

Multiple Model Estimation : The EM Algorithm & Applications

Princeton University
COS 429 Lecture

Nov. 13, 2007

Harpreet S. Sawhney
hsawhney@sarnoff.com



Recapitulation

- Problem of motion estimation
- Parametric models of motion
- Direct methods for image motion estimation
- Camera models & parametric motion
- Image & video mosaicing as an application
- Quasi-parametric model-based motion & structure estimation : Depth and Pose
- Image-based Rendering
- Hi-res stereo sequence synthesis as an IBMR application

Plan

- Motivate Image-based Modeling & Rendering (IBMR)
 - Change in viewpoint, IMAX app
- Parameterize motion & structure for video
 - Euclidean case
 - Direct Estimation
- Plane+Parallax
 - Formulation
 - Direct Estimation
- IMAX app.
- Tweening app.
- Model-to-video pose estimation
- Video Flashlights

Real-World Apps of IBR

- The Matrix
- What Dreams May Come
- Titanic

Application : Dynamic New View Rendering

The Matrix

Flow-based New View Rendering



Original 8 frames



Tweened 71 frames

Enhanced Visualization

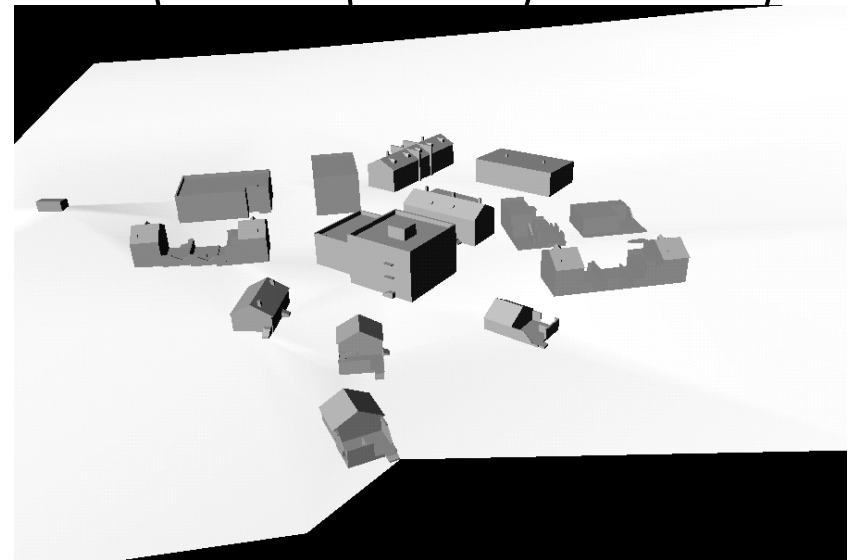
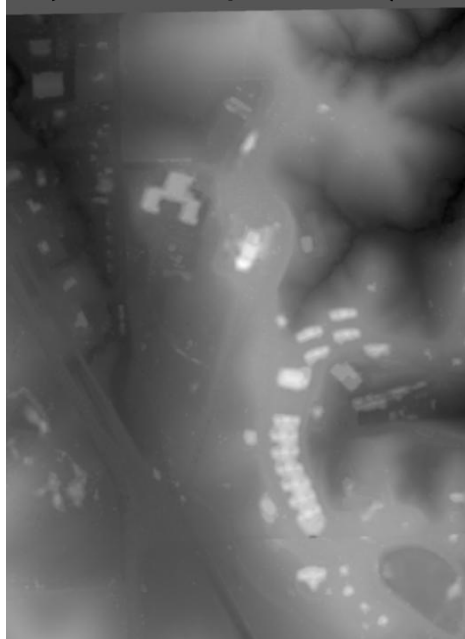
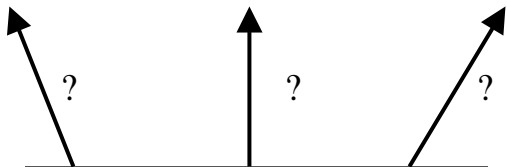


3D Model-based Direct Camera Pose Estimation and Video Visualization

[Hsu et al. '00]

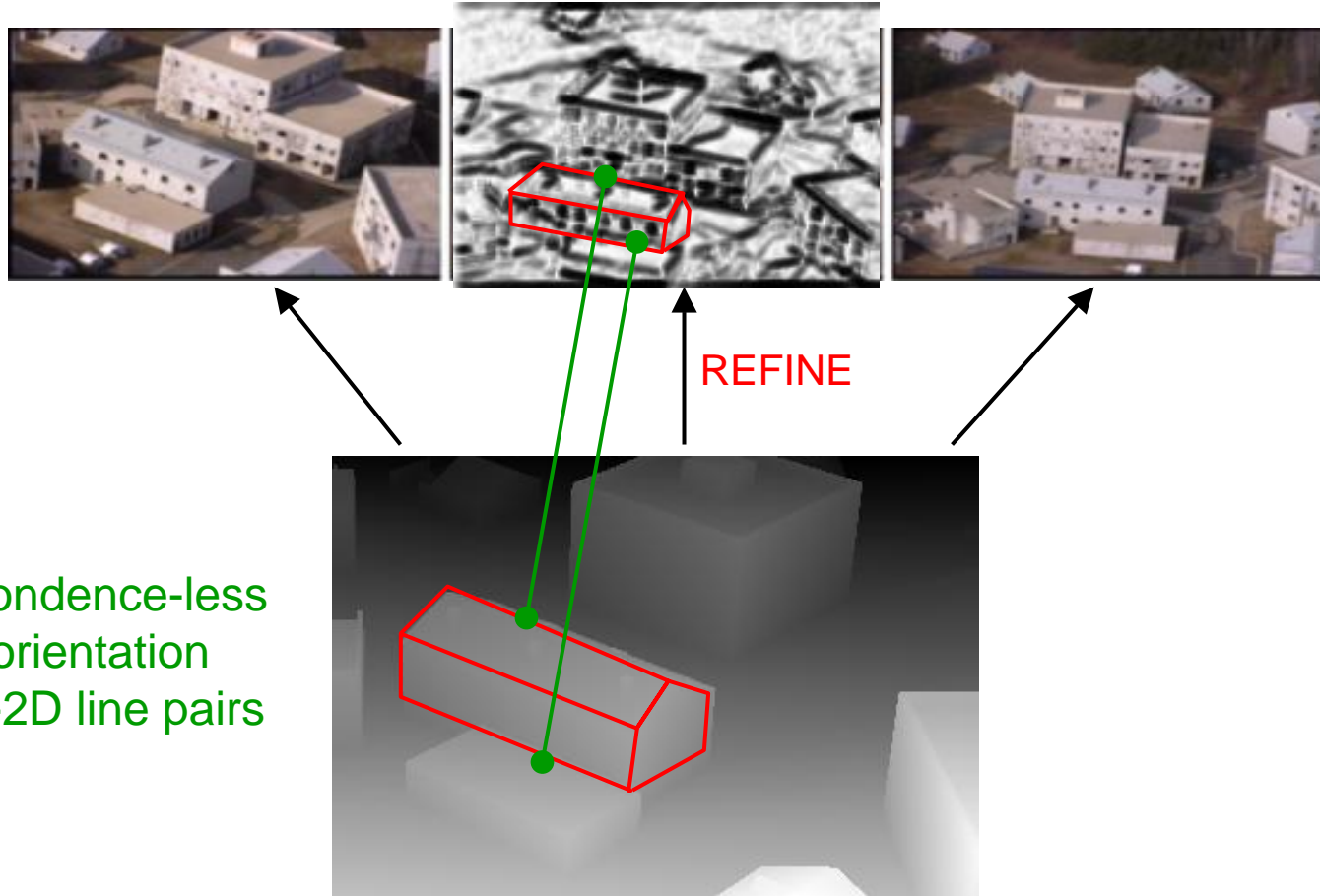
Pose Estimation

...when only shape of 3D scene is known ...



Video to Site Model Alignment

- **Model to frame alignment**

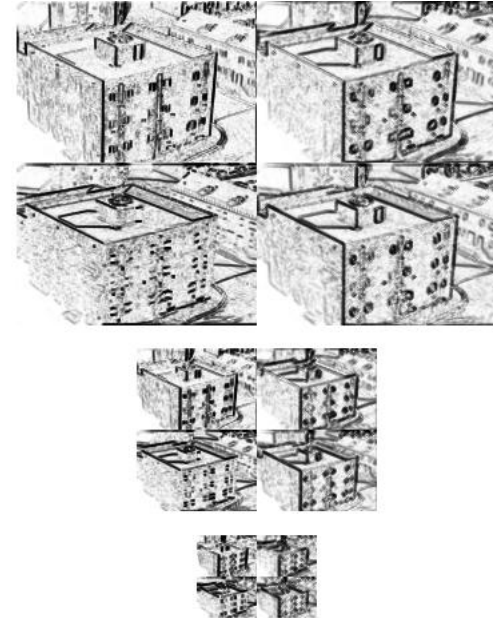
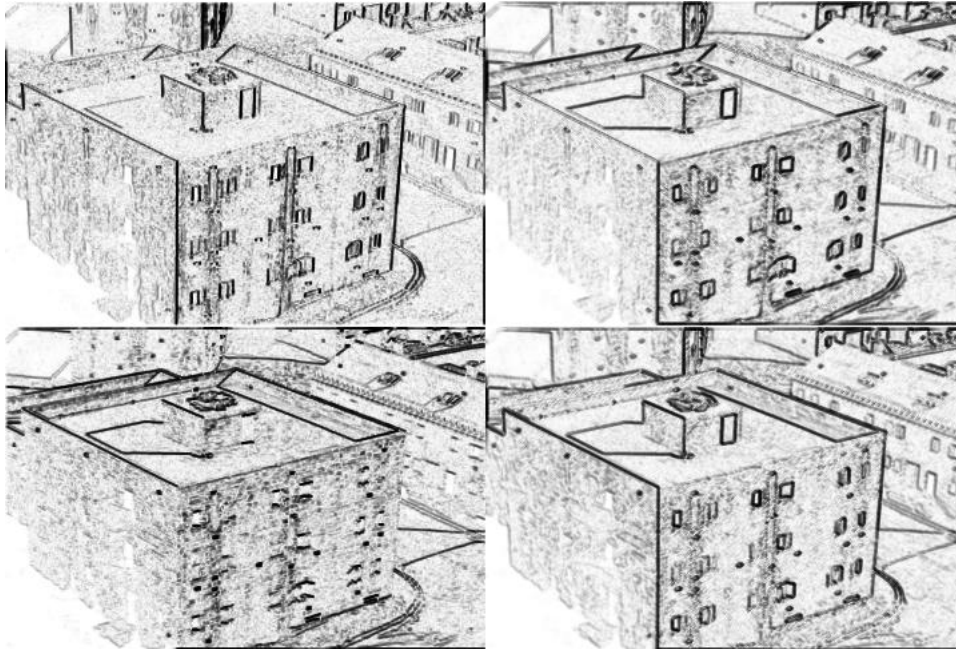


The REGSITE Algorithm

... aligning site model edges to image edges...

