Chapter 9 Localization and Mapping

Part 2

9.2 Visual Localization and Motion Estimation

Outline

- 9.2 Visual localization and Motion Estimation
  - 9.2.1 Introduction
  - 9.2.2 Aligning Signals for Localization and Motion Estimation
  - 9.2.3 Matching Features for Localization and Motion Estimation
  - 9.2.4 Searching for the Optimal pose
  - Summary

Outline

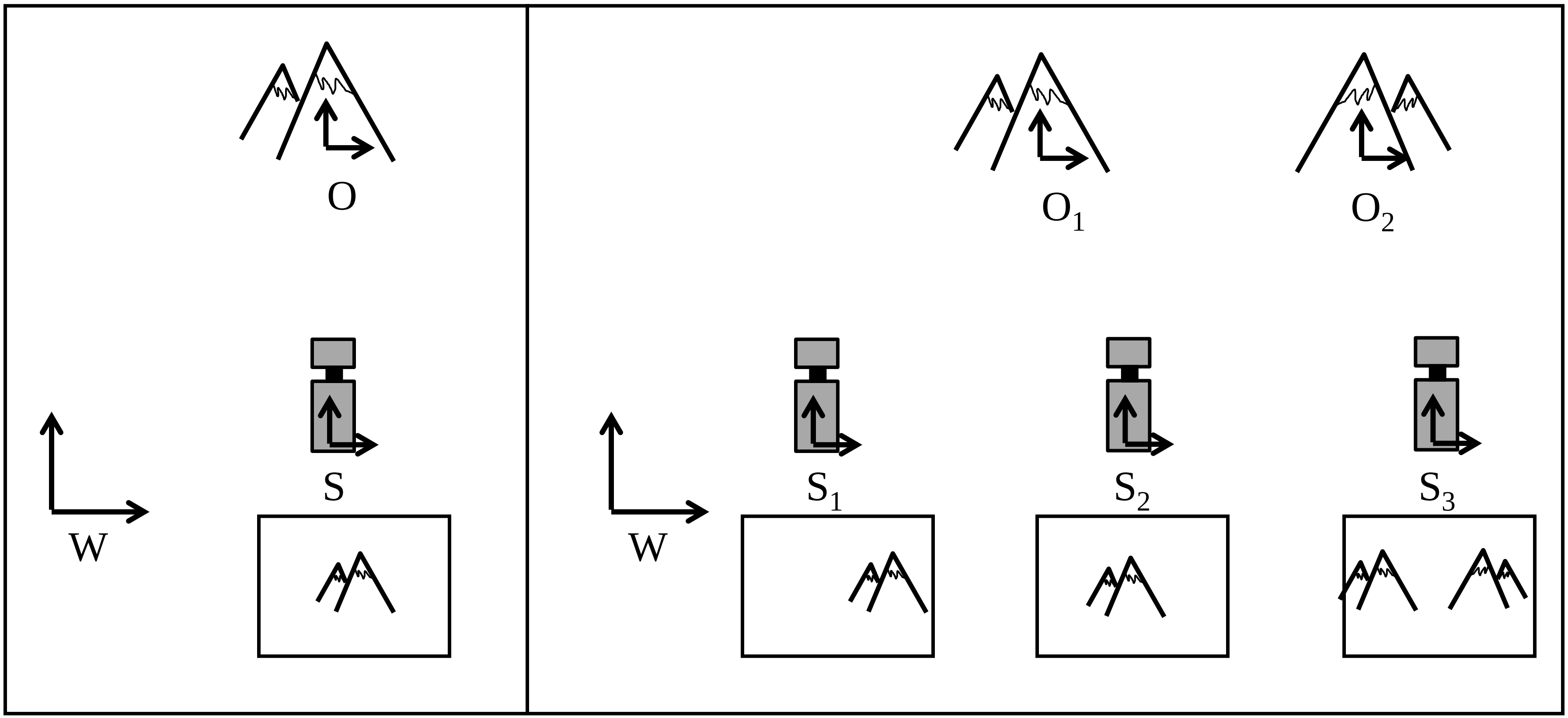
- 9.2 Visual localization and Motion Estimation
  - 9.2.1 Introduction
  - 9.2.2 Aligning Signals for Localization and Motion Estimation
  - 9.2.3 Matching Features for Localization and Motion Estimation
  - 9.2.4 Searching for the Optimal pose
  - Summary

Overall Framework

- Three related problems.
  - Localize robot based on a map
  - Measuring motion based on imagery
  - Measuring object positions from imagery
- Last two can be combined to construct maps.

9.2.1 Introduction

- All of the mechanisms we will consider are summarized in this figure.



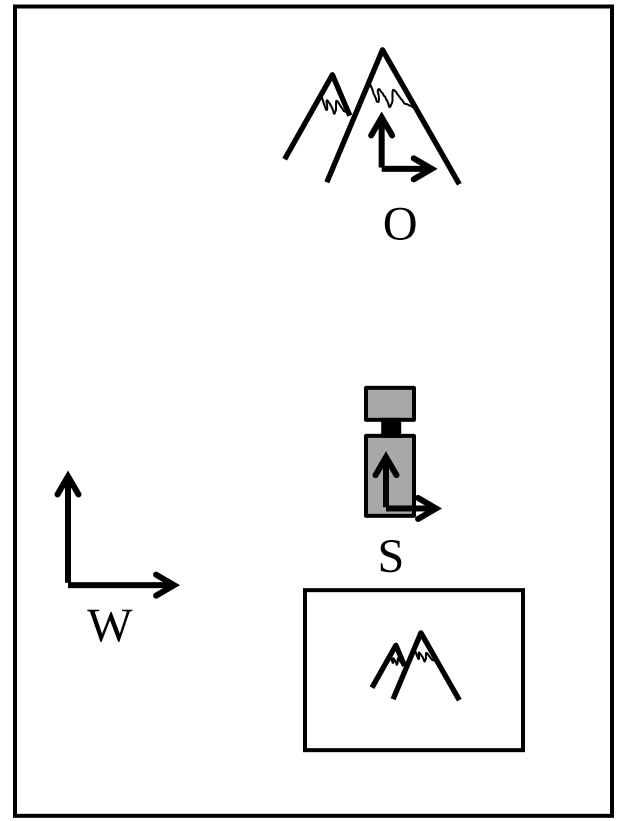
9.2.1.1 Canonical Problems
(Mapping and Localization)

- Simple Mapping

$$\underline{\rho}_O^W = \underline{\rho}_S^W * \underline{\rho}_O^S$$

- Localization

$$\underline{\rho}_S^W = \underline{\rho}_O^W * \underline{\rho}_S^O$$



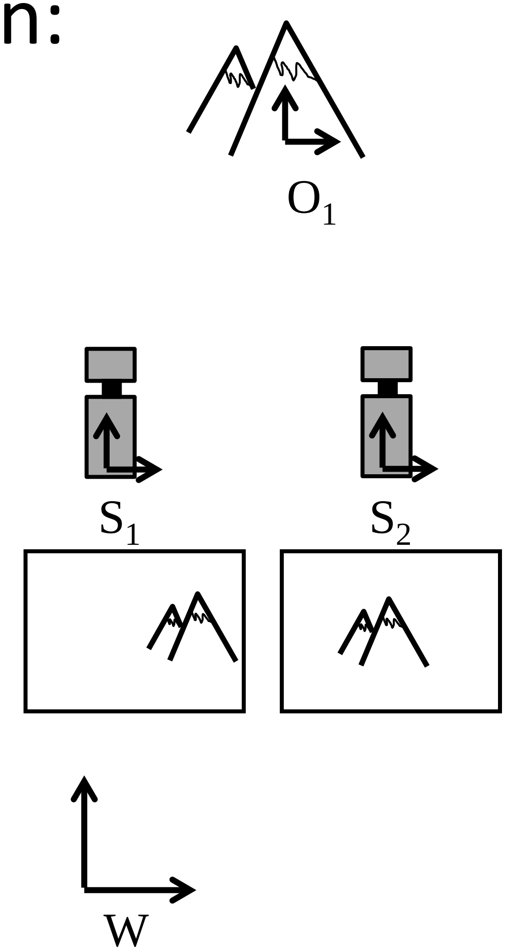
Carnegie Mellon
THE ROBOTICS INSTITUTE

9.2.1.1 Canonical Problems

(Motion Estimation)

- Motion Estimation
- Observe object O1 twice, then:

$$\underline{\rho}_{S2}^{S1} = \underline{\rho}_{O1}^{S1} * \underline{\rho}_{S2}^{O1}$$



9.2.1.1 Canonical Problems
(SLAM)

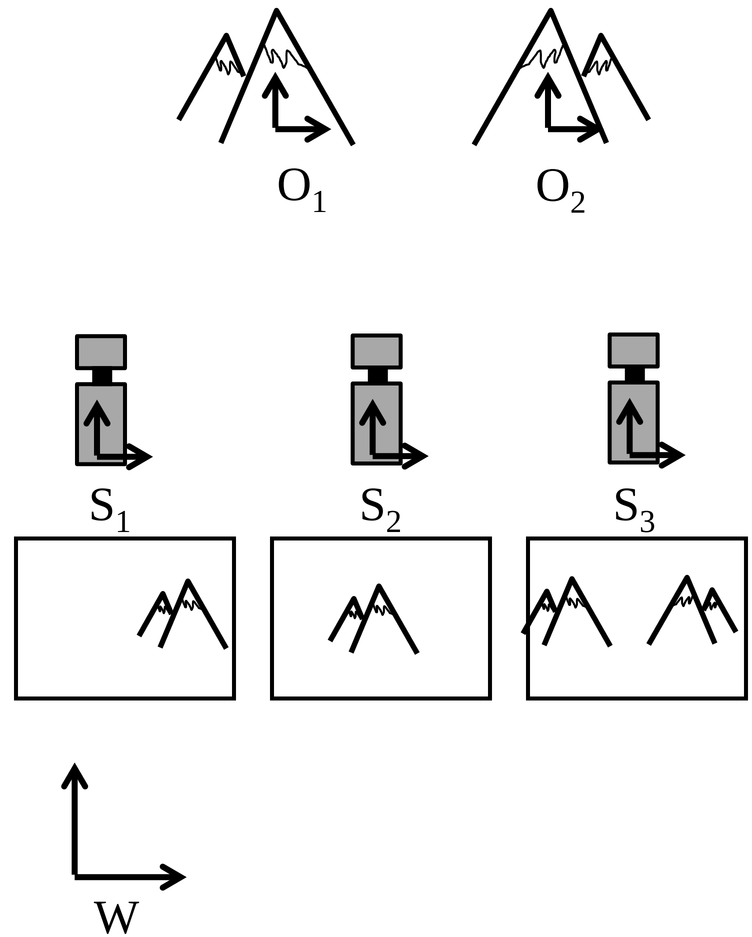
- Mapping
- Also Observe object O2 and then:

$$\underline{\rho}_{S3}^{S2} = \underline{\rho}_{O2}^{S2} * \underline{\rho}_{S3}^{O2}$$

- Treat O1 as origin:

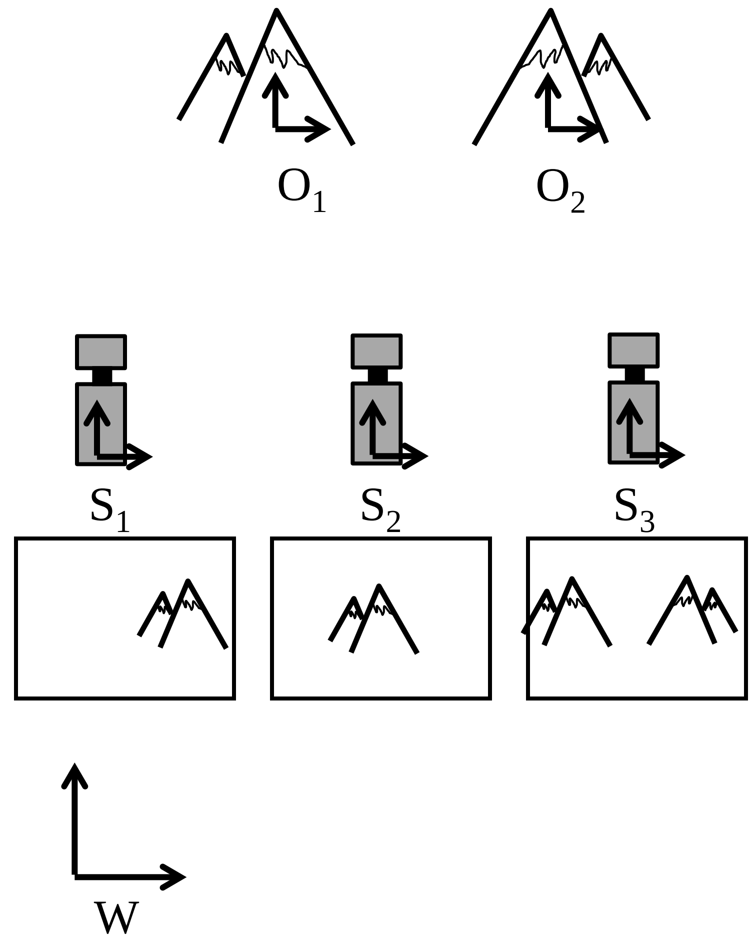
$$\underline{\rho}_{O2}^{O1} = \underline{\rho}_{S1}^{O1} * \underline{\rho}_{S2}^{S1} * \underline{\rho}_{O2}^{S2}$$

- Etc.
- Note how error accumulates.



9.2.1.1 Canonical Problems
(Consistent Mapping)

- Suppose robot sees O2 at step 100.
  - Calls it O100.
  - Also $\underline{\rho}_{O2}^{O1} \neq \underline{\rho}_{100}^{O1}$
- Have to go back and fix all of $\underline{\rho}_{S2}^{S1}$ through $\underline{\rho}_{S100}^{S99}$



9.2.1.2 Visual Localization

- Compare what robot sees to what it expects to see.
- GPS is an example where "perception" sensor is a multi-channel radar.

- Nomenclature:
  - Scene = the real world
  - Image = pixels in a computer

9.2.1.2.1 Image Formation

- When the pose of the object with respect to the sensor ($\underline{\rho}_O^S$) is known, model frame points can be transformed into sensor frame points…

$$\underline{X}^s = T_o^s(\underline{\rho}_O^S)\underline{X}^m$$

Extrinsic Parameters

- For example, in detail, this may be…

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}^s = \begin{bmatrix} c\psi c\theta & (c\psi s\theta s\phi - s\psi c\phi) & (c\psi s\theta c\phi + s\psi s\phi) & u \\ s\psi c\theta & (s\psi s\theta s\phi + c\psi c\phi) & (s\psi s\theta c\phi - c\psi s\phi) & v \\ -s\theta & c\theta s\phi & c\theta c\phi & w \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}^m$$

9.2.1.2.1 Image Formation

- Substituting the second result into the first gives:

$$\underline{x}_i = P\underline{X}^s = PT_o^s(\underline{\rho}_O^S)\underline{X}^m = T(\underline{\rho}_O^S)\underline{X}^m = \underline{h}(\underline{\rho}_O^S, \underline{X}^m)$$

- This complete model of a camera looking at an object tells us where points on the object (mode) appear in the image.

