Efficient Algorithms for Matching

Matching is Fundamental

Matching is Fundamental

- [illegible]
  - *[illegible]*
    matching is the hard part, once this is solved everything else is just equation shuffling…
  - *Object Recognition;*
    having established optimal correspondence between features on the image and within a model one can determine the appropriateness of the model.

Tutorial Overview

- Section 1: Generative models for matching in object recognition and structure from motion
- Section 2: Algorithms for Matching.
- Section 3: ICP.

Section 1
Generative Matching Overview

Section 1
Generative Matching Overview

- 1.1 Explain Generative model of matching
  - Useful for structure and motion recovery
  - And object recognition
- 1.2 Probabilistic interpretation, likelihood of a model depends on the matching.
- 1.3 Marginalizing over the matching: either (a) for object recognition or (b) for learning the shape and appearance.
- 1.4 Strong priors on shape.

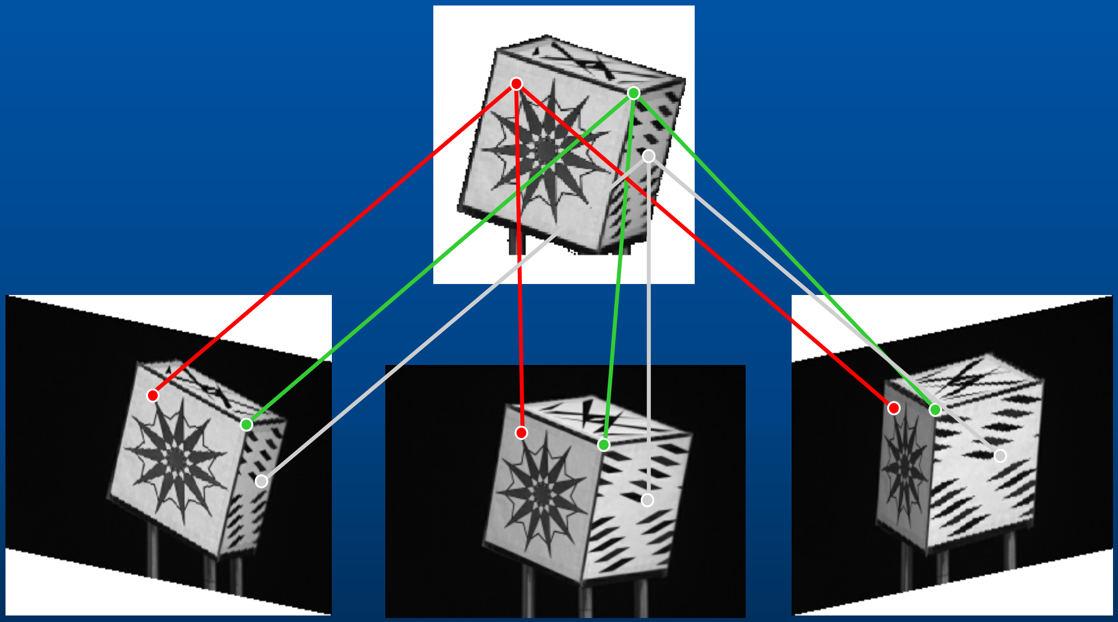
Section 1.1
Generative Matching Introduction

Generative Model of Matching

One way to consider matching is the use of generative models:

- Features generated from some model
- Bayesian analysis easy: Analysis by Synthesis, [inspired by Grenander 1970], why is this good: can learn appearance and shape!.

Generative model



- Patches on the model generate patches in the image; together with some score for goodness of match.

Feature Generation

- Flow is to generate features in an image and detect objects based on this.

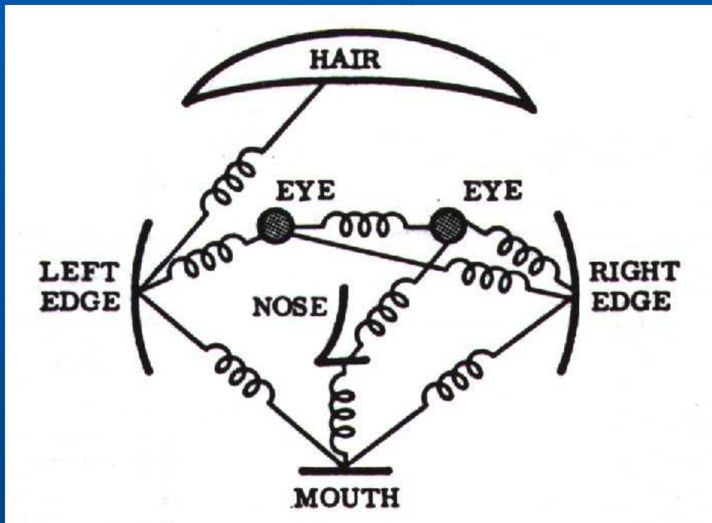
- Features need to be
  - Discriminative.
  - Reproducible (appear on same part of the object in different scenes).
  - Rich, i.e. the more the better, don’t throw away information

Types of Features

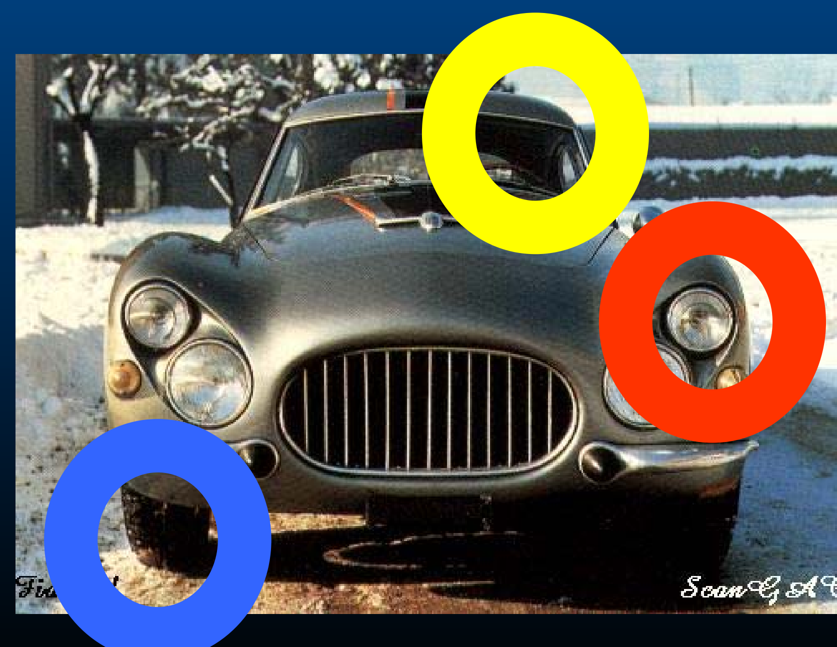
- Typical Features include:
  - Harris corners
  - Canny edges
  - SIFT operator (Lowe)
  - Entropy operator (Kadir and Brady).
  - Maximally Stable Extremal Regions.
  - Learnt Templates, specific to object (e.g nose, eyes)
  - Etc.

- Learning which features useful is an interesting topic of research.

Fergus et al



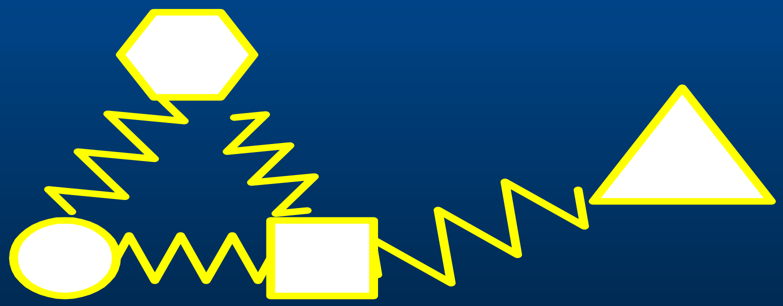
Fischler & Elschlager, 1973



- Yuille, □91
- Brunelli & Poggio, □93
- Lades, v.d. Malsburg et al. □93
- Cootes, Lanitis, Taylor et al. □95
- Amit & Geman, □95, □99
- Perona et al. □95, □96, □98, □00

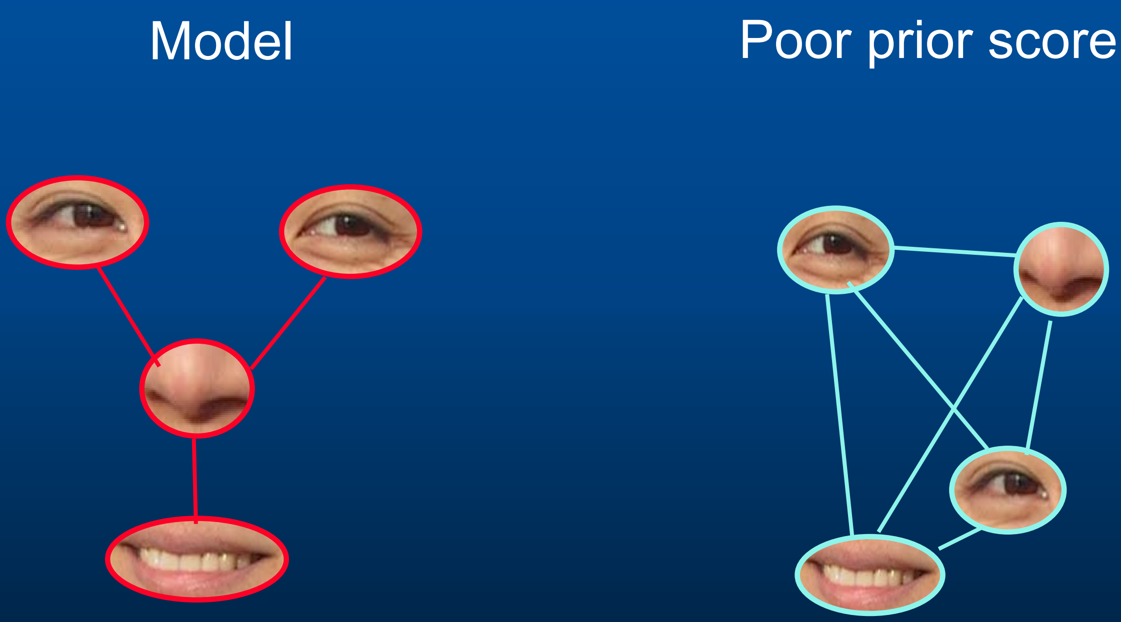
Generative Model for Object Recognition

Once features extracted, what is relation between them?



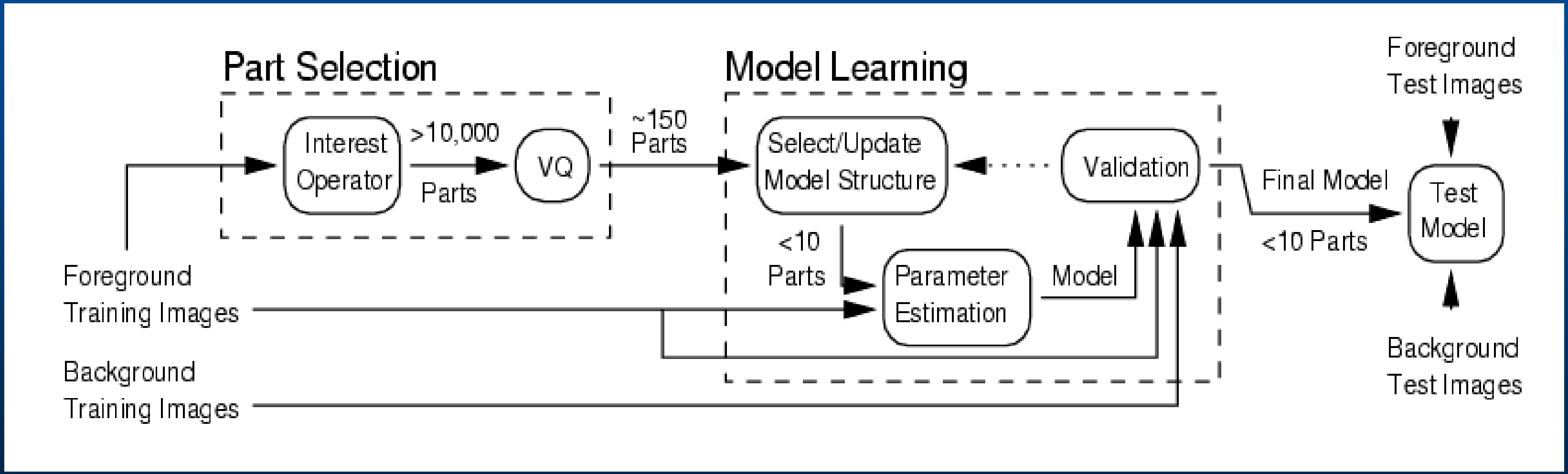
Choose: 2D relation, or rigid 3D relation?

Examples of Relations for faces



- Weber suggests learning features…

Block Diagram of Weber Method to learn features.



- Extract features, apply VQ.

Possible Features

Section 1.2
Generative Model, Probabilistic Interpretation

Probabilistic Formulation

- Weber ECCV 2000

$$p(X^o, \mathbf{x}^m, \mathbf{h}).$$

- Means: joint probability of
  - X^O the responses of our feature detectors
  - x^m the position of the object parts
  - h an indicator as to whether a feature is foreground or background.

Probabilistic Formulation

- Fergus et al 2003

$$p(\mathbf{X},\mathbf{S},\mathbf{A}|\theta) = \sum_{\mathbf{h}\in H} p(\mathbf{X},\mathbf{S},\mathbf{A},\mathbf{h}|\theta) =$$

$$\sum_{\mathbf{h}\in H} \underbrace{p(\mathbf{A}|\mathbf{X},\mathbf{S},\mathbf{h},\theta)}_{Appearance} \underbrace{p(\mathbf{X}|\mathbf{S},\mathbf{h},\theta)}_{Shape} \underbrace{p(\mathbf{S}|\mathbf{h},\theta)}_{Rel.\ Scale} \underbrace{p(\mathbf{h}|\theta)}_{Other}$$

  - X Locations of features detected in the image.
  - S Scale of features
  - A appearance of features
  - ❑ parameters of model

Is an object in the Scene?

- Fergus et al: calculate ratio R: if R>1 then yes:

$$
\begin{aligned}
R &= \frac{p(\text{Object}|\mathbf{X},\mathbf{S},\mathbf{A})}{p(\text{No object}|\mathbf{X},\mathbf{S},\mathbf{A})} \\
&= \frac{p(\mathbf{X},\mathbf{S},\mathbf{A}|\text{Object})\; p(\text{Object})}{p(\mathbf{X},\mathbf{S},\mathbf{A}|\text{No object})\; p(\text{No object})} \\
&\approx \frac{p(\mathbf{X},\mathbf{S},\mathbf{A}|\,\theta)\; p(\text{Object})}{p(\mathbf{X},\mathbf{S},\mathbf{A}|\theta_{bg})\; p(\text{No object})}
\end{aligned}
$$

- **Should really marginalize over ❑📪**

Section 1.3
Generative Model, Marginalizing out matches…

To Marginalize or Maximize the matching?

- So given a generative model AND a matching we can say how likely our image is under our model for it.
- By evaluating this for a set of models we can determine which model is best.
- However we could also marginalize out the matches.

Why marginalize

- If all we are interested in is whether an object is present then we do not really care about what matches what so we marginalize out the matching (tricky, more later).
- No direct analogue with SFM

Learning Shape without matching

- If we want to learn the appearance and shape of the model then we could also marginalize out the matches.
- Interestingly this can be done both for object recognition and for SFM as explored by Dellaert, and also Davidson.

Roadmap

- Next we describe how matches might be marginalized out.
- The following features the work of Dellaert et al to do this, first ignoring uniqueness of matches and then second using MCMC to include matching uniqueness.
- The conclusion is that it doesn’t work too well for structure estimation so matching is not irrelevant!

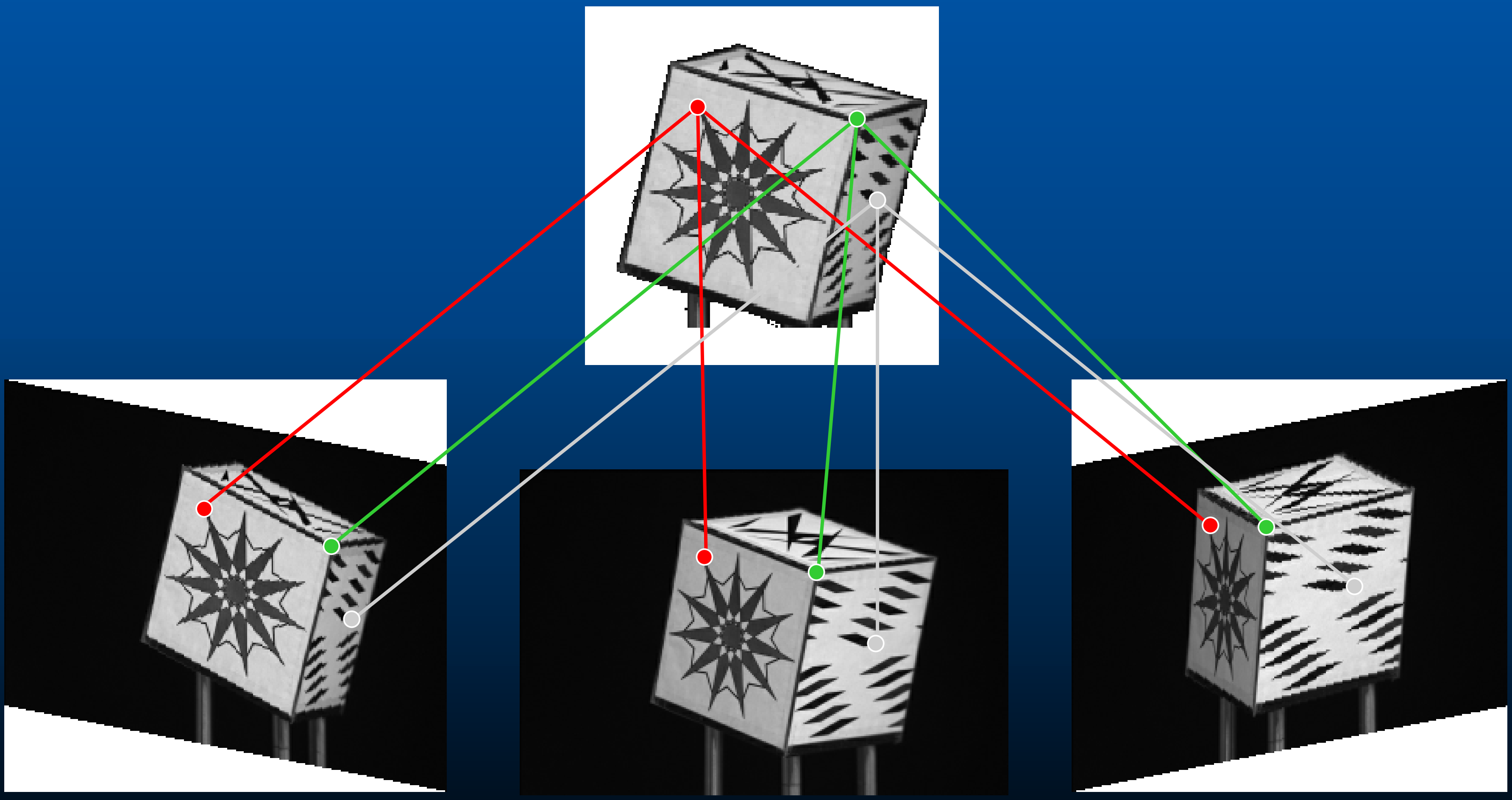
Structure from Motion

Traditionally: 2 Problems !

Correspondence

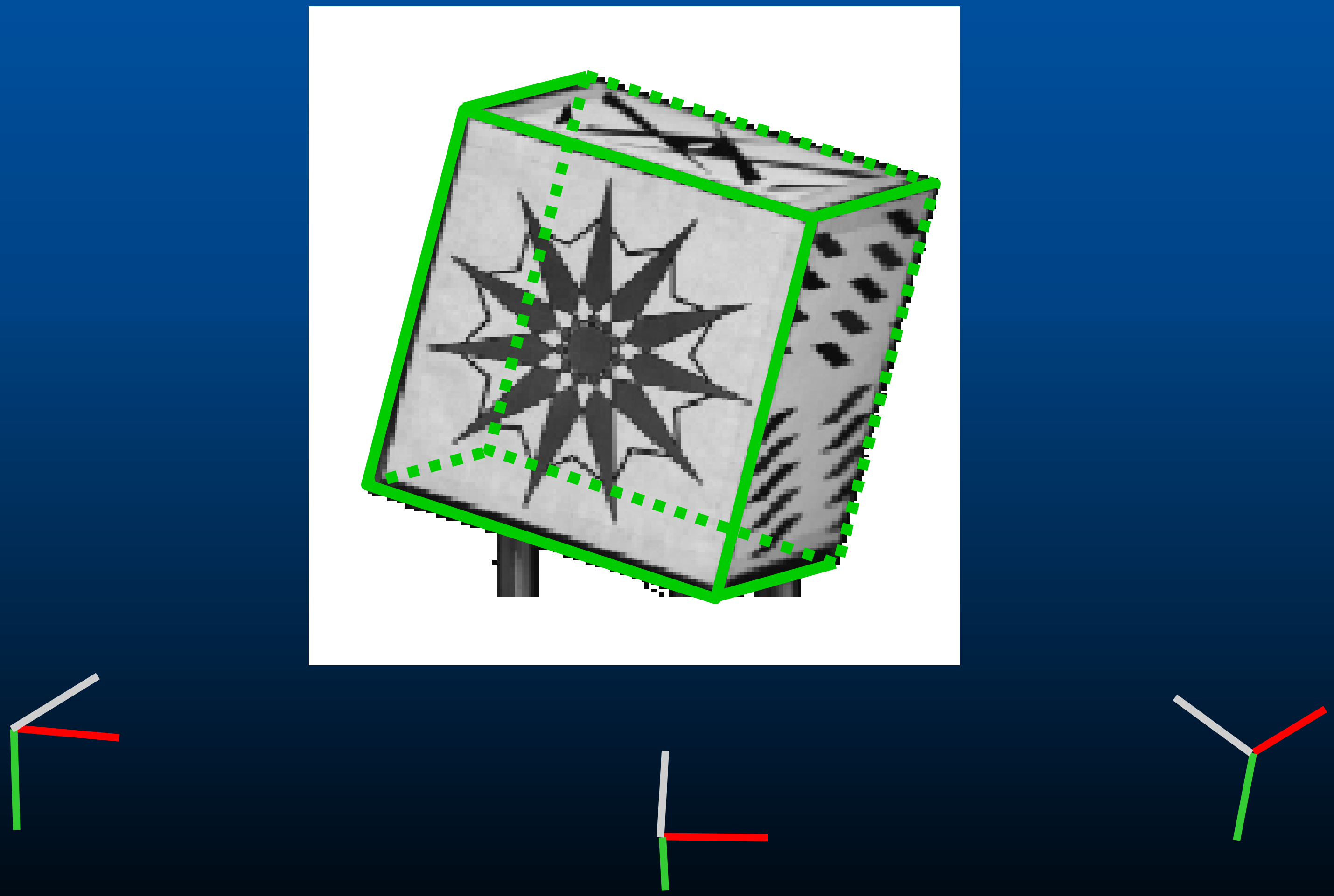
Optimization

A Correspondence Problem



An Optimization Problem

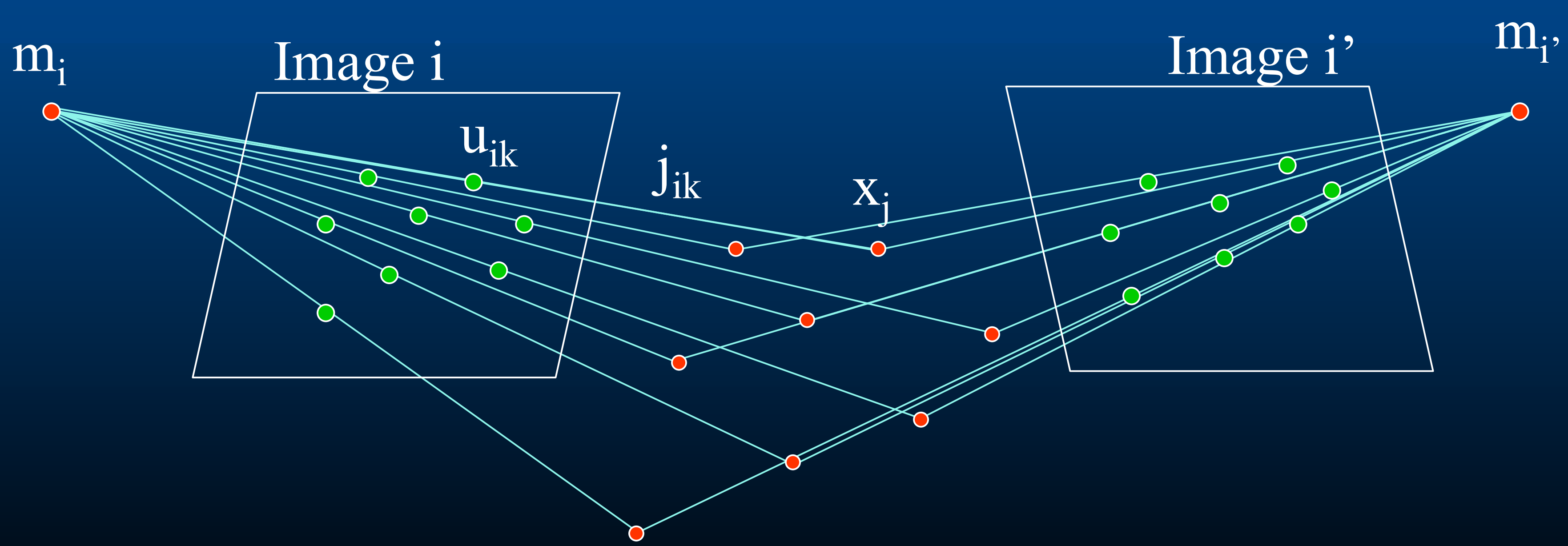
- Find the most likely structure and motion Θ



Optimization

=Non-linear Least-Squares !

$$\sum_{i=1}^{m} \sum_{k=1}^{K_i} \|\mathbf{u}_{ik} - \mathbf{h}(\mathbf{m}_i, \mathbf{x}_{\mathbf{j}_{ik}})\|^2$$



Big Question !

How can we recover structure and motion with **unknown correspondence** ?

Combinatorial Explosion

- In general, #J is combinatorial in m,n

- 3 images, 4 features: $4!^3$=13,824
- 5 images, 30 features: $30!^5$=1.3131e+162
- (number of stars:1e+20, atoms: 1e+79)

- Total Likelihood = intractable !

EM for marginalizing

1. **E-step:** Calculate the expected log likelihood $Q^t(\Theta)$:

$$Q^t(\Theta) = \sum_{\mathbf{J}} f^t(\mathbf{J}) \log L(\Theta; \mathbf{U}, \mathbf{J}) \qquad (6)$$

2. **M-step:** Find the ML estimate Θ^{t+1} for structure and motion, by maximizing $Q^t(\Theta)$:

$$\Theta^{t+1} = \underset{\Theta}{\operatorname{argmax}} \; Q^t(\Theta)$$

Clever Observation!