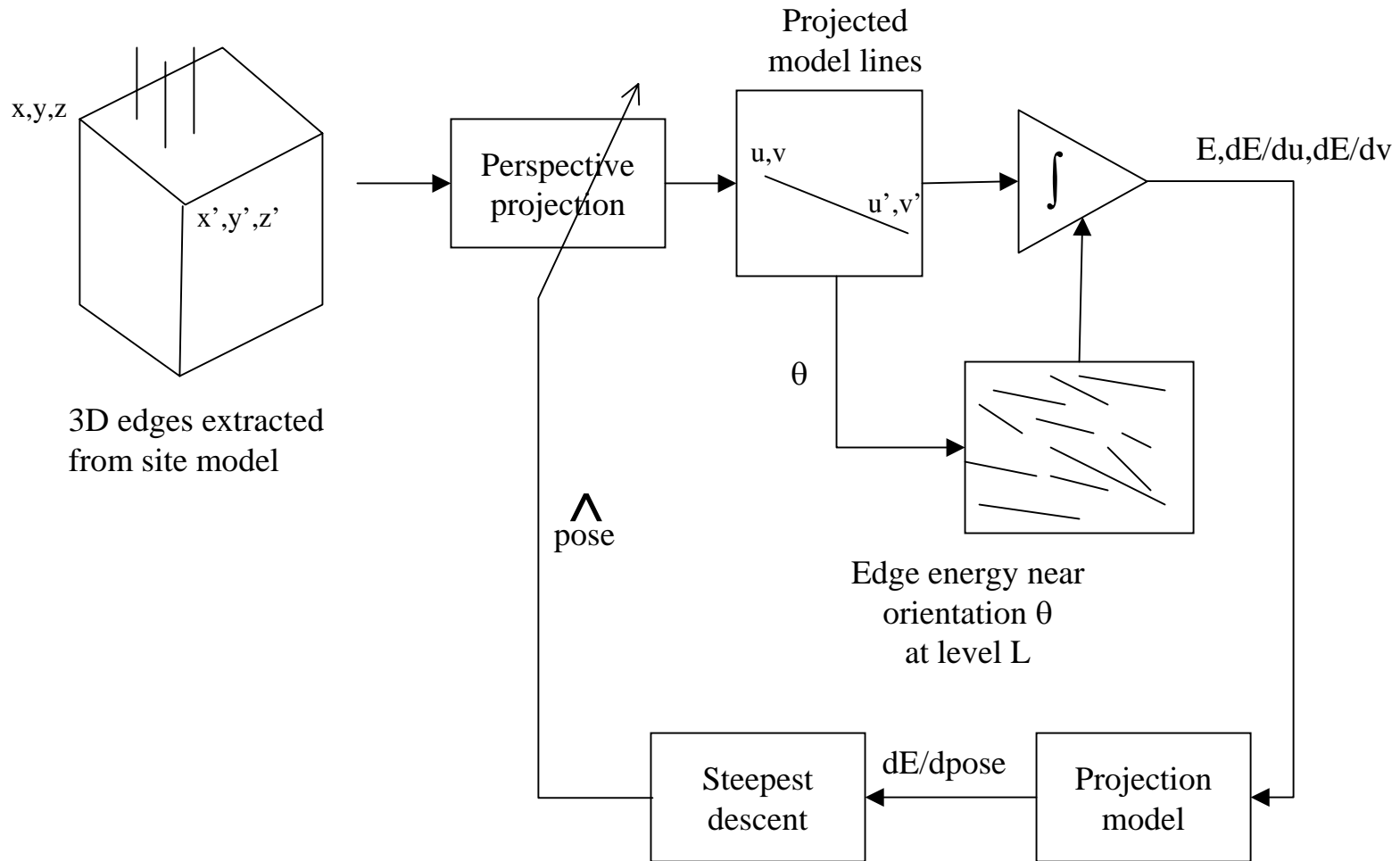
- **Inputs:**
 - Predicted pose of camera
 - Un-textured (Open Inventor) site model
 - Video frame
- **Output:**
 - Estimated pose of camera
- **Premise**
 - Discontinuities in 3D depth are correlated with brightness edges in the video frame (most of the time)
- **Approach**
 - Oriented energy image pyramids highlight image edges
 - Extract edges (depth discontinuities) from 3D site model
 - Adjust camera pose to maximize overlap of model and image edges
 - refinement is done using coarse to fine strategy over image pyramid

Oriented Energy Pyramid



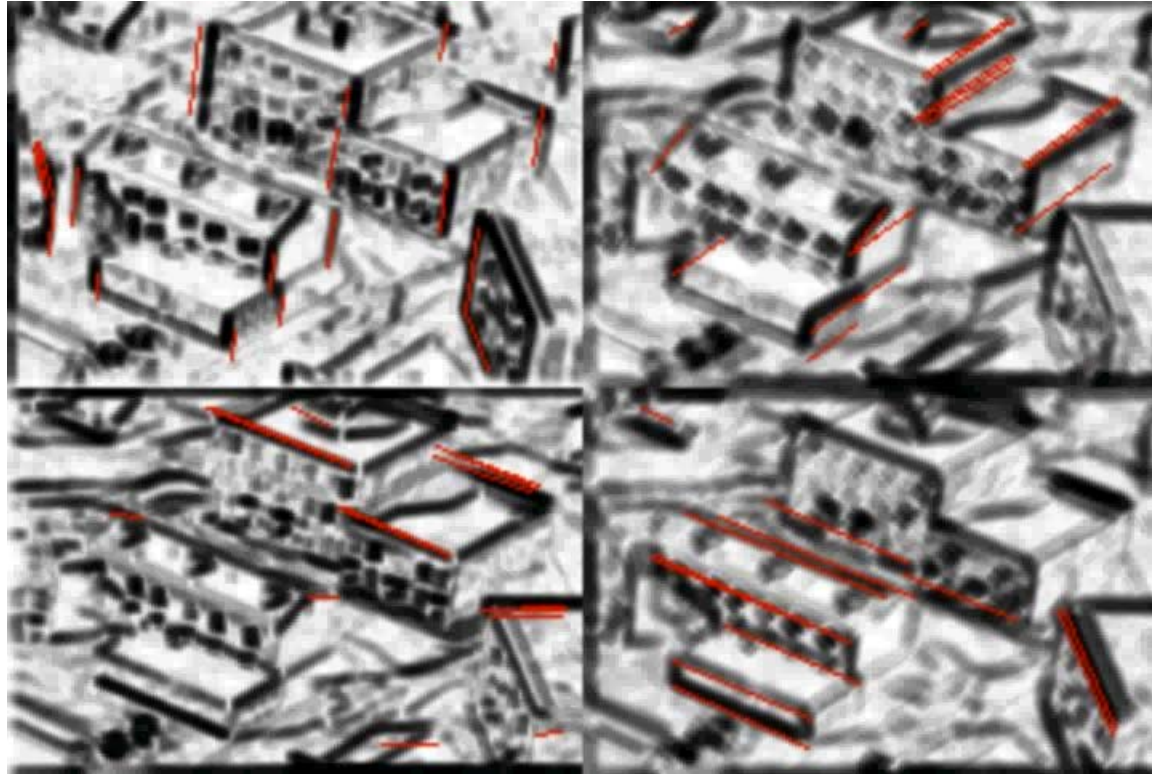
Oriented Energy Pyramid: 4 Orientation Bands 0 deg., 45 deg., 90 deg., 135 deg.

Pose Refinement Procedure



Pose Refinement Results

Iterative coarse-to-fine adjustment of pose



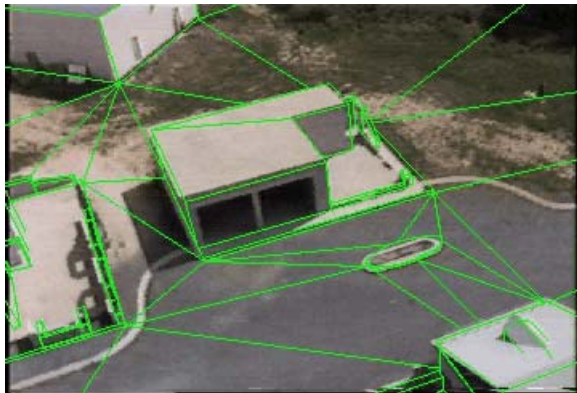
Geo-registration of Video Sequence from Draper Helicopter to 3D Site Models



Original Video



**Model rendered from the pose of the helicopter sensor.
Pose recovered after geo-registration process**



Overlay of site model on video



World as seen from the view-point of the runner

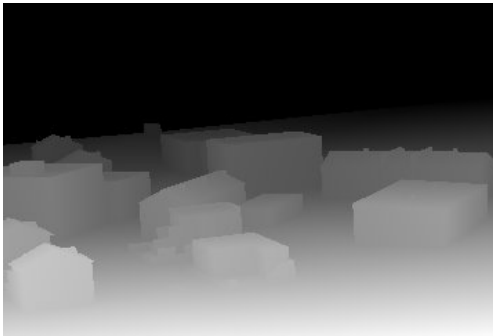
Re-projection & Enhanced Visualization of Video

