Math 695 Project: Motion Analysis

MAN HUY VU

Department of Mathematics and Statistics
California State University, Long Beach
mhvu@hotmail.com

May 7, 2009

Outline

Background of Motion Analysis

Motion Analysis is used to analyze qualitative and quantitative behaviors of objects in motion.

Some areas that uses Motion Analysis are:

1. Human Motion Analysis
2. Manufacturing
3. Military Uses

Background of Motion Analysis

Motion Analysis is used to analyze qualitative and quantitative behaviors of objects in motion.

Some areas that uses Motion Analysis are:

1. Human Motion Analysis
2. Manufacturing
3. Military Uses

Background of Motion Analysis

Motion Analysis is used to analyze qualitative and quantitative behaviors of objects in motion.

Some areas that uses Motion Analysis are:

1. Human Motion Analysis
2. Manufacturing
3. Military Uses

Applications

1. Human Motion Analysis
   - Automatic security monitoring
   - Gait or posture optimization
2. Manufacturing
   - Analyze projectiles to test products
3. Military Uses
   - Long range weapons like missiles

Applications

1. Human Motion Analysis
   - Automatic security monitoring
   - Gait or posture optimization
2. Manufacturing
   - Analyze projectiles to test products
3. Military Uses
   - Long range weapons like missiles

Applications

1. Human Motion Analysis
   - Automatic security monitoring
   - Gait or posture optimization
2. Manufacturing
   - Analyze projectiles to test products
3. Military Uses
   - Long range weapons like missiles

Applications

1. Human Motion Analysis
   - Automatic security monitoring
   - Gait or posture optimization
2. Manufacturing
   - Analyze projectiles to test products
3. Military Uses
   - Long range weapons like missiles

Applications

1. Human Motion Analysis
   - Automatic security monitoring
   - Gait or posture optimization
2. Manufacturing
   - Analyze projectiles to test products
3. Military Uses
   - Long range weapons like missiles

Applications

1. Human Motion Analysis
   - Automatic security monitoring
   - Gait or posture optimization
2. Manufacturing
   - Analyze projectiles to test products
3. Military Uses
   - Long range weapons like missiles

Applications

1. Human Motion Analysis
   - Automatic security monitoring
   - Gait or posture optimization
2. Manufacturing
   - Analyze projectiles to test products
3. Military Uses
   - Long range weapons like missiles

Applications

1. Human Motion Analysis
   - Automatic security monitoring
   - Gait or posture optimization
2. Manufacturing
   - Analyze projectiles to test products
3. Military Uses
   - Long range weapons like missiles

Applications

1. Human Motion Analysis
   - Automatic security monitoring
   - Gait or posture optimization
2. Manufacturing
   - Analyze projectiles to test products
3. Military Uses
   - Long range weapons like missiles

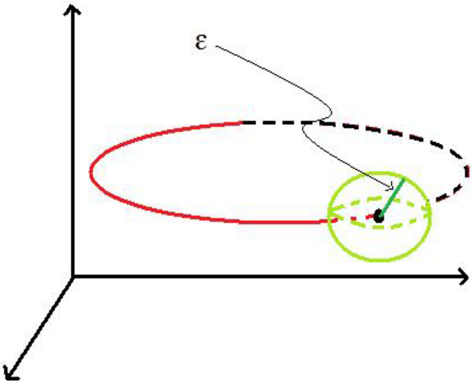
Introduction

Based on "Motion Analysis Using The Novelty Filter" by
E. Ardizzone, A. Chella, S.Gaglio, and F.Sorbello

- Use Novelty Filter and apply to a training set.
- Apply a new frame snapshot to the filter.

Introduction

- For a given threshold ε determine if the position is on the trajectory.
- If it is not on the trajectory then update the new training set.



Problems

Draw Backs: Lose some wanted data in 2D.

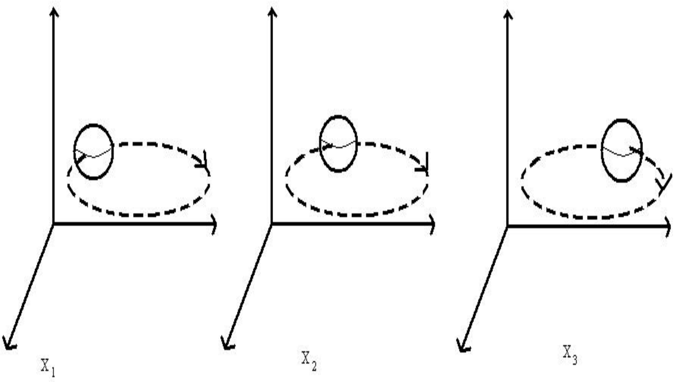
i.e.

