{k-1}$$

In continuous time this is represented by the differential equation:

$$\frac{d\vec{x}}{dt} = A\vec{x}$$

Which has solutions:

$$x(\mathrm{t}) = \sum_i v_i \, e^{\lambda_i t}$$

Where v_i and λ_i are the eigenvectors and eigenvalues of A

PCA + State space

Imagine building a matrix A such that:

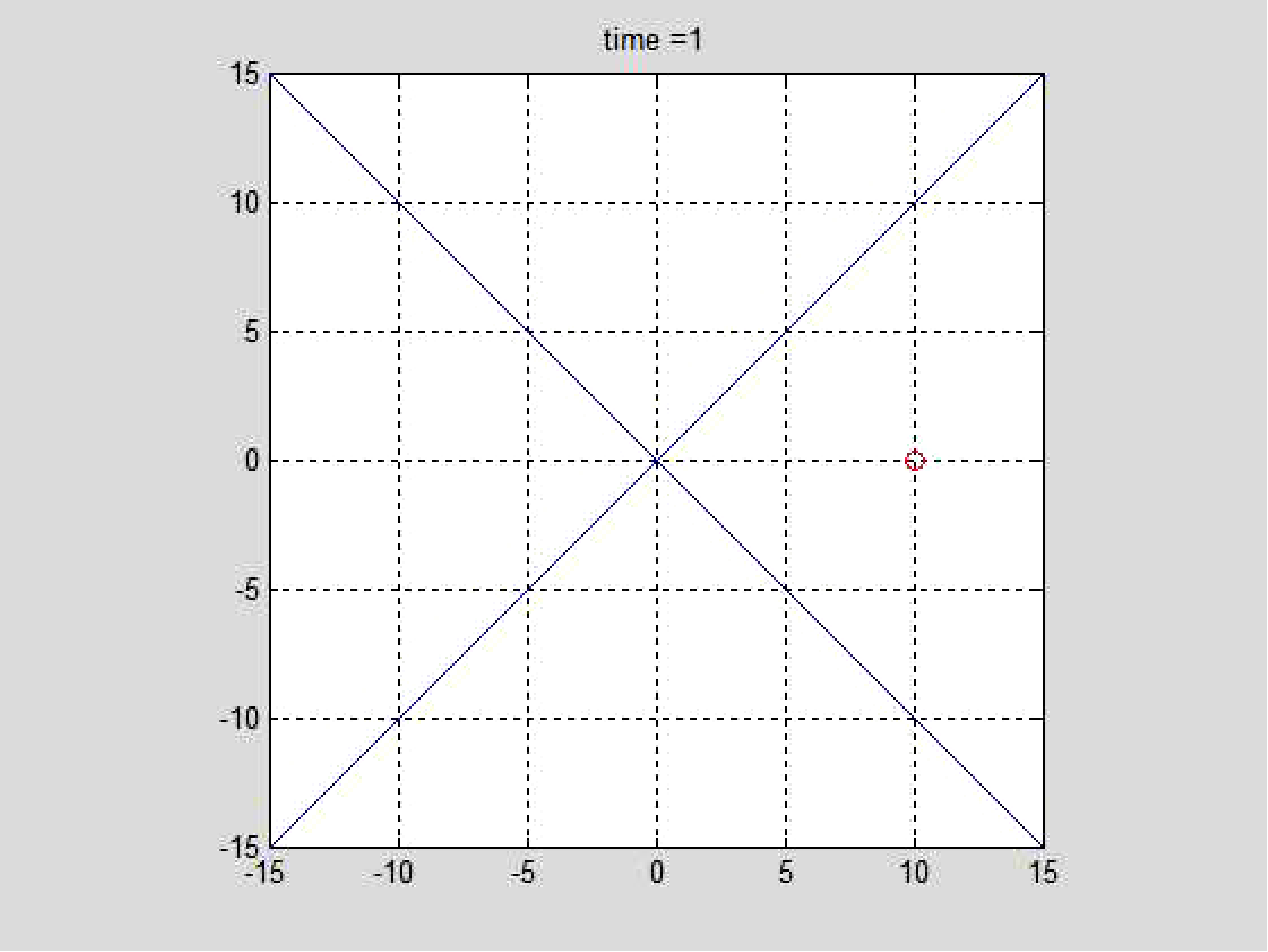
- the ***eigenvectors*** are the first ***principal components*** that we saw earlier
- The ***eigenvalues*** are the ***negative of the inverse of the coefficients***