Geo-registration of video to site models

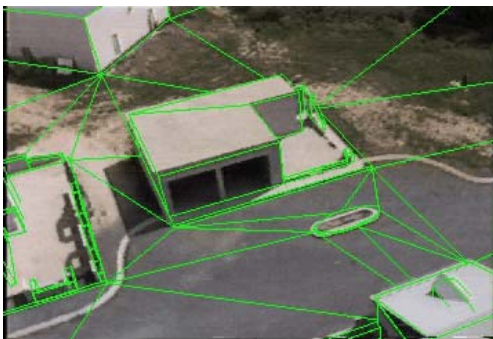
**Original
Video**



**Site
model**



**Geo-
registration
of video to
site model**

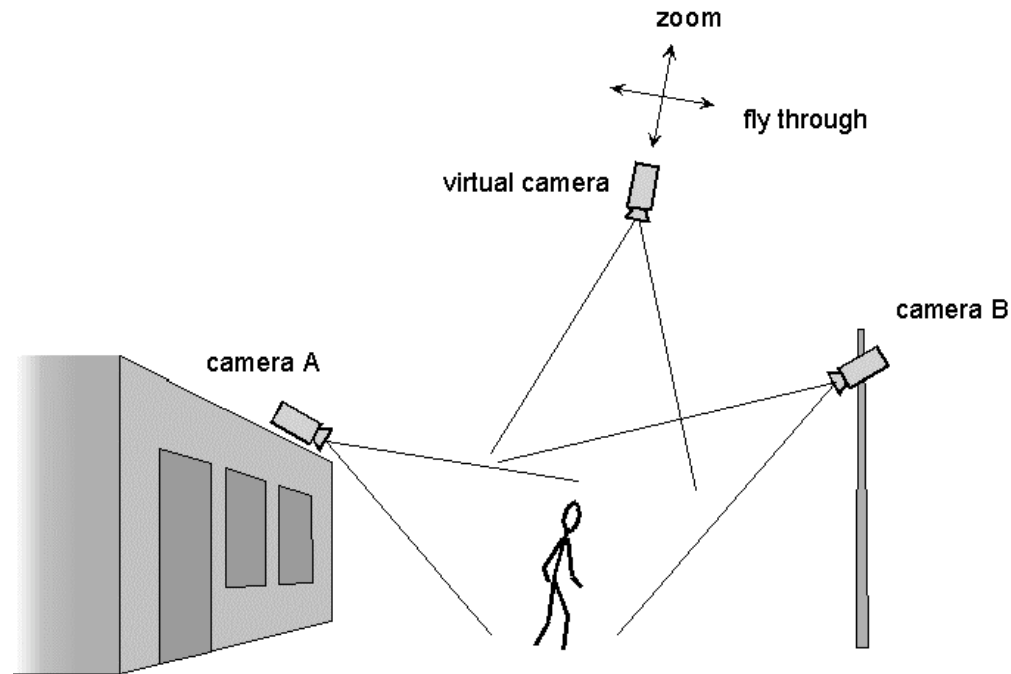


**Re-projection of video after merging
with model.**

Application : Model-based Video Visualization

Immersive and Interactive Telepresence

Total Facilities Visualization



Multiple cameras are merged to form a unified 3D scene representation. Each observer views the scene with his own “virtual” camera.

Distributed 2D Cameras



It is difficult to interpret activities viewed by multiple cameras

Video Flashlights Concept

[EGWR'02]

A tool for Global Visualization of Dynamic Environments

- **2D Video Flashlights:**
 - Project multiple 2D videos on a site model.
- **Moving Object Cued Video Flashlights:**
 - Project multiple 2D videos with automatically detected moving objects on a site model.
- **3D Video Flashlights:**
 - Project automatically extracted dynamic object models from multiple videos on a site model.

Video Flashlights: Moving Target Cueing

Moving objects (humans & vehicles) are detected and segmented from live camera videos
Shown as color coded dynamic visual cues from a bird's eye view
Accurate dynamic positioning w.r.t. the model provides a global context for the action



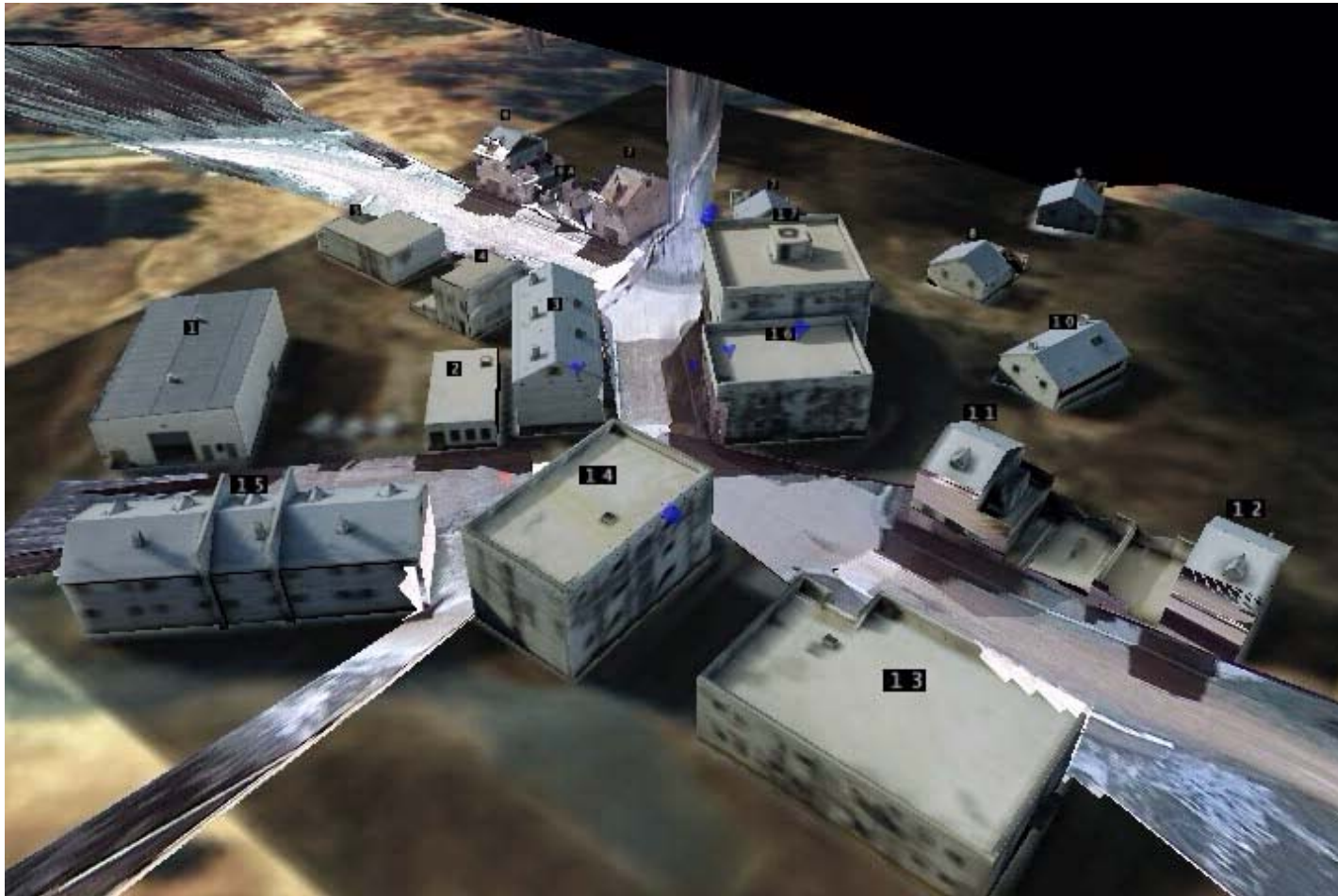
[VIDEO]

Moving Target Indication (MTI)

Video Flashlights

Live video streams are draped over a site model in real-time

Live videos are being viewed in the context of the model from a bird's eye view



Accurate Projection of multiple video streams onto the site model

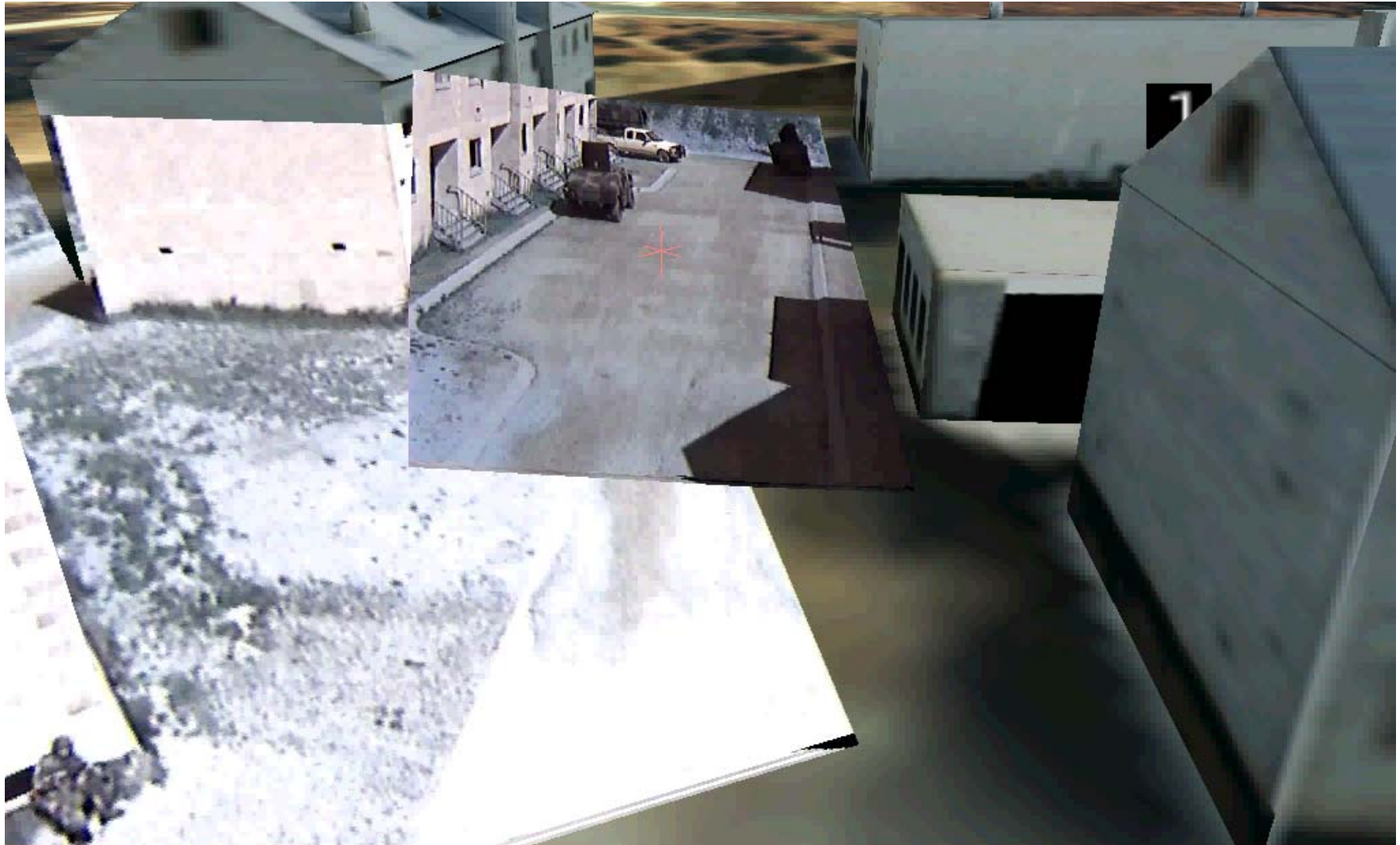
Enables interpretation of visual action in the global context of the model