1. http://anthonydevito.net/orbit1.swf
2. http://anthonydevito.net/orbit2.swf

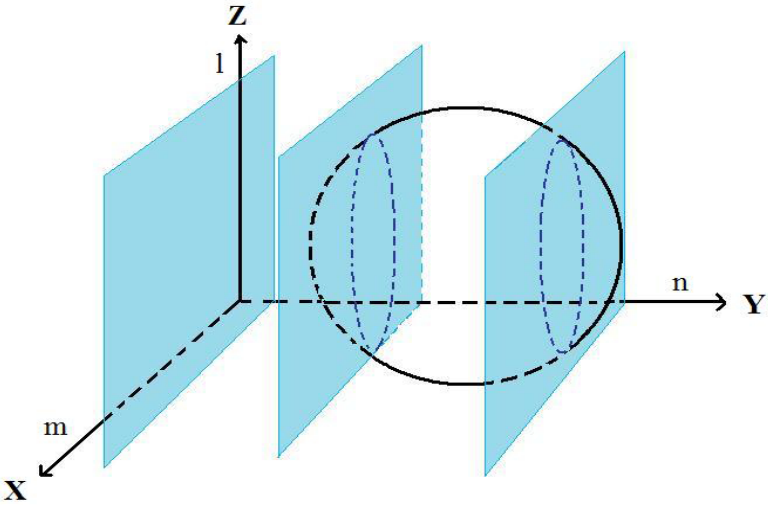
Proposed Modification: Do a 3D motion analysis with the same idea.

Setting Up Training Set

The training set will contain a set of 3D images.

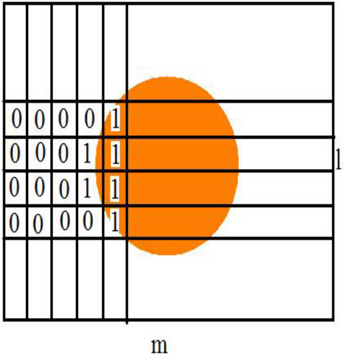


Slice it ALL Up



Each slice along the *y* will be a single 2D image.

A Single Slice



$$y_i \in \Re^{lm}, i = 1, 2, ..., n$$

Concatenating the Concatenated

$$X_j = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

where $X_j \in \Re^{lmn}, j = 1, 2, ..., t$

$$S = \{X_1 \mid X_2 \mid X_3 \mid ... \mid X_t\} \text{ so } S \in \Re^{lmnxt}$$

Kohonen's Novelty Filter

*Based on **Geometric Data Analysis** by Michael Kirby*

Let V be the subspace spanned by the training set.

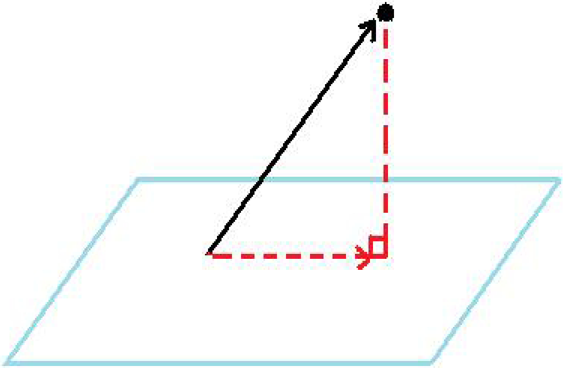
$$V = Span\{X_1, X_2, \ldots, X_t\}$$
$$= Span\{m_1, m_2, \ldots, m_t\}$$

where $\{m_i\}_{i=1}^{t}$ is the orthonormal basis.