9.2.1.2.1 Image Formation

$$\underline{X}^s = T_o^s(\underline{\rho}_O^S)\underline{X}^m$$

- Use a camera projection matrix to see where the point falls on the image plane:

$$\underline{x}_i = P\underline{X}^s$$

$$\begin{bmatrix} x_i \\ y_i \\ z_i \\ w_i \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \frac{1}{f} & 0 & 1 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix}$$

Intrinsic Parameters

9.2.1.2.2 Localization of Objects

- The familiar measurement relationship has a few more arguments:

image formation process

object model

$$\underline{z}(\underline{x}) = \underline{h}(\underline{x}, \underline{\rho}_O^S, \underline{Z})$$

image

image coords

pose

- Image could be 640 X 480 color pixels or 1024 X 64 range pixels etc.

9.2.1.2.2 Localization of Objects
(Predicting Images)

- Key points:
  - 1:Object model can be defined as a signal $\underline{Z}(\underline{X})$ over scene coordinates:
  - 2: Image and scene coordinates are related by a low dimensional transformation.
  - 3: Once transform is known, entire image is predictable from the model
- That's what computer graphics is.....

9.2.1.2.2 Localization of Objects

(Predicting Images)

- Consider a color camera and suppose transform depends only on rel. pose $\underline{x} = T(\underline{\rho}_O^S)\underline{X}$
- Substituting into our model: $\underline{z}(\underline{x}) = \underline{z}[T(\underline{\rho}_O^S)\underline{X}] = \underline{Z}(\underline{X})$
- Imaging process copies information from scene to corresponding point in image.
- Give the transform $T(\underline{\rho}_O^S)$ and the model $\underline{Z}(\underline{X})$ <u>we know what colors to put where</u>.

9.2.1.2.3 Basic Approaches and Issues

(Basic Approaches)

- First: search for the pose that explains the image:

$$\underline{z}_{pred}(\underline{x}) = \underline{h}(\underline{x}, \underline{\rho}_O^S, \underline{Z})$$

- Second: search for the pose which aligns coordinates:

$$\underline{x}_{pred} = T(\underline{\rho}_O^S)\underline{X}$$

- A predictable set of issues arise ….

9.2.1.2.3 Basic Approaches and Issues
(Issues)

Data Association

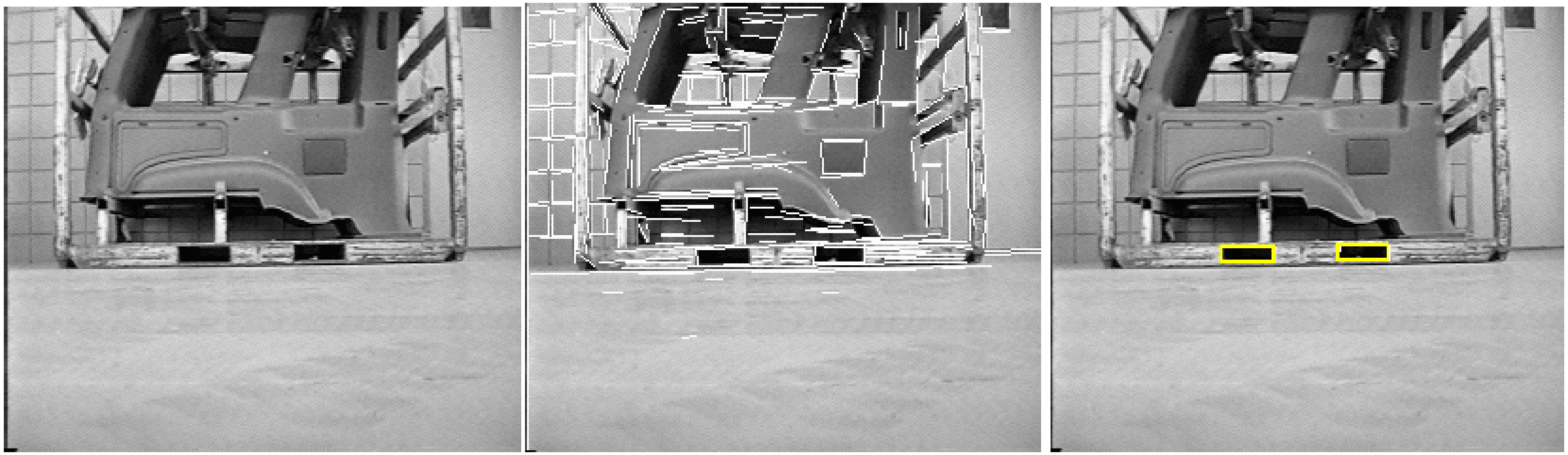
- MATCHING
- Which features are to be paired with which?

Equations

- ALIGNMENT
- Is the solution pose unique?
- Have initial estimate?
- How good is the data?
- How much time/computing available?

We have solved these problems with Kalman Filters already

9.2.1.2.5 Feature Example: Find The Pallet



- Reduce image to intensity edges.
- Match edges to model of fork holes.
- Find the pose.

Video

Carnegie Mellon
THE ROBOTICS INSTITUTE

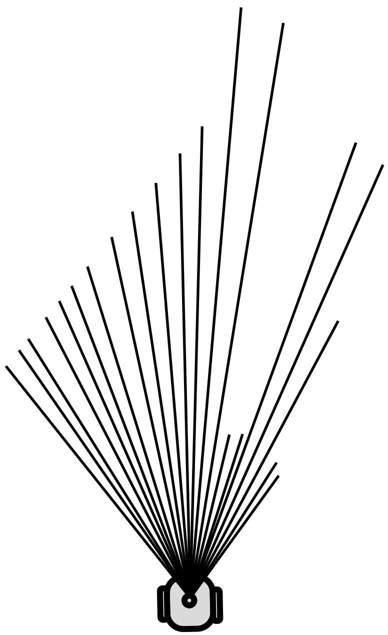
9.2.1.2.5 Robot localization

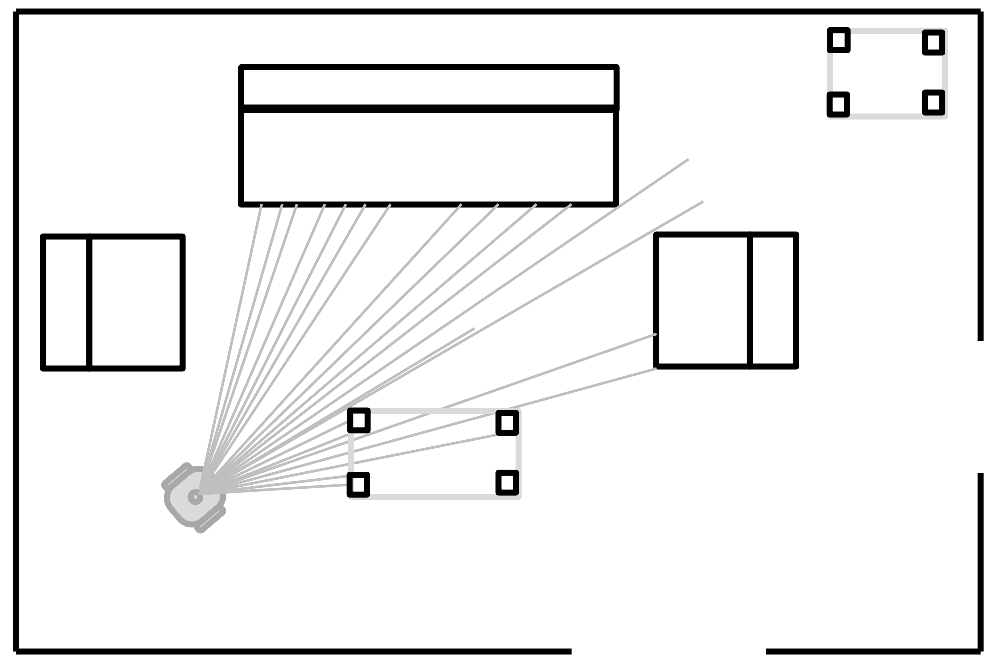
- Localizing an object with respect to a sensor is mathematically identical to localizing a robot with respect to a map.
- Now the model is of the form …

$$\underbrace{\underline{z}_{pred}}_{\text{image}}(\underline{x}) = \overbrace{\underline{h}}^{\text{image formation}}(\underbrace{\underline{x}}_{\text{image coords}}, \underbrace{\underline{\rho}_S^W}_{\text{pose}}, \overbrace{\underline{Z}}^{\text{map}})$$

9.2.1.2.6 Example: Find the Robot From Lidar

- Its not obvious !!





9.2.1.3 Visual Motion Estimation

- When there is no map, we can still estimate motion by matching past images to present ones.
- If the scene to image transform is invertible then, its easy:

$$\underline{x}_2 = T(\underline{\rho}_O^{S2})\underline{X} = T(\underline{\rho}_O^{S2})T^{-1}(\underline{\rho}_O^{S1})\underline{x}_1 = T(\underline{\rho}_O^{S1}, \underline{\rho}_O^{S2})\underline{x}_1$$

- Sometimes, this can be written in terms of a relative sensor pose:

$$\underline{x}_2 = T(\underline{\rho}_O^{S1}, \underline{\rho}_O^{S2})\underline{x}_1 = T(\underline{\rho}_{S1}^{S2})\underline{x}_1$$

- Or even in terms of an image-to-image transform:

$$\underline{x}_2 = T_S(\underline{\rho}_{S1}^{S2})\underline{x}_1 = T_I(\underline{\rho}_{I1}^{I2})\underline{x}_1$$

9.2.1.4 Fundamental Algorithms

- We have seen that there are three basic comuter vision algorithms that can be used to localize and estimate motion:
  - Align signals in two images
  - Match features to create correspondences
  - Compute relative pose in scene from rlative pose in image.

Outline

- 9.2 Visual localization and Motion Estimation
  - 9.2.1 Introduction
  - 9.2.2 Aligning Signals for Localization and Motion Estimation
  - 9.2.3 Matching Features for Localization and Motion Estimation
  - 9.2.4 Searching for the Optimal pose
  - Summary

9.2.2.1 Signal-Based Objective Function

- Define the predicted signal:

object
model
or map

image
formation
process

$$\underline{z}_{pred}(\underline{x}, \underline{\rho}, \underline{Z}) = \underline{h}(\underline{x}, \underline{\rho}, \underline{Z})$$

image

image
coords

pose

- We want to find the pose that aligns the observed and predicted signal.
- Form the <u>residual</u>:

$$\underline{r}(\underline{x}, \underline{\rho}, \underline{Z}) = \underline{z}_{obs}(\underline{x}) - \underline{z}_{pred}(\underline{x}, \underline{\rho}, \underline{Z})$$

9.2.2.1 Signal-Based Objective Function

- Compute the pose that minimizes the squared residual:

$$\underline{\rho}^* = argmin \left[f(\underline{\rho}) = \frac{1}{2} \sum_{\underline{x} \in W} \underline{r}^T(\underline{x}, \underline{\rho}, \underline{Z}) \underline{r}(\underline{x}, \underline{\rho}, \underline{Z}) \right]$$

- Now, order all the elements in the residual based on x and then the <u>x</u> argument can be removed:

$$\underline{\rho} = argmin \left[f(\underline{\rho}) = \frac{1}{2} \underline{r}^T(\underline{\rho}, \underline{Z}) \underline{r}(\underline{\rho}, \underline{Z}) \right]$$

- It may be advisable to normalize video images before computing residuals.

9.2.2.2 Aligning Video In Image Plane

- An example approach is correlation of monochrome video.
  - Perform exhaustive search over a search window
- The transformation of feature positions is:

$$\underline{y}(\underline{x}, \underline{p}) = \underline{T}(\underline{x}, \underline{p}) = \begin{bmatrix} x + p_1 \\ y + p_2 \end{bmatrix}$$

- The pixel residuals are:

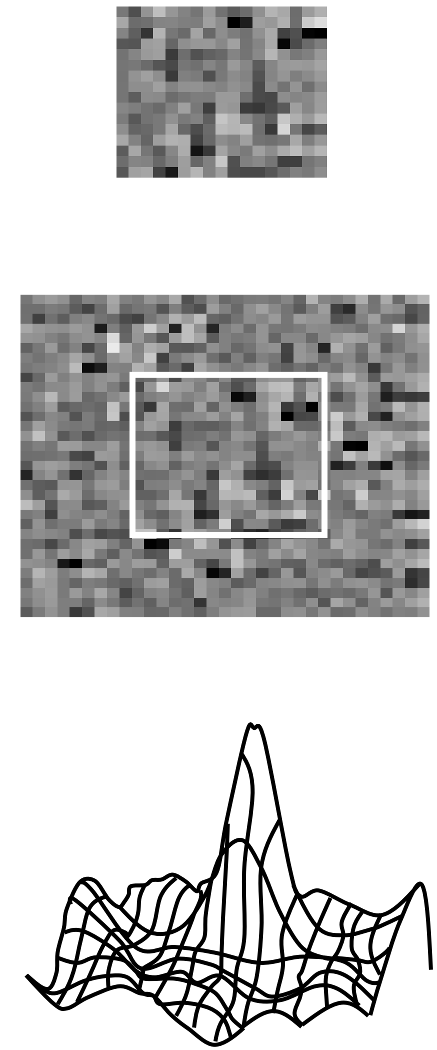
$$\underline{r}(\underline{x}, \underline{p}, \underline{Z}) = \underline{z}_{obs}(\underline{x}) - \underline{z}_{pred}(\underline{x}, \underline{p}, \underline{Z}) = \underline{z}_{obs}(\underline{x}) - \underline{Z}[\underline{y}(\underline{x}, \underline{p})]$$

- Solve as linear least squares:

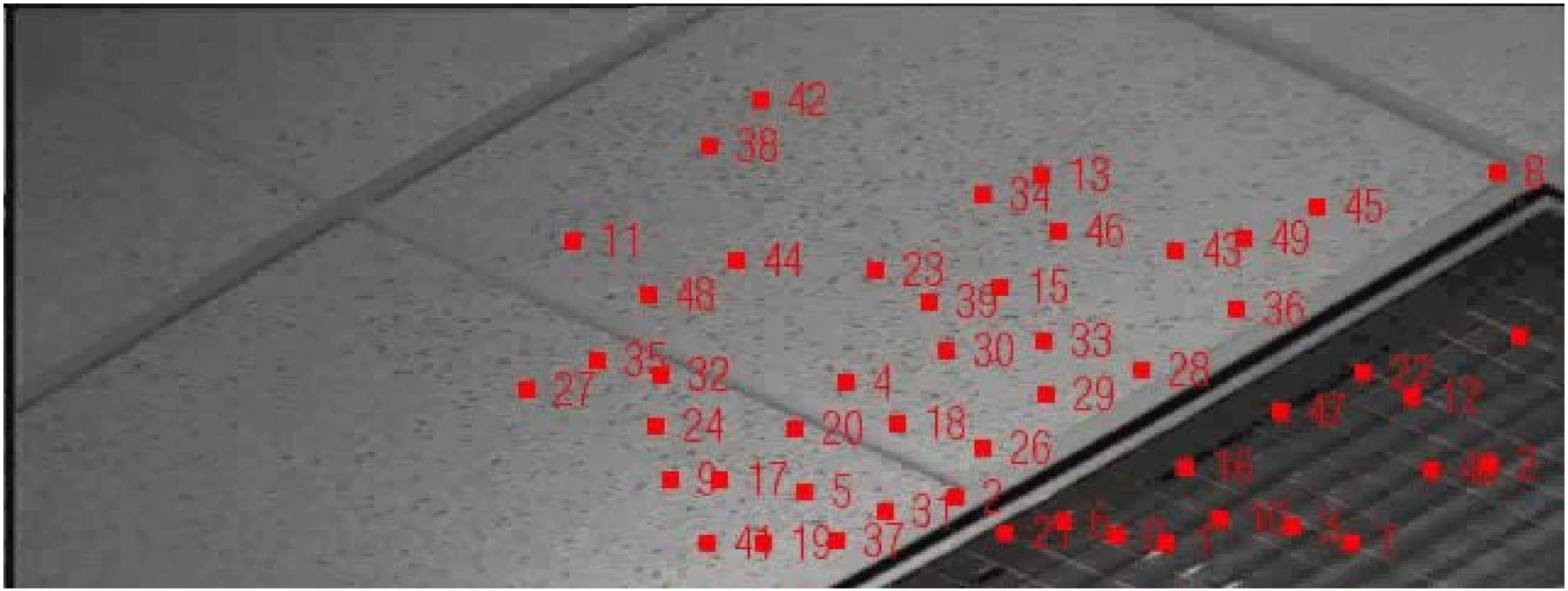
$$\underline{p} = argmin \left[f(\underline{p}) = \frac{1}{2}\underline{r}^T(\underline{p}, \underline{Z})\underline{r}(\underline{p}, \underline{Z}) \right]$$

9.2.2.2 Aligning Video In Image Plane

- Exhaustive search
  - Checks correspondence over a limited regions of possible displacements.
  - Muddies distinction between pose refinement and data association.
- Correlation is a matched filter so there is no better noise rejection around.

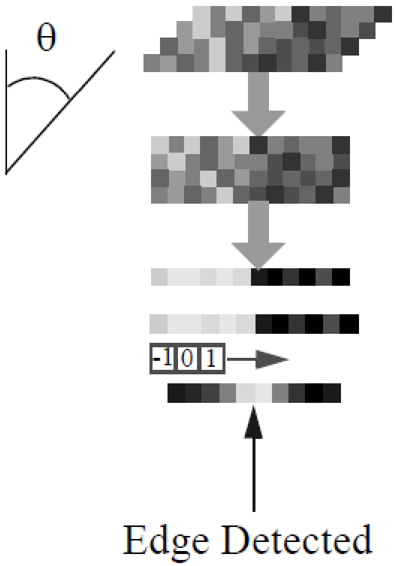


Video : Lucas Kanade Tracker



9.2.2.3 Example. Lane Tracking

- Transform incoming video based on:
  - Constant perspective foreshortening
  - Variable crosstrack offset
  - Variable road curvature
- Collapse columns into a linear image
- Search over a series of transforms for optimal signal match.



9.2.2.3 Example. Lane Tracking
(RoadFollowing / Lanetracking)

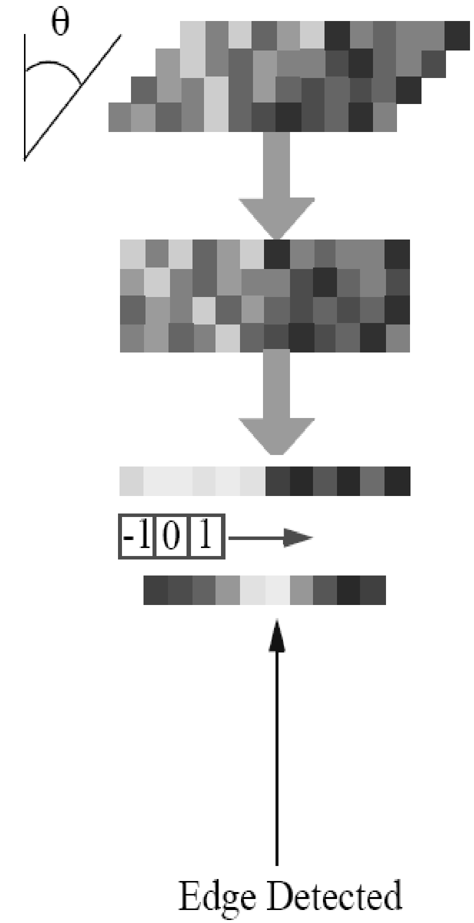
- About 15,000 people die each year in just the US in single vehicle roadway departure accidents.
- This is a visual servoing application.
- Lane tracking used for:
  - Lane Departure Warning (LDW)
  - Adaptive Cruise Control (ACC)

Carnegie Mellon
OTICS INSTITUTE

9.2.2.3 Example. Lane Tracking

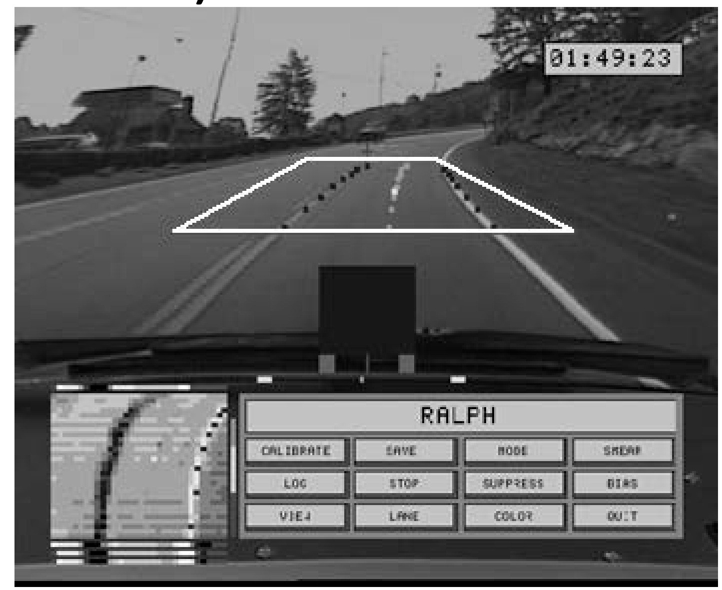
(Warping Approach: Dickmanns)

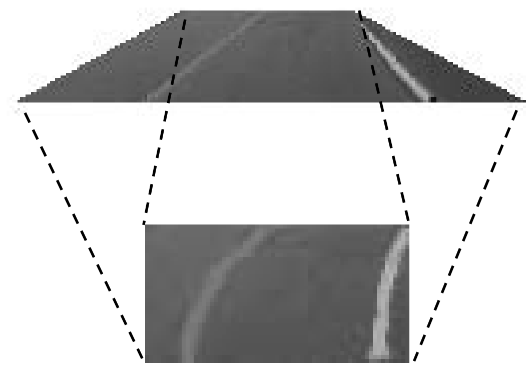
- Prewarps image regions based on expectations for both edge position and orientation.
- Sums along columns in order to enhance edges and reduce noise.
  - Summing is the simplest kind of filter. The random parts of the signal tend to cancel whereas the dc part continues to grow with the sum.
- Runs an edge detector on the resulting column sum.



9.2.2.3 Example. Lane Tracking
(Warping Approach: RALPH)

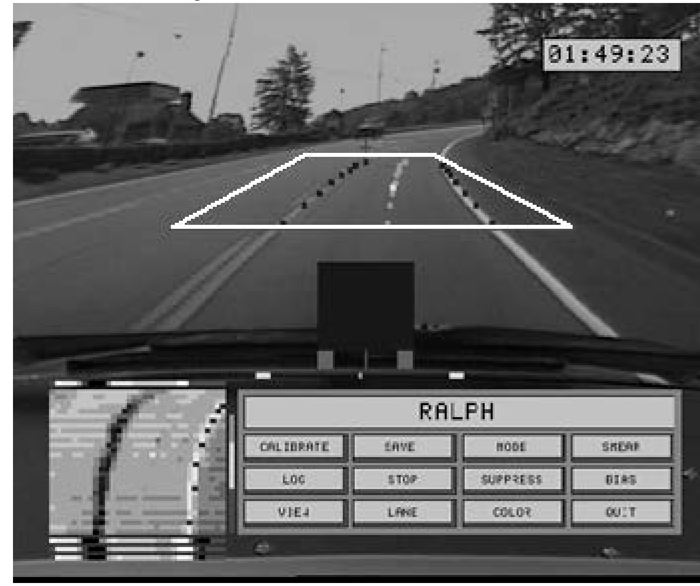
- Extends this idea to the entire road piece in view.
- Drove 2850 miles across the US.
- Tries to minimize the amount of explicit road modeling information used.
- Accomplishes lane detection in three steps:
  – sample the image
  – compute the curvature
  – compute the lateral offset

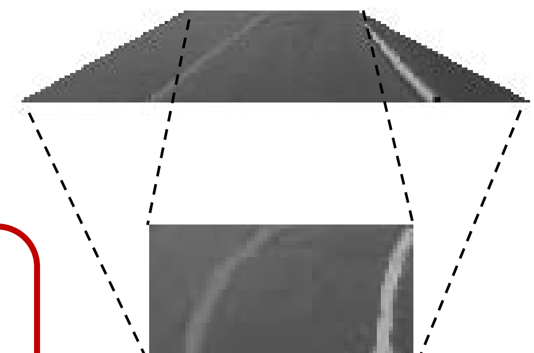




9.2.2.3 Example. Lane Tracking
(Warping Approach: RALPH)

- For the trapezoidal ROI:
  - start and end depends on the velocity.
  - width at all ranges is identical on the groundplane.
  - produces a rectangular “aerial image” of 30(h) X 32(w) pixels.



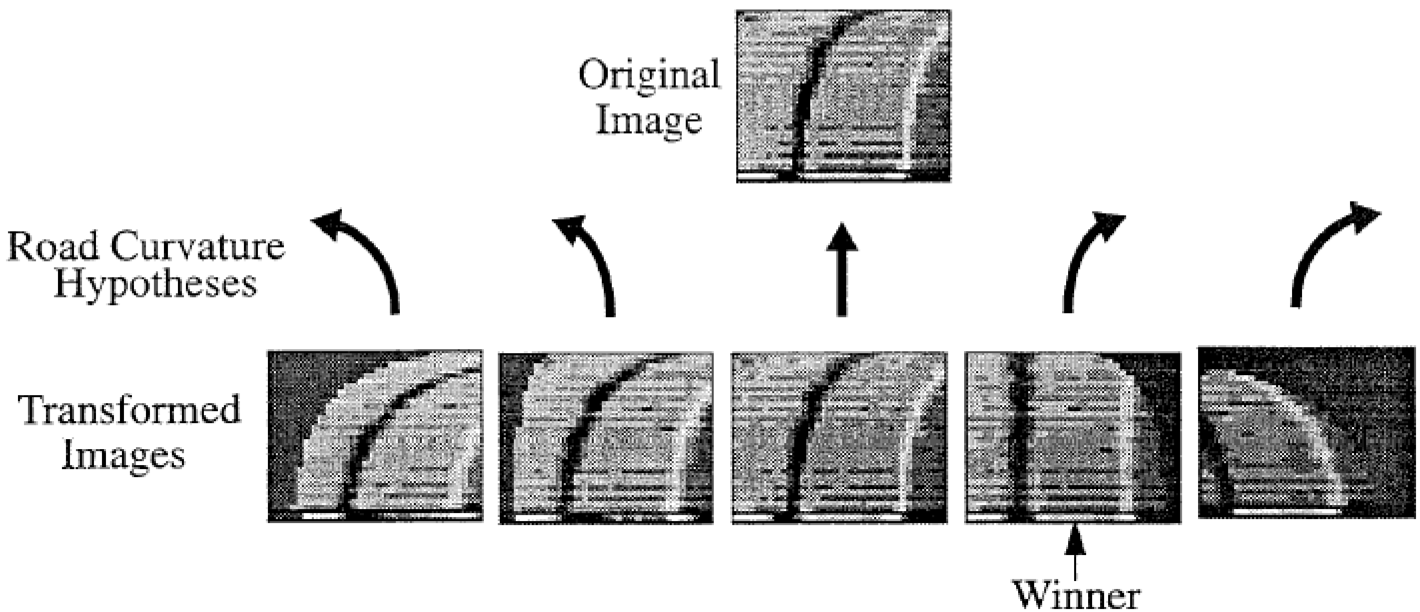


Called “Inverse Perspective Mapping”

9.2.2.3 Example. Lane Tracking

(Finding Curvature in RALPH)

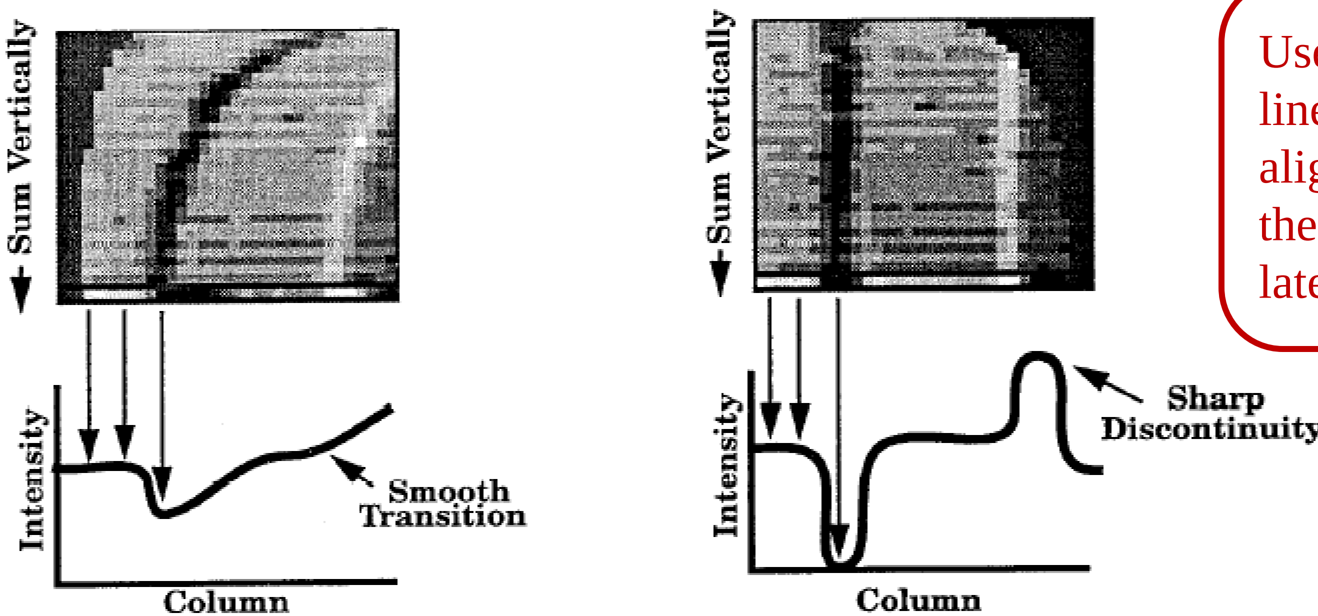
- Hypothesize a number of possible curvatures.
- Straighten the aerial image based on each assumption.
- The transformed aerial image which is straightest is the winner.