Provides photo-realistic sky-to-street views at arbitrary scales and viewpoints

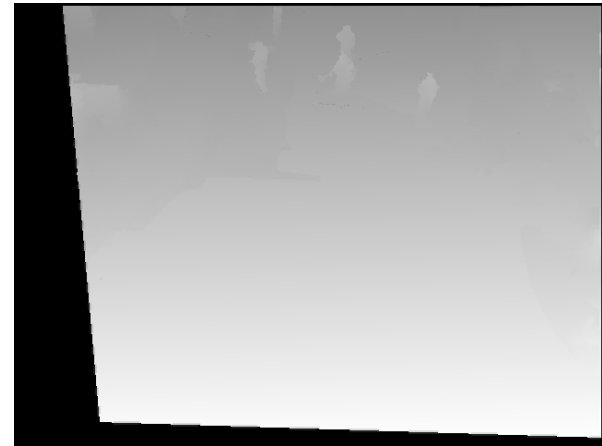
Video Flashlights

Close up view of multiple video streams draped over the site model

Close up view allows zooming onto action that is happening over multiple video cameras

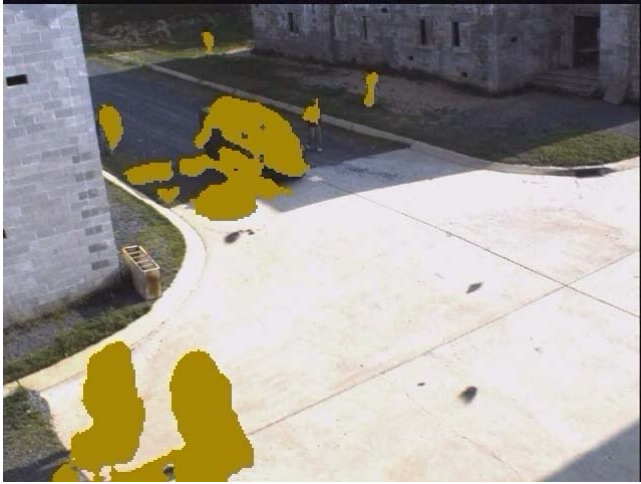


Depth Computation for continuous video streams

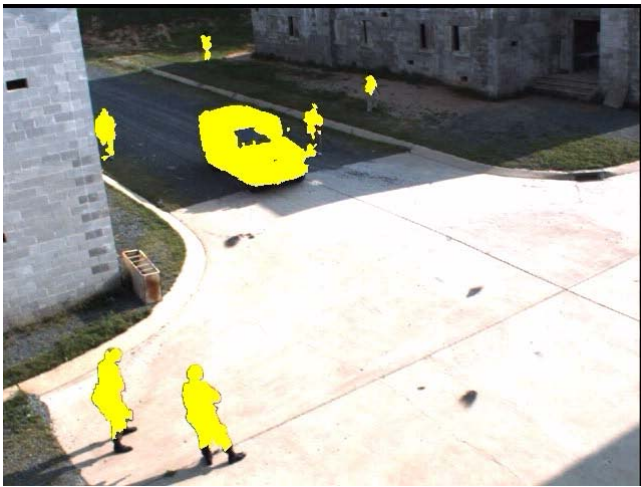


2D and 3D MTI : Results

- 3D method can separate shadows from moving objects
- 3D method provides better delineation of moving objects



2D MTI



3D MTI

Camera 1

Camera 2

Where are we headed ?

From Pixels
to
Intermediate Representations
for

Immersive Visualization

Immersive Communications

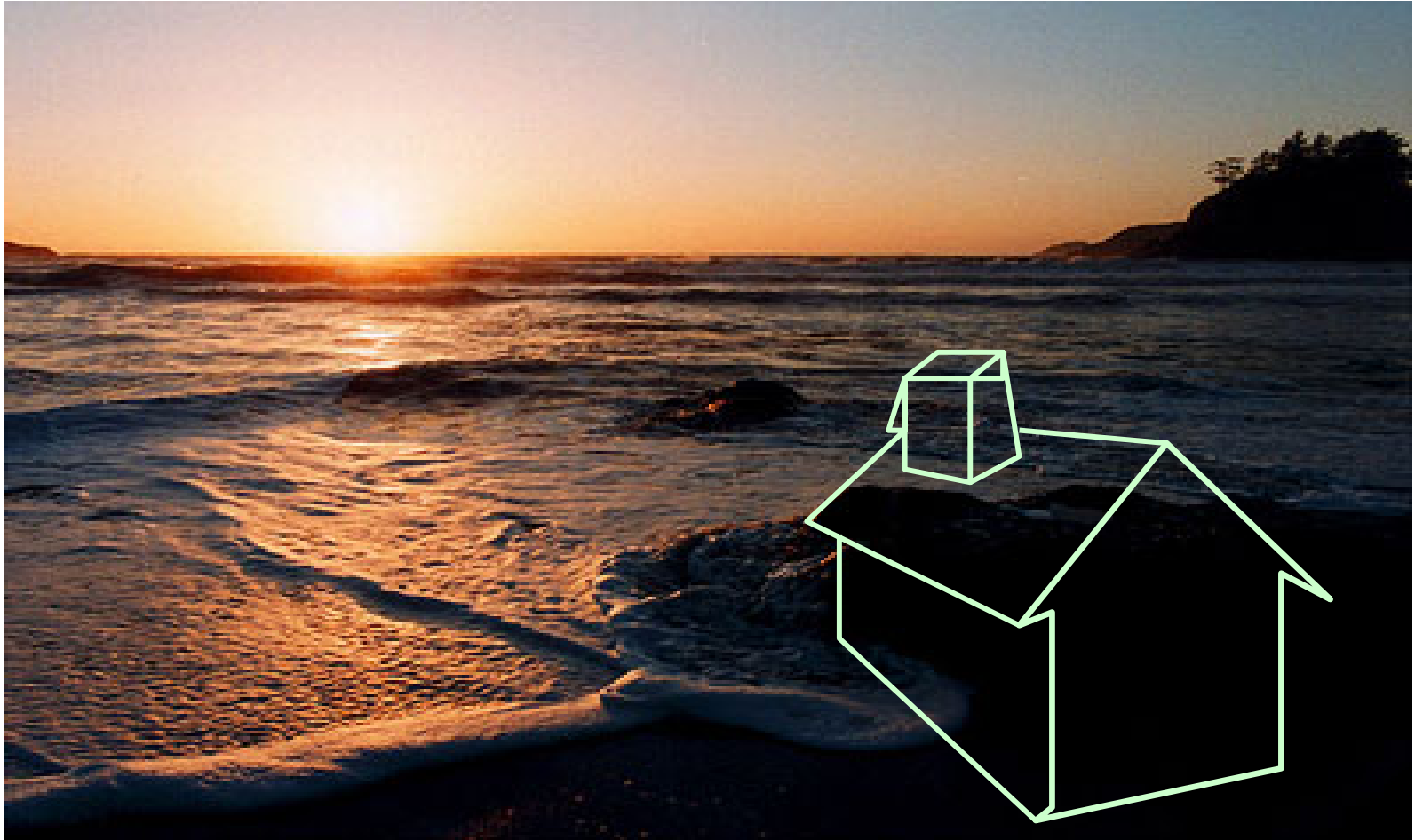
Perception

Object / Activity Recognition

Pattern Discovery

Cognition

AN IMMERSIVE IBMR GRAND CHALLENGE



AND IF WE DO IT RIGHT



Handling Moving Objects in 2D Parametric Alignment & Mosaicing

Multiple Motions : Robust Regression



Find the dominant motion while rejecting outliers

Estimating the Mean

$$r_i = d_i - \mu$$

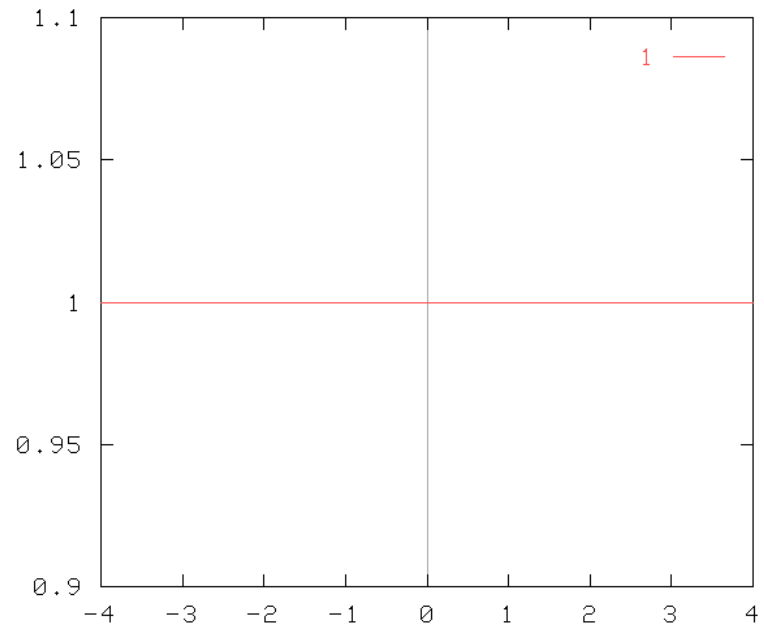
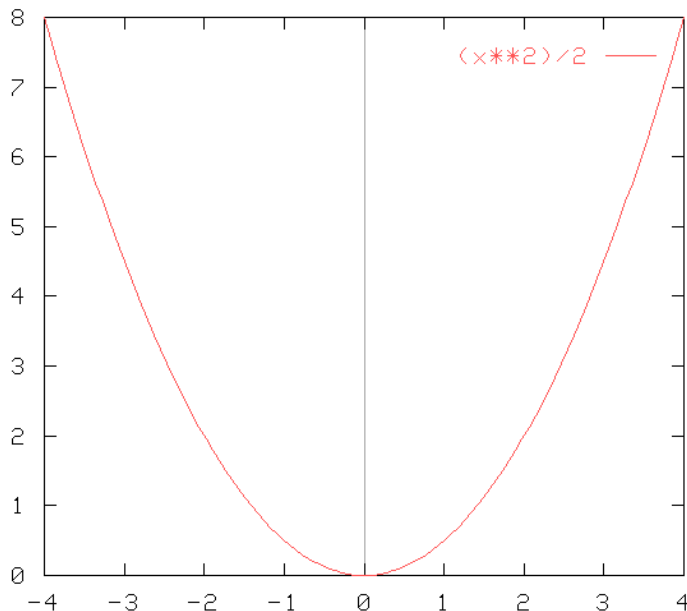
$$\min_{\mu} \sum_i (\rho(r_i) = r_i^2)$$

$$\mu = \frac{1}{N} \sum_i d_i$$

- Mean is the “least squares” estimate
- But each measurement is given the same weight
- Influence of an outlier: $\frac{\delta}{N}$

→ Can be arbitrarily large

Influence is given by : $\frac{\rho(r)}{r}$



Generalized M-Estimation

$$\min_{\Theta} \sum_i \rho(r_i; \sigma), \quad r_i = l_2(p_i) - l_1(p_i - u(p_i; \Theta))$$

- Given a solution $\Theta^{(m)}$ at the m th iteration, find $\delta\Theta$ by solving :

$$\sum_l \sum_i \left(\frac{\dot{\rho}(r_i)}{r_i} \right) \frac{\partial r_i}{\partial \theta_k} \frac{\partial r_i}{\partial \theta_l} \delta \theta_l = - \sum_i \left(\frac{\dot{\rho}(r_i)}{r_i} \right) r_i \frac{\partial r_i}{\partial \theta_k} \quad \forall k$$

\mathbf{w}_i

- \mathbf{w}_i is a weight associated with each measurement. Can be varied to provide robustness to outliers.