1. **E-step:** Calculate the weights f_{ijk}^t from the distribution over assignments. Then, in each of the m images calculate n virtual measurements $\mathbf{v}_{ij}^t$.

2. **M-step:** Find the structure and motion estimate Θ^{t+1} that minimizes the (weighted) re-projection error given the virtual measurements:

$$\Theta^{t+1} = \underset{\Theta}{\operatorname{argmin}} \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{1}{2(\sigma_{ij}^t)^2} ||\mathbf{v}_{ij}^t - \mathbf{h}(\mathbf{m}_i, \mathbf{x}_j)||^2$$

Clever Observation

- In other words we can compute a set of virtual measurements (virtual projections of the model into the image) and minimizing the distance to these is the same as minimizing the marginalized, over matches log likelihood.

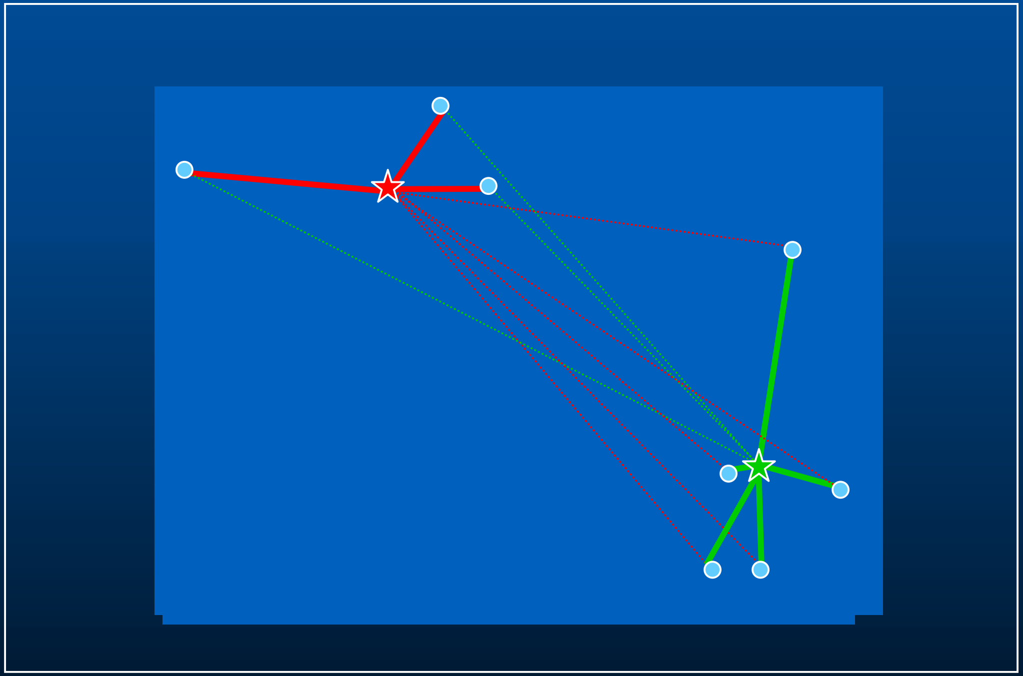
- The virtual measurement are simply the weighted sum of the features.

$$\mathbf{v}_{ij}^t \triangleq \frac{\sum_{k=1}^{K_i} f_{ijk}^t \mathbf{u}_{ik}}{\sum_{k=1}^{K_i} f_{ijk}^t}, \quad (\sigma_{ij}^t)^2 \triangleq \frac{\sigma^2}{\sum_{k=1}^{K_i} f_{ijk}^t}$$

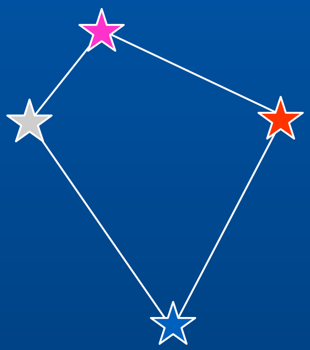
Pseudo Code

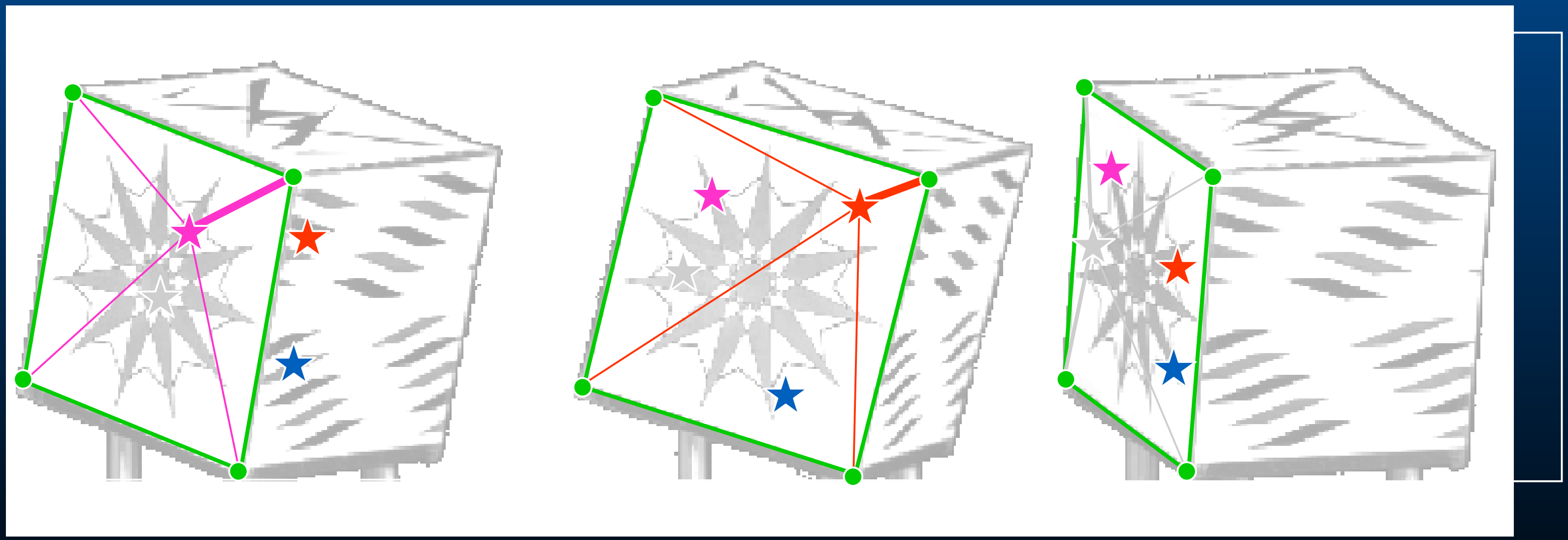
1. Generate an initial structure and motion estimate Θ^0.
2. Given Θ^t and the data $\mathbf{U}$, run the Metropolis sampler in each image to obtain approximate values for the weights f_{ijk}^t, using equation (15).
3. Calculate the virtual measurements $\mathbf{v}_{ij}^t$ with (11).
4. Find the new estimate Θ^{t+1} for structure and motion using the virtual measurements $\mathbf{v}_{ij}^t$ as data. This can be done using any SFM method compatible with the projection model assumed.
5. If not converged, return to step 2.

Expectation Maximization



E-Step: Soft Correspondences

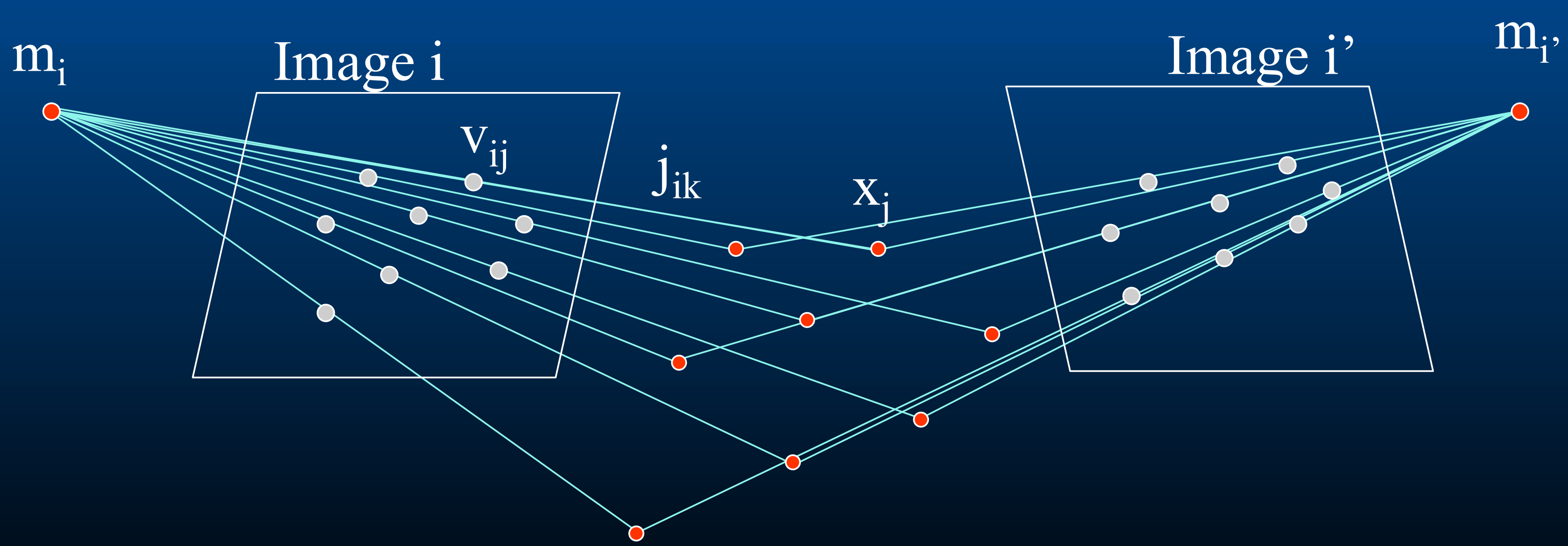




M-Step: Optimization

using virtual measurements !

$$\sum_{i=1}^{m} \sum_{j=1}^{n} \| \mathbf{v}_{ij} - \mathbf{h}(\mathbf{m}_i, \mathbf{x}_j) \|^2$$



Structure from Motion without Correspondence via EM:

1. Generate an initial structure and motion estimate Θ^0.

2. In each image, calculate the n^2 "soft correspondences" f_{ijk}

3. Calculate the virtual measurements $\mathbf{v}_{ij}^t$.

4. Find the new estimate Θ^{t+1} for structure and motion using the virtual measurements $\mathbf{v}_{ij}^t$.

5. If not converged, return to step 2.

Incorporating Appearance

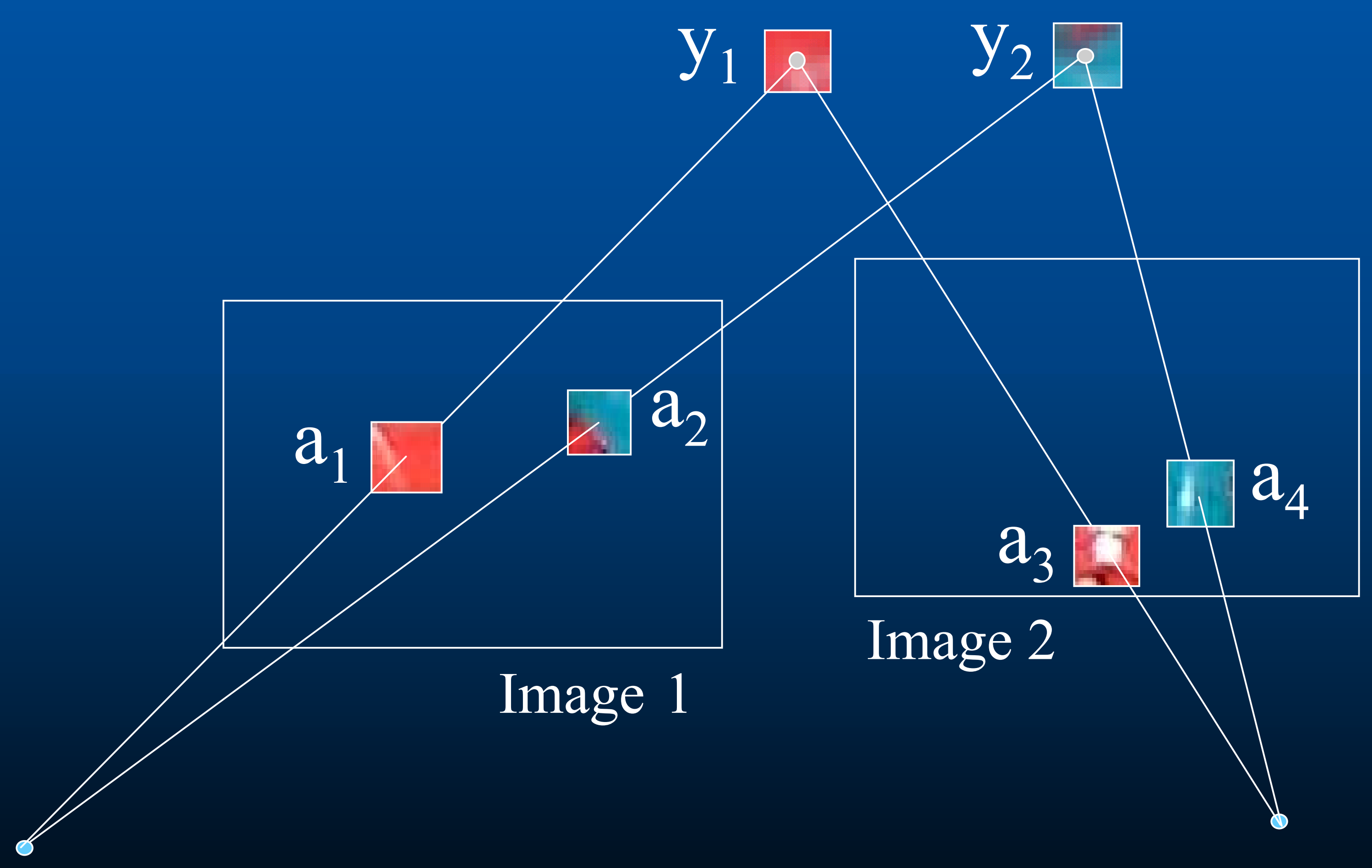
Appearance Models



- Templates
- Color Histograms
- Color Invariants
- Symbolic

“Toy” Example

Appearance Measurements

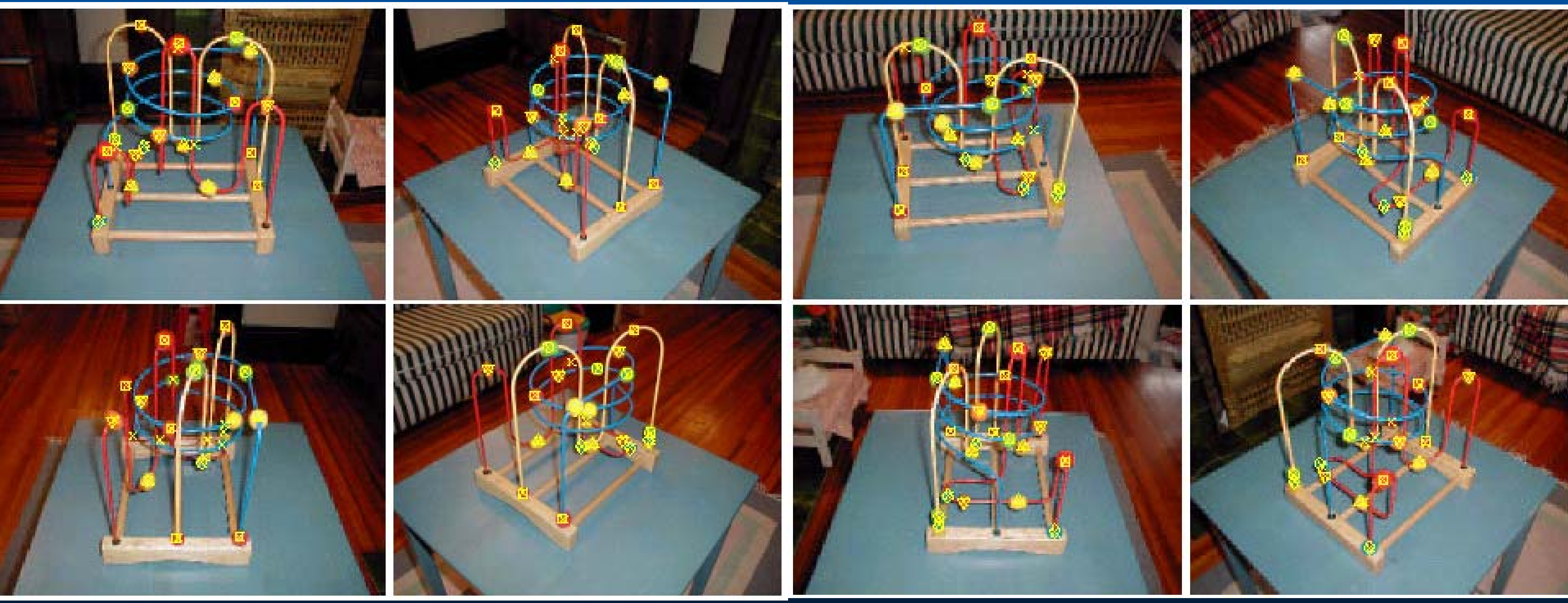


EM with Appearance

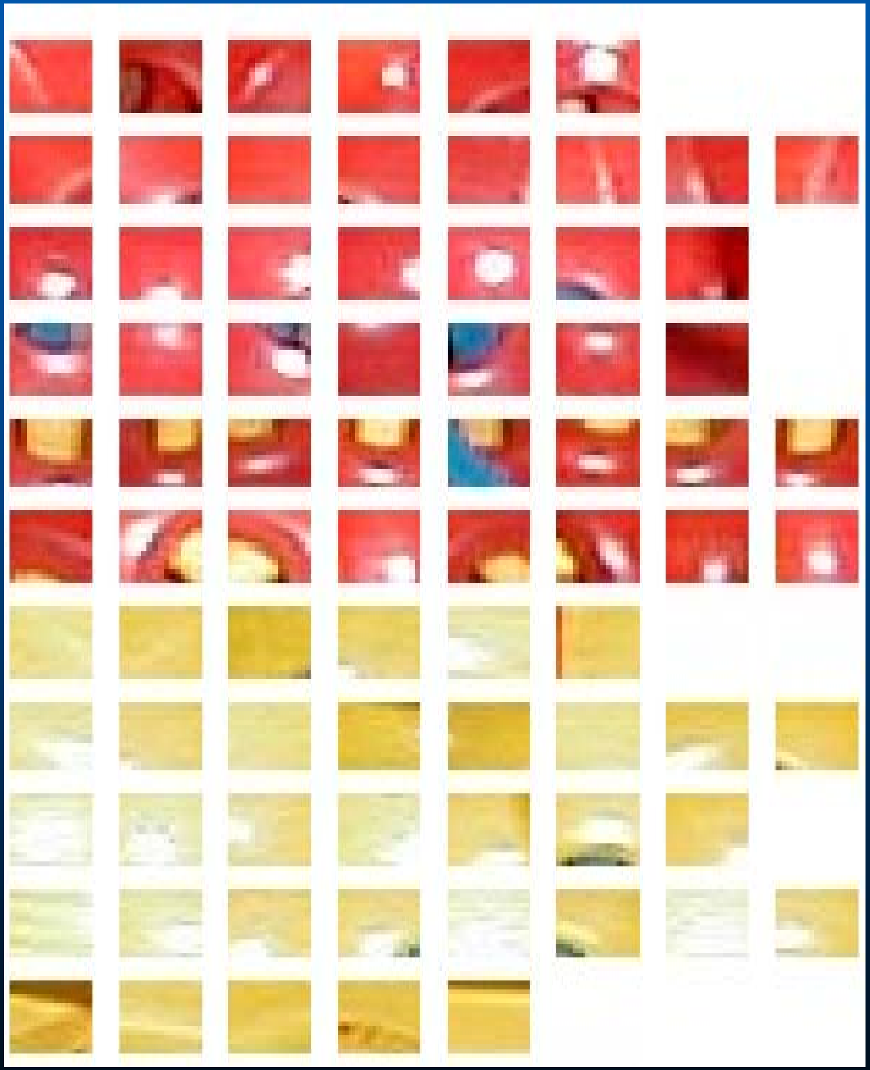
- M-step:
  re-estimate appearance (templates)

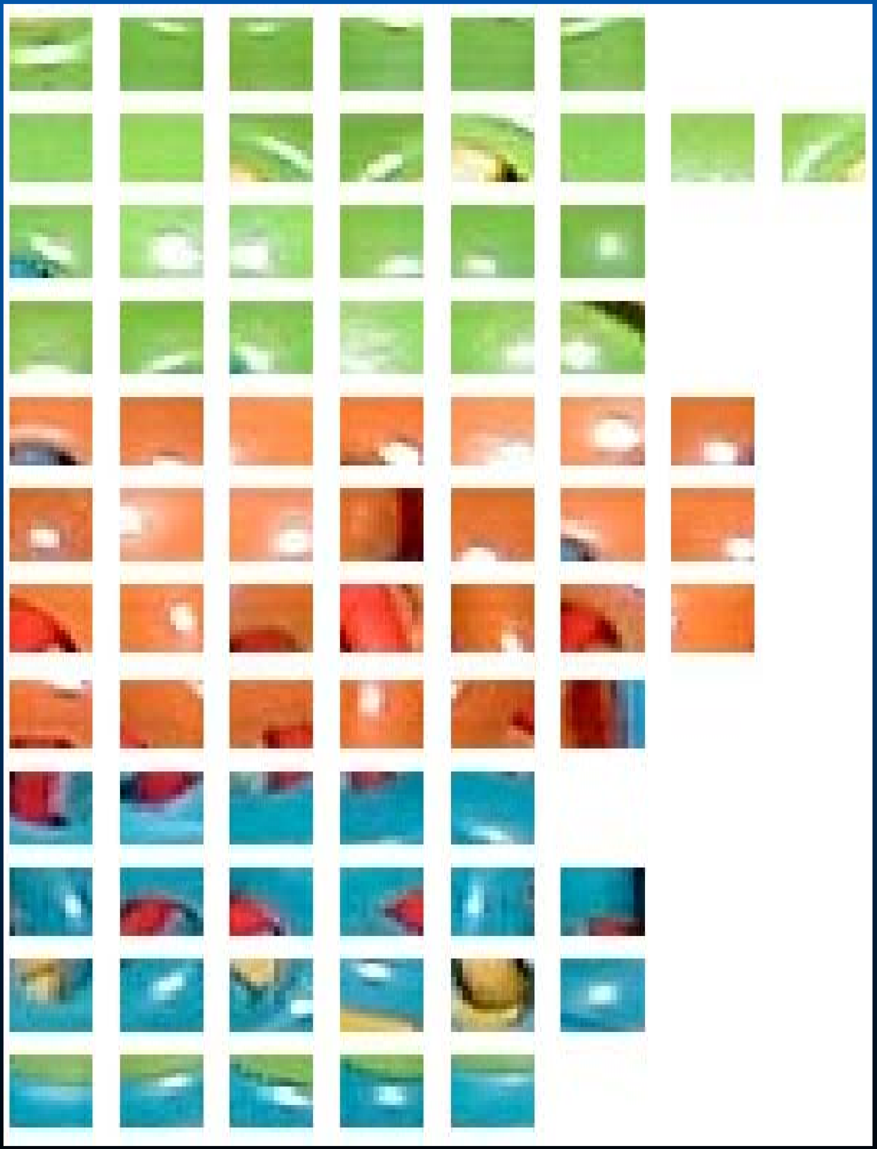
- E-step:
  use appearance to constrain matches

“wiretoy” Image Sequence

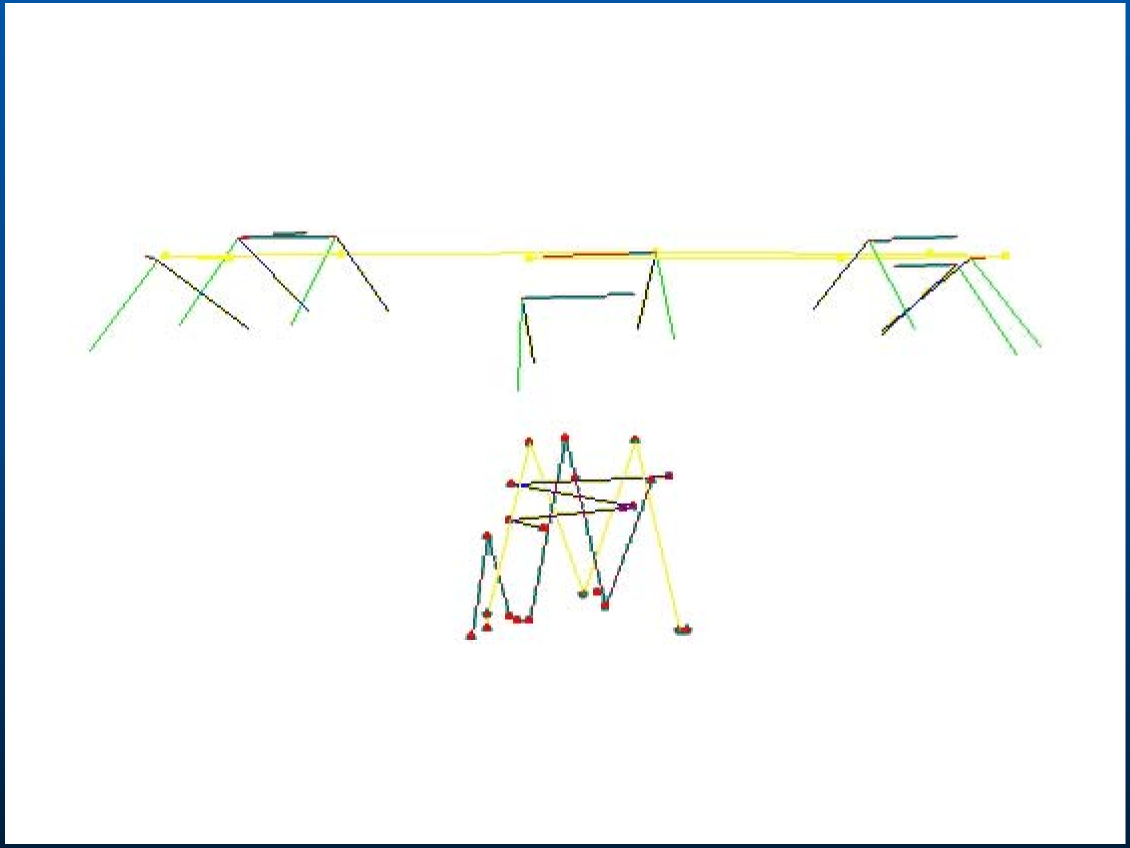


Appearance Measurements





Recovered 3D Structure



Critique

- The Dellaert et al algorithm seems to produce poorer matches than standard techniques; why?
- One argument is that matching IS structure so do we want to marginalize over matching?
- Anyway, better results seem to be achieved by maximization so far…which justifies the next section about algorithms for matching!! (just as well).

Section 1.4
Strong Priors on Shape:

Combing OR and SAM.
(ORSAM)

A quick thought: Stronger Prior shape models

- If strong prior models are used object recognition and structure from motion meet.
- If we recognize that the images arise from a certain class of objects might we want not use that information to refine our estimates of shape?