9.2.2.3 Example. Lane Tracking

(Which is “Straightest”)

- The column summed image has the sharpest peaks when the hypothesis is correct
  - Has edgiest intensity profile



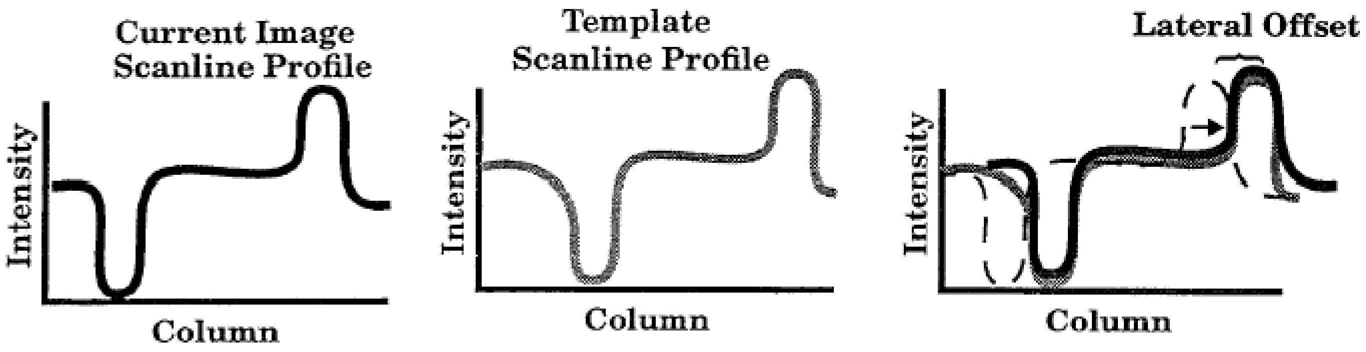
Uses any linear features aligned with the road at any lateral offset

Carnegie Mellon THE ROBOTICS INSTITUTE

9.2.2.3 Example. Lane Tracking

(Finding Lateral Offset)

- Column summing produces 32 element vector called the scanline intensity profile.
- As in GPS, a correlation search produces a peak at the correct offset.



This profile needs only to be unique and correlateable.

9.2.2.3 Example. Lane Tracking
(Outlook)

- Adaptation to multiple roadtypes can be accomplished by correlating with multiple road signatures simultaneously.
- Learning can be done at several levels:
  - Supervised: Operator presses a button to save the present profile as a template.
  - Unsupervised: Modify the template in use to incorporate a small percentage of the present profile.
  - Predictive: Assume curvature is continuous and extract a new template from the top of the image (the road far ahead).

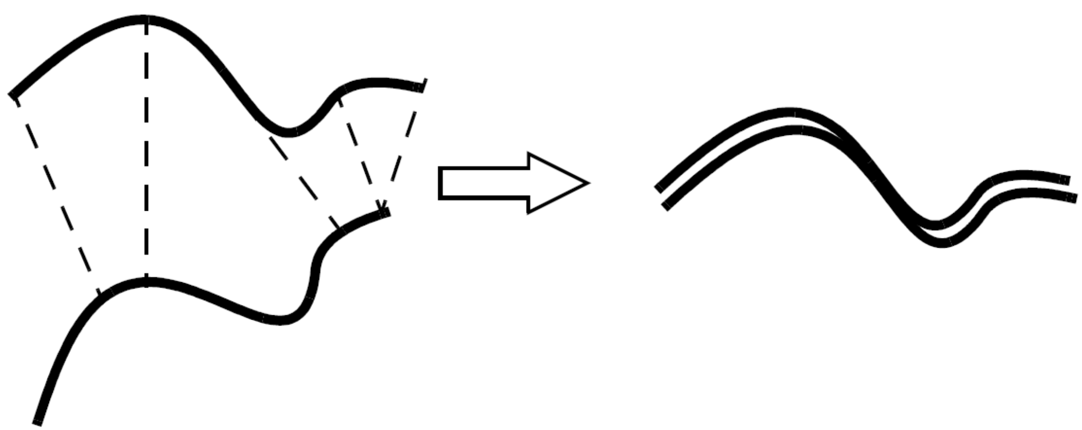
Video

2.2.4 Other forms of Maps

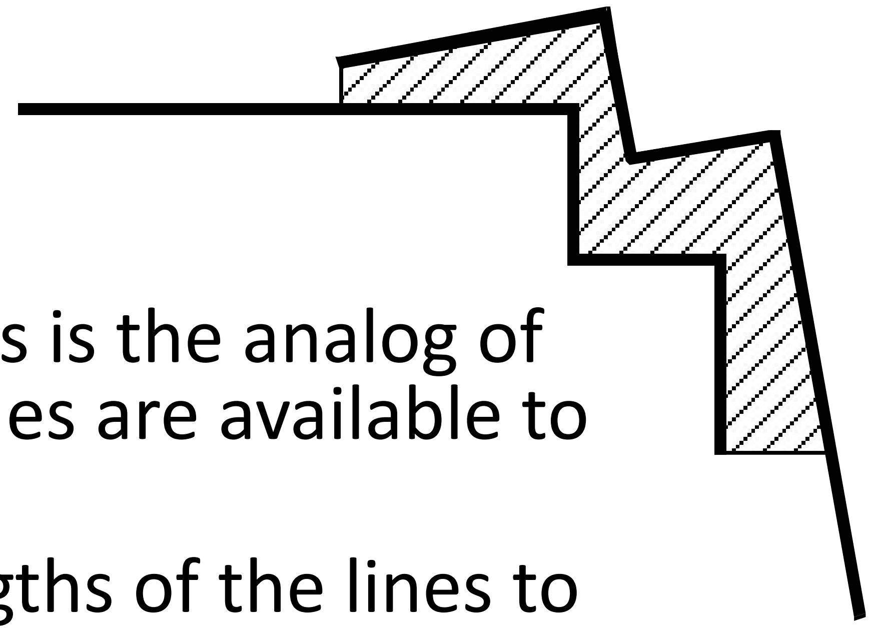
- Any field of perceptual expectations $\underline{h}(\underline{\rho})$ over the space of poses can be used as a map.
- Some options
  - Remember VO features in spatially indexable form
  - Lidar intensity signatures of roads
  - Video of factory floors
  - Aerial elevation maps
  - Range data of building walls

9.2.2.5 Aligning Surface Geometry in Range Imagery

- The equivalent of video alignment is curve or surface alignment.
- We usually assume that the two curves or surfaces are undistorted but search over distortion is possible.
- The area between the two surfaces is one way to express the registration error.



9.2.2.5 Aligning Surface Geometry in Range Imagery

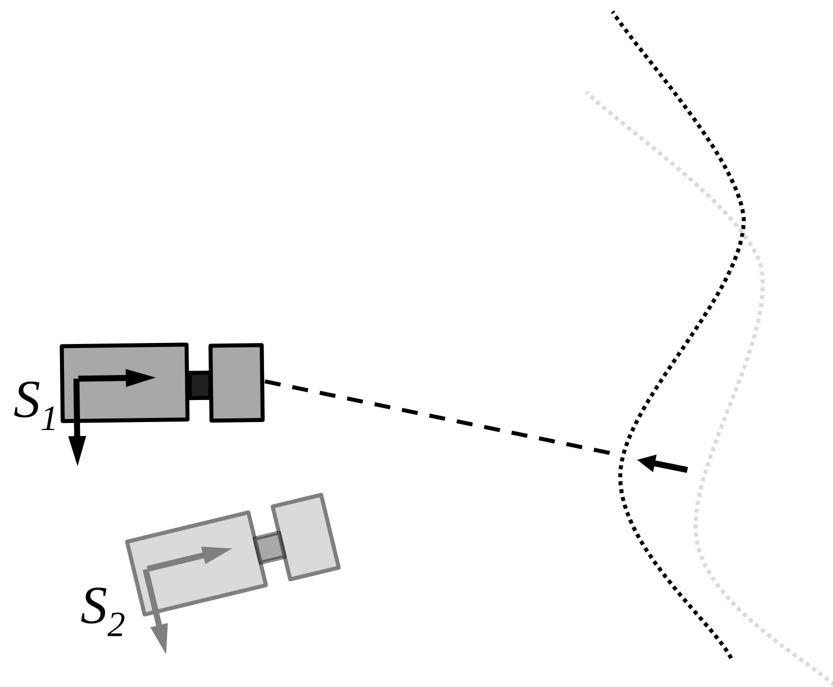


- Area/Volume between scans is the analog of SSD of video. Several schemes are available to estimate this area.
- ICP uses the sum of the lengths of the lines to closest points as the residual.
  – Each point on one scan has a closest neighbor on the other.
- Feature-based schemes do the same:
  – but then the points actually correspond.

9.2.2.5 Aligning Surface Geometry in Range Imagery

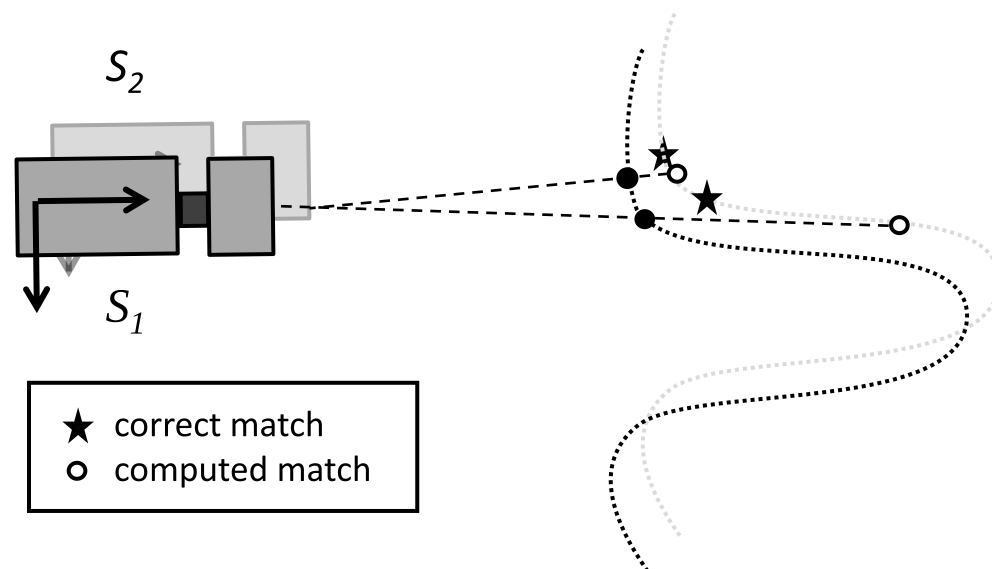
(Projective Association – Matching in Image Coordinates)

- Standard association is n^2 computations for n points and its done every iteration.
- Projective association is order n.



9.2.2.5 Aligning Surface Geometry in Range Imagery
(Projective Association – Glancing Incidence Pathology

- Projective association is fast but it can be pretty wrong at glancing incidence.
- Residuals are small near the answer and this helps a lot.



Outline

- 9.2 Visual localization and Motion Estimation
  - 9.2.1 Introduction
  - 9.2.2 Aligning Signals for Localization and Motion Estimation
  - <u>9.2.3 Matching Features for Localization and Motion Estimation</u>
  - 9.2.4 Searching for the Optimal pose
  - Summary

Matching Features

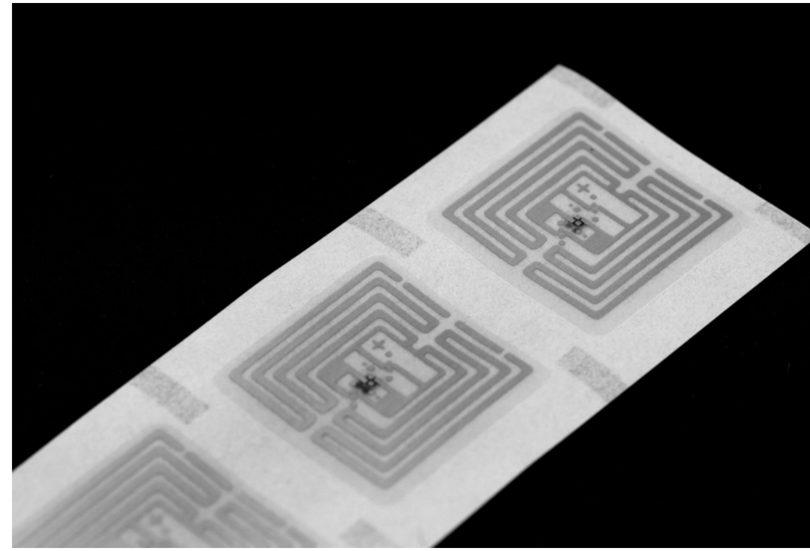
- In signal alignment:
  - Data Is already ordered
  - Data at corresponding positions in signal is assumed to correspond.
- Matching features only:
  - Reduces the amount of data to match
  - BUT introduces a data (feature) association problem.

9.2.3.1 Segmentation and Features

- Reducing imagery to features has the advantage of:
  - boosting the signal content.
  - making the minimum of cost function as sharp as possible.
- Features can be many things
  - Edges or regions in video
  - Points of high curvature in range data

9.2.3.2 Objective Function / 9.2.3.3 Feature Attributes

- An image $\underline{z}_{obs}$ contains more information than its signal amplitudes
  - because the individual amplitudes occur somewhere in particular.
- Often, the useful info is not the amplitudes but where they occur.
- Features may retain some of the original signal or they may be stripped of everything but their locations.
- Retained attributes may be:
  - Id, barcode etc.
  - Block of surrounding pixels
  - Eigenvalues of Harris corners
  - Curvature or spin images



Commercial RFID Tag

9.2.3.4 Typical Features and Objective Functions

- A residual can be formed from predicted and observed locations of features.

$$\underline{r}_k(\underline{\rho}, \underline{X}_k) = \underline{x}_k - \underline{h}(\underline{\rho}, \underline{X}_k)$$

- Collect them all into a single vector and drop k.