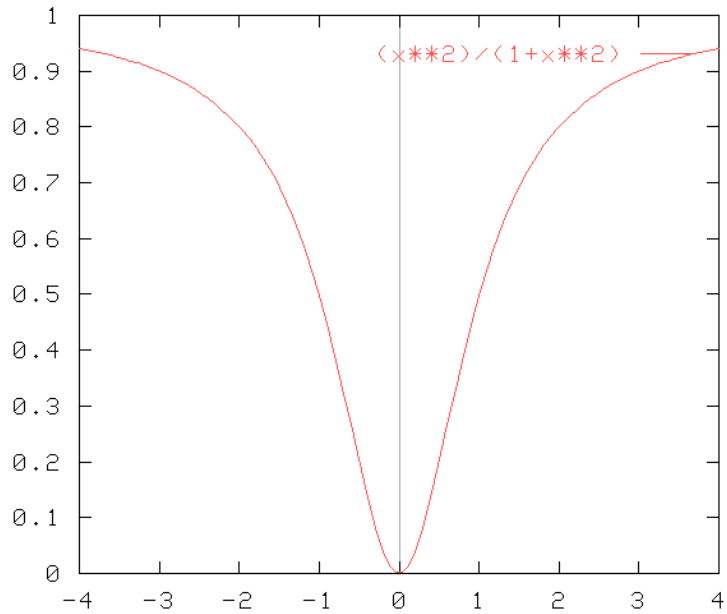
Choices of the $\rho(r_i; \sigma)$ function:

$$\rho_{SS} = \frac{r^2}{2\sigma^2} \quad \rho_{GM} = \frac{r^2/\sigma^2}{1+r^2/\sigma^2}$$

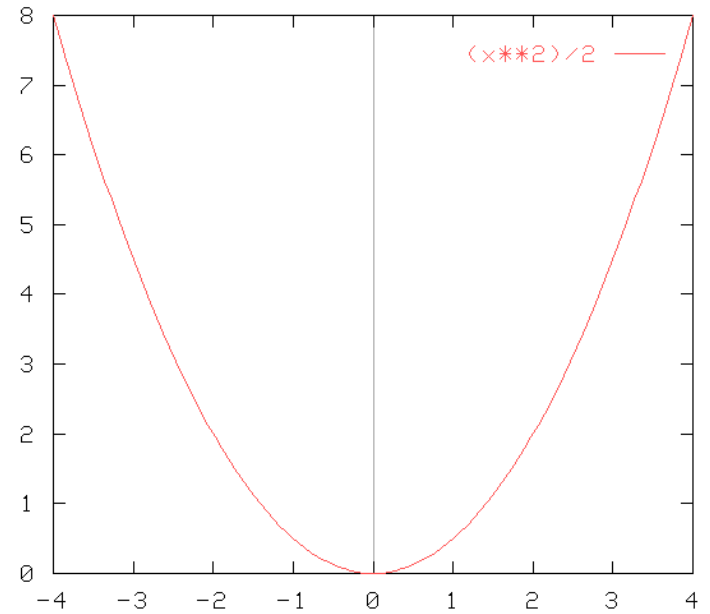
$$\frac{\dot{\rho}_{SS}(r)}{r} = \frac{1}{\sigma^2}$$

$$\frac{\dot{\rho}_{GM}(r)}{r} = \frac{2\sigma^2}{(\sigma^2 + r^2)^2}$$

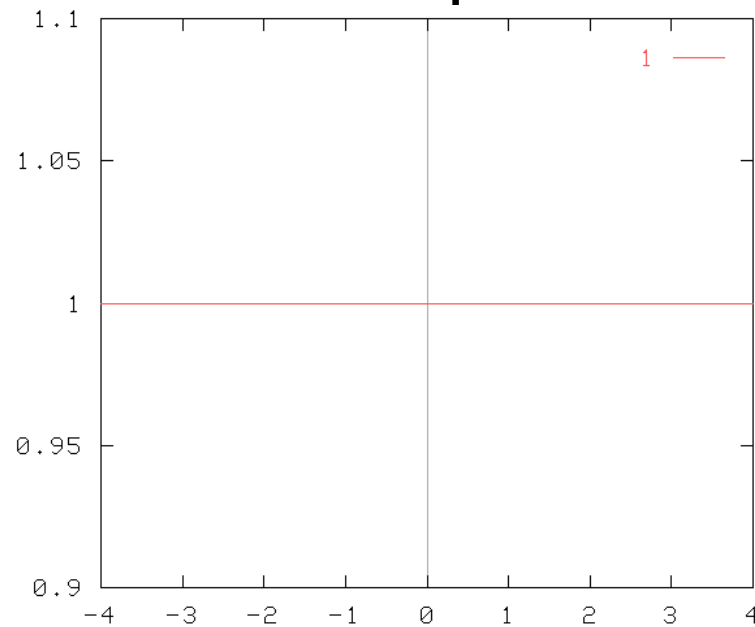
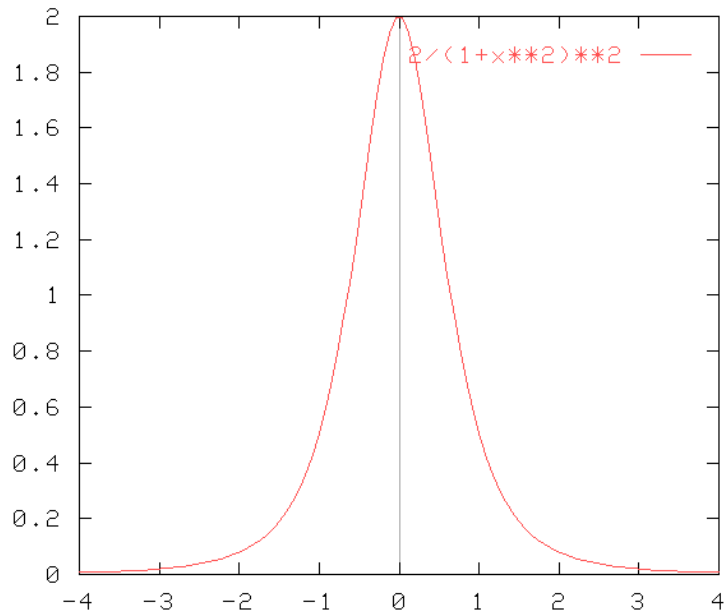
Optimization Functions & their Corresponding Weight Plots