Aim Structure From 2-6 views

- Problem; SFM often under constrained i.e. homogeneous regions, occlusions
  - Generic Smoothness prior often used (Szeliski 2002)-traditional dense stereo reached limit of performance.
- Solution; Combine recognition and SFM to go much further in resolving ambiguity.
  - Recognition allows for more functional models e.g. opening doors, transparent windows.

Example: Parameterizing buildings

- The form of a prior for a building is far from obvious
  - Generative/explicit distribution hard to formulate.
- Previous work (Dick et al) constructed parameterized models of building parts e.g. doors etc
  - Problem how to combine these sub parts?

- Define an unnormalized prior via a cost function
- We can explore/test validity of this prior by Reversible Jump Metropolis Hastings, MCMC.

Example of Primitives

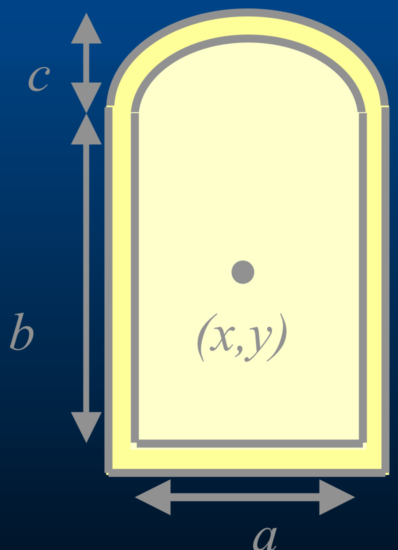
- Reconstruction and recognition of architecture

Shape representation

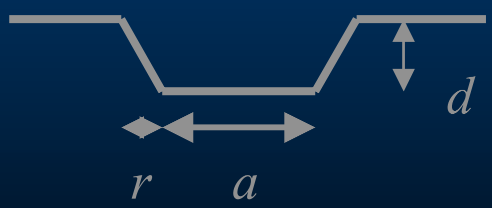
- Model is a collection of “wall” planes
- Each wall plane may contain primitives defined by 4 – 8 parameters

E.g.:
- Window
- Door
- Pediment
- Pedestal
- Entablature
- Column
- Buttress
- Drainpipe

Example shape (window)



Front view

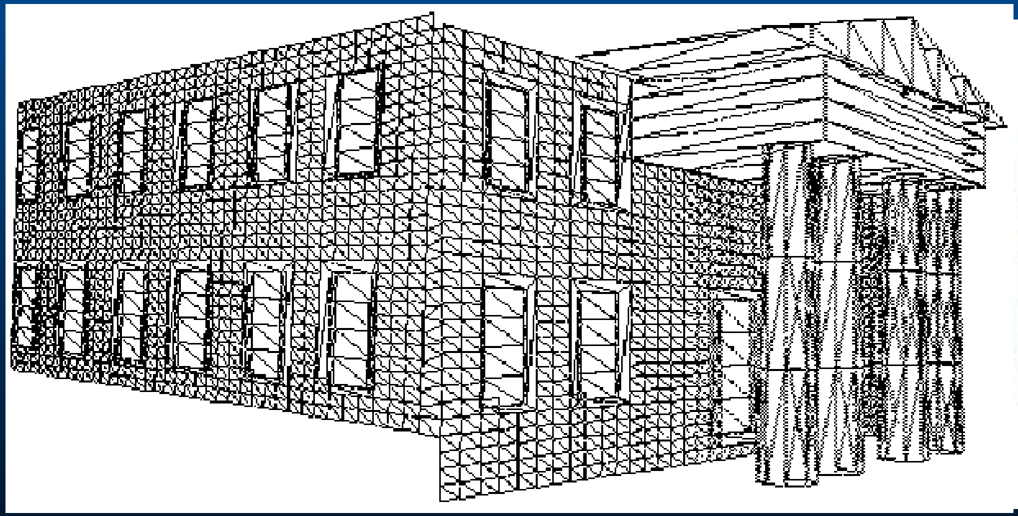


Overhead view

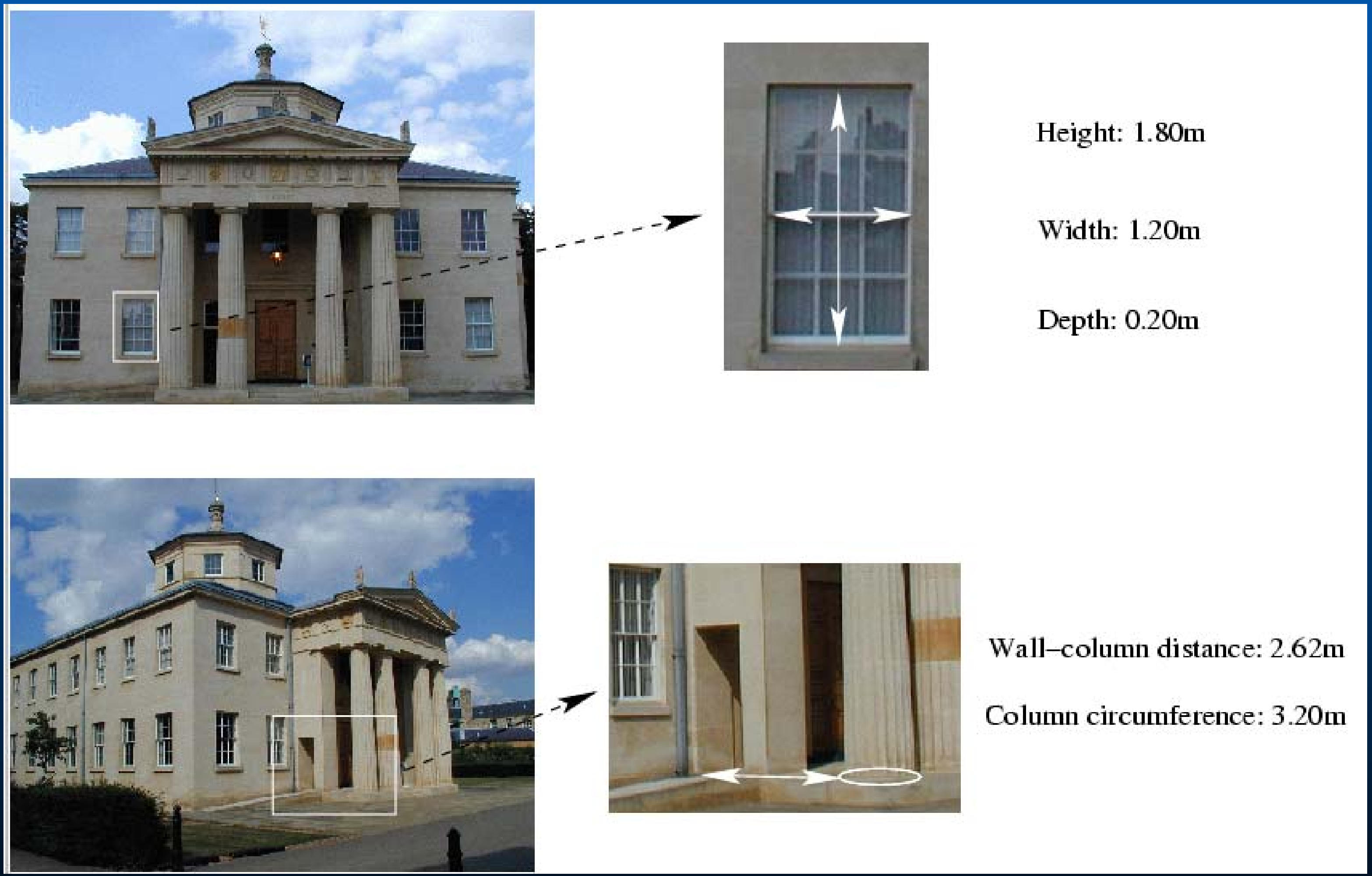
Model estimation

- Initial shape estimate obtained via existing structure and motion algorithms
  - Extract and match corners and lines
  - Self-calibrate cameras
  - Plane fitting RANSAC to estimate walls
- Search for likely primitives on each wall [ICCV01]
  - This produces seed points for the MCMC process
  - Likelihood measure is based on sum squared error of reprojected pixels
    - Assumes Lambertian model

Reconstructed model



Ground truth



Section 2
Algorithms For Feature Matching

Overview

- RANSAC
- Problems with RANSAC
- MAPSAC

Section 2.1
Random Sampling Methods

Section 2.1
Random Sampling Methods

- If the features are related by some sort of global relation then we can use this to guide the matching.
- Basic Idea is to use some sort of correlation to get putative matches.
- Then randomly sample from these, estimate the relation and see how many other features agree.

Object Recognition

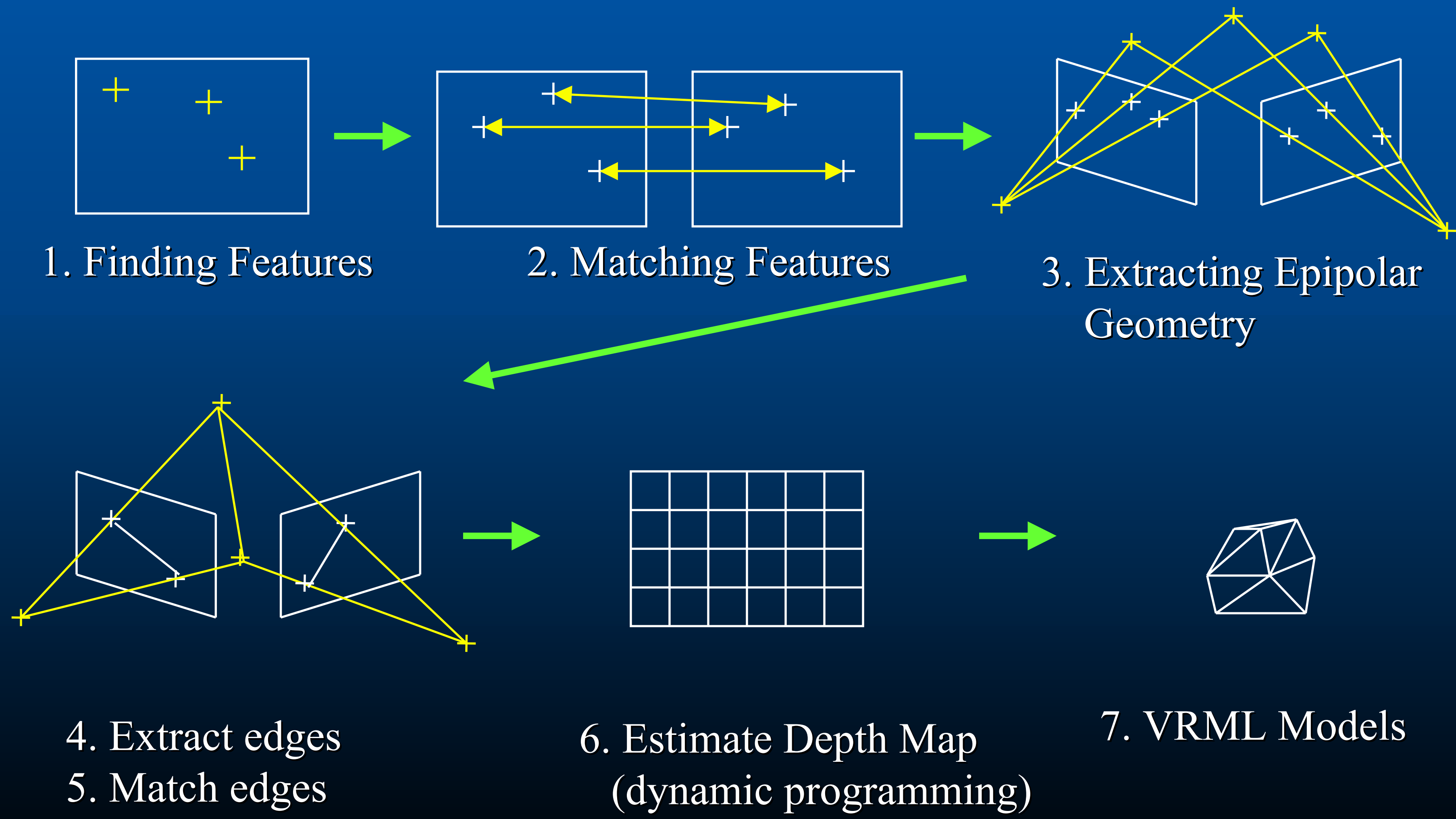
- Paradigm for the past 40 years has been [Roberts 65]:
  - Extract features in image.
  - Match features in model to image.

Structure and Motion Recovery

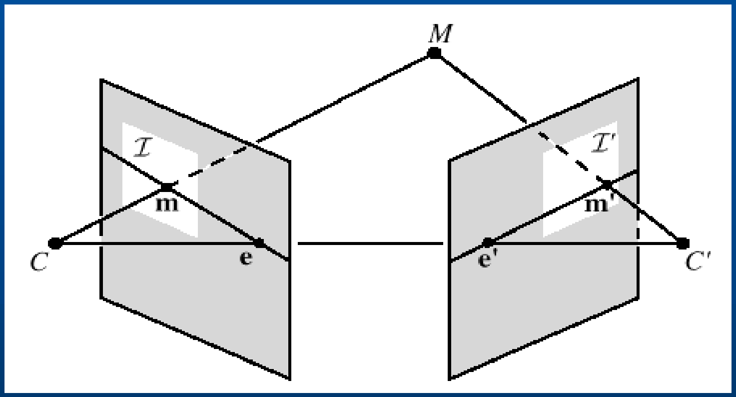
- Repeat:
  - Match features between images,
  - Infer image relation based on feature matches,
  - Rematch under guidance from image relation.
- NEXT: we illustrate RANSAC with respect to feature matching for SAM.

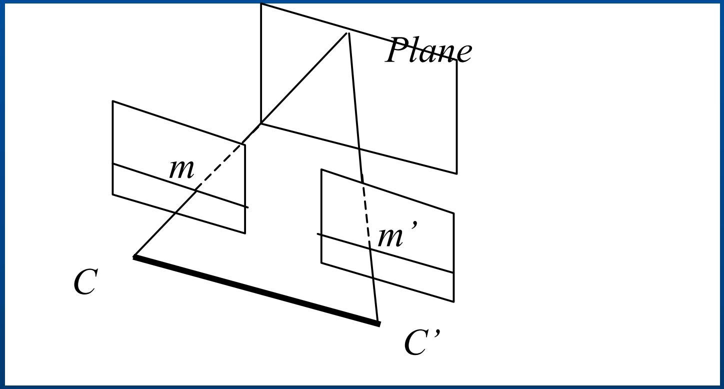
A RANSAC system for SAM

Structure and Motion Recovery



Guide matches with Geometry





$$\mathbf{x}^t \mathbf{F} \mathbf{x}' = 0$$

$$\begin{pmatrix} x & y & 1 \end{pmatrix} \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix} \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = 0$$

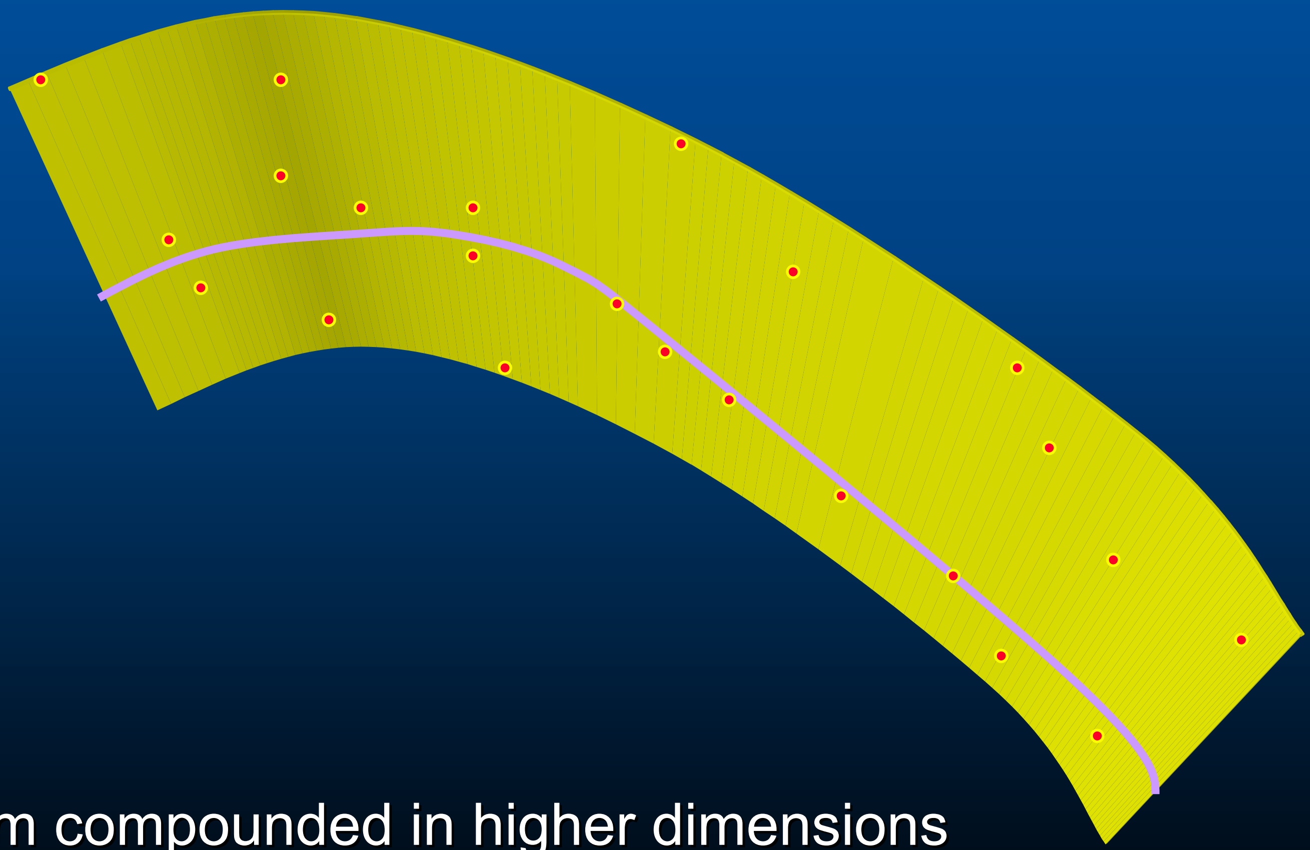
$$\mathbf{x}' = \mathbf{H}\mathbf{x}$$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Concatenated Image space

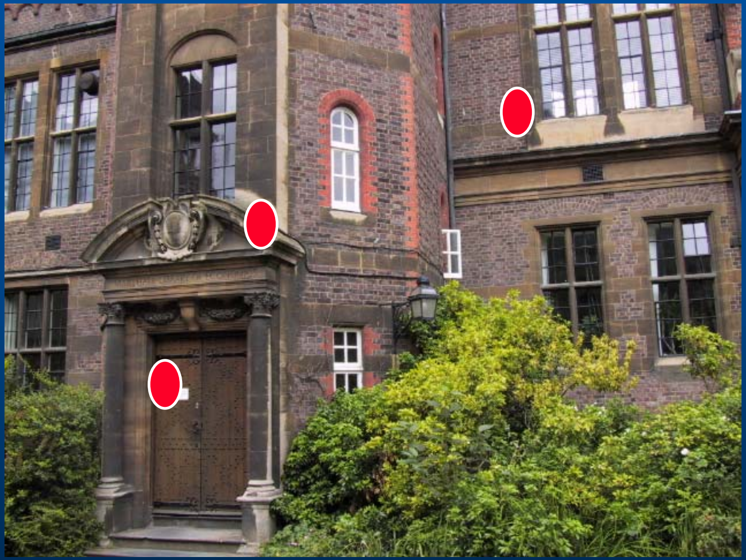
- ***2 Views- consider 4D space of image coordinates (x,y,x’,y’).***
- ***Fundamental matrix is a 3D manifold in this space.***
- ***Homography is a 2D manifold in this space.***

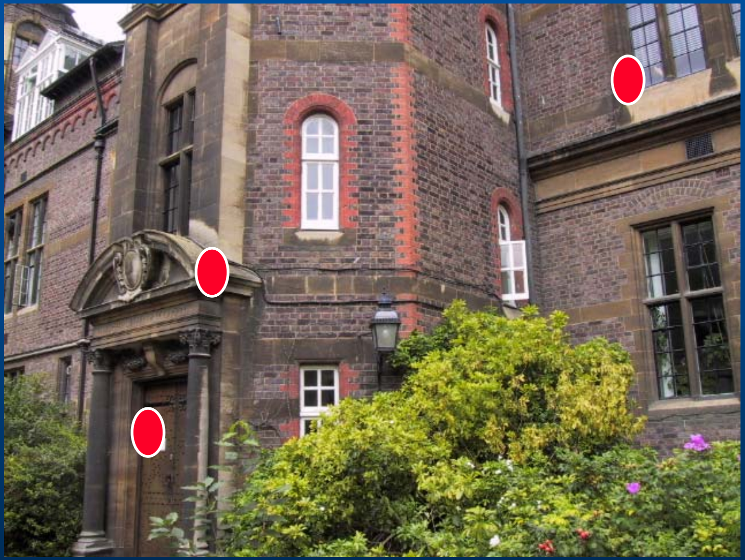
Estimation of Motion model like fitting a manifold to space of 4D image points in two images:



- Problem compounded in higher dimensions

Stage 1 Corner Detection





Images of the same scene from different viewpoints

Feature Detectors need to consistently locate the position within the image of a landmark on the 3D object.

Typical Features Detected

Stage 2 Feature Matching

Images of the same scene from different viewpoints

Initial Feature correspondence via Cross Correlation.

Stage 2 Feature Matching

Initial Feature correspondence via Cross Correlation
Many outliers.

Stage 3 Estimation of Epipolar Geometry

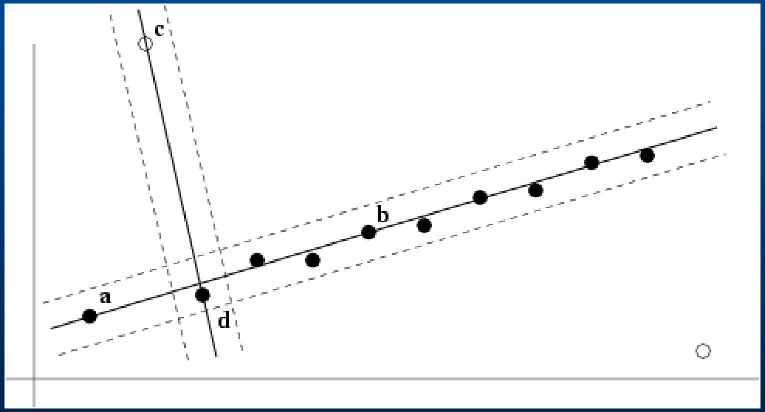
Images of the same scene from different viewpoints

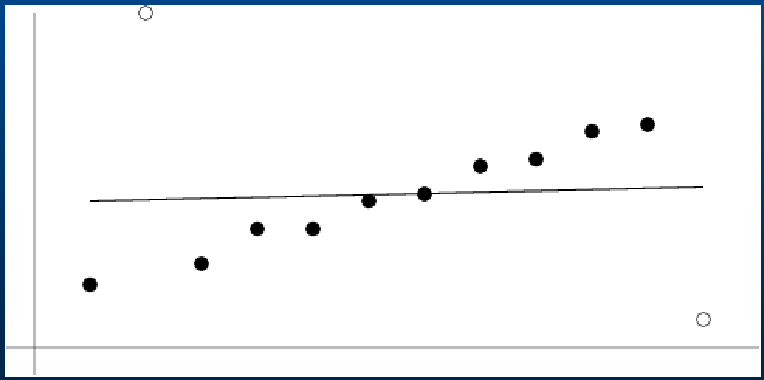
Corresponding features must lie on corresponding epipolar lines.

All epipolar lines intersect at a common point.

Robust estimation

- What if set of matches contains gross outliers?





RANSAC

<u>Objective</u>

Robust fit of model to data set S which contains outliers

<u>Algorithm</u>

(i) Randomly select a sample of *s* data points from S and instantiate the model from this subset.

(ii) Determine the set of data points S_i which are within a distance threshold *t* of the model. The set S_i is the consensus set of samples and defines the inliers of S.

(iii) If the subset of S_i is greater than some threshold *T*, re-estimate the model using all the points in S_i and terminate

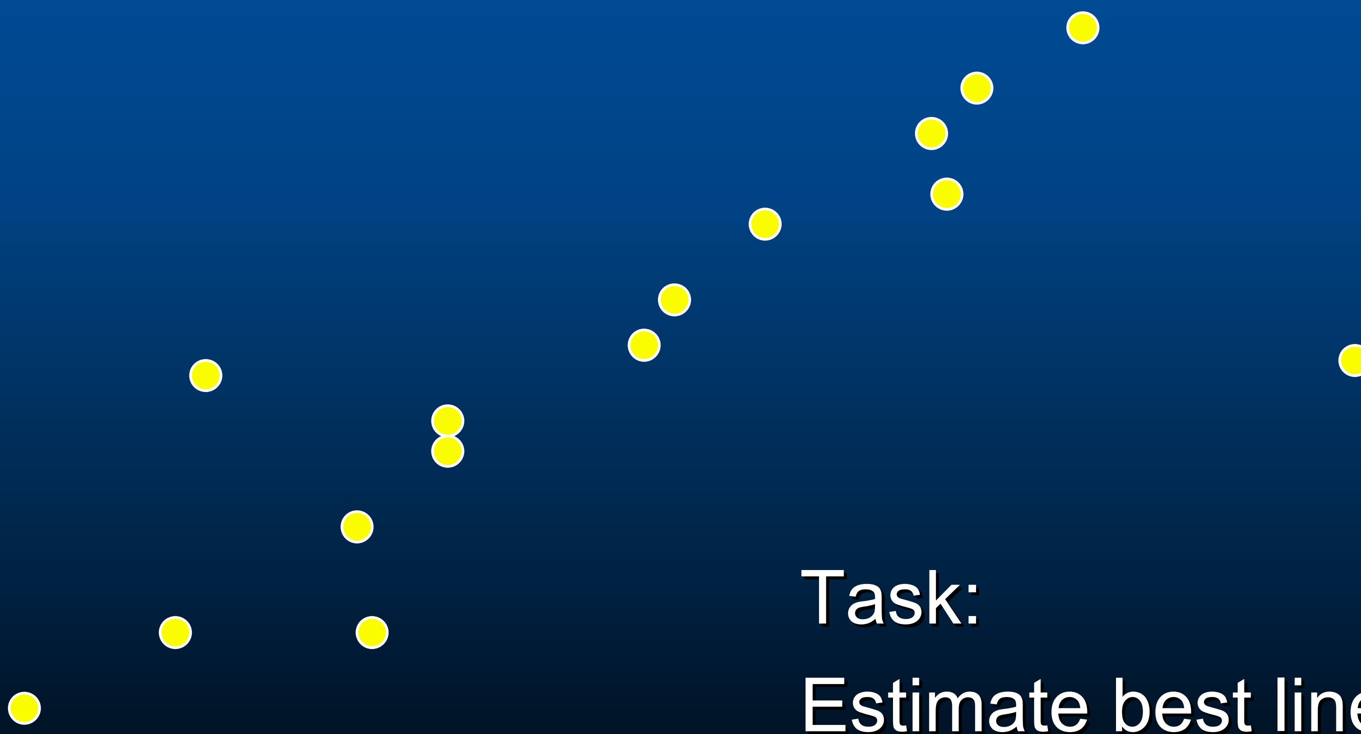
(iv) If the size of S_i is less than *T*, select a new subset and repeat the above.