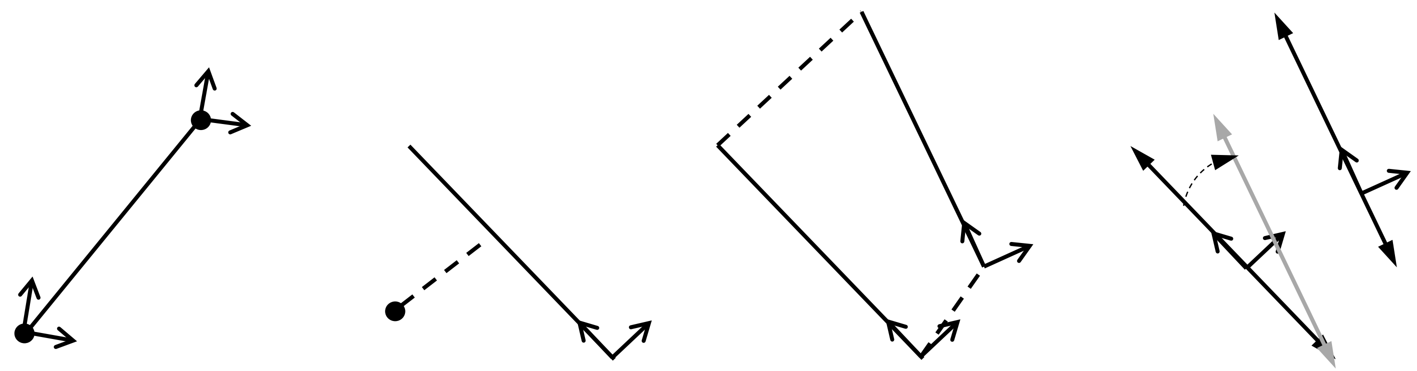
$$\underline{r}(\underline{\rho}, \underline{X}) = \underline{x} - \underline{h}(\underline{\rho}, \underline{X})$$

- We could then find the location correspoding to the residual norm

$$\underline{\rho} = argmin \left[f(\underline{\rho}) = \frac{1}{2}\underline{r}^T(\underline{\rho}, \underline{Z})\underline{r}(\underline{\rho}, \underline{Z}) \right]$$

9.2.3.4 Typical Features and Objective Functions

- The distance between corresponding points is only one option.
- Examples of other planar correspondences are shown below.



point-to-point
(2 cost dofs)

point-to-segment
point-to-line
(1 cost dof)

segment-to-segment
(2 cost dofs)

line-to-line
(1 cost dof)

9.2.3.5 Search For Associations

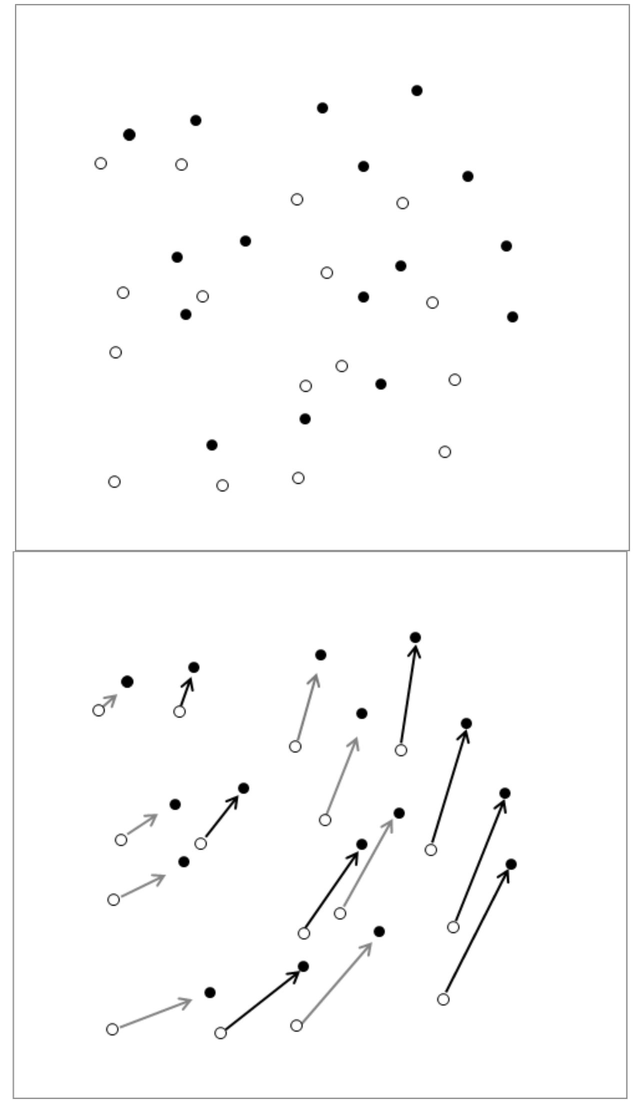
- Based on no other info, the data association problem is factorial complexity.
- However pose knowledge constrains correspondences so the two problems are coupled.
- Features attributes also help constrain the search.
- Generally, all of the following information can be brought to bear:
  - Richness : feature attributes
  - Pose Estimates
  - Spatial Separation (reduced ambiguity)
  - Consensus
  - Conditioning (some incorrect correspondences may be OK)

9.2.3.6 RANSAC

- Short for RANdom Sample Consensus
- Useful when data set contains outliers that do not fit the model.
  - 1: Choose a random sample of data of sufficient size to fix all of the parameters of the model (pose). These are hypothetical inliers.
  - 2: Test all other points against the hypothetical model and reject points as outliers that do not fit.
  - 3: Re-estimate the model from all remaining inliers.
  - 4: If there are sufficient inliers, remember this model if it is the best fit so far.
  - 5: Terminate after n iterations or a good enough fit is achieved.

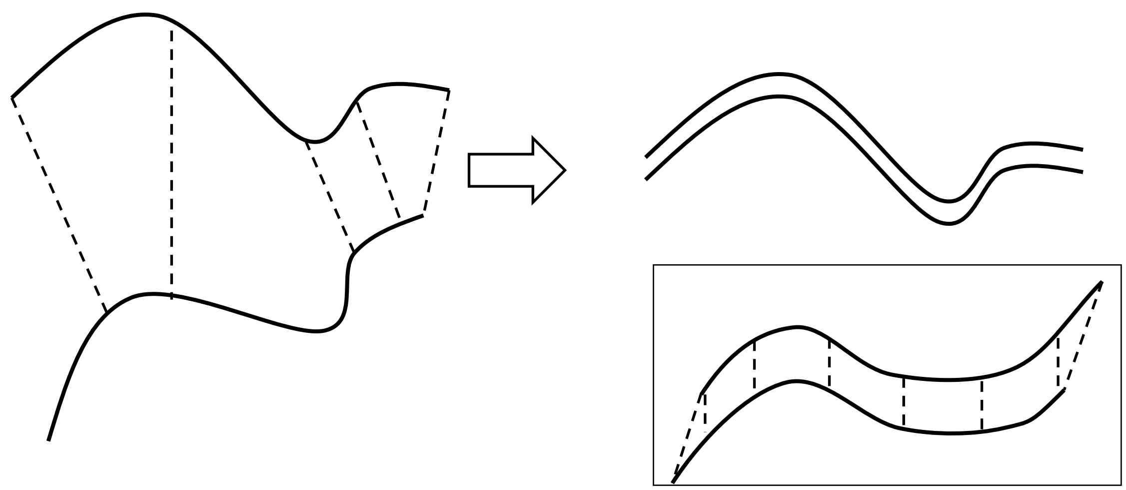
9.2.3.6.1 Example RANSAC in Image

- Suppose 16 features and 50% outliers in the data.
- It takes only two features to fix the pose in 2D.
- The probability of selecting 2 inliers is 25% so <u>it takes only 4 attempts on average</u> to find the right pose.



9.2.3.7 Closest Point Association in Range Data

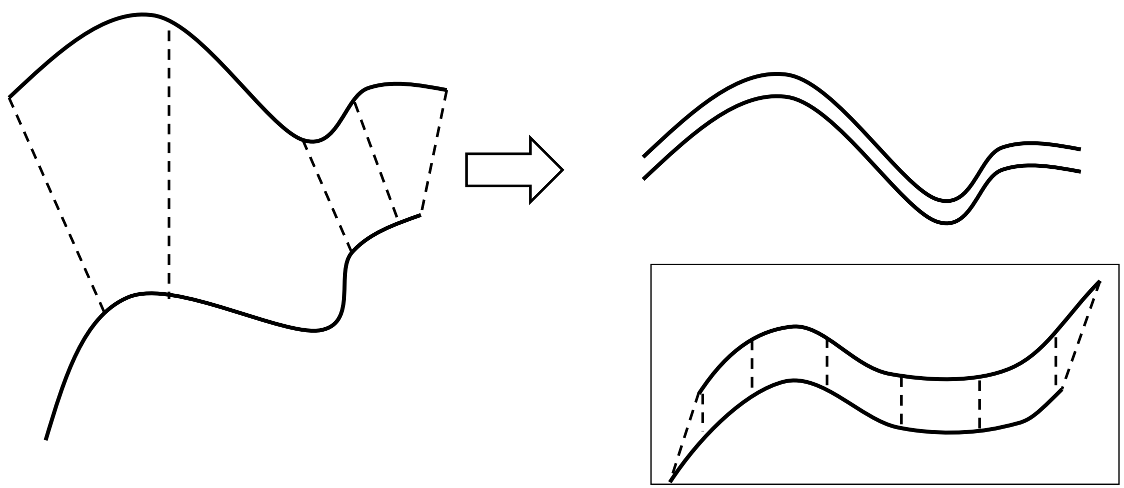
- Avoids explicit solution of the correspondence problem.
  - Solution emerges as the iteration proceeds
- Used for range imagery. Makes sense when:
  - Two partial views of the same shape are available.
  - They are largely undistorted.
  - An initial estimate of relative position is available.
  - They are free form surfaces.



9.2.3.7 Closest Point Association in Range Data
(Basic Algorithm)

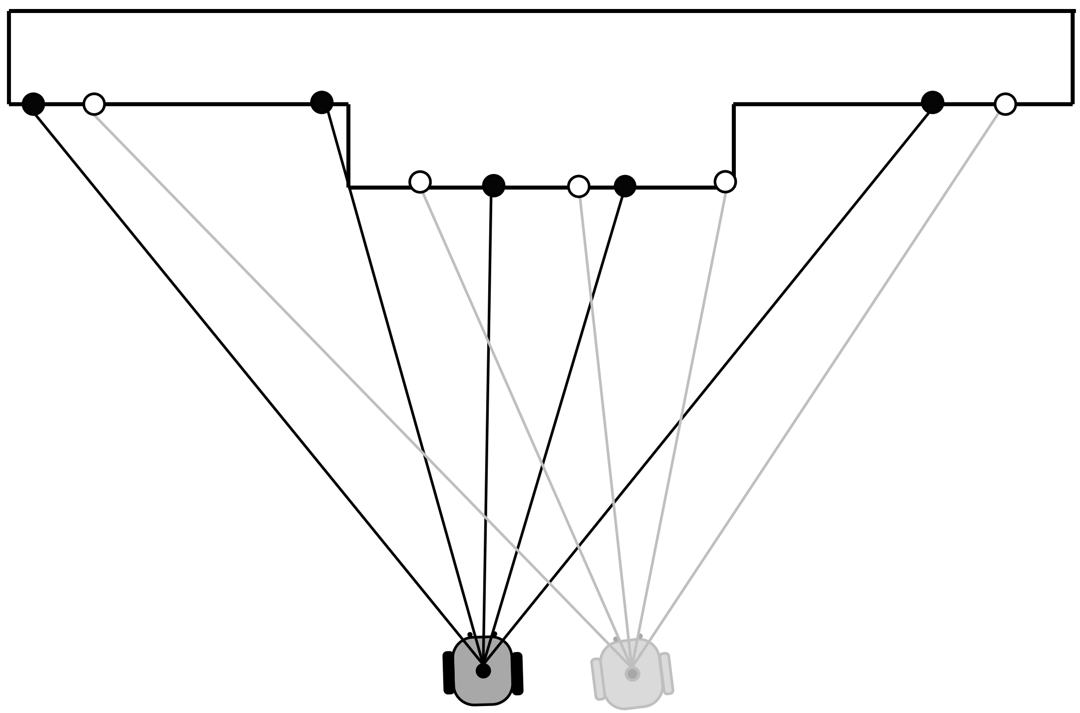
- Temporarily associate each point on scan1 with its closest neighbor on scan2.

  - For each point in scan 2:
    - Compute the distance to each point in scan1.
    - Associate the closest point in scan1 with the original point in scan 2.



9.2.3.7 Closest Point Association in Range Data
(Interpolation)

- Discrete samplings need not line up at each point.
- Need to interpolate.



Sensors produce equally spaced angular pixels.

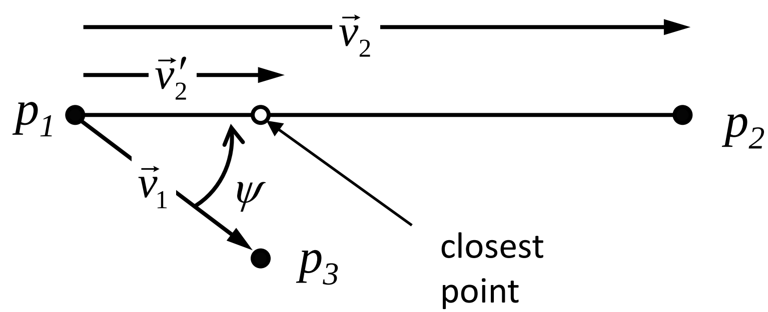
9.2.3.7 Closest Point Association in Range Data
(Interpolation)

- A point on the line from p1 to p2 is closer to p3 if:

$$\vec{v}_1 \bullet \vec{v}_2 > 0$$

- If so, then the closest point is:

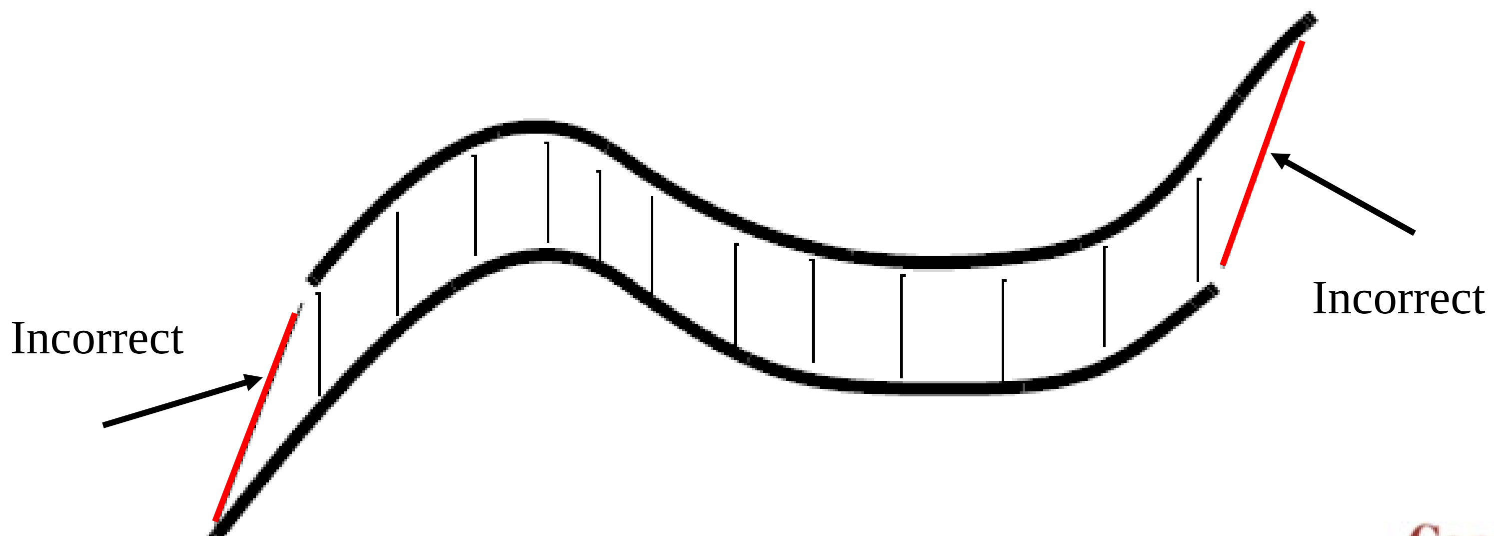
$$\vec{v}_2{}' = \frac{(\vec{v}_1 \bullet \vec{v}_2)}{(\vec{v}_2 \bullet \vec{v}_2)}\vec{v}_2$$



9.2.3.7 Closest Point Association in Range Data

(Endpoint Pathology)

- Unless you actively avoid it, ICP will associate points when there is no real association.
- I.E. Endpoints when scans overlap only partially.