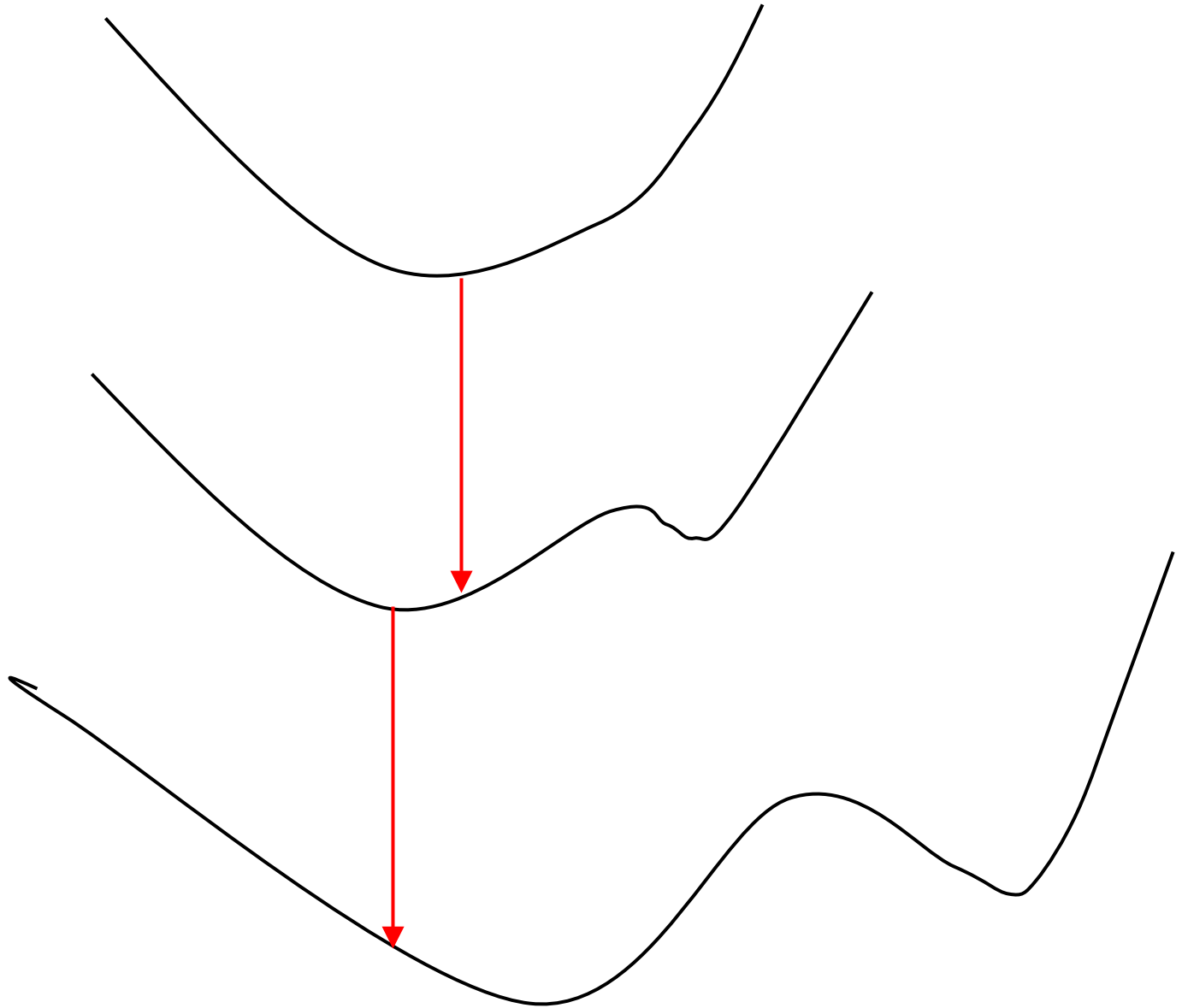
Geman-McClure



Sum-of-squares



Continuation Method: Coarse-to-fine



With Robust Functions Direct Alignment Works for Non-dominant Moving Objects Too



Original two frames



Background Alignment

Object Deletion with Layers

Original Video



**Video Stream with
deleted moving object**



DYNAMIC MOSAICS

Original Video



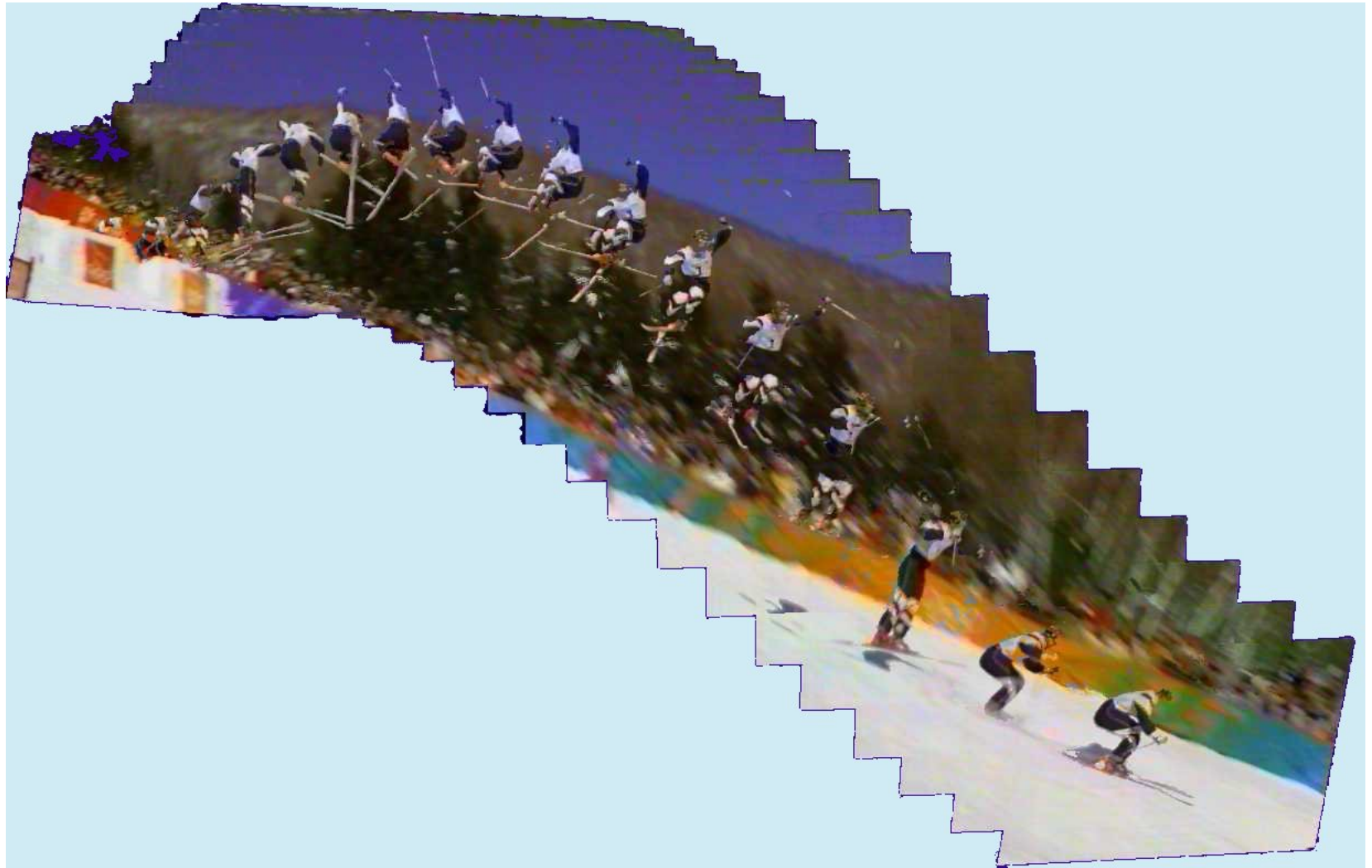
**Video Stream with
deleted moving object**



Dynamic Mosaic Video



SYNOPSIS MOSAICS



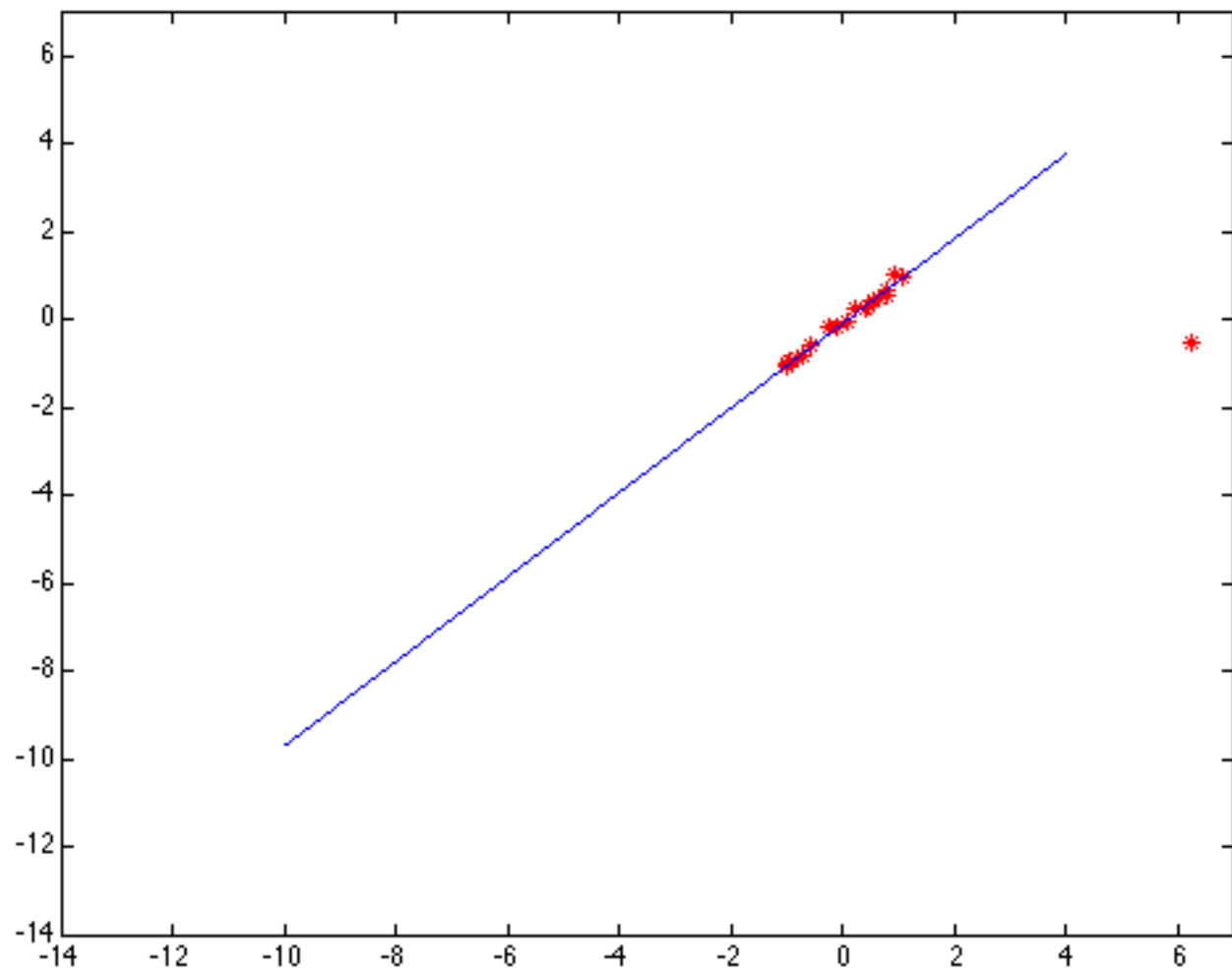
Problem

- Assumption:
 - Constraints that do not fit the dominant motion are treated as outliers : Extreme noise
- Problem:
 - But they are not noise
 - There indeed are multiple motions present in the scene