(v) After *N* trials the largest consensus set S_i is selected, and the model is re-estimated using all the points in the subset S_i

RANSAC

- Repeat M times:
  - Sample minimal number of matches to estimate two view relation.
  - Calculate number of inliers or posterior likelihood for relation.
  - Choose relation to maximize number of inliers.

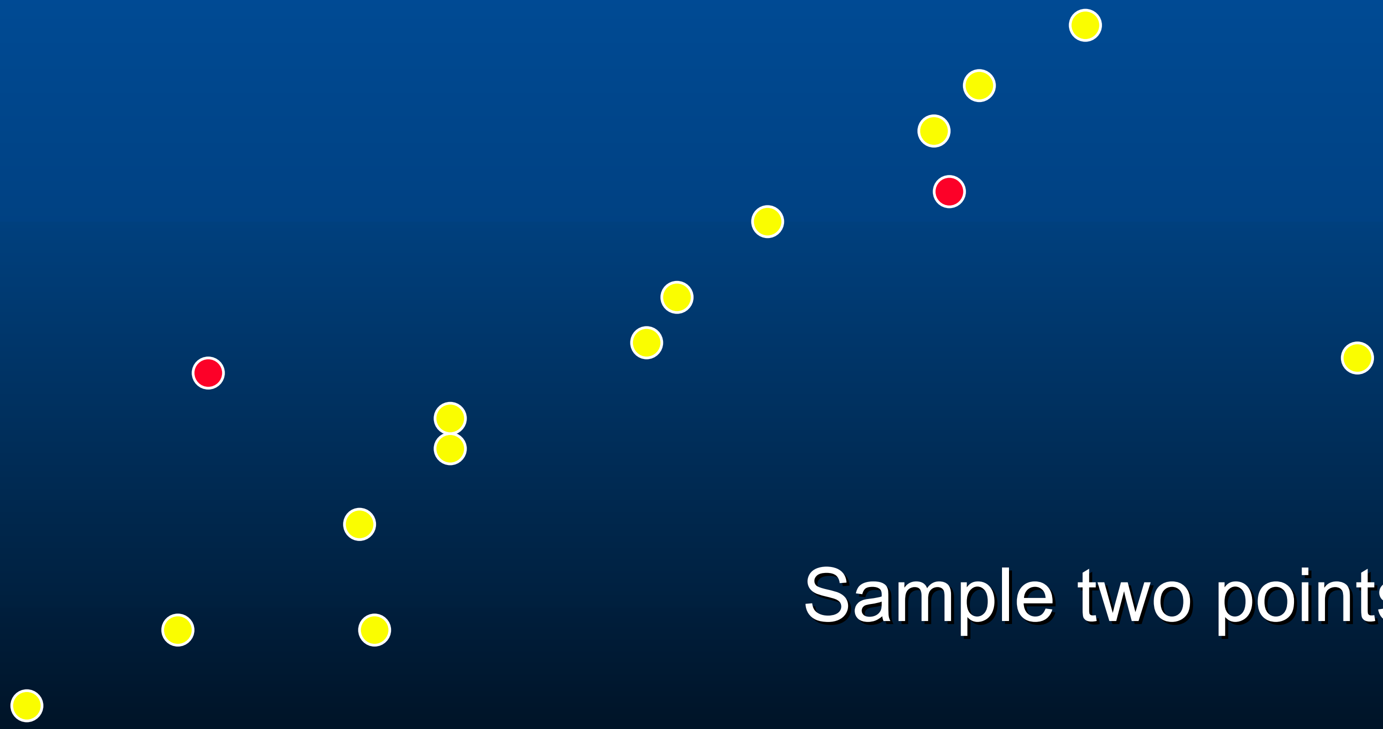
RANSAC line fitting example



Task:

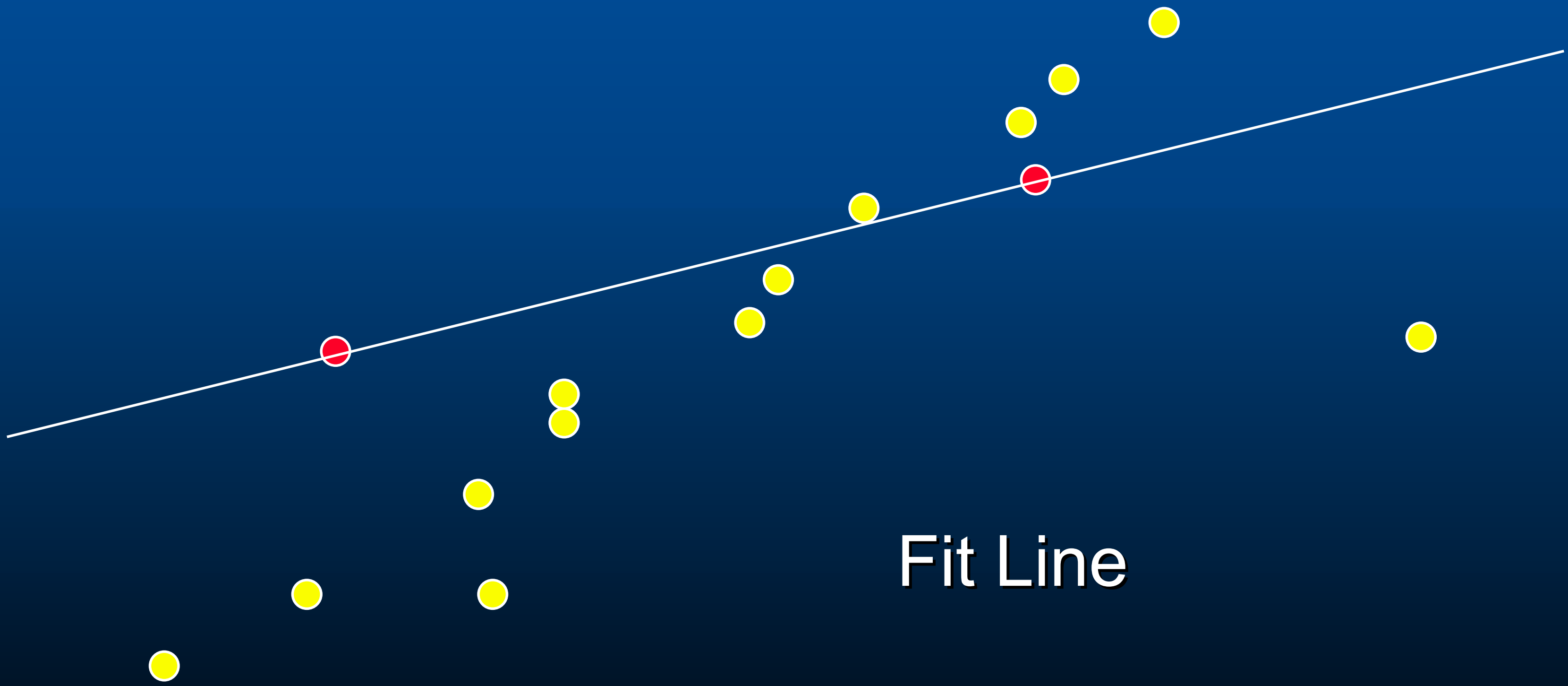
Estimate best line

RANSAC line fitting example

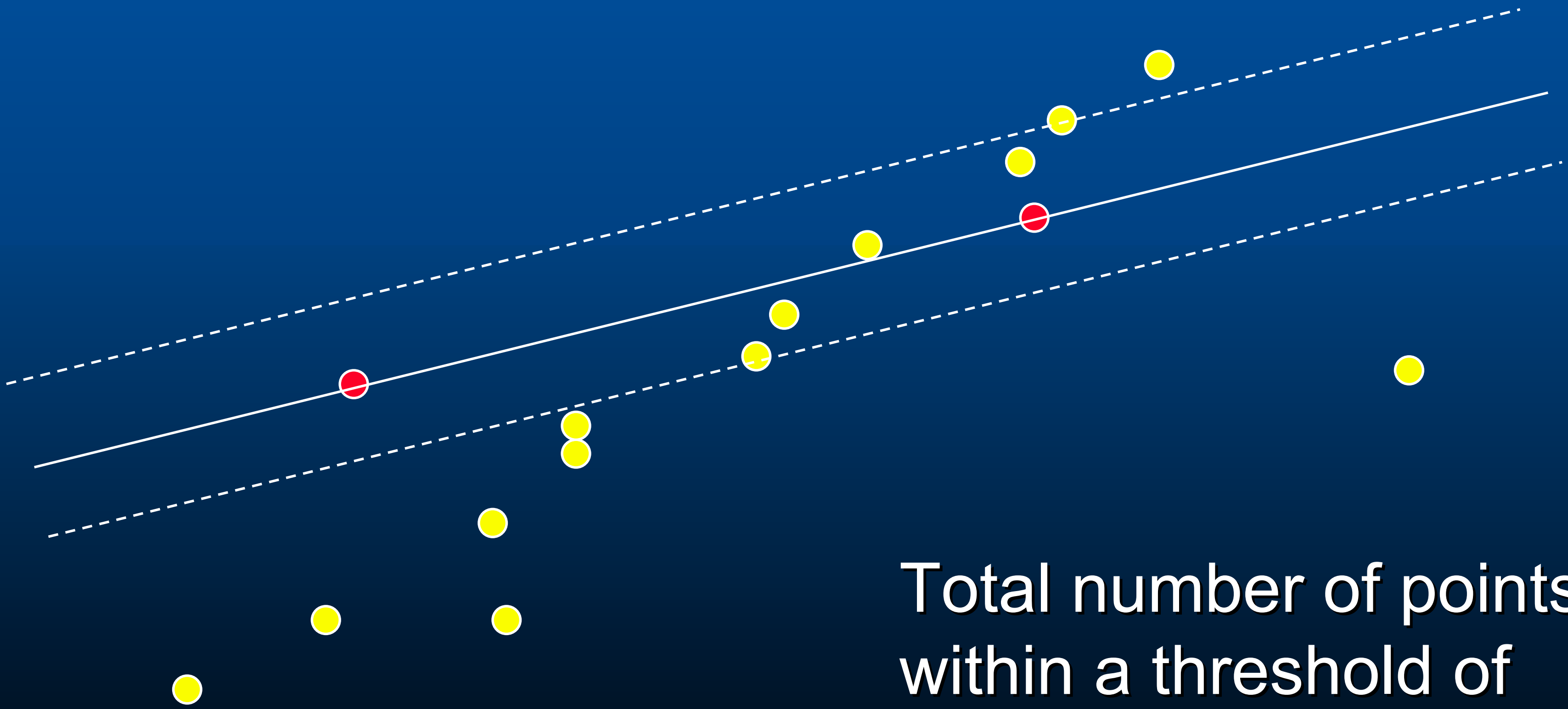


Sample two points

RANSAC line fitting example



RANSAC line fitting example

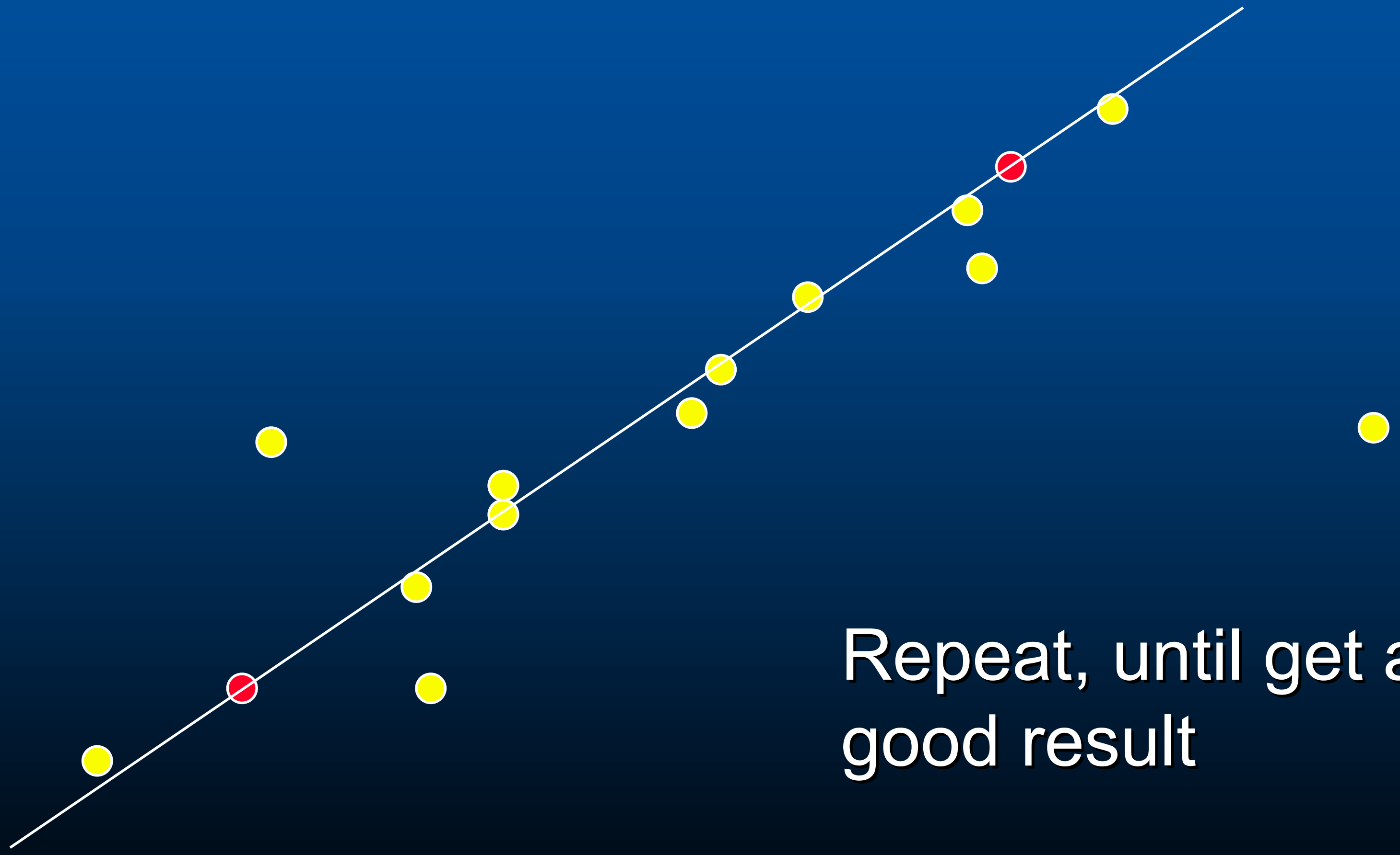


Total number of points within a threshold of line.

RANSAC line fitting example

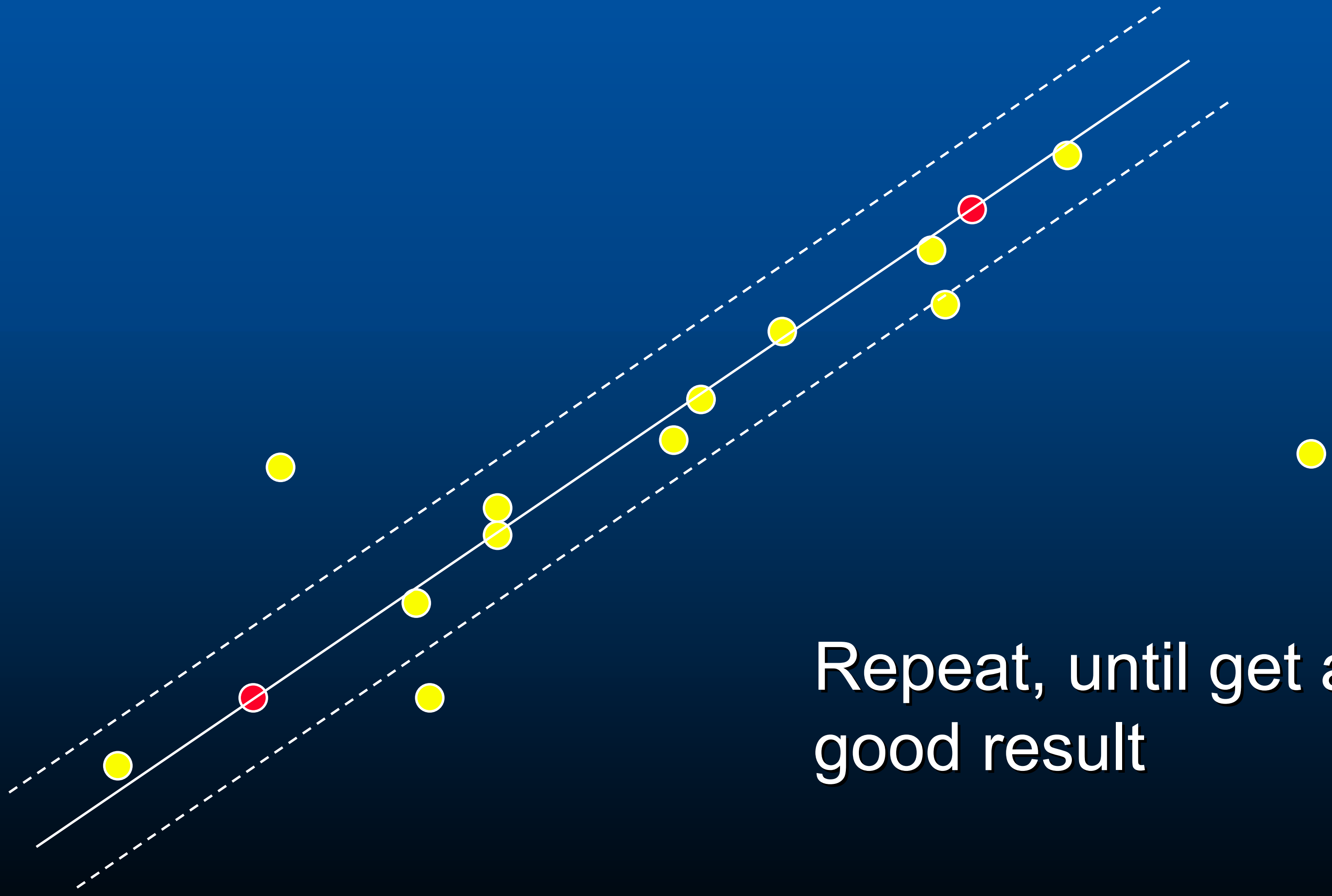
Repeat, until get a good result

RANSAC line fitting example



Repeat, until get a good result

RANSAC line fitting example



Repeat, until get a good result

How many samples?

Choose *N* so that, with probability *p*, at least one random sample is free from outliers. e.g. *p*=0.99

$$\left(1-\left(1-e\right)^s\right)^N = 1-p$$

$$N = \log\left(1-p\right)/\log\left(1-\left(1-e\right)^s\right)$$

	proportion of outliers e						
s	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

Adaptively determining the number of samples

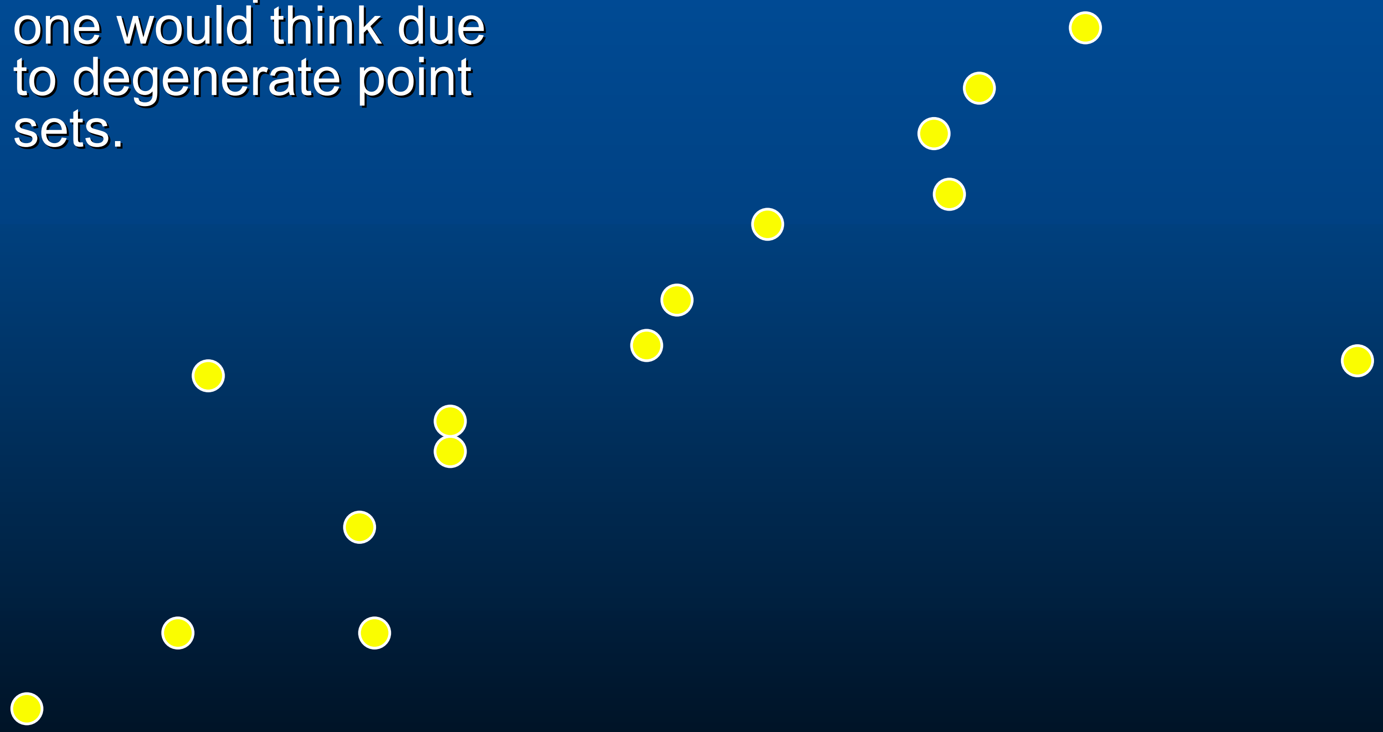
e is often unknown a priori, so pick worst case, e.g. 50%, and adapt if more inliers are found, e.g. 80% would yield e=0.2

- $N=\infty$, *sample_count* =0
- While N >*sample_count* repeat
  - Choose a sample and count the number of inliers
  - Set e=1-(number of inliers)/(total number of points)
  - Recompute N from e
  - Increment the *sample_count* by 1
- Terminate

$$\left(N = \log(1-p)/\log\left(1-(1-e)^s\right)\right)$$

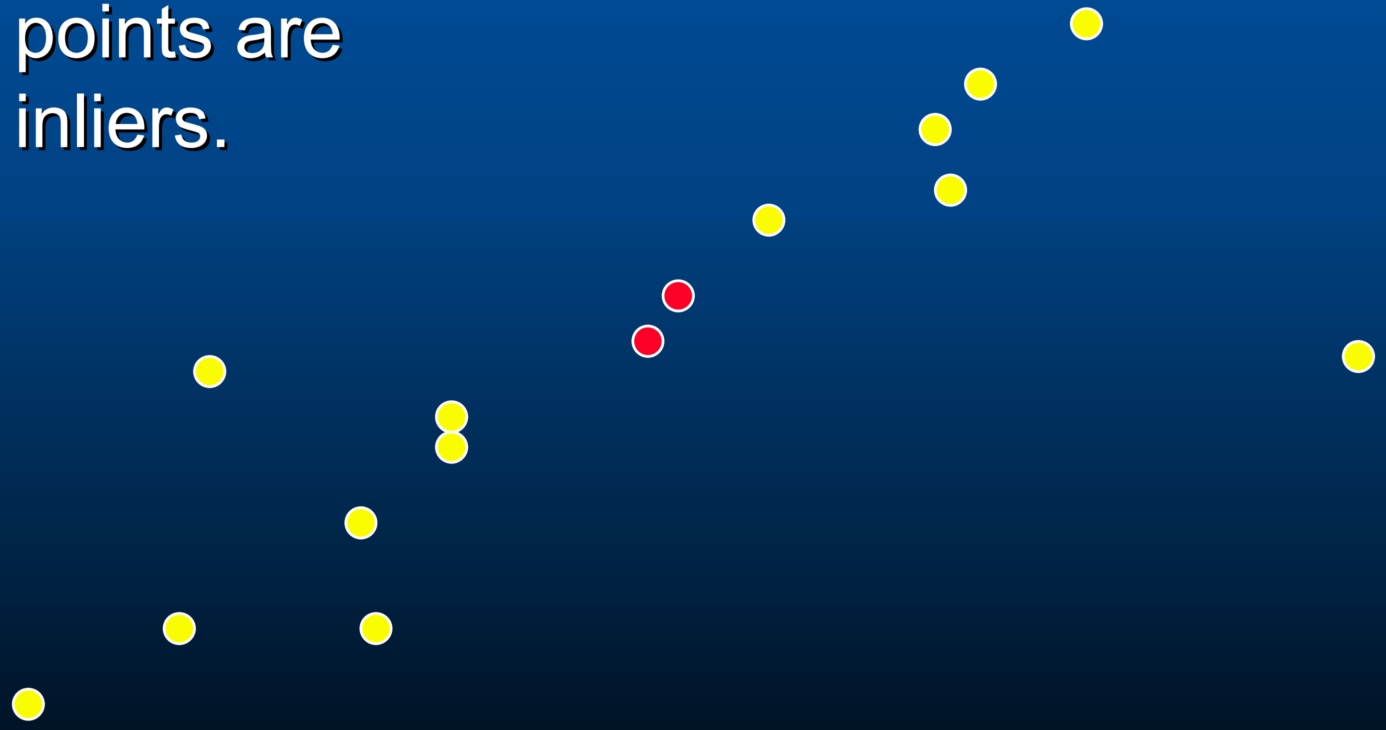
Number of Samples II

- Make take many more samples than one would think due to degenerate point sets.