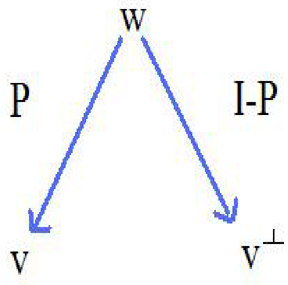
Seeking a Projection

- Let M be the matrix consisting of orthonormal vectors of S.
- We can find P, the **projection matrix**, by multiplying M and M^T.

$$P = M * M^T$$

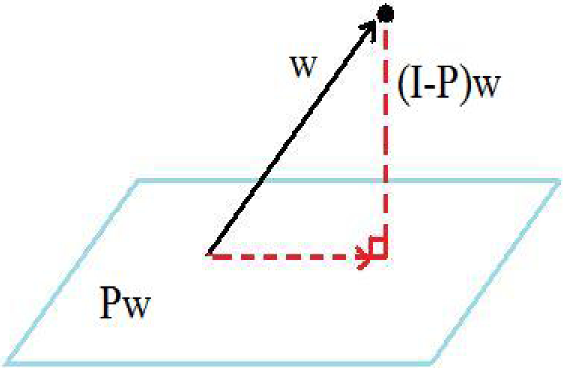


The Novel and the Not-so-Novel: I



- $Pw = n \in \Re^T$
- $(I - P) = n^\perp \in \Re^{lmn-t}$

The Novel and the Not-so-Novel: II



$$w = n + n^{\perp}$$

Implementing

We want to find the **residual(novelty)** of each incoming snapshot $w \in \Re^{lmn}$ by applying the matrix

$$E = I - P$$

$$\Rightarrow Ew = n$$

where $n \in \Re^{lmn}$

Implementing

We want to find the **residual(novelty)** of each incoming snapshot $w \in \Re^{lmn}$ by applying the matrix

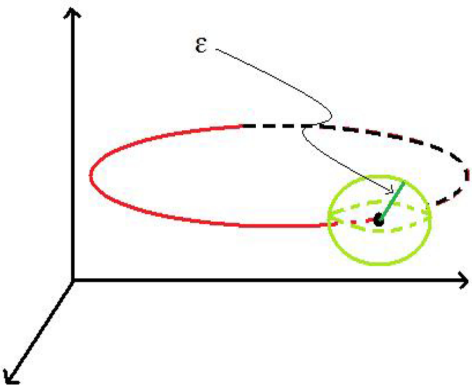
$$E = I - P$$

$$\Rightarrow Ew = n$$

where $n \in \Re^{lmn}$

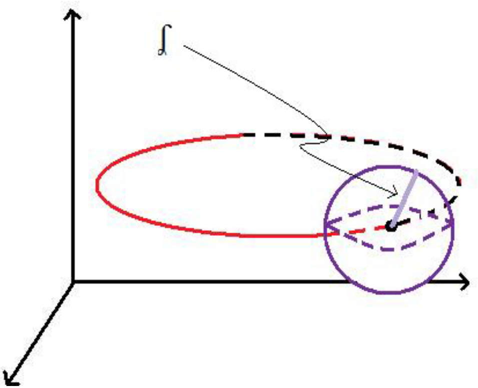
On or Off

- Determine where the new snapshot is already on the trajectory.
- If not, is it less than the set threshold?



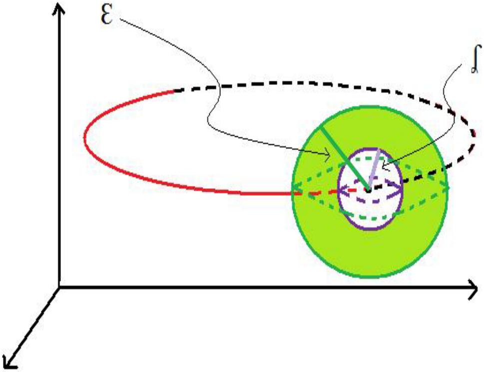
Case 1

If $\|n\| \leq \delta$, where δ is small almost zero, then w is on the trajectory.



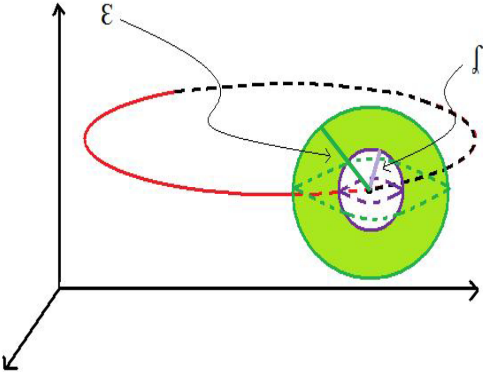
Case 2

If $\delta \leq \|n\| \leq \varepsilon$, where ε is the threshold, then w is not on the current trajectory but w can be taken as a new trajectory.