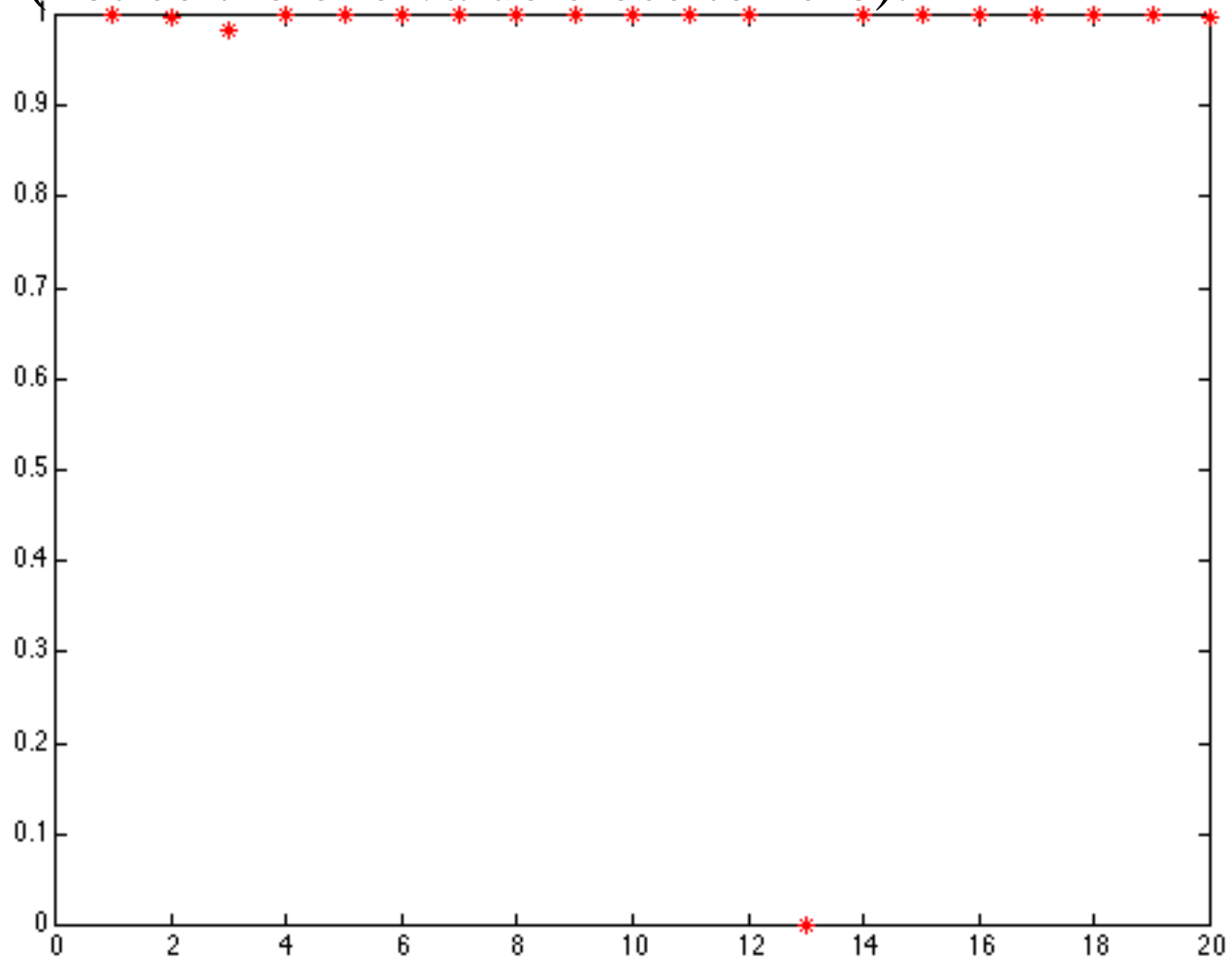
Motivate Simultaneous Multiple Model Estimation

Motivating Multiple Models

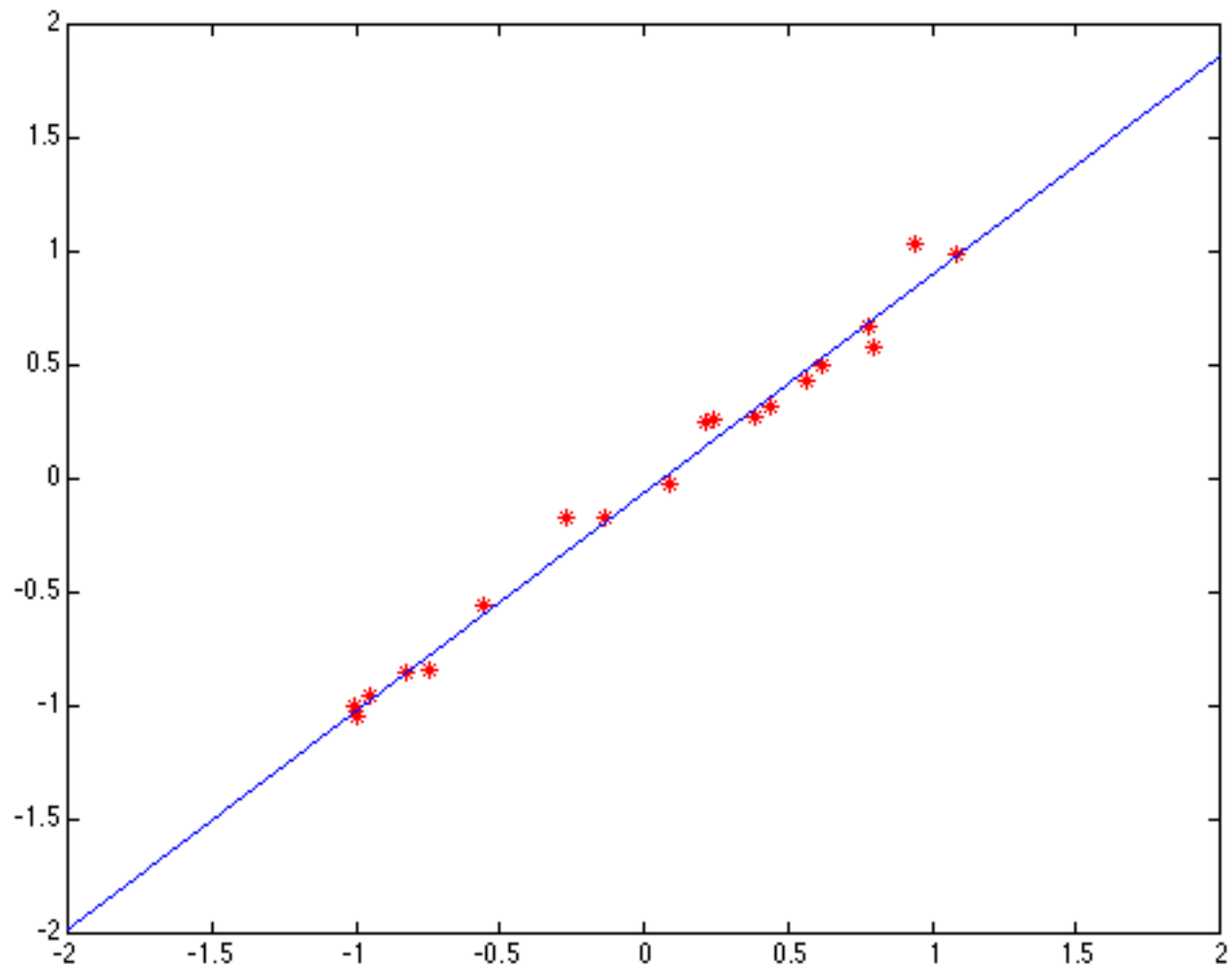
Line Fitting



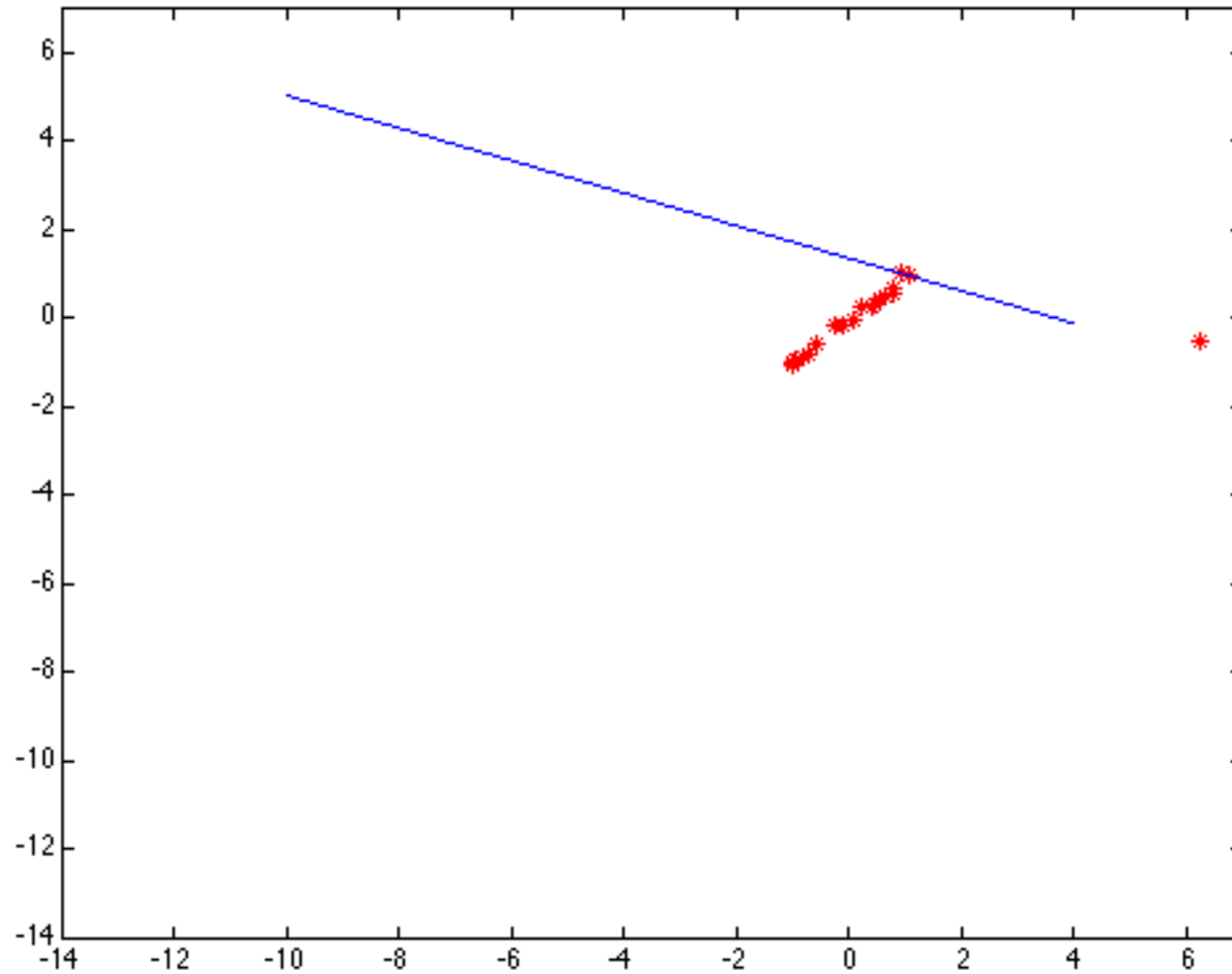
The expected values of the deltas at the maximum
(notice the one value close to zero).