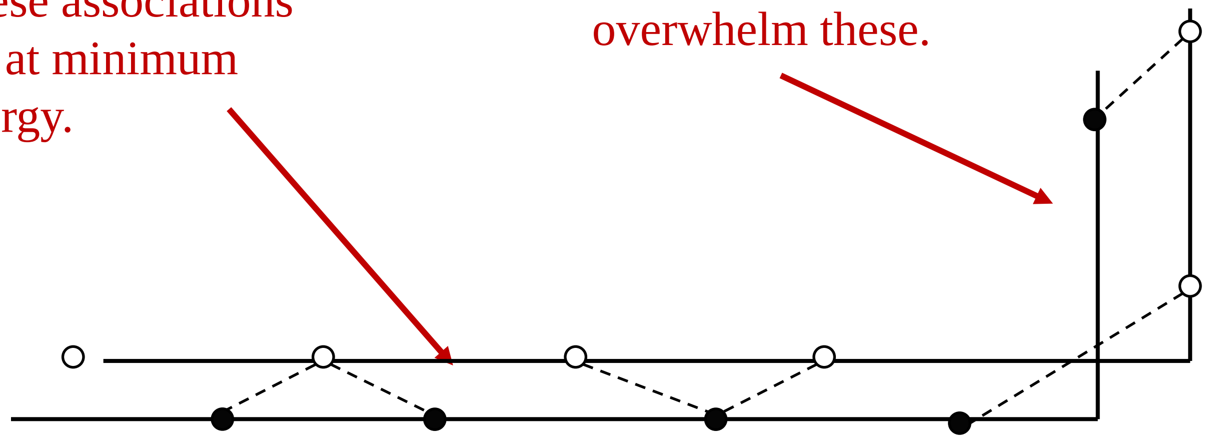


9.2.3.7 Closest Point Association in Range Data
(Internal Tension Pathology)

- Motion in the tangential direction is resisted by almost any set of associations
  - Because when some association lines are shortened, others are lengthened.

These associations are at minimum energy.

They will overwhelm these.



Straight point-to-point ICP does poorly when curves have a lot of straight segments in them.

Tangential associations are not measuring area.

9.2.3.7 Closest Point Association in Range Data

(Other Improvements)

- Basic issue:
  – Closest does not always equal corresponding.
- Other ideas for fixes:
  – Associations in the local normal direction.
  – Associations only at points of high curvature.
  – Associations of only “compatible” (similiar curvature) points.
- All are a kind of shift toward a feature based approach.

Outline

- 9.2 Visual localization and Motion Estimation
  – 9.2.1 Introduction
  – 9.2.2 Aligning Signals for Localization and Motion Estimation
  – 9.2.3 Matching Features for Localization and Motion Estimation
  – <u>9.2.4 Searching for the Optimal Pose</u>
  – Summary

9.2.4 Searching for the Optimal Pose
(3 SubProblems)

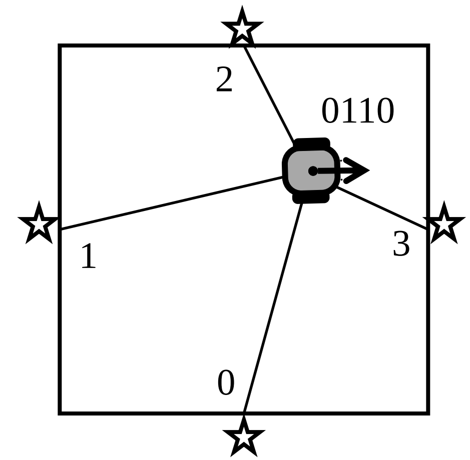
- Pose Determination
  – Find pose with no prior information
- Pose Refinement
  – Find pose with initial estimate inside radius of convergence.
- Pose Tracking
  – Find pose with initial estimate very near by.

9.2.4.1 Pose Determination

- Also called the insertion problem (AGVs).
- Vision part of the problem is place recognition.
- Fundamentally, this is minimization of an objective function with many local minima.
- Two ideas for proceeding…
  - Sampling will work when there are few local minima
  - Lookup tables can be used to find a good initial guess from the data.

9.2.4.1.1 Example Place Recognition in Bearing Data

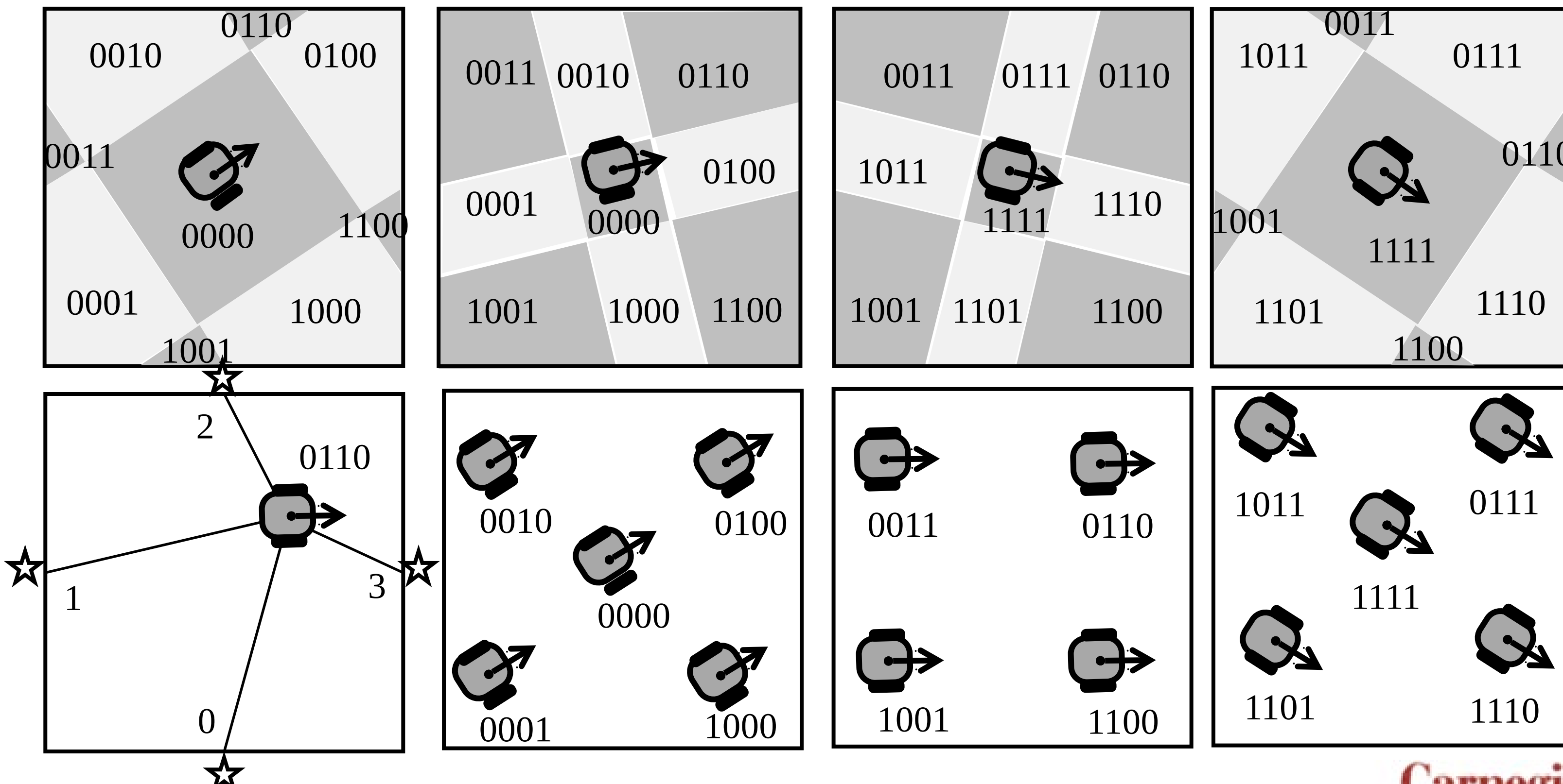
- Problem: process a single scan of 4 fiducial bearings and determine, roughly, where the robot is.
- Account for symmetry using supplied heading quadrant.
  - 4 solutions for any given scan



Carnegie Mellon
THE ROBOTICS INSTITUTE

9.2.4.1.1 Example Place Recognition in Bearing Data

- Solution: reduce each scan to a 4 digit binary number.
  - 0 iff bearing < 45°
  - 1 iff bearing > 45°



Carnegie Mellon
THE ROBOTICS INSTITUTE

9.2.4.2 Pose Refinement

- Assume a signal matching approach.

- Let the unknown pose and warp be defined by a set of parameters $\underline{p}$.

- The residual would be the (vectorized version of):

$$\underline{r}(\underline{x}, \underline{p}, \underline{Z}) = \underline{z}_{obs}(\underline{x}) - \underline{z}_{pred}(\underline{x}, \underline{p}, \underline{Z}) = \underline{z}_{obs}(\underline{x}) - \underline{Z}[\underline{y}(\underline{x}, \underline{p})]$$

9.2.4.2 Pose Refinement

- If an initial estimate is available, it may be close enough to justify the use of gradient information to find a local minimum.
- Recall that the (unweighted) Newton step takes the form:

$$\Delta \underline{p} = -[\underline{r}_{\underline{p}}^{T} \underline{r}_{\underline{p}}]^{-1} \underline{r}_{\underline{p}}^{T} \underline{r}(\underline{p})$$

- where $\underline{r}_p$ is the residual gradient wrt the parameters. In this case:

$$\underline{r}_{\underline{p}} = \underline{r}_{\underline{p}}(\underline{p}, \underline{Z}) = -\underline{h}_{\underline{p}}(\underline{p}, \underline{Z}) = -\underline{Z}_{\underline{p}}[\underline{y}(\underline{p})]$$

9.2.4.2 Pose Refinement

- Recall from last slide: $\underline{r}_{\underline{p}} = \underline{r}_{\underline{p}}(\underline{p}, \underline{Z}) = -\underline{h}_{\underline{p}}(\underline{p}, \underline{Z}) = -\underline{Z}_{\underline{p}}[\underline{y}(\underline{p})]$
- By the chain rule:

$$\underline{Z}_{\underline{p}} = \frac{\partial}{\partial \underline{p}}\{\underline{Z}(\underline{y}(\underline{p}))\} = \left(\frac{\partial \underline{Z}}{\partial \underline{y}}\right)\left(\frac{\partial \underline{y}}{\partial \underline{p}}\right)$$

- The two components of the gradient are:
  - $(\partial \underline{z}/\partial \underline{y})$ the gradient of the image evaluated at $\underline{y}(\underline{p})$.
  - $(\partial \underline{z}/\partial \underline{p})$ the parameter Jacobian of the transform evaluated at $\underline{p}$.

9.2.4.2 Pose Refinement

- If instead we were matching features, the Newton step is more simply:

$$\Delta \underline{\rho} = -[\underline{r}_{\rho}{}^{T} \underline{r}_{\rho}]^{-1} \underline{r}_{\rho}^{T} \underline{r}(\underline{\rho})$$

- The pose gradient is:

$$\underline{r}_{\rho} = \underline{r}_{\rho}(\underline{\rho}, \underline{X}) = -\underline{h}_{\rho}(\underline{\rho}, \underline{X})$$

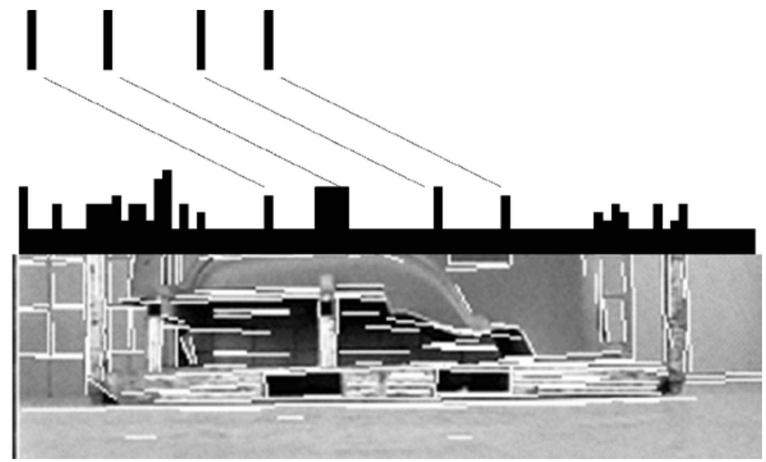
- We can substitute this into the Newton step to produce:

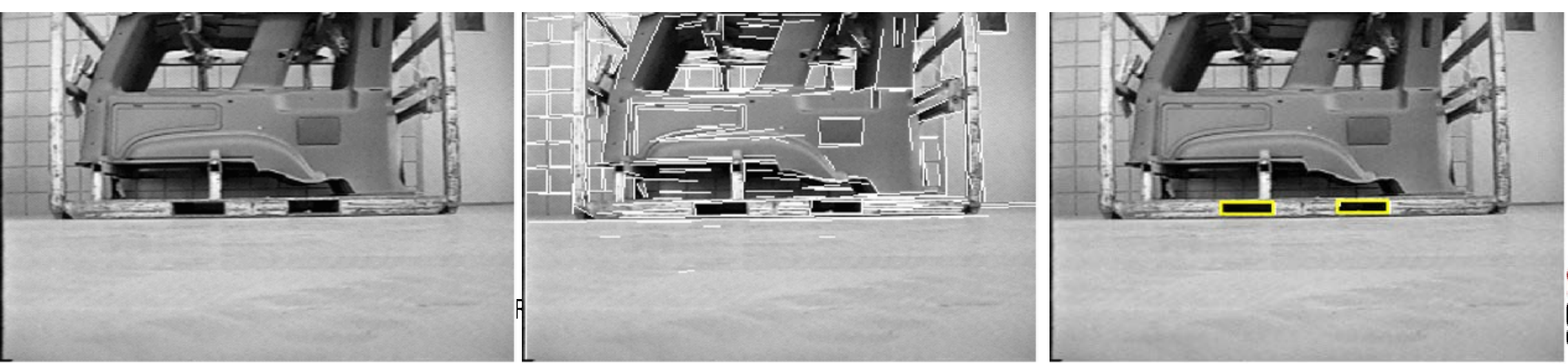
$$\Delta \underline{\rho} = -[\underline{r}_{\rho}{}^{T} \underline{r}_{\rho}]^{-1} \underline{r}_{\rho}^{T} \underline{r}(\underline{p}) = [\underline{h}_{\rho}{}^{T} \underline{h}_{\rho}]^{-1} \underline{h}_{\rho}^{T} \underline{r}(\underline{\rho})$$

- Which is just the left pseudoinverse. In vision problems, the equations are typically overdetermined.