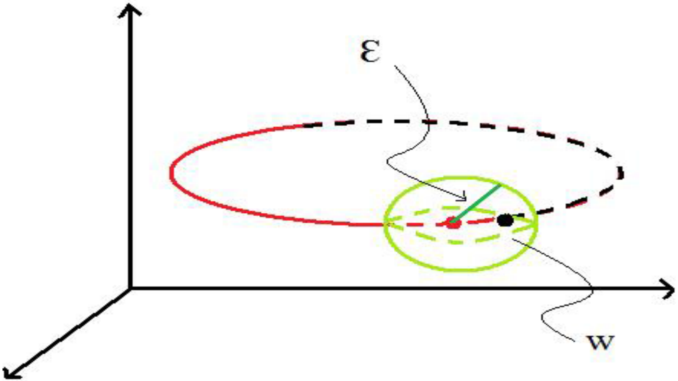


Case 2

For a more accurate trajectory ε must be small.



Case 2

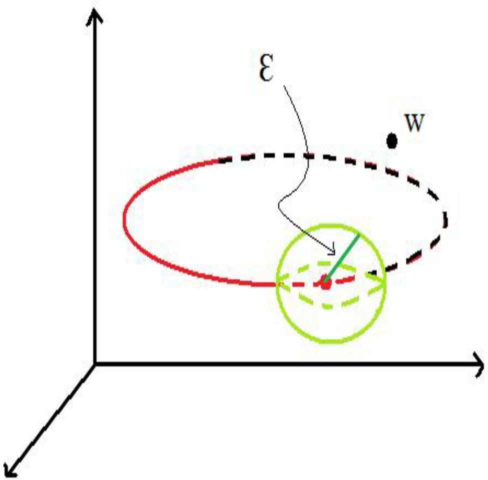


Update the training with this new w.

$$S_{old} = \{X_1 \mid X_2 \mid X_3 \mid \ldots \mid X_t\}$$
$$S_{new} = \{X_1 \mid X_2 \mid X_3 \mid \ldots \mid X_t \mid w\}$$

Case 3

If $\|n\| \gneq \varepsilon$ then disregard w, otherwise the trajectory will be inaccurate.



Pseudocode

1. Initialize all given data: $\delta, \varepsilon, X_1, X_2, ..., X_t, w$.
2. Create $S = \{X_1 \mid X_2 \mid X_3 \mid ... \mid X_t\}$ via horizontal concatenation.
3. Orthonormalize S via Gram-Schmidt or QR (M=orth(S)).
4. Create $P = M * M^T$.

Pseudocode

1. Initialize all given data: $\delta, \varepsilon, X_1, X_2, ..., X_t, w$.
2. Create $S = \{X_1 \mid X_2 \mid X_3 \mid ... \mid X_t\}$ via horizontal concatenation.
3. Orthonormalize S via Gram-Schmidt or QR (M=orth(S)).
4. Create $P = M * M^T$.

Pseudocode

1. Initialize all given data: $\delta, \varepsilon, X_1, X_2, ..., X_t, w$.
2. Create $S = \{X_1 \mid X_2 \mid X_3 \mid ... \mid X_t\}$ via horizontal concatenation.
3. Orthonormalize S via Gram-Schmidt or QR (M=orth(S)).
4. Create $P = M * M^T$.

Pseudocode

1. Initialize all given data: $\delta, \varepsilon, X_1, X_2, ..., X_t, w$.
2. Create $S = \{X_1 \mid X_2 \mid X_3 \mid ... \mid X_t\}$ via horizontal concatenation.
3. Orthonormalize S via Gram-Schmidt or QR (M=orth(S)).
4. Create $P = M * M^T$.

Pseudocode

5. Create $E = I - P$.
6. Find Novelty: $Ew = n$.
7. If $\|n\| \leq \delta$, w is on trajectory. Done.
8. If $\|n\| \gneq \varepsilon$, w is not on trajectory and outside of threshold. Done.

Pseudocode

5. Create $E = I - P$.
6. Find Novelty: $Ew = n$.
7. If $\|n\| \leq \delta$, w is on trajectory. Done.
8. If $\|n\| \gneq \varepsilon$, w is not on trajectory and outside of threshold. Done.

Pseudocode

5. Create $E = I - P$.
6. Find Novelty: $Ew = n$.
7. If $\|n\| \leq \delta$, w is on trajectory. Done.
8. If $\|n\| \gneq \varepsilon$, w is not on trajectory and outside of threshold. Done.

Pseudocode

5. Create $E = I - P$.
6. Find Novelty: $Ew = n$.
7. If $\|n\| \leq \delta$, w is on trajectory. Done.
8. If $\|n\| \gneq \varepsilon$, w is not on trajectory and outside of threshold. Done.