$$A = UDU^{\top}$$

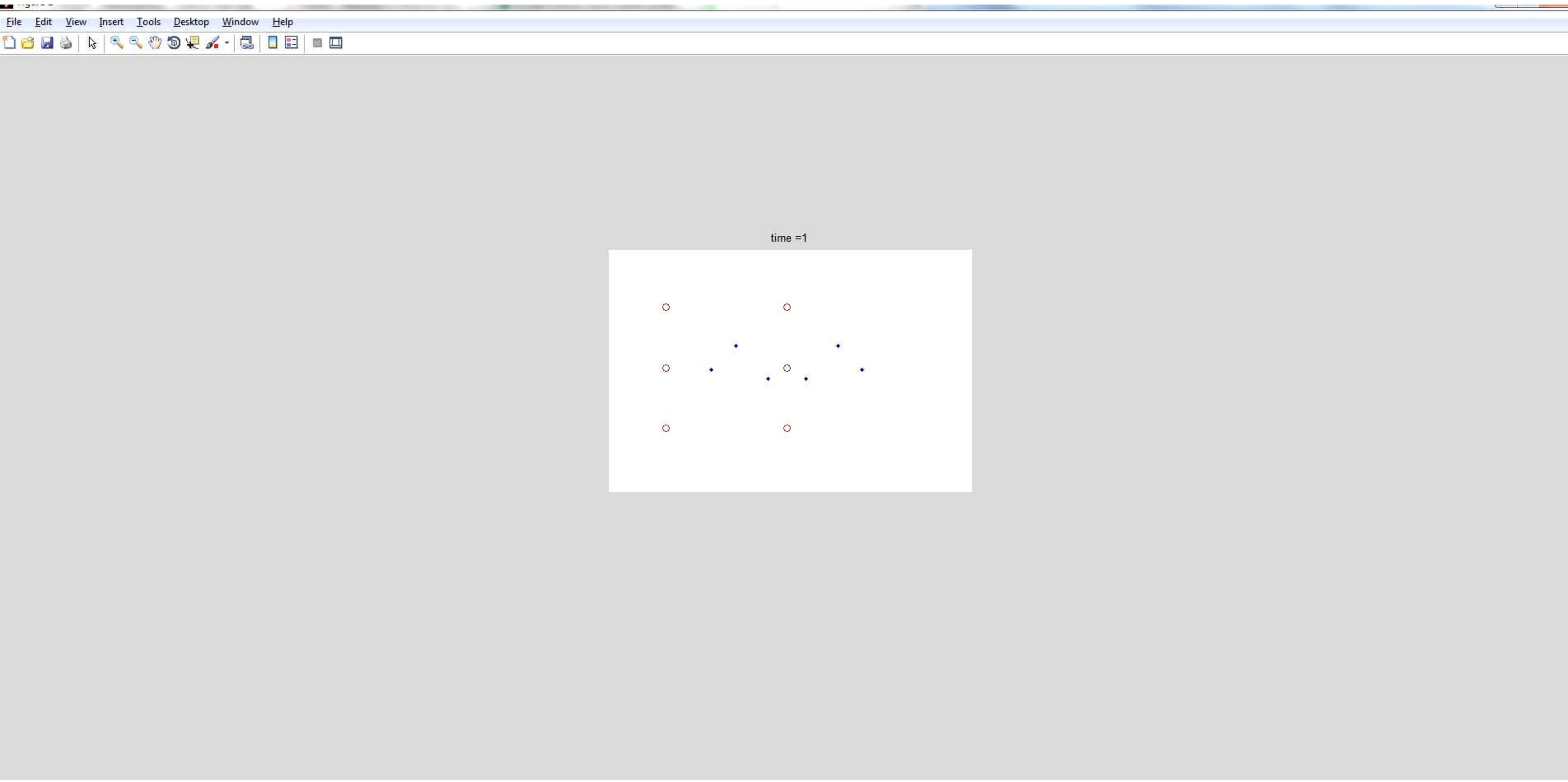
This should give a propagation equation such that:

- in the ***absence of noise*** and ***measurements***
    - The state vector will ***quickly*** converge onto the ***principal components*** with highest coefficients
    - And it will then ***slowly*** move towards the ***mean shape***

NOTE: We have to be careful about the mean. PCA is done after subtracting the mean pose. It has to be added back in.



Video courtesy of Nathanael Kuo



Video courtesy of Nathanael Kuo