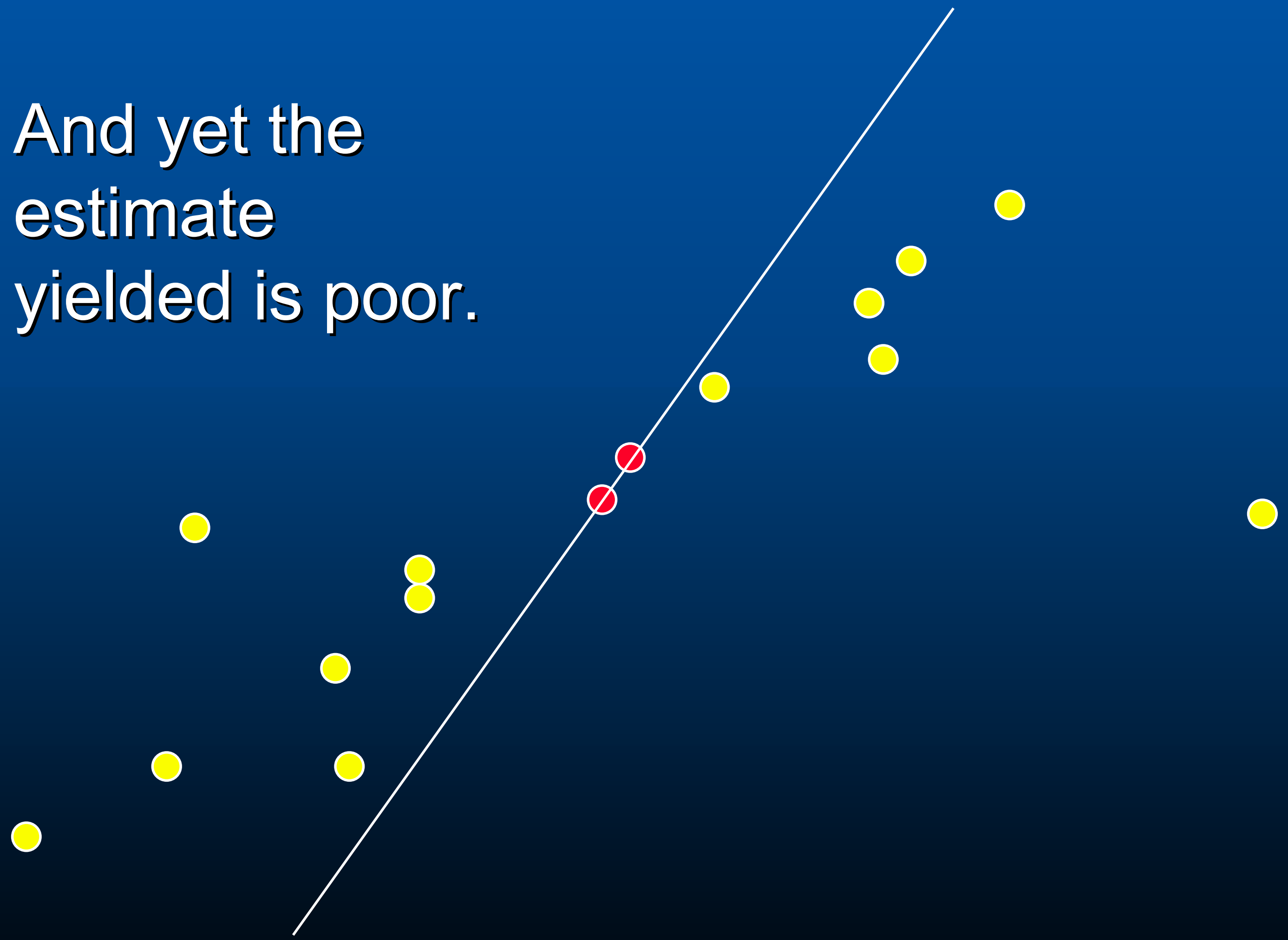
Number of Samples II

- These two points are inliers.



Number of Samples II

- And yet the estimate yielded is poor.



Automatic computation of H

Objective

Compute homography between two images

Algorithm

(i) **Interest points:** Compute interest points in each image

(ii) **Putative correspondences:** Compute a set of interest point matches based on some similarity measure

(iii) **RANSAC robust estimation:** Repeat for N samples

(a) Select 4 correspondences and compute H

(b) Calculate the distance $d_\perp$ for each putative match

(c) Compute the number of inliers consistent with H ($d_\perp < t$)

Choose H with most inliers

(iv) **Optimal estimation:** re-estimate H from all inliers by minimizing ML cost function with Levenberg-Marquardt

(v) Guided matching: Determine more matches using prediction by computed H

Optionally iterate last two steps until convergence

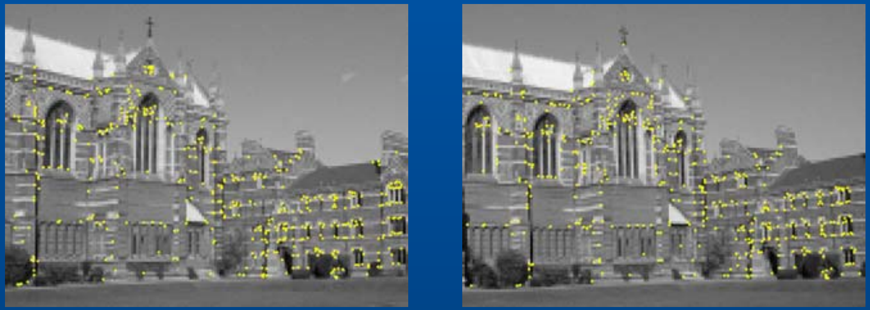
Determine putative correspondences

- Compare interest points
  - Similarity measure:SAD, SSD, NCC on small neighborhood

- NOTE: we can use correlation score to bias the selection of the samples selecting matches with a better correlation score more often (Tordoff et al).

- NOTE multiple matches for each point can be RANSAC'ed on (although this increases the proportion of outliers).

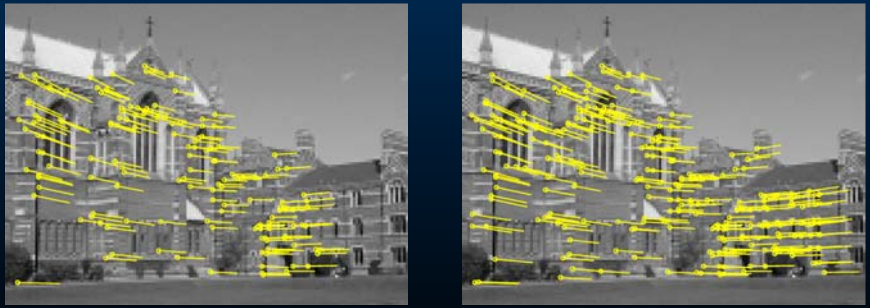
Example: robust computation



Interest points (500/image)

Putative correspondences (268)

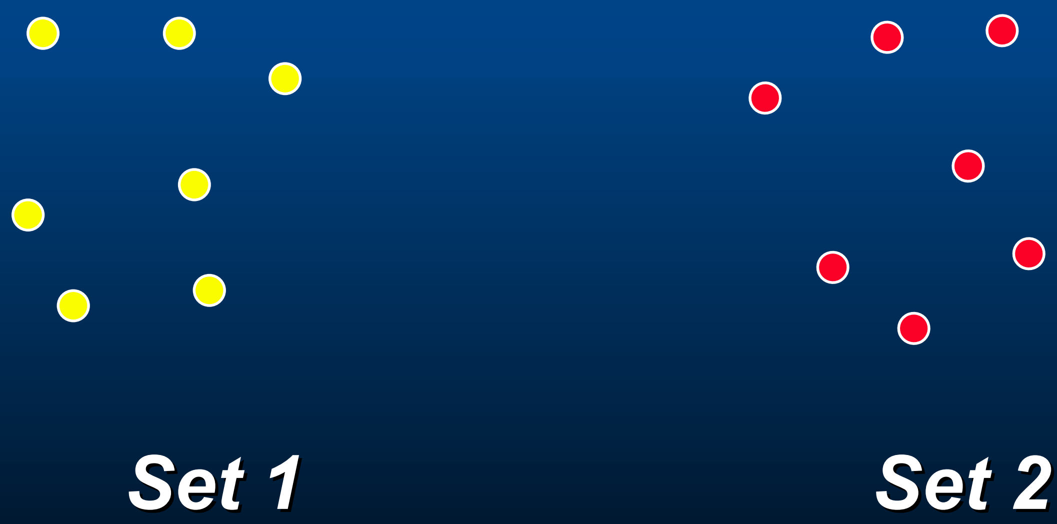
Outliers (117)



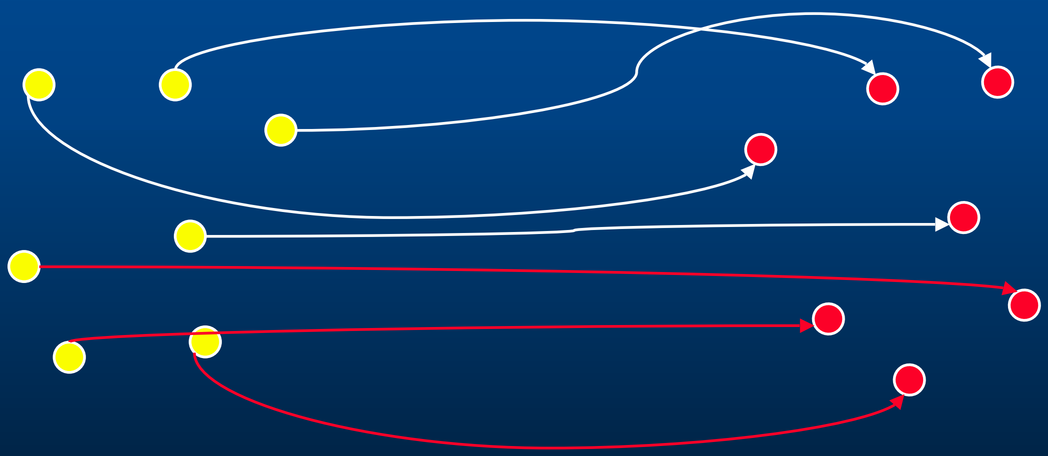
Inliers (151)

Final inliers (262)

Example; 2D Similarity Transformation



Example; 2D Similarity Transformation

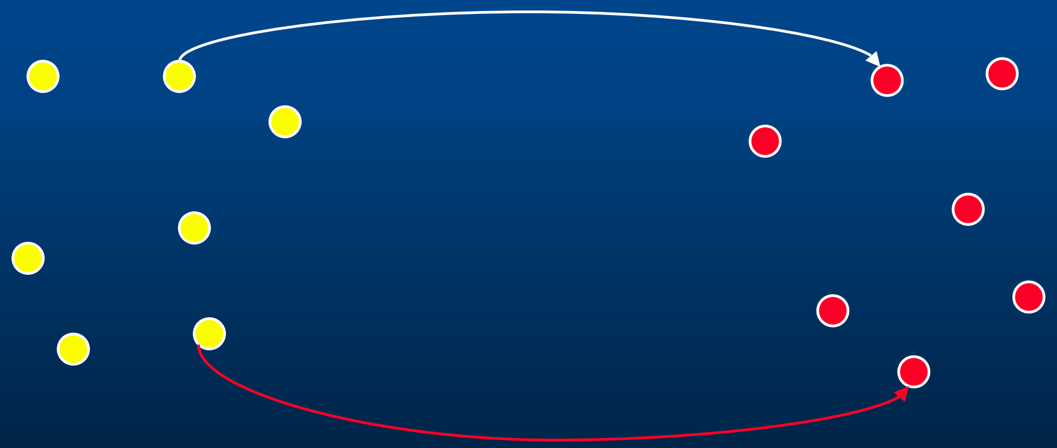


Set 1 Set 2

Set of matches from some correlation function.

Some are incorrect (shown in red)

Example; 2D Similarity Transformation

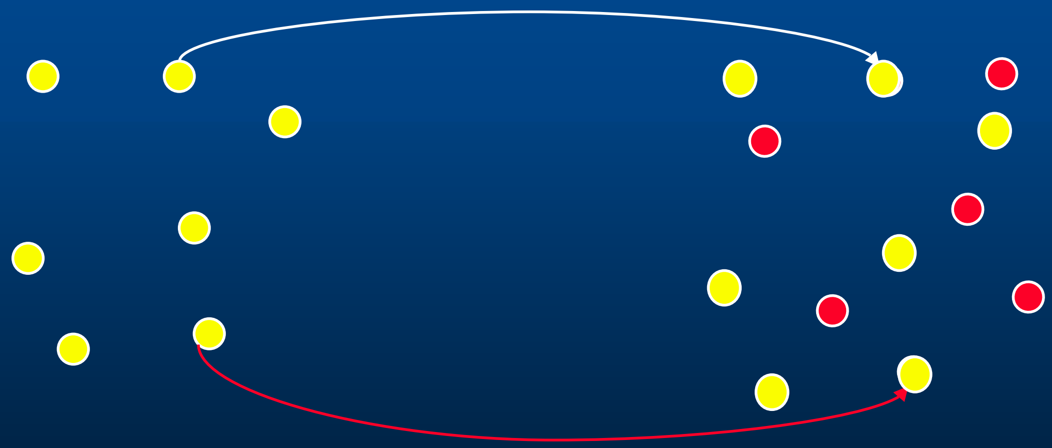


Set 1 Set 2

Two matches, used to infer transform,

Here: Top match correct, bottom incorrect

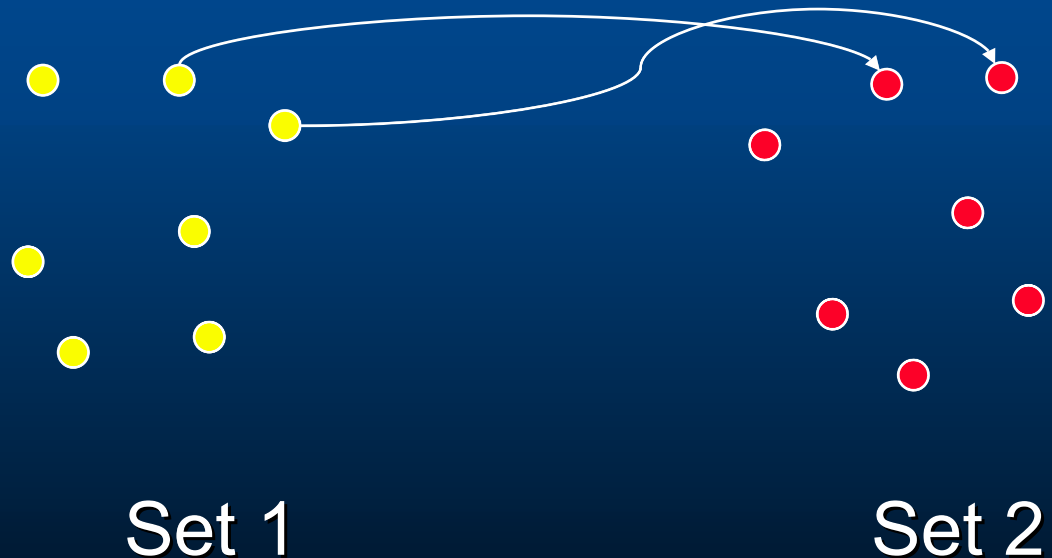
Example; 2D Similarity Transformation



Set 1 Set 2

Features mapped under transform do not align well.

Example; 2D Similarity Transformation



On the other hand, if we pick two correct matches (modulo noise).

Example; 2D Similarity Transformation

Set 1 Set 2

Alignment is good!

Problems and Improvements to RANSAC

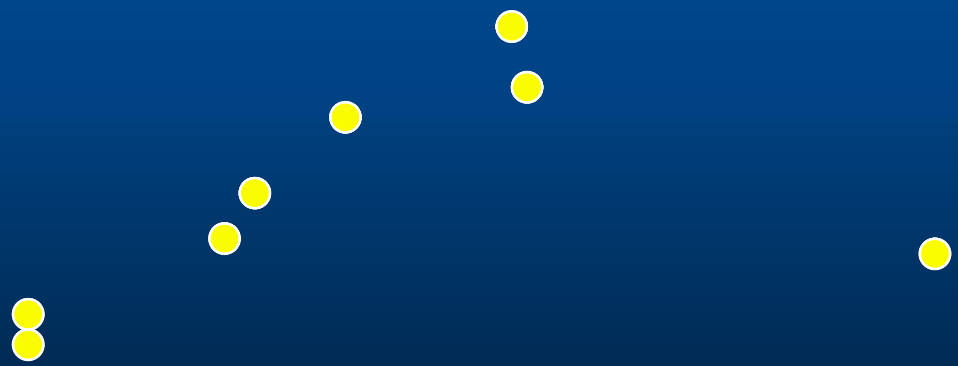
- Problem 1, cost function.
- Problem 2, what model to fit?

Problem 1; cost function

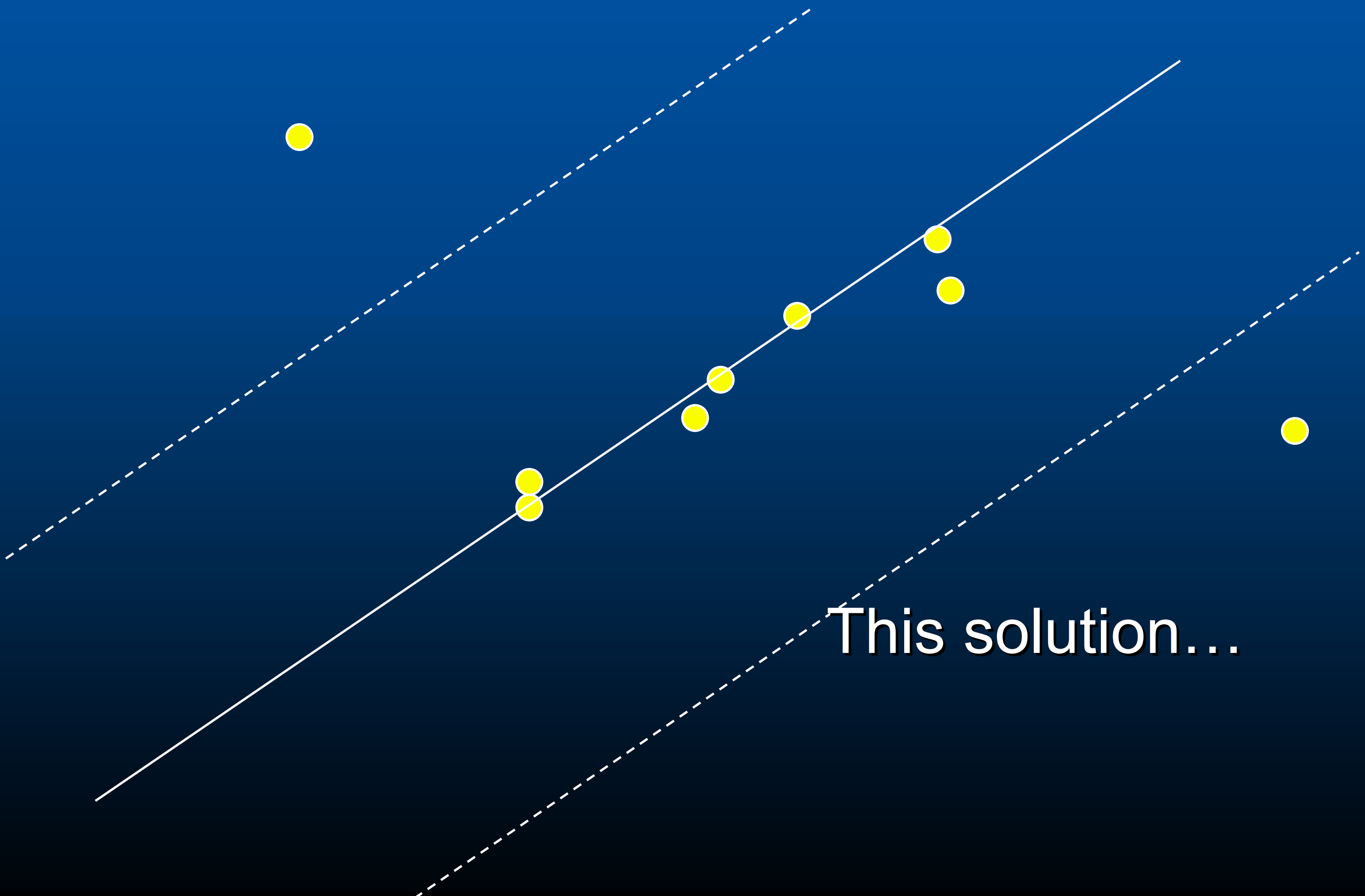
- RANSAC can be vulnerable to the correct choice of the threshold:
  - Too large all hypotheses are ranked equally.
  - Too small leads to an unstable fit.

- The interesting thing is that the same strategy can be followed with any modification of the cost function.

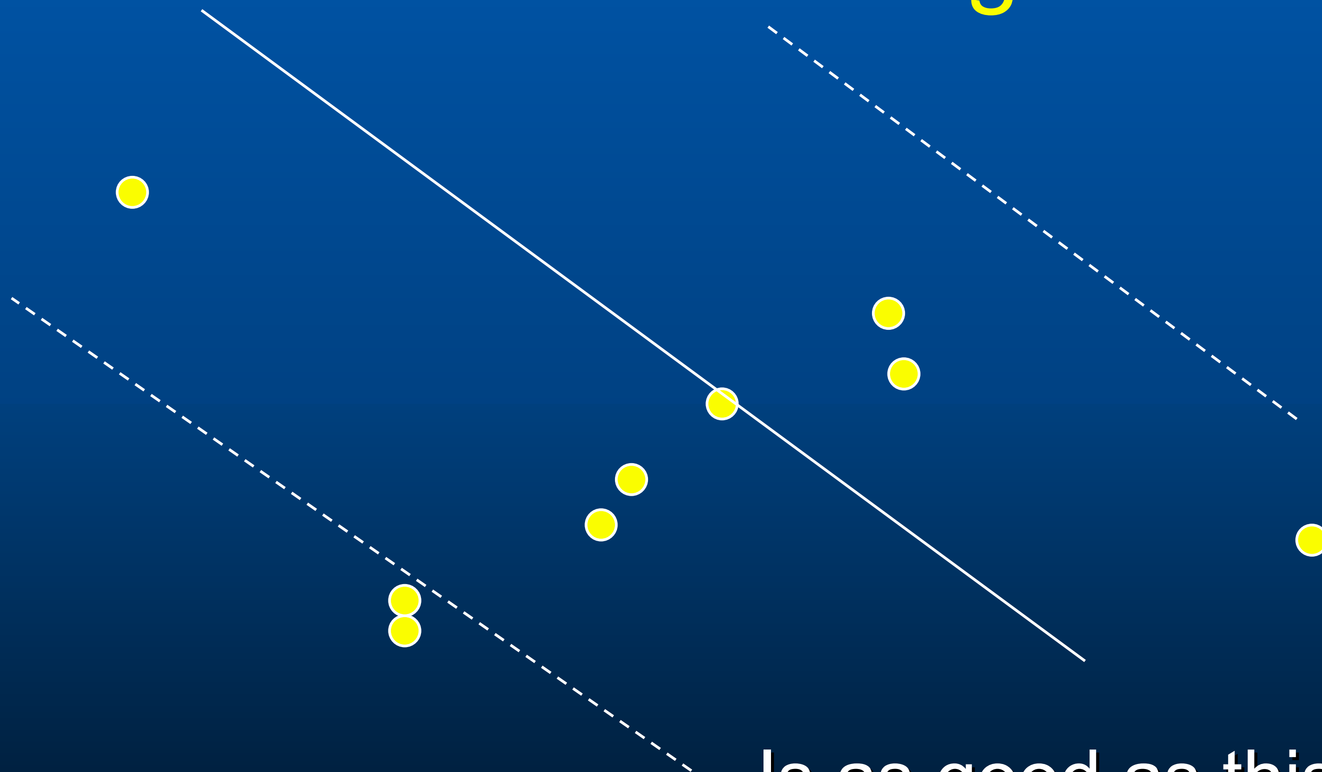
Problem with RANSAC; threshold too high



Problem with RANSAC; threshold too high

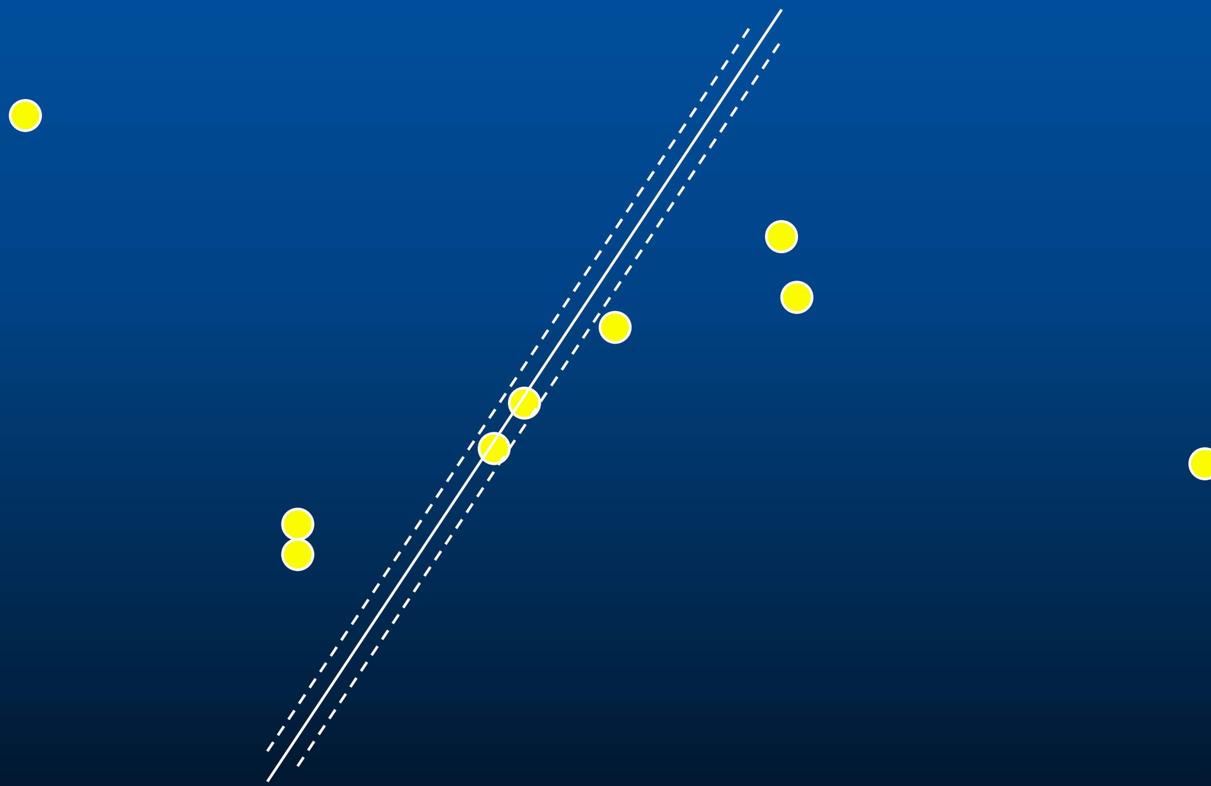


Problem with RANSAC; threshold too high



Is as good as this solution

Problem with RANSAC; threshold too low-no support



Problem 1; cost function

- Examples of other cost functions
  - Least Median Squares; i.e. take the sample that minimized the median of the residuals.
  - MAPSAC/MLESAC use the posterior or likelihood of the data.
  - MINPRAN (Stewart), makes assumptions about randomness of data

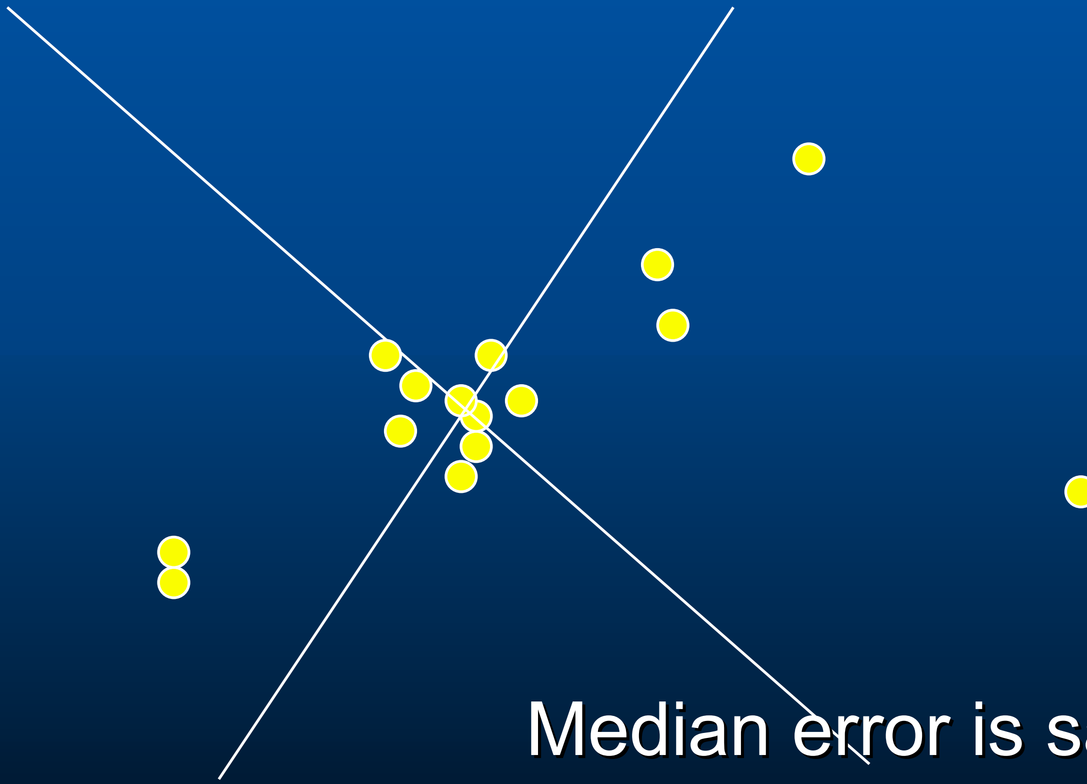
LMS

- Repeat M times:
  – Sample minimal number of matches to estimate two view relation.
  – Calculate error of all data.
  – Choose relation to minimize median of errors.