Pseudocode

9. If $\delta \leq \|n\| \leq \varepsilon$

10. Vertical concatenate w with old training set.

$$S=\{X_1 \mid X_2 \mid X_3 \mid \ldots \mid X_t \mid w\}$$

11. Repeat 3-10 with every new snapshot.

Pseudocode

9. If $\delta \leq \|n\| \leq \varepsilon$

10. Vertical concatenate w with old training set.

$$S=\{X_1 \mid X_2 \mid X_3 \mid \ldots \mid X_t \mid w\}$$

11. Repeat 3-10 with every new snapshot.

Pseudocode

9. If $\delta \leq \|n\| \leq \varepsilon$
10. Vertical concatenate w with old training set.

$$S=\{X_1 \mid X_2 \mid X_3 \mid \ldots \mid X_t \mid w\}$$

11. Repeat 3-10 with every new snapshot.

Pros and Cons of 2D Case

Pros:

- Cheap and easy to compute
- Data is easy to obtain

Cons:

- Lose some wanted information

Pros and Cons of 3D Case

Pros:

- Maintains all necessary information

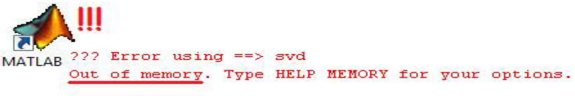
Cons:

- Expensive to compute
- Data is hard to acquire

Fear the "Ginormous" Matrix

The matrix $P = M * M^T \in \Re^{lmnxlmn}$ is extremely large.

For example: take a snapshot to be 100x100x100 where each slice is a 100x100 picture. then $P \in \Re^{1,000,000x1,000,000}$



Do you have it in X-Small?

1. We can reduce the size of P by doing a Pyramidal Decomposition.
   - Decompose every news incoming snapshot to the same level.
   - Proceed to analyze motion via Novelty Filter on the approximation portion.

Do you have that in X-Small?

2. Modify the size of P by using Singular Value Decomposition (SVD) on S.

- Use SVD on $(S^T * S) \in \Re^{txt}$ and orthonormalize the bases.
- Do the analysis using this new P and project back up to original dimension.

Predictions

Can you predict the entire trajectory with the training set?No.

However, we can tell if a guessed position will be on the trajectory or not by calculating

$$\frac{\varepsilon}{\|n\|}$$

Predictions

Can you predict the entire trajectory with the training set?No.

However, we can tell if a guessed position will be on the trajectory or not by calculating

$$\frac{\varepsilon}{\|n\|}$$

Predictions

Can you predict the entire trajectory with the training set?No.

However, we can tell if a guessed position will be on the trajectory or not by calculating

$$\frac{\varepsilon}{\|n\|}$$

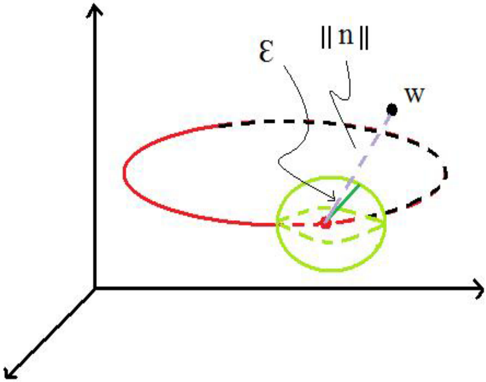
Predictions

Can you predict the entire trajectory with the training set?No.

However, we can tell if a guessed position will be on the trajectory or not by calculating

$$\frac{\varepsilon}{||n||}$$

Predictions



If $\frac{\varepsilon}{||n||}$ is small then there is a low probability that the guessed position will be on the trajectory and vice versa.

Conclusion

- Given a training set of positions of an object
- Use novelty filter to find the difference betweent he training set and the new snapshot
- Analyze whether the new snapshot can be added onto the trajectory
- Update the new trajectory

Conclusion

- Given a training set of positions of an object
- Use novelty filter to find the difference betweent he training set and the new snapshot
- Analyze whether the new snapshot can be added onto the trajectory
- Update the new trajectory

Conclusion

- Given a training set of positions of an object
- Use novelty filter to find the difference betweent he training set and the new snapshot
- Analyze whether the new snapshot can be added onto the trajectory
- Update the new trajectory

Conclusion

- Given a training set of positions of an object
- Use novelty filter to find the difference betweent he training set and the new snapshot
- Analyze whether the new snapshot can be added onto the trajectory
- Update the new trajectory