9.2.4.2.2 Example: Locate a Pallet

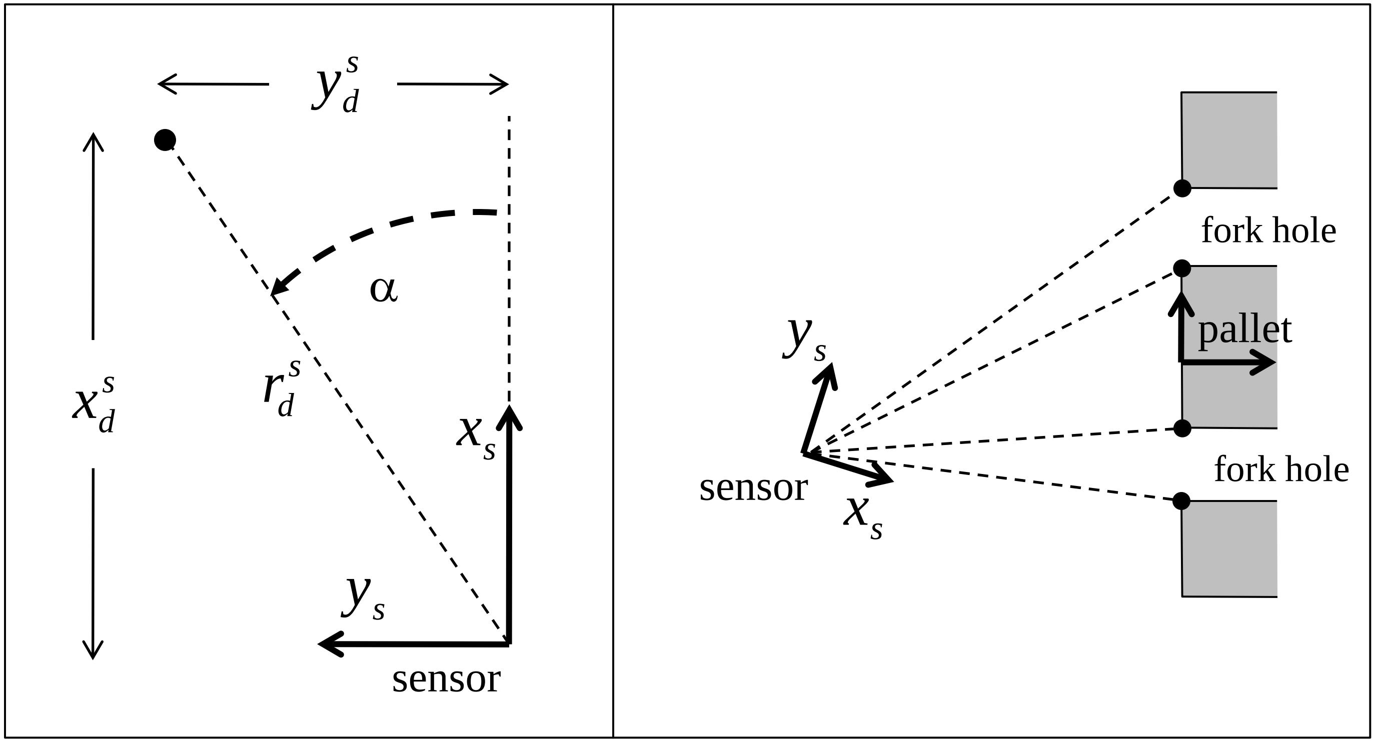
- Vertical fork hole edges must
  - be principally oriented in the image
  - occur in darkening / lightening pairs
- Different templates can be correlated to identify the pallet type.
- Models encode the (scale independent) ratio of hole width to hole separation.





9.2.4.2.2 Example: Locate a Pallet

- Assume that the 4 vertical edges of the pallet holes have been found. Localize the pallet.



9.2.4.2.2 Example: Locate a Pallet

- Let m denote model frame and s denote sensor.
- The measurement model is simply: $y_d^i = (fy_d^s)/x_d^s$
- Which is proportional to the bearing angle α.
- The vector of scene coordinates is: $\underline{r}_d^s = \left[x_d^s \; y_d^s\right]^T$
- Then, the measurement Jacobian is:

$$H_{sd}^{id} = \frac{\partial y_d^i}{\partial \underline{r}_d^s} = \left[-\frac{fy_d^s}{(x_d^s)^2} \; \frac{f}{x_d^s} \right]$$

9.2.4.2.2 Example: Locate a Pallet

- Attach a frame to each of the four feature points.
- Then the Jacobian is a compound-left pose Jacobian:

$$H_{sm}^{sd} = \frac{\partial \underline{\rho}_d^s}{\partial \underline{\rho}_m^s} = \begin{bmatrix} 1 & 0 & -(s\psi x_d^m + c\psi y_d^m) \\ 0 & 1 & (c\psi x_d^m - s\psi y_d^m) \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -(y_d^s - y_m^s) \\ 0 & 1 & (x_d^s - x_m^s) \\ 0 & 0 & 1 \end{bmatrix}$$

- We will use only the first two lines

$$H_{sm}^{sd} = \frac{\partial \underline{r}_d^s}{\partial \underline{\rho}_m^s} = \begin{bmatrix} 1 & 0 & -(s\psi x_d^m + c\psi y_d^m) \\ 0 & 1 & (c\psi x_d^m - s\psi y_d^m) \end{bmatrix} = \begin{bmatrix} 1 & 0 & -(y_d^s - y_m^s) \\ 0 & 1 & (x_d^s - x_m^s) \end{bmatrix}$$

9.2.4.2.2 Example: Locate a Pallet

- The complete solution for an assumed initial value for the pose $\underline{\rho}_m^s$ is:

$$\underline{r}_d^s = T_m^s(\underline{\rho}_m^s) * \underline{r}_d^m$$

$$y_d^i = (f y_d^s) / x_d^s$$

$$H_{sm}^{id} = \left(\frac{\partial y_d^i}{\partial \underline{\rho}_m^s}\right) = \left(\frac{\partial y_d^i}{\partial \underline{\rho}_d^s}\right)\left(\frac{\partial \underline{\rho}_d^s}{\partial \underline{\rho}_m^s}\right) = H_{sd}^{id} H_{sm}^{sd} = \left[-\frac{f y_d^s}{(x_d^s)^2} \;\; \frac{f}{x_d^s} \right] \begin{bmatrix} 1 & 0 & -(y_d^s - y_m^s) \\ 0 & 1 & (x_d^s - x_m^s) \end{bmatrix}$$

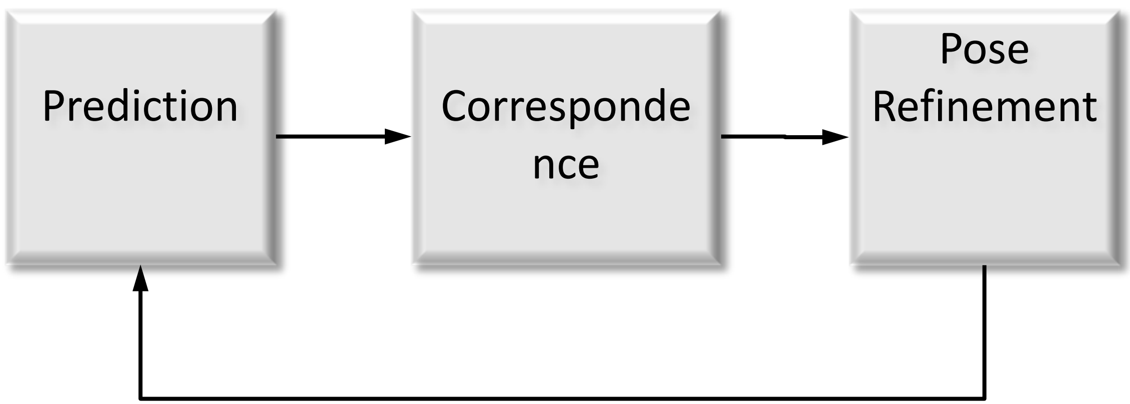
- 4 measurements are stacked to form the residual:

$$\underline{r}_k(\underline{\rho}, \underline{X}_k) = \underline{x}_k - \underline{h}(\underline{\rho}, \underline{X}_k)$$

- and its gradient points the way in line search.

9.2.4.3 Pose Tracking

- In the most general case:
  - Feature locations are predicted, possibly based on secondary estimates of motion.
  - Corresponding features are passed to a pose refinement algorithm.



9.2.4.3.1 Feature Velocities

- If we want velocites and there are no secondary estimates available …
- We can compute camera velocity from feature velocity. First linearize the measurement model wrt sensor motion.

$$\Delta \underline{x}_k = H(\underline{\rho}, \underline{X}_k)\Delta \underline{\rho}$$

- Then divide by Δt and pass to the limit:

$$\dot{\underline{x}}_k = H(\underline{x}, \underline{\rho}, \underline{Z})\dot{\underline{\rho}}$$

9.2.4.3.2 Making and Tracking Floor Mosaics

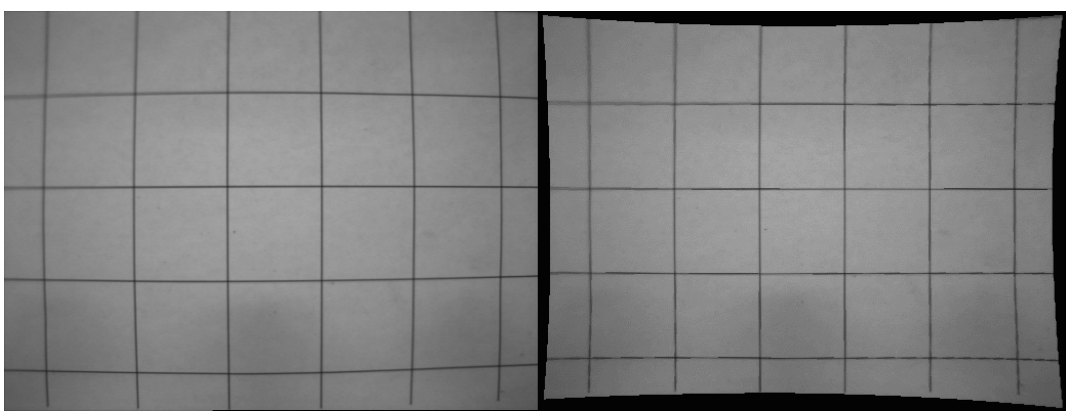
- All systems which use a map to localize are (model-based) visual trackers.
- Hence visual tracking is one of the most important algorithms in mobile robotics.
  - The Kalman filter system model amounts to the estimate of intervening motion.
  - Measurement model is the prediction mechanism.

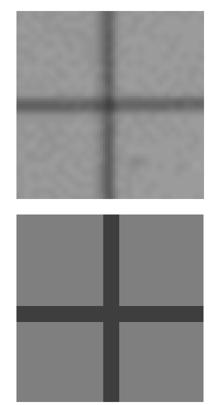
9.2.4.3.2 Making and Tracking Floor Mosaics (Tracking Mosaics)

- Floor mosaics are used as the map.
- Features in imagery are correlated with map (mosaic) based predictions.
- Submillimeter precision and 60 mph speeds are possible.

9.2.4.3.2 Making and Tracking Floor Mosaics (Lens Distortion Removal)

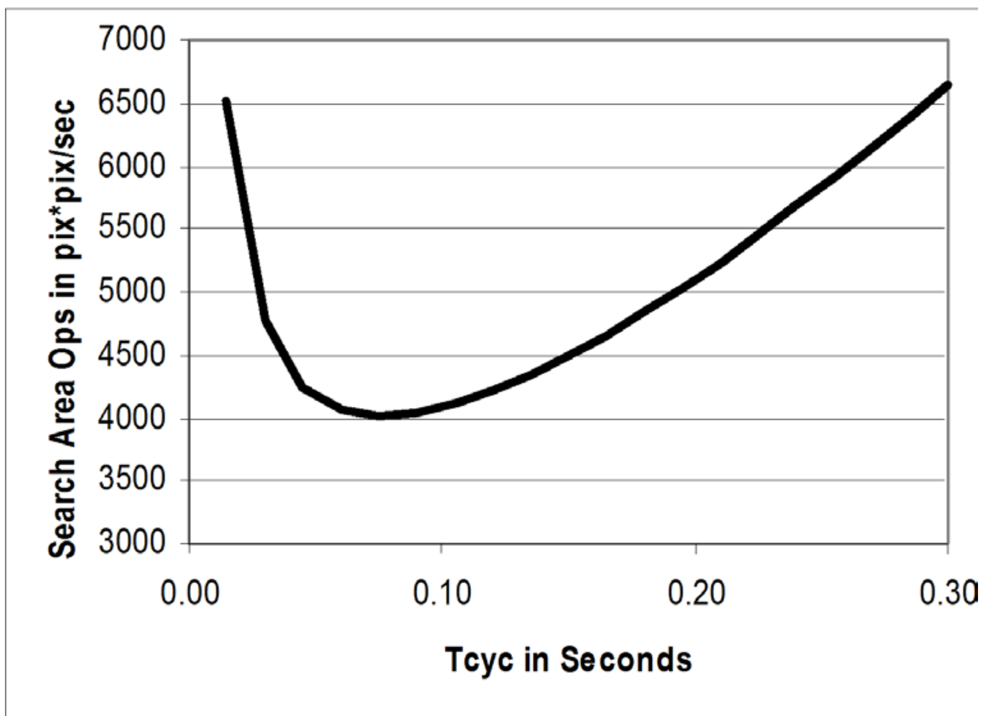
- Wide FOV lens is necessary because cmaera is so close to floor and large image footprint is required.
- Technique: Image a grid and compute the lens distortion function that explains it.
- Then, invert the distortion to rectify the images.