Pros and Cons LMS

- PRO
  - Do not need any threshold for inliers.
- CON
  - Cannot work for more than 50% outliers.
  - Problems if a lot of data belongs to a submanifold (e.g. dominate plane in the image)

Con: LMS, subspace problem



Median error is same for two solutions.

Con: LMS, subspace problem

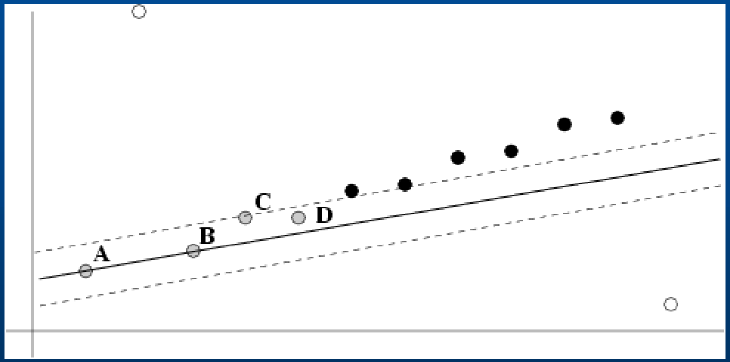
No good solution if the number of outliers >50%

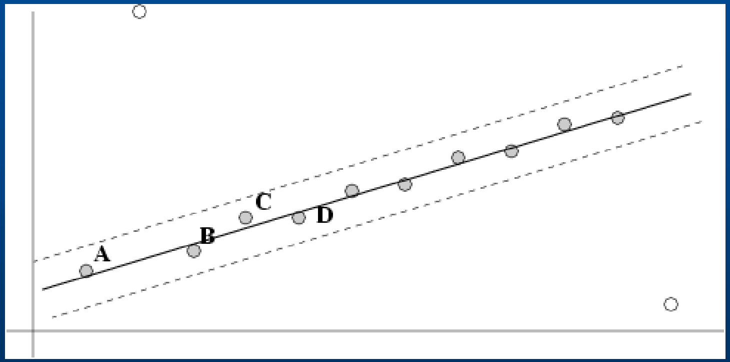
Pros LMS

- One major advantage of LMS is that it can yield a robust estimate of the variance of the errors.
- But care should be taken to use the right formula, as this depends on the distribution of the errors, and degrees of freedom in the errors (codimension).

Robust Maximum Likelihood Estimation

Random Sampling can optimize any function:



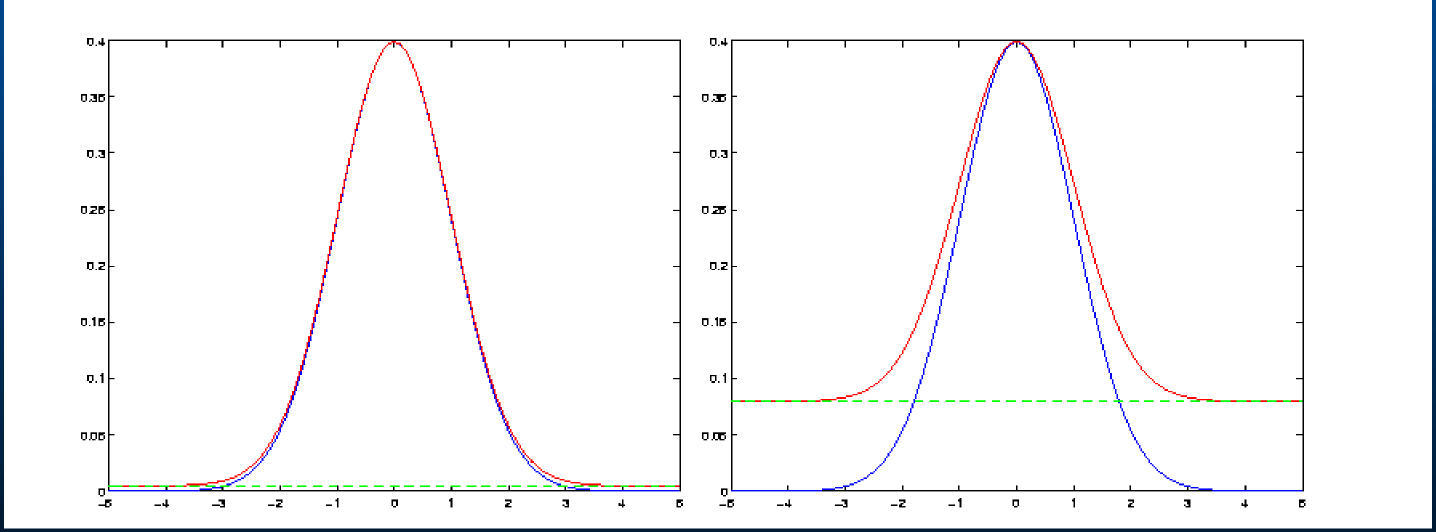


Better, robust cost function, MLESAC

$$\mathcal{R} = \sum_{i} \rho\left(d_{\perp i}\right) \text{ with } \rho(e) = \begin{cases} e^2 & e^2 < t^2 \text{ inlier} \\ t^2 & e^2 > t^2 \text{ outlier} \end{cases}$$

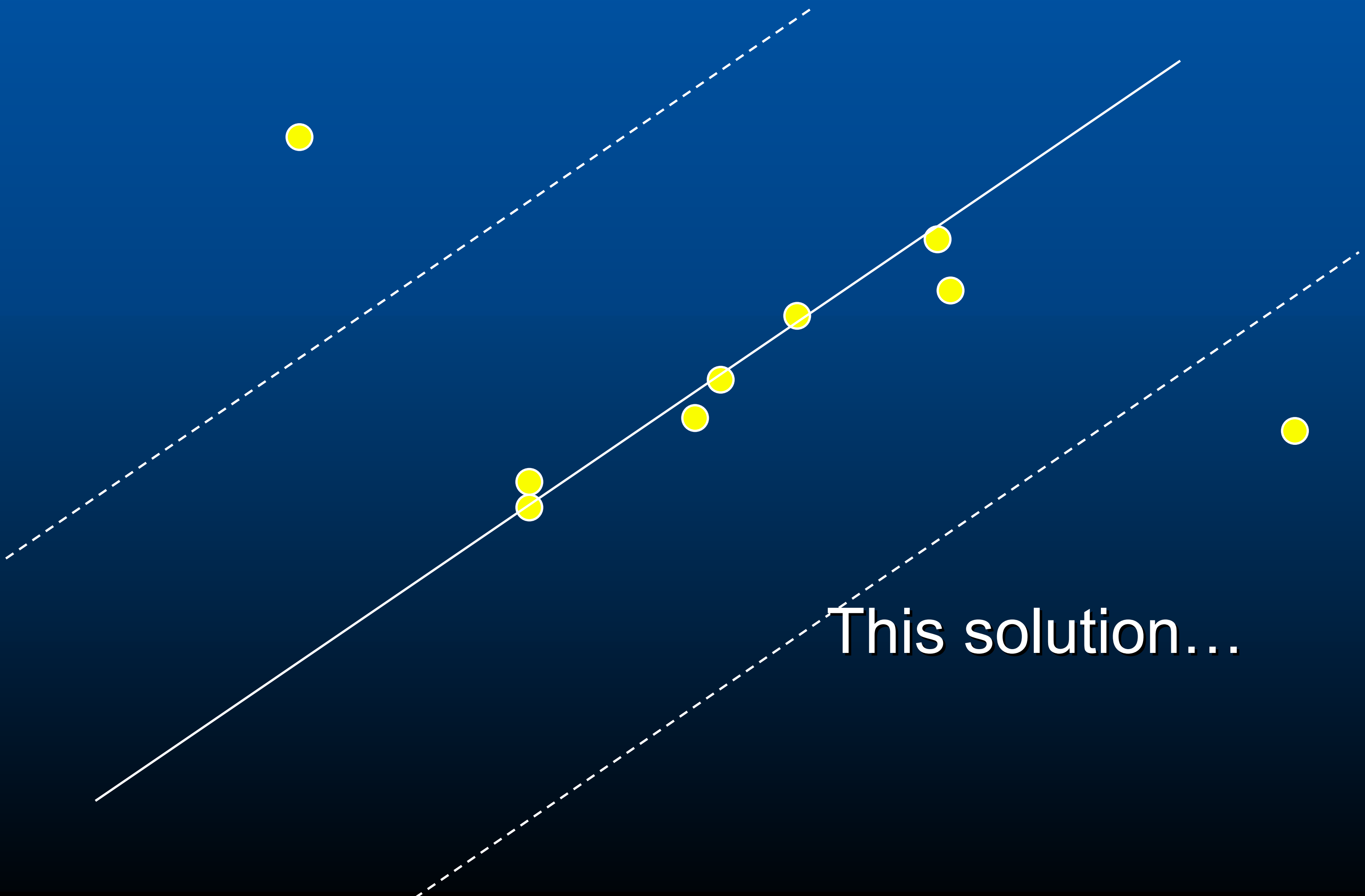
Mixture (Maxture) of Gaussian/Uniform?

$$\rho_2\left(\frac{e^2}{\sigma^2}\right) = \begin{cases} \frac{e^2}{\sigma^2} & \frac{e^2}{\sigma^2} < T \\ T & \frac{e^2}{\sigma^2} \geq T \end{cases} .$$

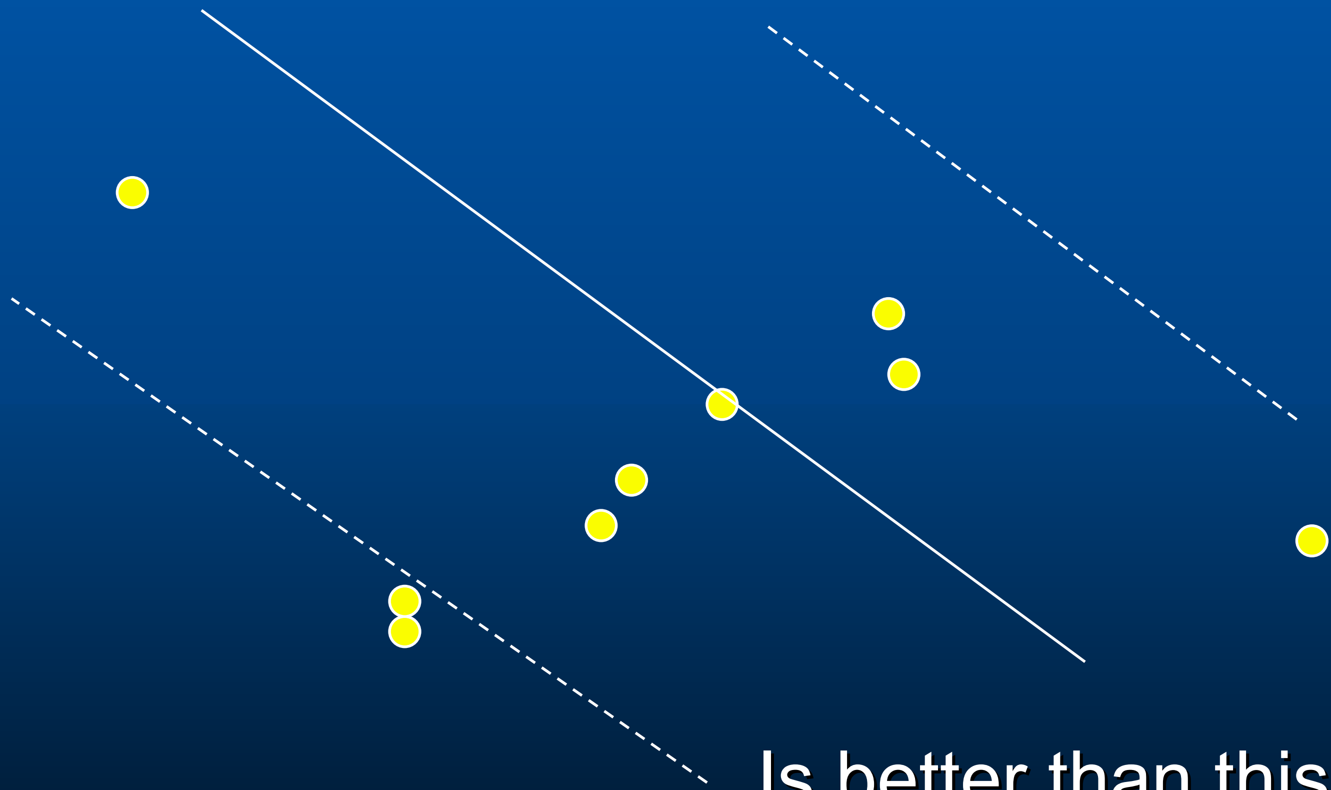


- Red-mixture, green-uniform, blue-Gaussian.

MLESAC/MAPSAC



MLESAC/MAPSAC



Is better than this solution

MAPSAC

- Add in prior to get to MAP solution
- Interesting thing is that with MAPSAC one could sample less than the minimal number of points to make an estimate (using prior as extra information).
- Any posterior can be optimized; random sampling good for matching AND FUNCTION OPTIMIZATION! e.g. MAPSAC is a cheap way to optimize objective functions regardless of outliers or not.

MAPSAC

- Once the benefits of MAPSAC are seen there is no reason to continue to use RANSAC;
  - in many situations the improvement in the solution can be marked
  - Especially if want to use prior information (e.g. the F matrix changing smoothly over time).
  - Gives an optimized solution

AT NO EXTRA COST!

Problem 2, what model to fit?

- There are many cases when we do not know the relation between the images, there may a choice of many.
- In this case a Bayesian solution might be to evaluate the likelihood of each.

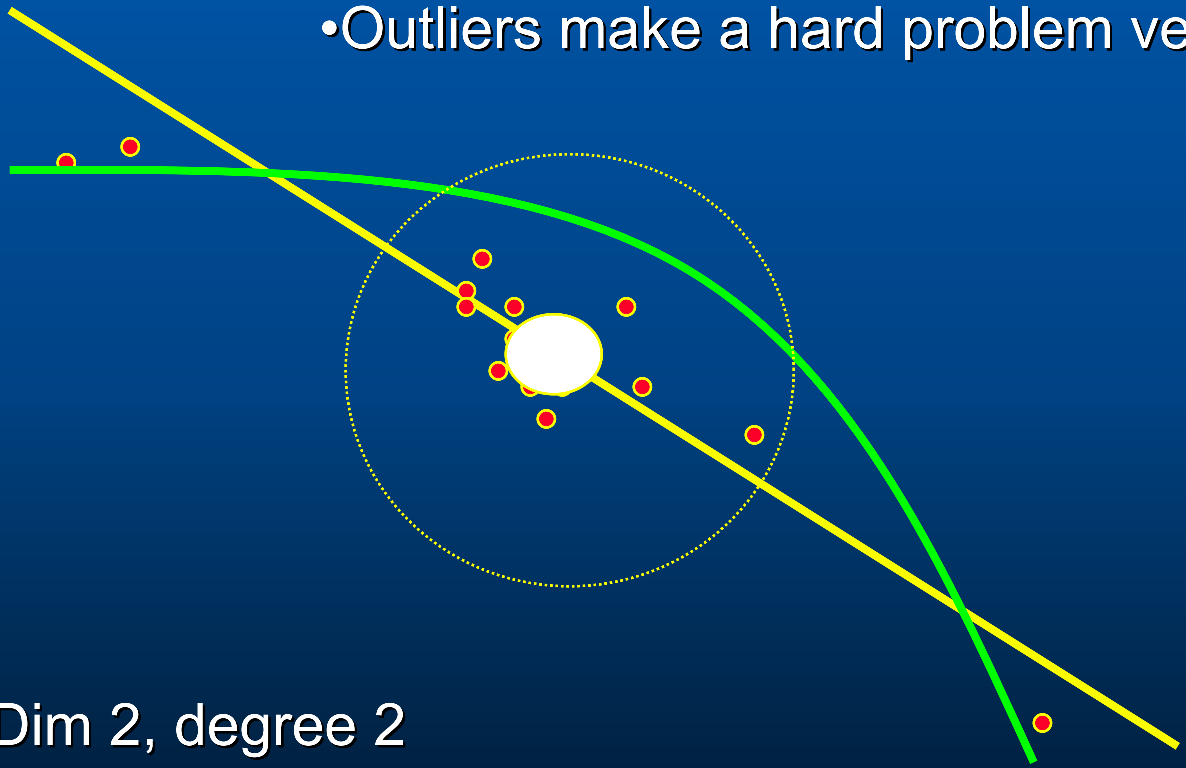
There are many possible two view relations, e.g.

Relation, $\mathcal{R}$	c	k	d	Q	Constraints, $g_q(x, y, x', y'; \boldsymbol{\theta}) = 0$	Parameters, $\boldsymbol{\theta}$
General	7	7	3	1	$\mathbf{x}^{2\top}\mathbf{F}\mathbf{x}^1 = 0$	$\mathbf{F} = \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix}$
Affine $\mathbf{F}_A$	4	4	3	1	$\mathbf{x}^{2\top}\mathbf{F}_A\mathbf{x}^1 = 0$	$\mathbf{F}_A = \begin{bmatrix} 0 & 0 & f_3 \\ 0 & 0 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix}$
Homography	4	8	2	2	$\mathbf{x}^2 = \mathbf{H}\mathbf{x}^1$	$\mathbf{H} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$
Affinity	3	6	2	2	$\mathbf{x}^2 = \mathbf{H}_A\mathbf{x}^1$	$\mathbf{H}_A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ 0 & 0 & a_7 \end{bmatrix}$

TABLE 1. **Two View Relations;** *A description of the reduced models that are fitted to degenerate sets of correspondences. c is the minimum number of correspondences needed in a sample to estimate the constraint. k is the number of parameters in the relation; d is the dimension of the constraint, Q is the number of independent constraints* $g_k()$ *on the image coordinates.*

Robust Model Selection

•Outliers make a hard problem very hard!



•Curve Dim 2, degree 2

•Line Dim 1, degree 1

•Point Dim 0, degree 1

Model Selection outside scope of this work

- See papers by me, or Kanatani.

Chum and Matas possible speed ups

- Rather than test all the data given a hypothesis (which could be costly for large amounts of data)
  - Test against a subset: Randomized RANSAC.

Altered Match Selection strategies:

- Zhang suggest picking points far apart to avoid degeneracy of samples.
- Tordoff suggests selecting matches with a good correspondence more often.
- Chum and Matas suggest Hi-Lo RANSAC: each time a large consensus set is found RANSAC again within the set of inliers…

Section 2.3
Robust Registration of 2D and 3D Point Sets ICP

Thanks to Andrew Fitzgibbon

Section 2.3
Robust Registration of 2D and 3D Point Sets ICP

Introduction to point-set registration

The ICP algorithm

The Levenberg-Marquardt version

Comparisons and contrasts between the two.

The problem

Input:

Two point sets
$\mathcal{M} = \{\mathbf{M}_i\}$ and $\mathcal{D} = \{\mathbf{D}_j\}$

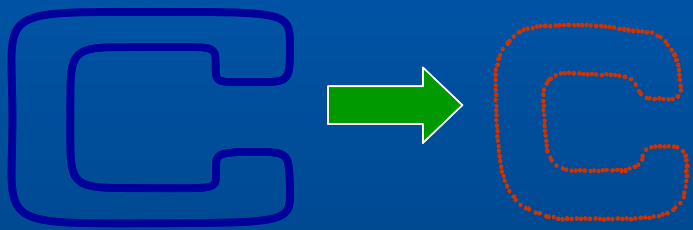
Assumption:

$\mathcal{D}$ is obtained by subjecting $\mathcal{M}$ to a transformation T, and measuring with error

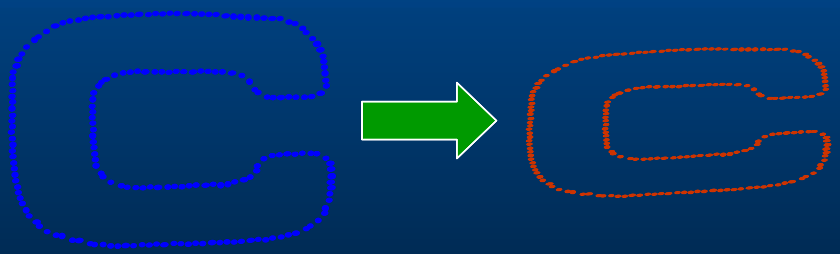
Task:

Determine T

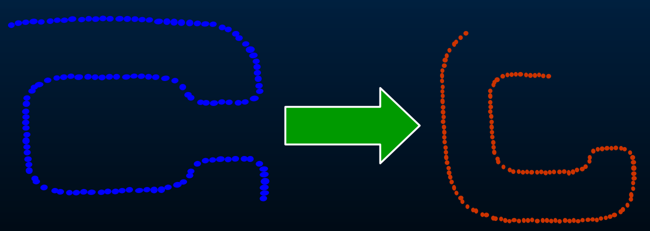
Problem variants



- Infinite point sets (curves & surfaces)



- Non-Euclidean transformations



- Incomplete data

The strategy

Find T which minimizes error between transformed model and data

For each datum

$$\epsilon(T) = -\log P(T) = \sum_j \min_i \; d(T * \mathbf{M}_i, \mathbf{D}_j)$$

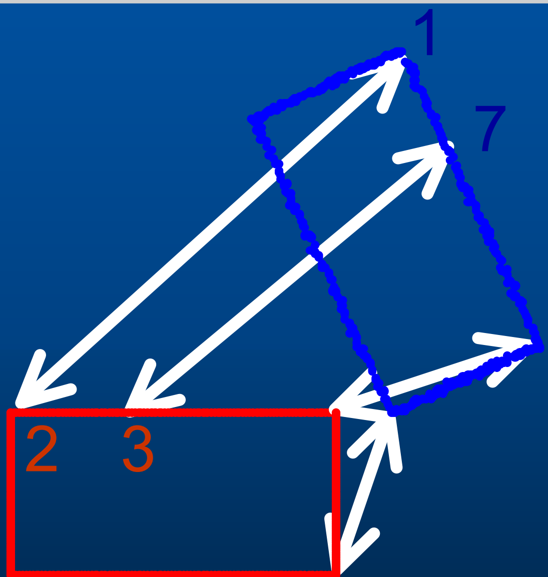
Distance to model

Where:

- $d(\mathbf{x}, \mathbf{y})$ is a distance between points $\mathbf{x}$ and $\mathbf{y}$.
- $T * \mathbf{x}$ applies the transformation to $\mathbf{x}$
  e.g. $T = (\theta, t_x, t_y)$ for 2D

$$T * \mathbf{x} = \begin{pmatrix} x \cos\theta + y \sin\theta + t_x \\ -x \sin\theta + y \cos\theta + t_y \end{pmatrix}$$

Known Correspondences



1 – 2

7 – 3

...

Hard:

$$\epsilon(T) = \sum_j \min_i d(T * \mathbf{M}_i, \mathbf{D}_j)$$

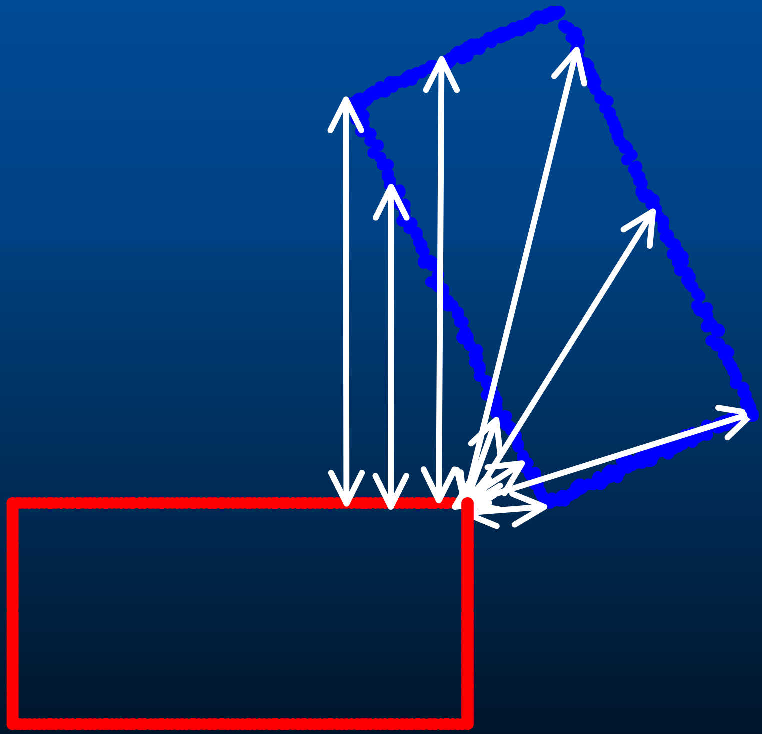
Easy:

Given *correspondences* $j \leftrightarrow \phi(j)$

Can minimize

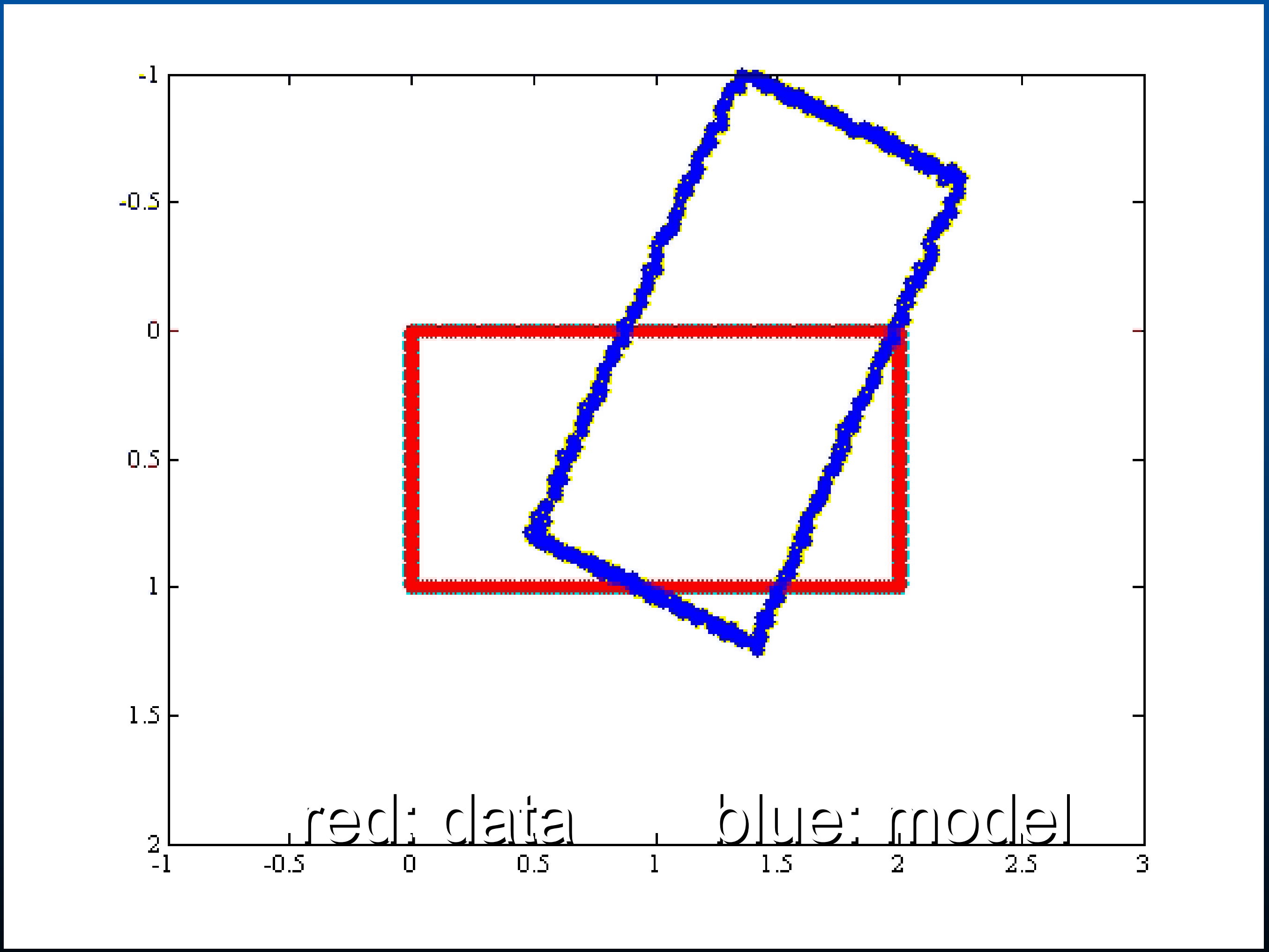
$$\sum_j d(T * \mathbf{M}_{\phi(j)}, \mathbf{D}_j)$$

But we don't know the correspondence



- That's OK, just choose the closest point...
- *Of course* it's wrong, but it will get us closer

Iterate these steps: ICP



Common problems with ICP

- ICP inaction
  - Slow convergence: let's see why
- Difficult to extend to include:
  - Robustness
    - M-estimation
  - Constraints
    - translation limits
  - *A-priori* information
    - priors on projective transformations

Convergence: ICP as optimization

EM-like version: task is to minimize over T and ϕ

$$-\log P(T;\phi) = \sum_j d(T * \mathbf{M}_{\phi(j)}, \mathbf{D}_j)$$

Fix T, compute $\{\phi(j)\}_{j=1}^{n}$:

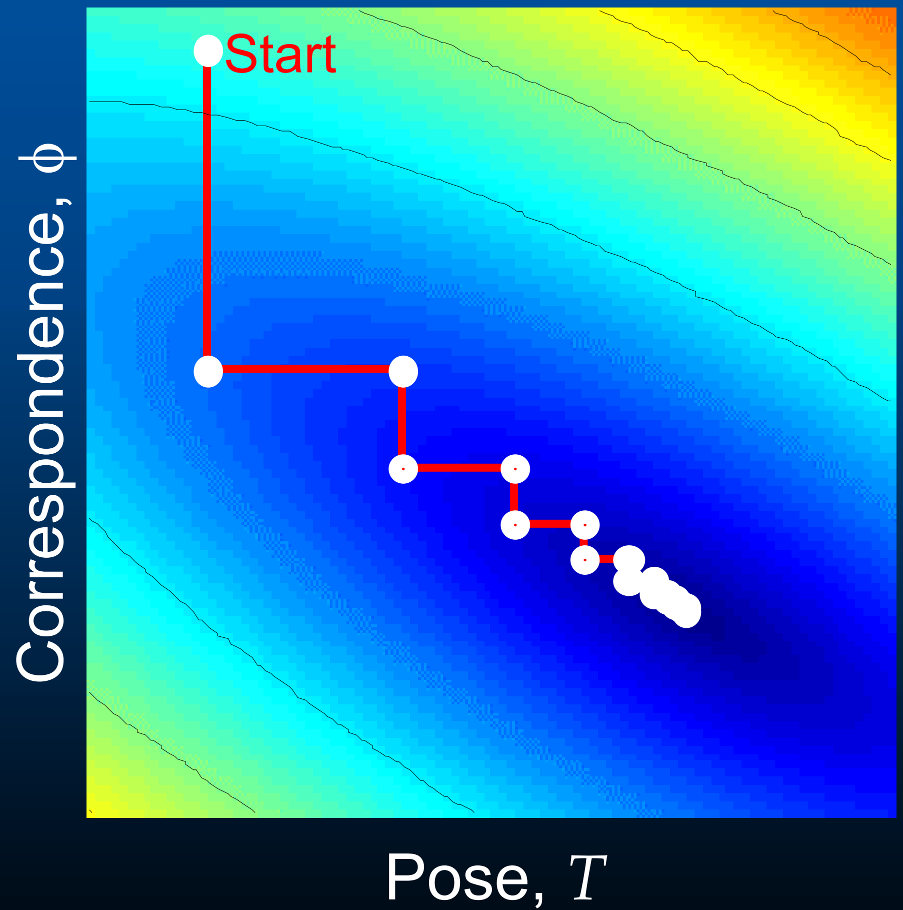
$$\phi(j) = \operatorname*{argmin}_i d(T * \mathbf{M}_i, \mathbf{D}_j)$$

Fix $\{\phi(j)\}_{j=1}^{n}$, compute T

$$T = \operatorname*{argmin}_T \sum_j d(T * \mathbf{M}_{\phi(i)}, \mathbf{D}_j)$$

ICP as optimization

Error is a function of *correspondence* and *pose* parameters



My proposal: LMICP

Insert our cost function $\epsilon(T)$ into a standard nonlinear optimizer...

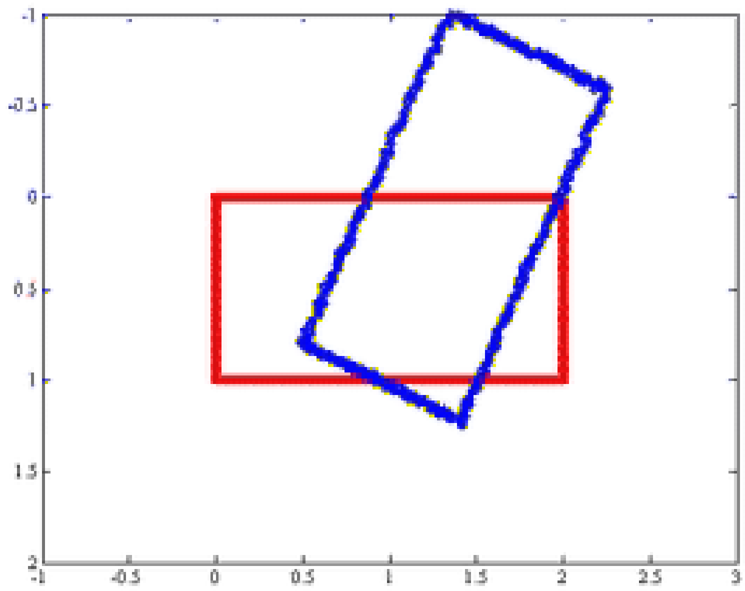
E.g. a Levenberg-Marquardt implementation such as Matlab's `lsqnonlin`

Don't use Numerical Recipes

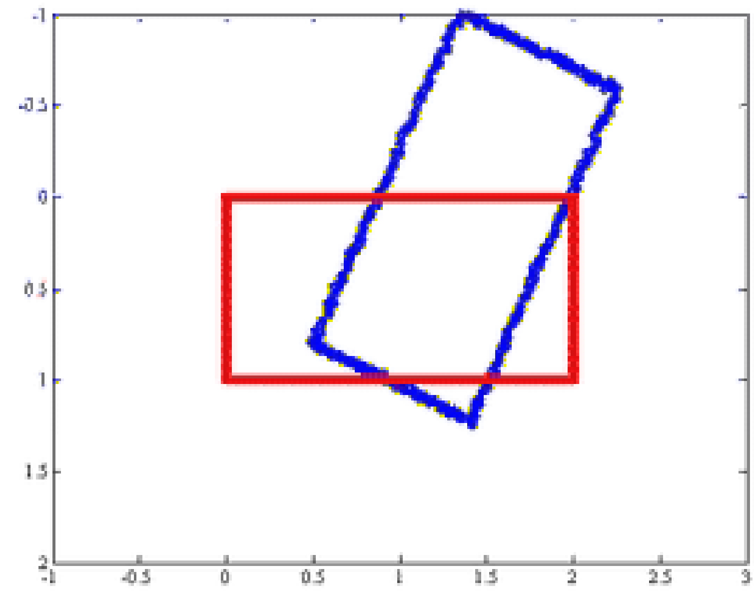
Advantages

- Obvious:
  - Easier to code
  - Easier to modify
- Nonobvious:
  - Runs faster
  - Wider basin of convergence

Example



ICP



LM-ICP

But what about derivatives?

Need to compute derivatives for nonlinear optimization

Option 1: [For home use only, will run slowly]

Finite differences for function $f(\mathbf{x})$

$$\left.\frac{df}{dx_k}\right|_{\mathbf{x}} = \lim_{h \to 0} \frac{f(\mathbf{x} + h\mathbf{e}_k) - f(\mathbf{x})}{h}$$

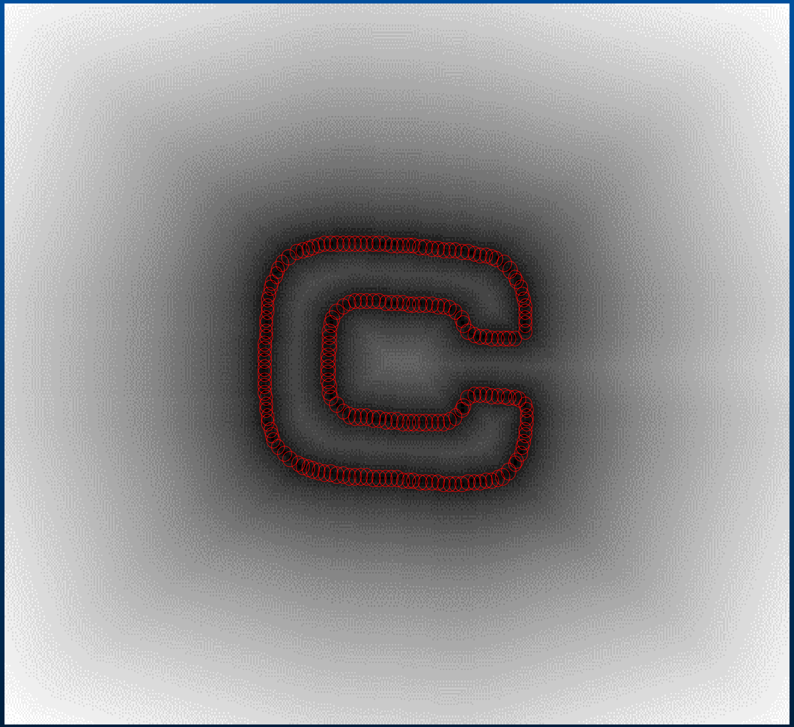
Option 2: [Later, let's speed it up first]

Speeding it up

As described, each iteration costs several closest-point computations.

These don't need to be accurate, so pre-compute the Distance Transform and query from that...

The Distance Transform



Discrete cache of distances to data.

$$L(x, y) = \min_j d(\mathbf{D}_j, (x, y))$$

So cost function is

$$\epsilon(T) = \sum_i L(T * \mathbf{M}_i)$$

Derivatives: Option 2

With distance transform, cost function is

$$\epsilon(T) = \sum_i L(T * \mathbf{M}_i)$$

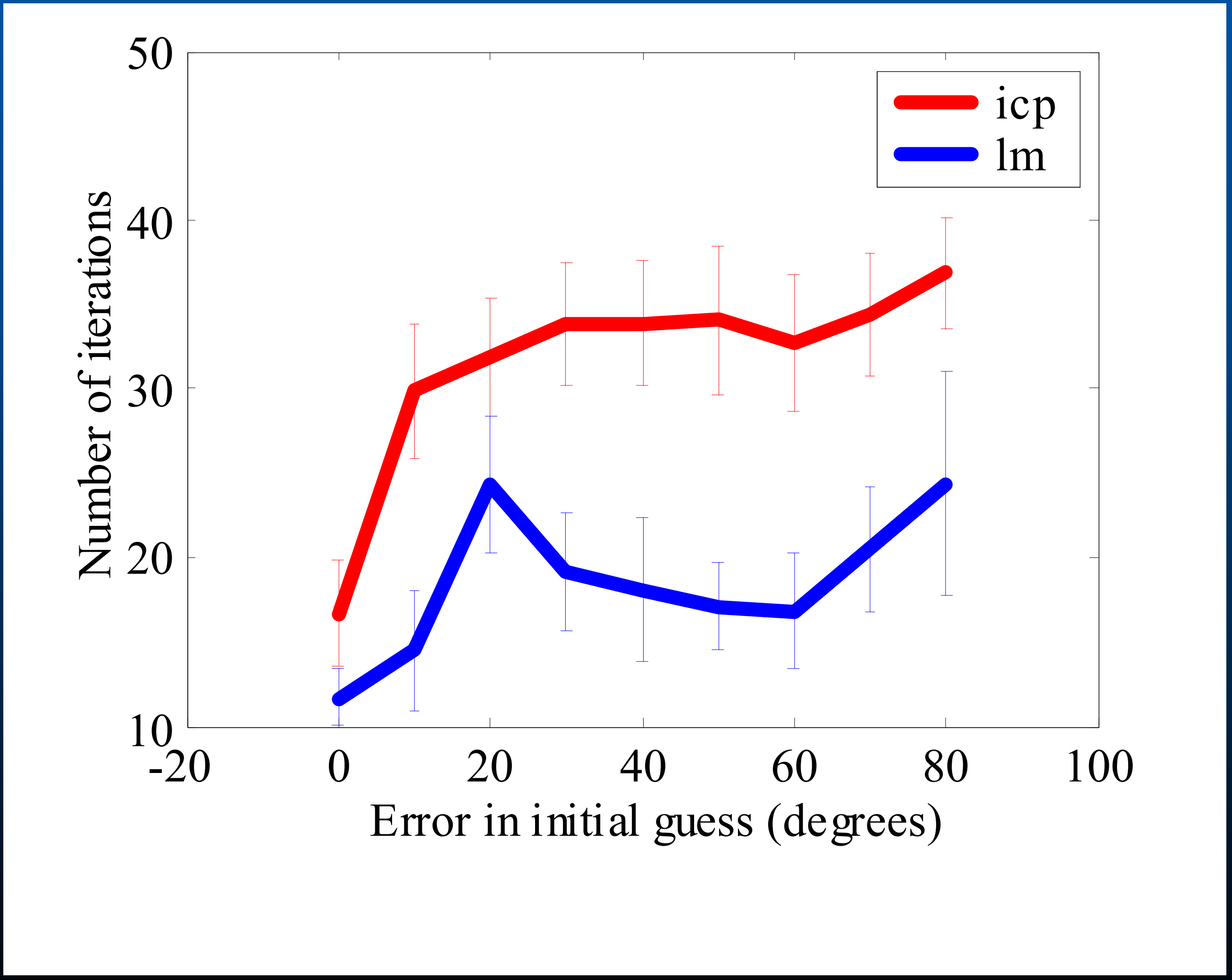
Differentiating wrt one param T_k, chain rule gives

$$\frac{d\epsilon(T)}{dT_k} = \sum_i \quad L_x(T * \mathbf{M}_i)\frac{d}{dx}T * \mathbf{M}_i + L_y(T * \mathbf{M}_i)\frac{d}{dy}T * \mathbf{M}_i$$

And L is an image, so we all know how to compute L_x and L_y

Performance: speed

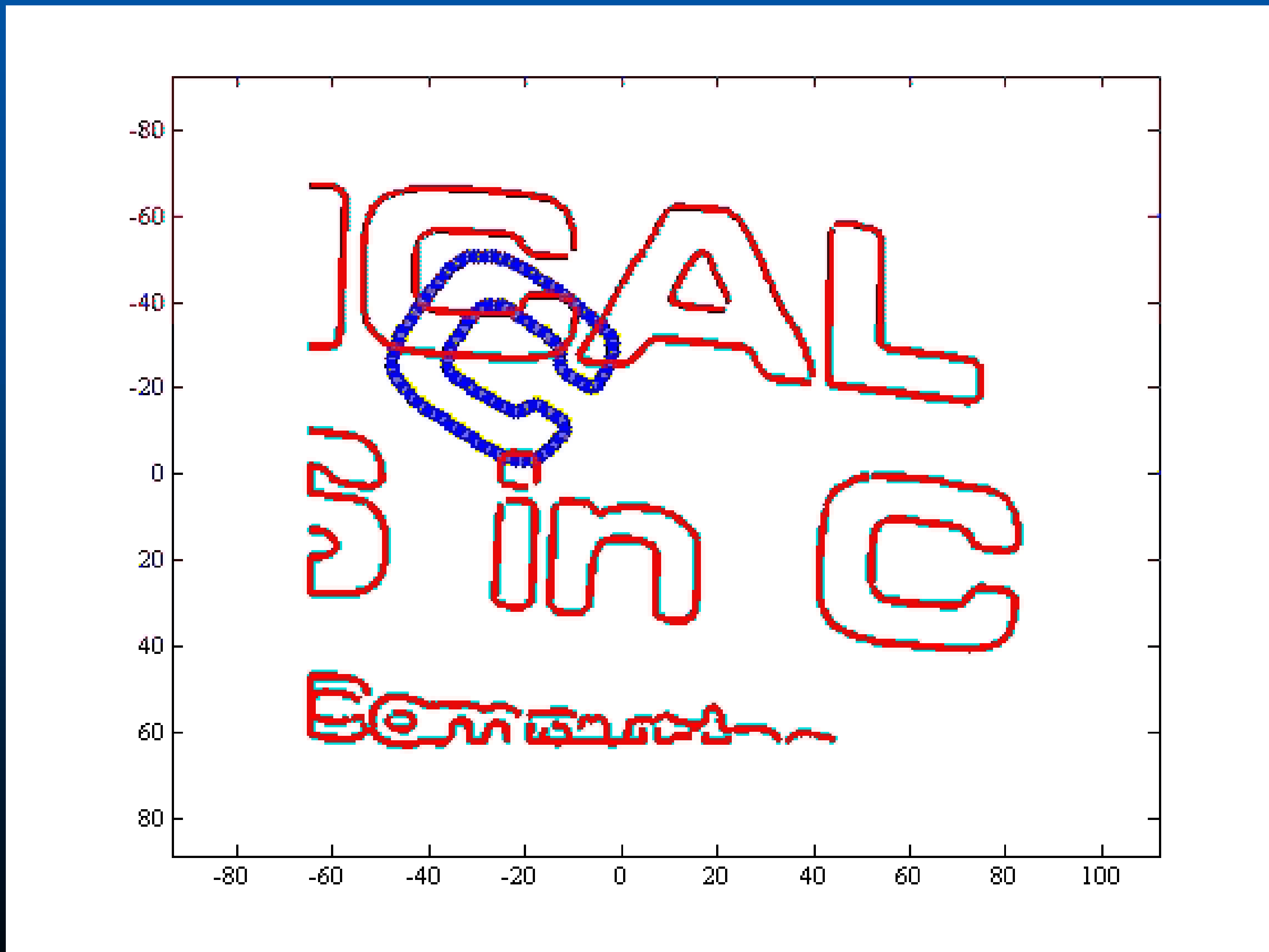
- Box-box registrat ion, 400 points



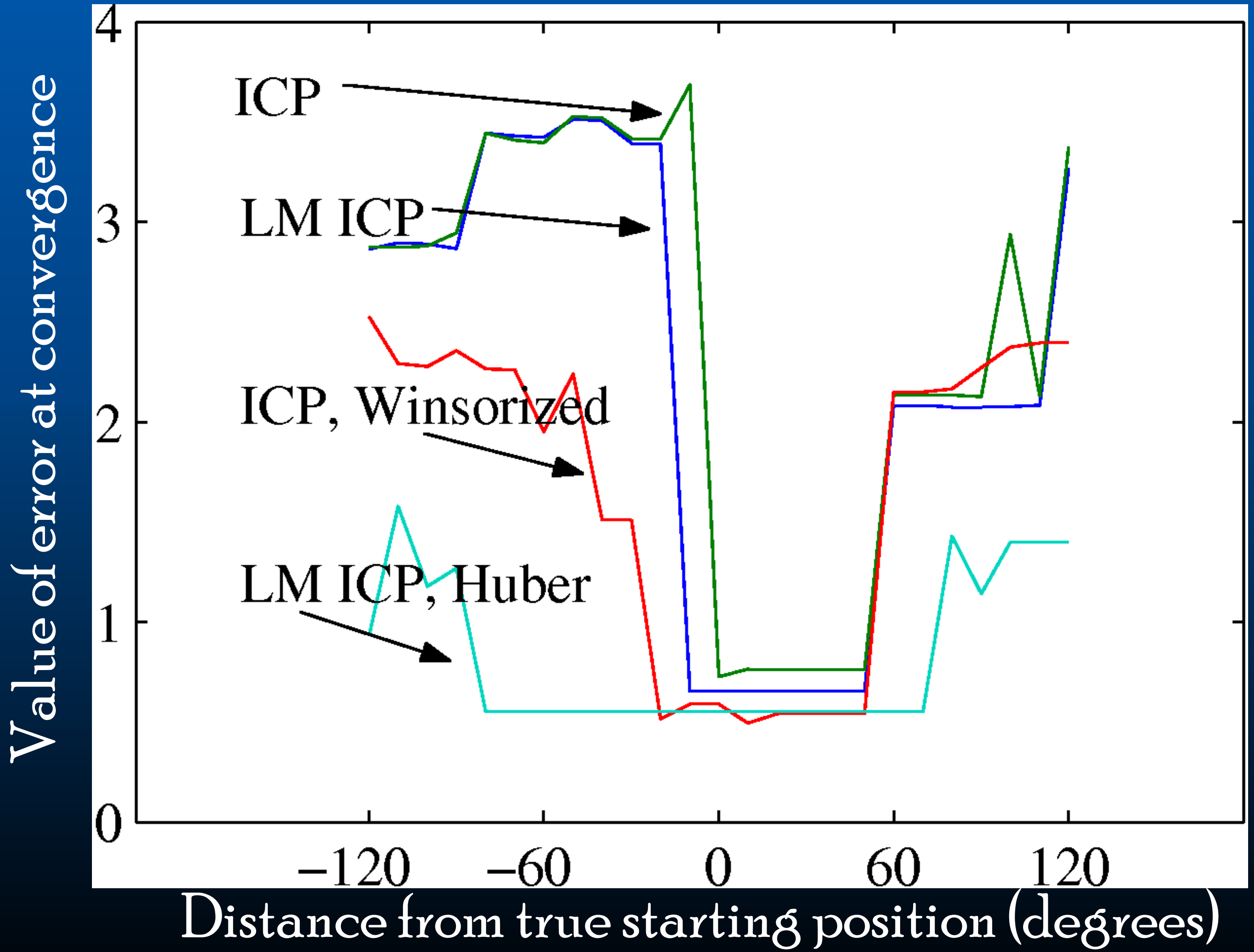
Robustness: Using an M-estimator

- Need robustness when data are
  - riddled with outliers
  - not a complete subset of model
    - (e.g. sampled model)
- ICP: Requires iteration at inner loop
  - Very expensive
- LM: Trivial to add to cost function
  - Distance transform easily modified

Performance: Examples



Performance: radius of



Conclusion

- LMICP is faster, more accurate, has a wider basin of convergence, is easier to code, easier to extend.
- ICP is easier to understand.
- ICP is slow because it hasn't had the benefit of 40 years of numerical analysis