9.2.4.3.2 Making and Tracking Floor Mosaics (Tracking Update Rate)

- A sweet spot exists.
  - Random error rewards slow updates.
  - Systematic error rewards fast updates.

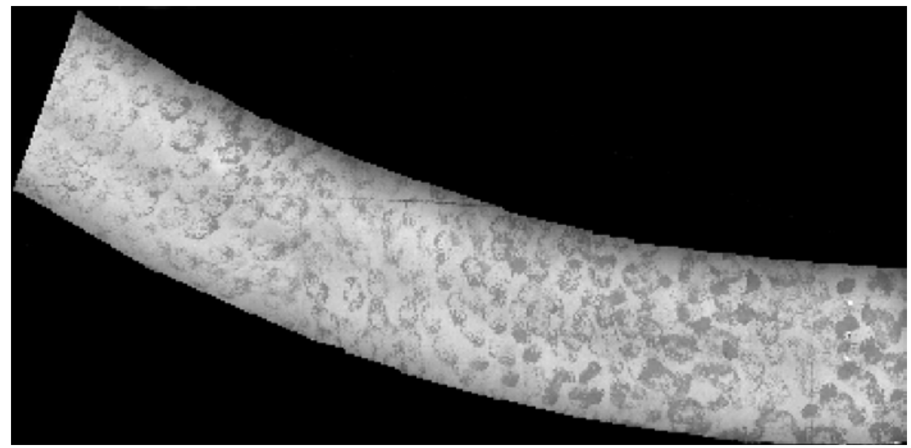


9.2.4.3.2 Making and Tracking Floor Mosaics (Making Mosaics)

- A very easy case because:
  - Ranges are known
  - Environment is flat (no disortion)
  - Motion is approximately known (odometry)

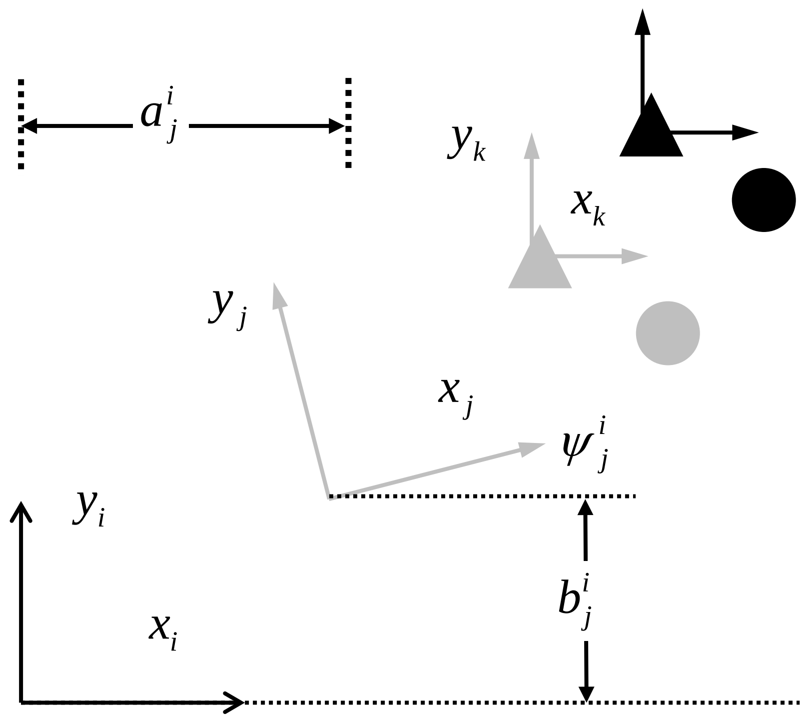
9.2.4.3.2 Making and Tracking Floor Mosaics (Registration)

- Bright centers of all images are merged to produce a single long thin image…



9.2.4.3.2 Making and Tracking Floor Mosaics (Registration)

- It is not necessary to solve for the rotations of each feature – it comes out at the pose level.
- The image alignment is accomplished with a compound-left pose Jacobian:



$$
\begin{bmatrix} \Delta a_k^i \\ \Delta b_k^i \end{bmatrix} = \begin{bmatrix} 1 & 0 & -(b_k^i - b_j^i) \\ 0 & 1 & (a_k^i - a_j^i) \end{bmatrix} \begin{bmatrix} \Delta a_j^i \\ \Delta b_j^i \end{bmatrix}
$$

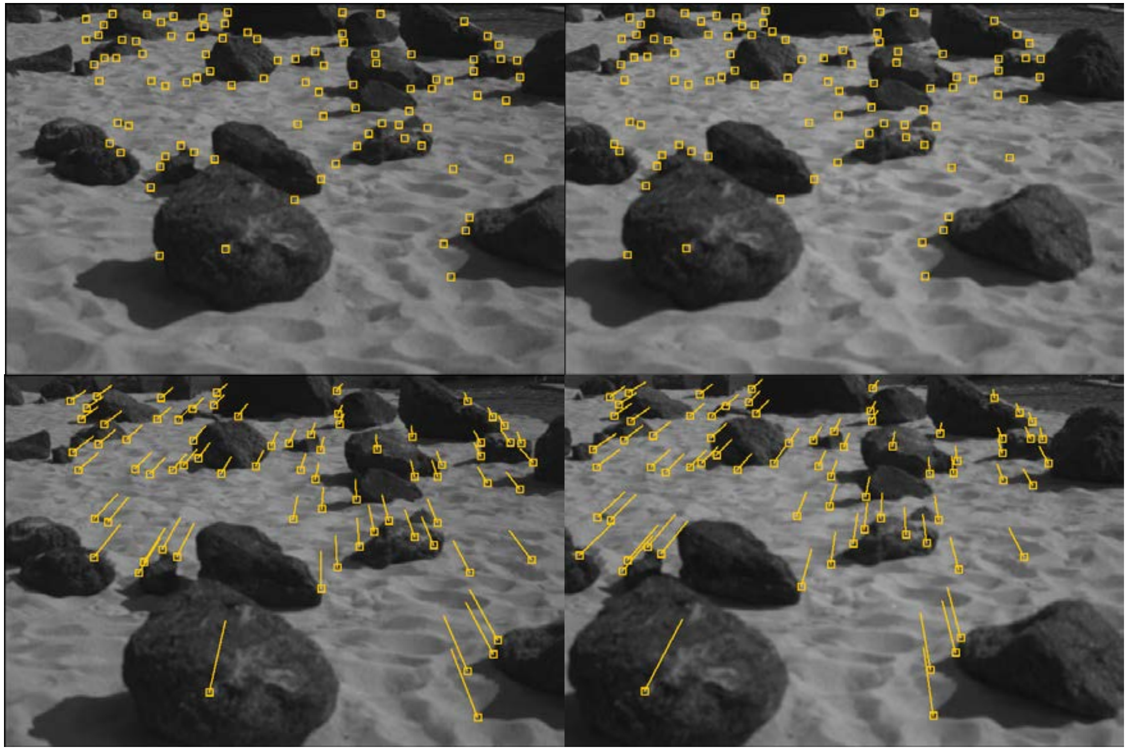
9.2.4.3.3 Visual Odometry

- Solve the same tracking problem but:
  - Flow is assumed to be caused by camera motion.
  - Goal is to find the camera motion.
- Essential mathematics are identical to pose refinement. BUT:
  - State vector represents
    - differential motion
    - in the scene
  - A secondary integration process usually computes position.

Features Everywhere

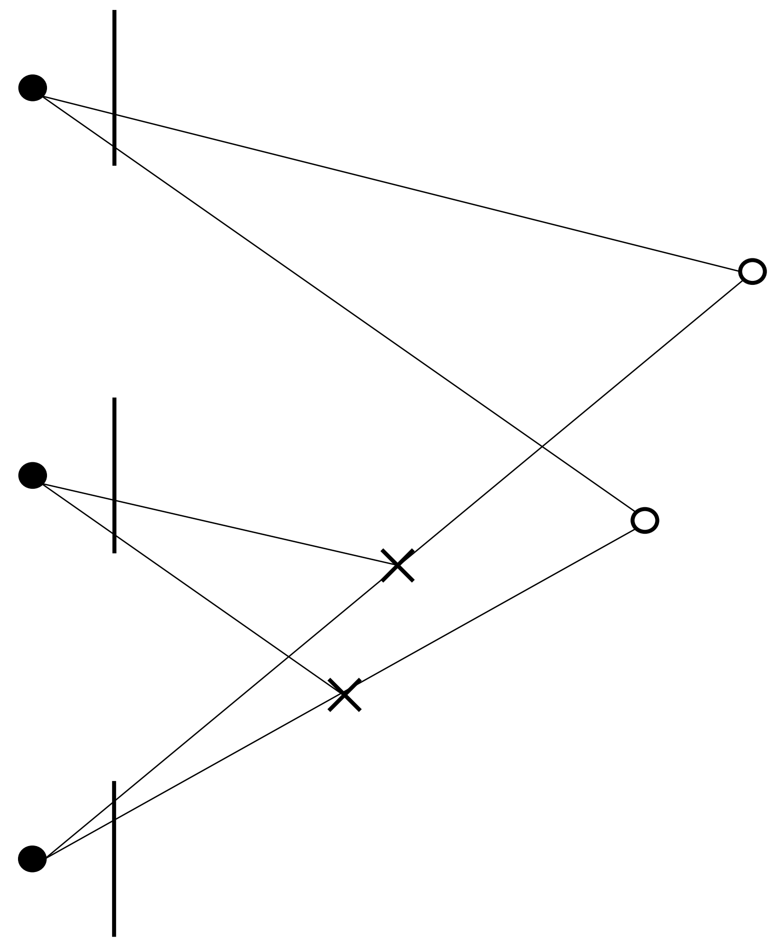
9.2.4.3.3 Visual Odometry

- In this case, the camera motion and the scene surfaces are 3D.
- It helps to have two cameras (stereo) to resolve the scale ambiguity problem.
- A secondary pose estiamate helps too.



9.2.4.3.3 Visual Odometry
(Projective Difficulties: [Monocular] Scale Ambiguity)

- Features at twice the depth are consistent with twice the translation.
  – no way to tell which of the top two cases is correct.
- However, orientation change can be measured without knowing depth.
  – If you knew the motion was rotation!
- Distinguishing rotation from translation is another problem.

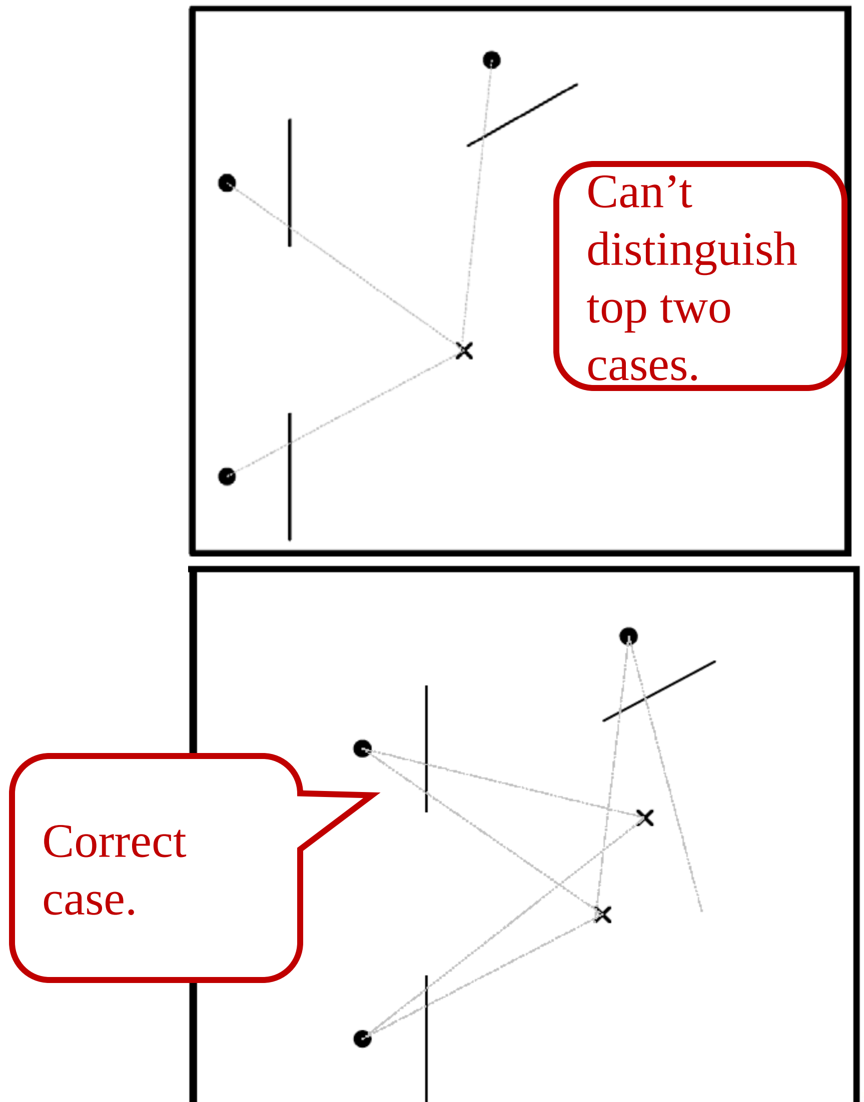


Carnegie Mellon
THE ROBOTICS INSTITUTE

9.2.4.3.3 Visual Odometry

(Projective Difficulties: Depth Determination)

- Some people use SFM to get the depth.
  - Vizodo is a special case where you “ignore” the shape output.
- Stereo is another alternative.
  - Need two features to determine 2D motion.
- A FOV wide enough to see well separated features helps for the problem of distinguishing rotation from translation.



Outline

- 9.2 Visual localization and Motion Estimation
  - 9.2.1 Introduction
  - 9.2.2 Aligning Signals for Localization and Motion Estimation
  - 9.2.3 Matching Features for Localization and Motion Estimation
  - 9.2.4 Searching for the Optimal pose
  - Summary

Summary

- Perception based positioning, rather than being esoteric, is a core capacity of capable mobile robots.
- The following four technologies are similar in substance but different in emphasis
  - Pose Refinement
  - Registration
  - Visual Tracking
  - Visual Odometry
- All rest on solutions to:
  - prediction
  - correspondence
  - registration

Summary

- Pose Refinement / Registration
  – Residuals between real and predicted features
- Visual Tracking / Odometry
  – Residuals between two sets of real feature locations.
- The existence of a prior map or model is a key distinction.
  – Prior maps make position estimation repeatable.
- ICP and template correlation are local association algorithms
  – use brute force search.