Section 2.3.1
Chamfer Distance

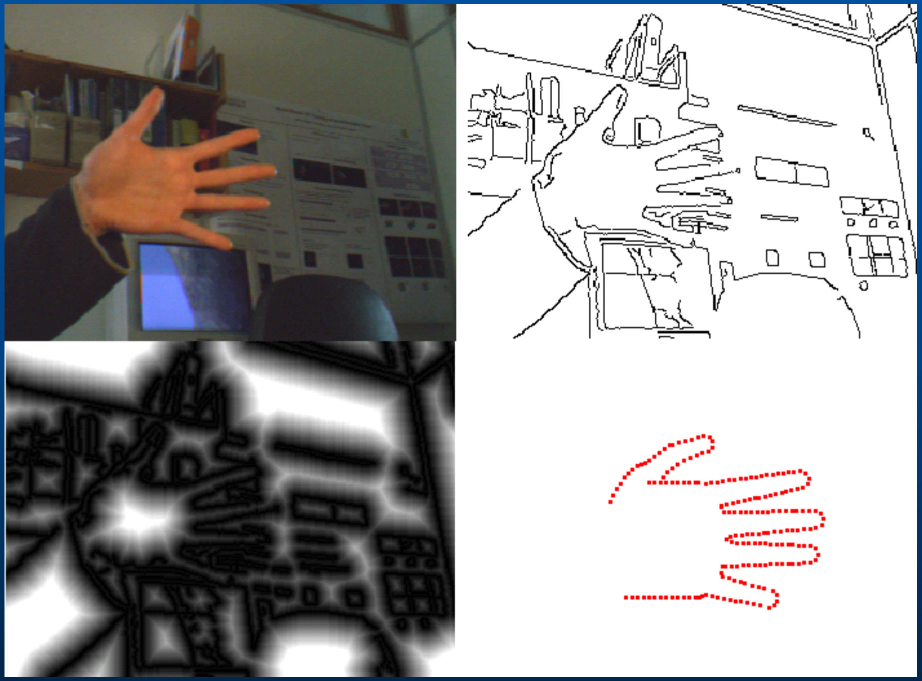
Thanks to

Arasanathan Thayananthan

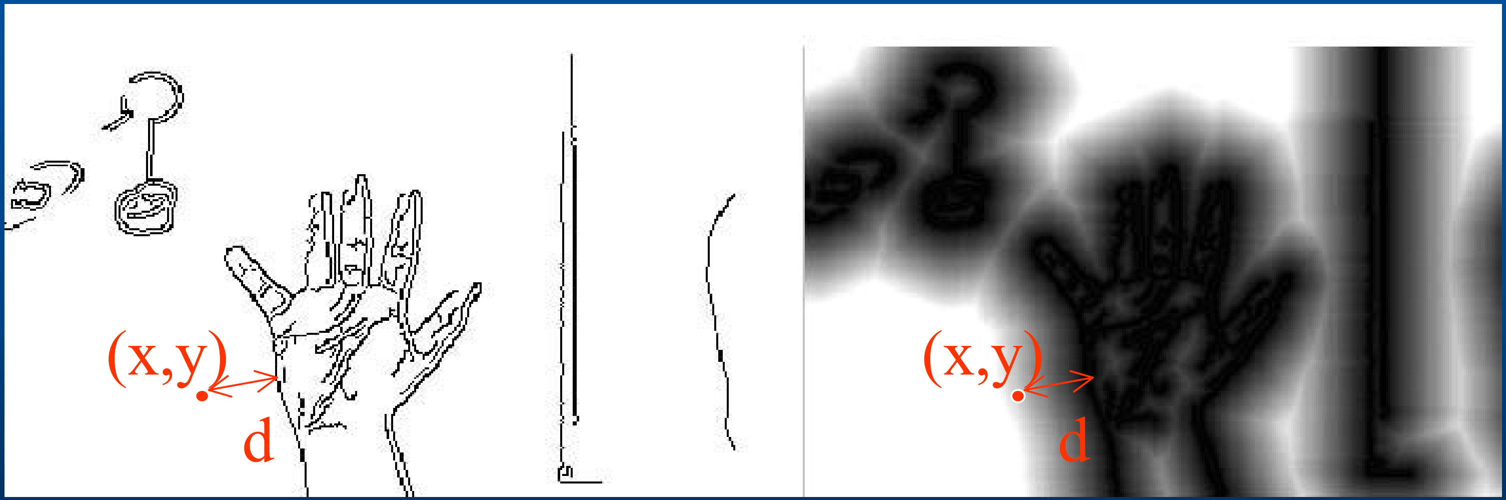
Bjorn Stenger

Chamfer Distance

- Left: Camera image
- Right: Canny edge map
- Left: Distance Transform of the canny edge map
- Right: Search templates (150-250 points)

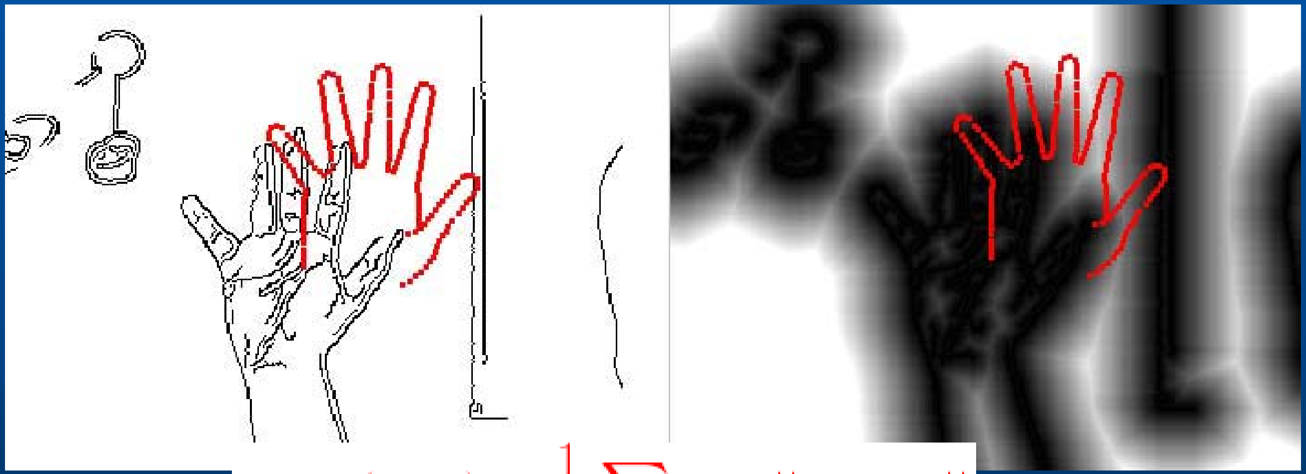


Chamfer Distance



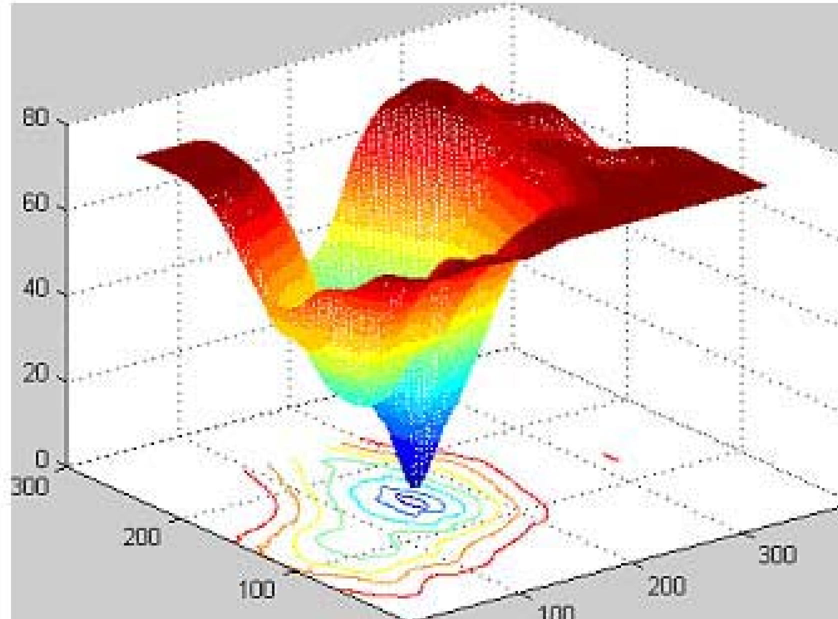
- Distance Image gives the distance to the nearest edge feature at every pixel location in the image.
- Calculated only once for each frame.

Chamfer Matching

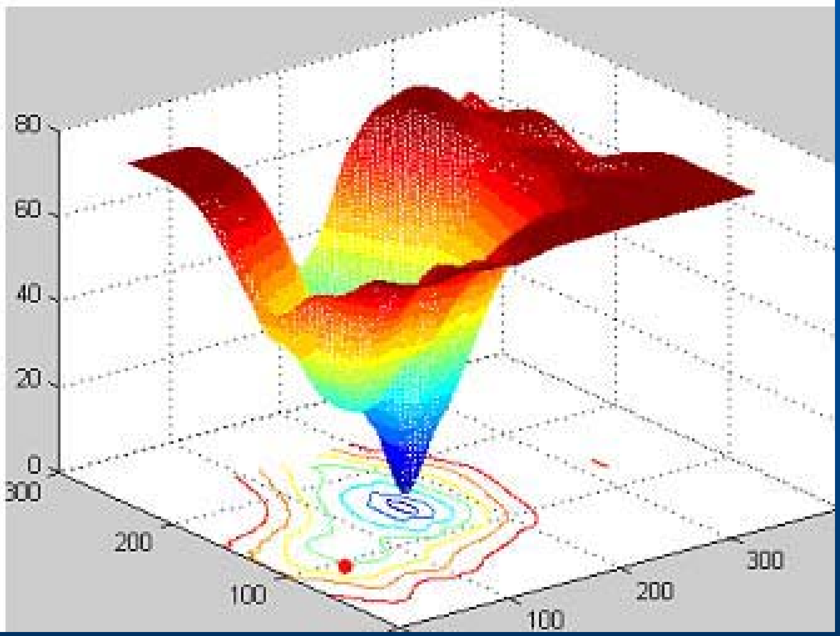


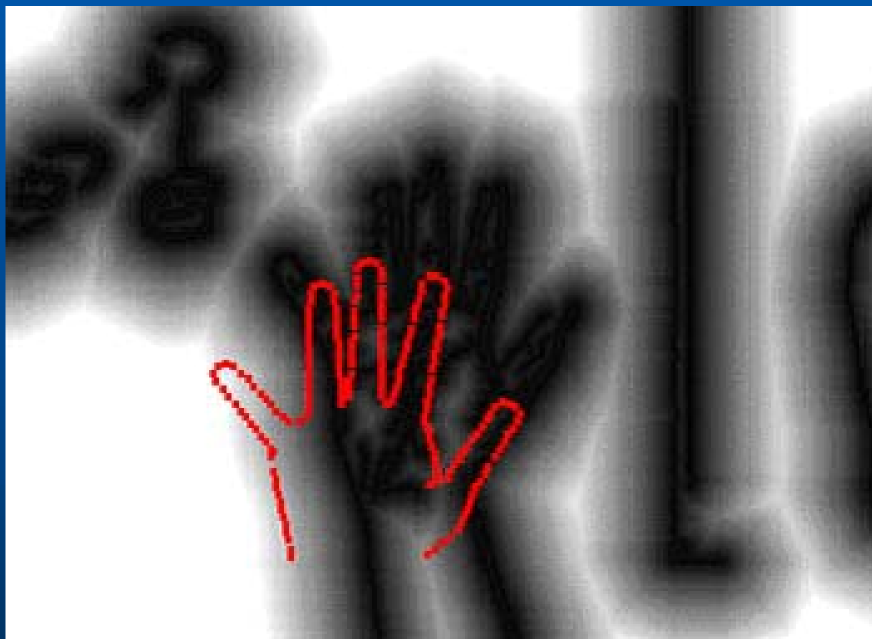
$$d_{cham}(\mathcal{U},\mathcal{V}) = \frac{1}{n}\sum_{u_i\in\mathcal{U}} \min_{v_j\in\mathcal{V}} ||u_i - v_j||$$

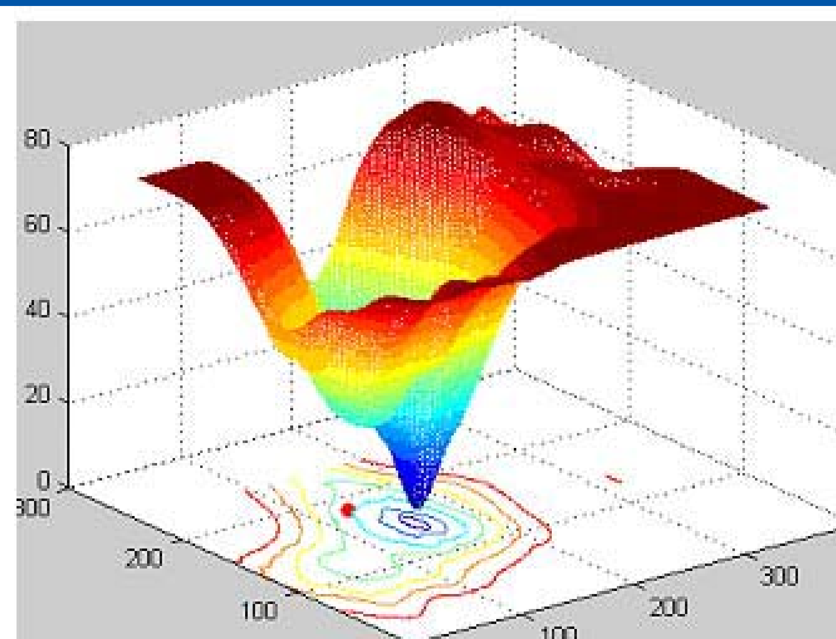
- The chamfer score is the average nearest distance from templates points to image points.
- The nearest distances are readily obtained from the distance image.
- Computationally inexpensive.

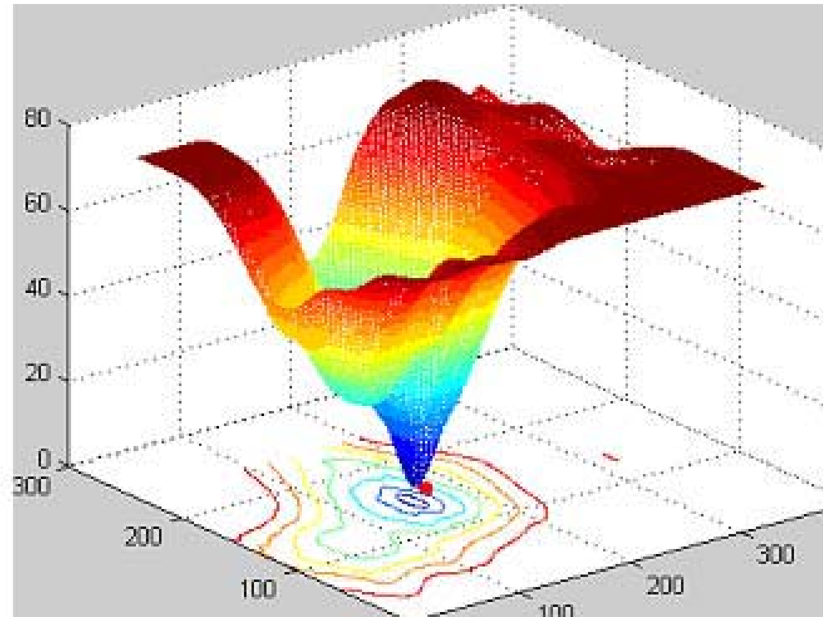


- Distance Image provides a smooth cost function.
- Efficient Searching techniques can be used to find the correct template.

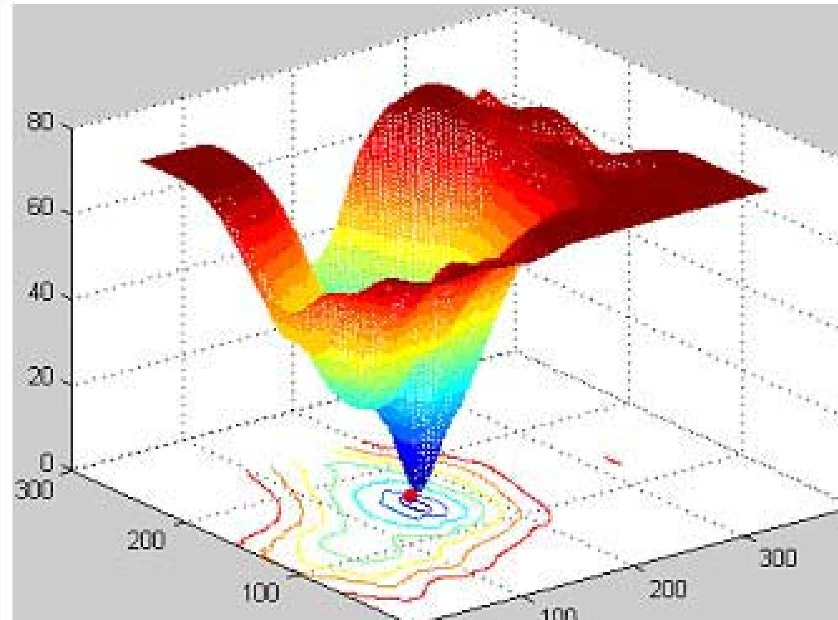

80
60
40
20
0
300
200
100
300
200
100

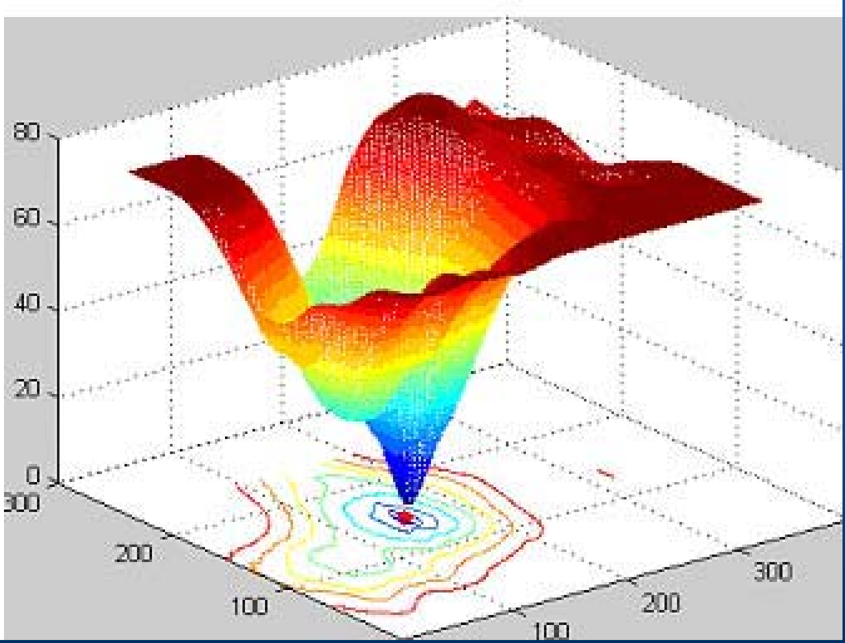



80
60
40
20
0
200
100
300
200
100



80
60
40
20
0
300
200
100
300
200
100

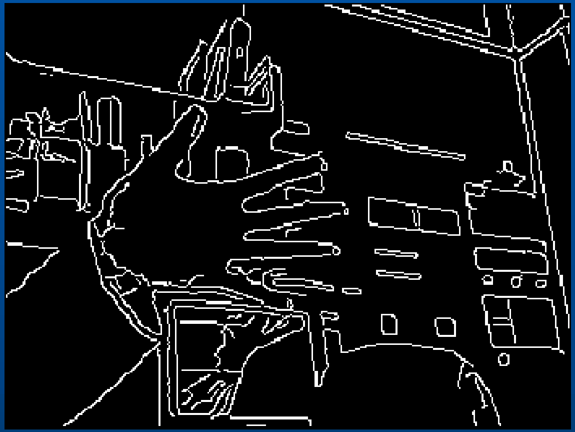

80
60
40
20
0
300
200
100
300
200
100

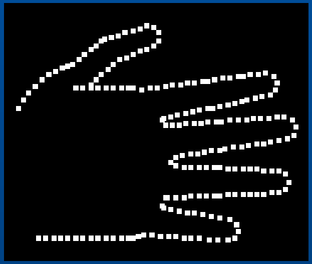


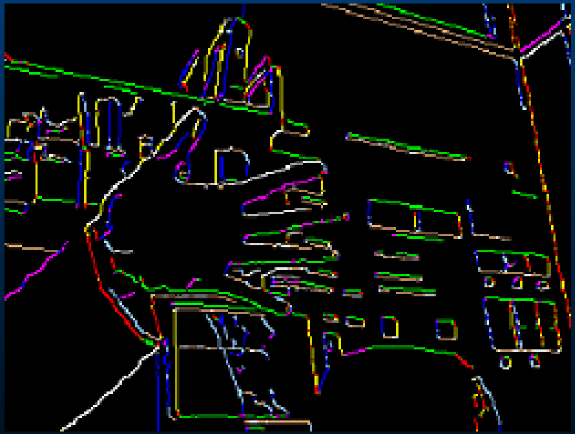
80
60
40
20
0
300
200
100
300
200
100

Multiple Edge Orientations

- Similar to Gavrila, Edge pixels are divided into 8 groups based on orientation
- Distance Transforms are calculated separately for each group
- Total matching score is obtained by adding individual chamfer scores







Applications: Hand Detection

- Initializing a hand model for tracking
  - Locate the hand in the image
  - Adapt model parameters
  - No skin color information used
  - Hand is open and roughly fronto-parallel

Results: Hand Detection

Original Shape Context

Shape Context with Continuity Constraint

Chamfer Matching

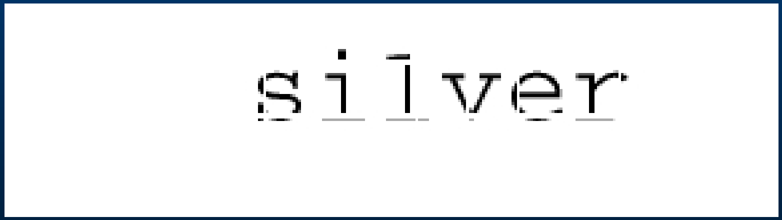
Applications: CAPTCHA

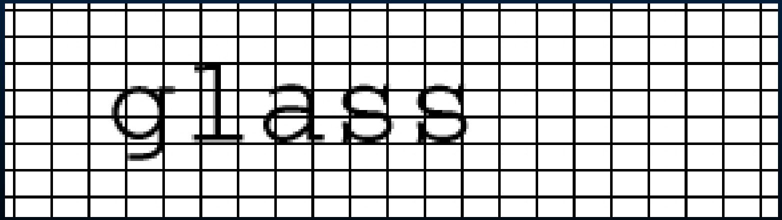
- C A P T
  C H A [Blum *et al.*, 02]

- Used in e-mail sign up for Yahoo accounts

-

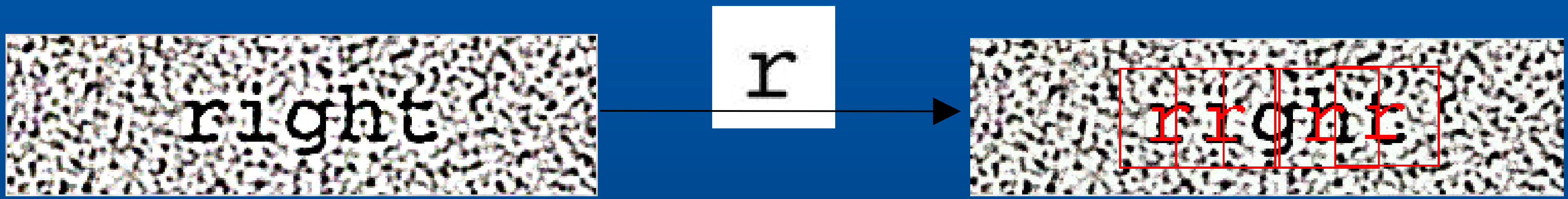
Examples:





EZ-Gimpy results

Chamfer cost for each letter template



Word matching cost: average chamfer cost
+ variance of distances

Top 3 matches (dictionary 561 words)

right 25.34

fight 27.88

night 28.42

93.2% correct matches using 2 templates per letter

Shape context 92.1% [Mori & Malik, 03]

Additional slides

THANKS!!!

- F. Dellaert, A. Fitzgibbon, J. Matas, P. Perez, B. Stenger, A. Thayananthan, B. Tordoff, Lexing Xie,

References

- Frank **Dellaert**, Steven M. Seitz, Charles E. Thorpe, and Sebastian Thrun. *Structure from motion without correspondence*. CVPR. IEEE Press, June 2000
- U. **Grenander**. *Elements of Pattern Theory*. Johns Hopkins University Press, New Baltimore, 1996.

References

- **M**.C. Burl, **M. Weber**, and P. Perona. *A probabilistic approach to object recognition using local photometry and global geometry*. In ECCV'98, pages 628--641, 1998.
- **M. Weber**, **M**. Welling, and P. Perona. *Unsupervised learning of models for recognition*. In Proc. 6th ECCV, Dublin, Ireland, to appear June 2000.

References

- B. **Tordoff** and D.W. Murray. *Guided sampling and consensus for motion estimation*. In Proc 7th European Conf on Computer Vision, Copenhagen, June 2002.
- **K**. **Kanatani**, *Geometric information criterion for model selection*, Int. J. Comp. Vision (1998), 26(3), 171--189.

References

- Jiri **Matas**, Ondrej Chum, Martin Urban, and Tomas Pajdla. *Robust wide baseline stereo from maximally stable extremal regions*. In Proceedings of British Machine Vision Conference, September 2002.
- **Z. Zhang**, R. Deriche, O. Faugeras, and Q. Luong. *A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry*. Artificial Intelligence, 78:87--119, 1995.

References

- P. H. S. **Torr** and D. W. Murray. *The development and comparison of robust methods for estimating the fundamental matrix*. Intl. J. of Computer Vision, 24(3):271--300, 1997
- P. Beardsley, P. **Torr**, and A. Zisserman. 3D *model acquisition from extended image sequences*. In Proc. European Conf. on Computer Vision, Vol. 2, pages 683--695, 1996

References

- P. H. S. **Torr** and A. Zisserman, "*MLESAC: A new robust estimator with application to estimating image geometry*," CVIU, vol. 78, pp. 138--156, 2000.
  P.H.S. Torr. Bayesian Model Estimation and Selection for Epipolar Geometry and Generic Manifold Fitting. In International Journal of Computer Vision, 50(1), pages 27—45, 2002.

References

- B. Stenger, A. Thayananthan, P.H.S. Torr, and R. Cipolla. *Bayesian Tracking using Tree-Based Density Estimation*, Accepted Ninth International Conference on Computer Vision, 2003.
- A. Thayananthan, B. Stenger, P.H.S. Torr, and R. Cipolla. *Shape Context and Chamfer Matching in Cluttered Scenes*. In Conference of Computer Vision and Pattern Recognition, pages 127-133, 2003.

References

- Rob Fergus, Pietro Perona, Andrew Zisserman, *Object class recognition using unsupervised scale-invariant learning CVPR,* 2003*.*
- **Fischler**, M.A., and R.A. **Elschlager**, "*The Representation and Matching of Pictorial Structures*," IEEE Trans. on Comp. 22(1), pp. 67--92 (January 1973).

References

- P.H.S. Torr and C. Davidson. *IMPSAC:* In *ECCV* pages 819—833, 2000.
- P.J **Rousseeuw** and A.M. Leroy. *Robust Regression and Outlier Detection*. J. Wiley, New York, New York, 1987.
- C. Stewart. MINPRAN: A new robust estimator for computer vision. IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(10):925--938, October 1995.

References

- P.H.S. Torr, A. Dick, and R. Cipolla. Layer Extraction with a Bayesian Model of Shapes. In *The Sixth European Conference on Computer Vision*, pages 273—289, 2000.
- A. R. Dick, P.H.S. Torr, and R. Cipolla. Automated 3D Modelling of Architecture. In *Proceedings British Machine Vision Conference,* pages 273—289, 2000.
- A. R. Dick, P.H.S. Torr, and R. Cipolla. Combining Recognition and Structure from Motion. In *IEEE Eighth International Conference on Computer Vision,*, pages 268—274, 2001.

References

- **Action Recognition by Shape Matching to Key Frames** S. Carlsson and J. Sullivan Workshop on Models versus Exemplars in Computer Vision, Kauai, Hawaii, USA December 14th, 2001
- M. A. **Fischler** and R. C. Bolles. *Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography*. Comm. Assoc. Comp. Mach., 24(6):381--395, 1981.

References

- C.V. Stewart, C.-L. Tsai and B. Roysam, "The Dual-Bootstrap Iterative Closest Point algorithm with application to retinal image registration", Department of Computer Science Technical Report RPI-CS-TR 02-9, June 2002.

(this is an interesting paper in which the performance of ICP is improved by not considering all the points in any iteration).

References

- G. Borgefors, "*Hierarchical **chamfer** matching: A parametric edge matching algorithm*," PAMI, vol. 10, pp. 849--865, Nov. 1988.
- G. Barrow et al., "*Parametric Correspondence and **Chamfer** Matching: two new techniques for image matching*," Proc. IJCAI, vol.2, pp.659-663, 1977.