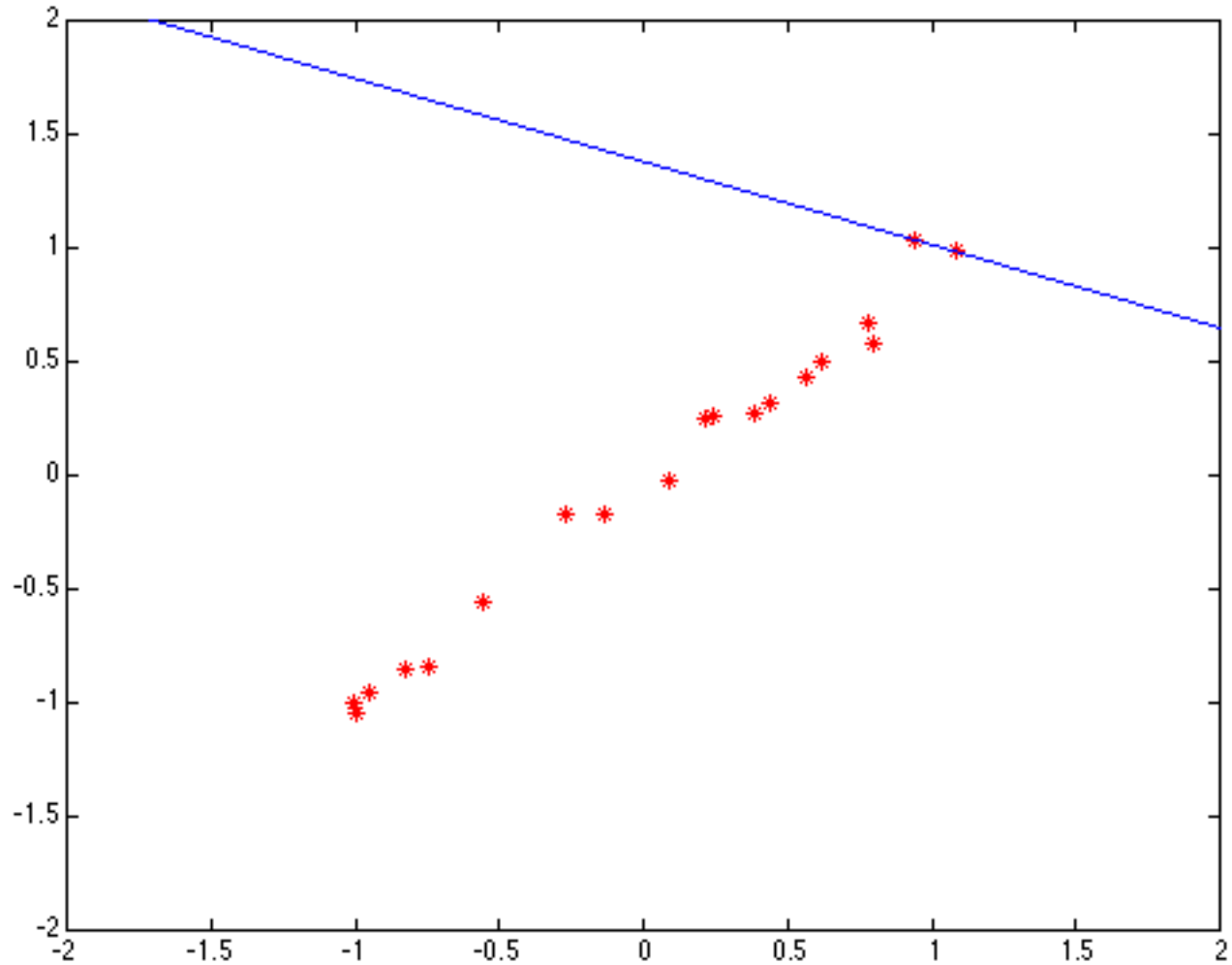
Closeup of the fit



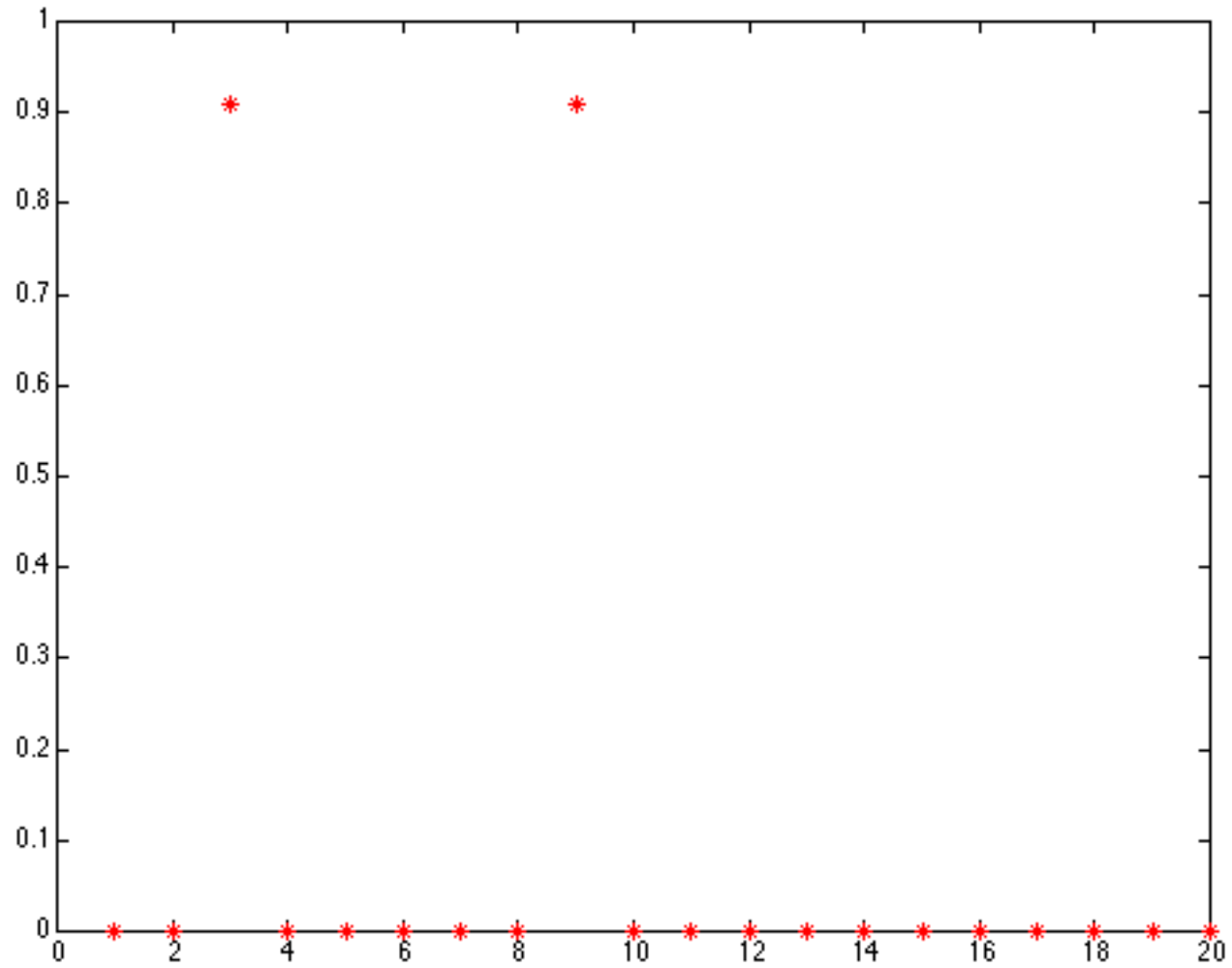
Local maximum



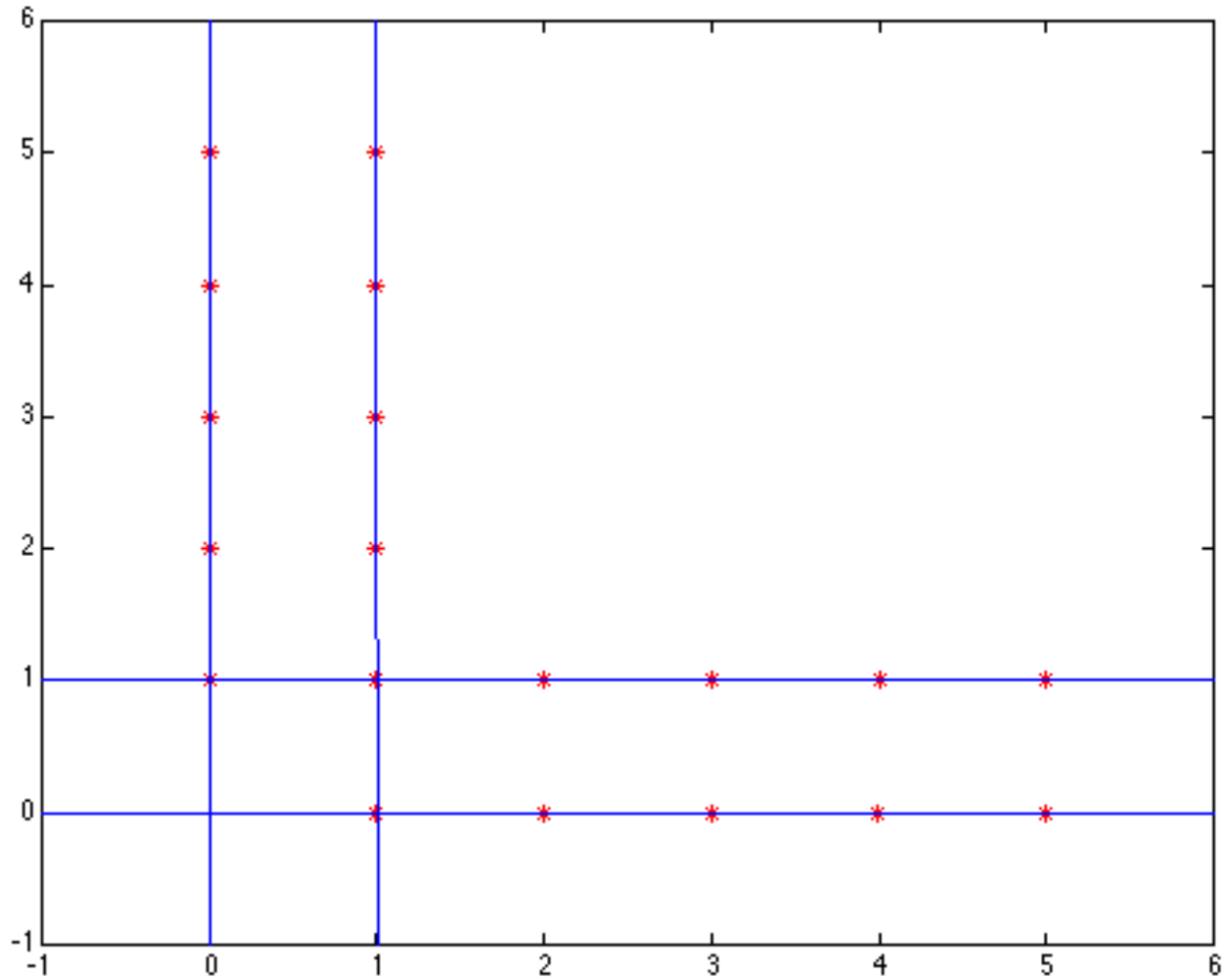
which is an excellent fit to some points



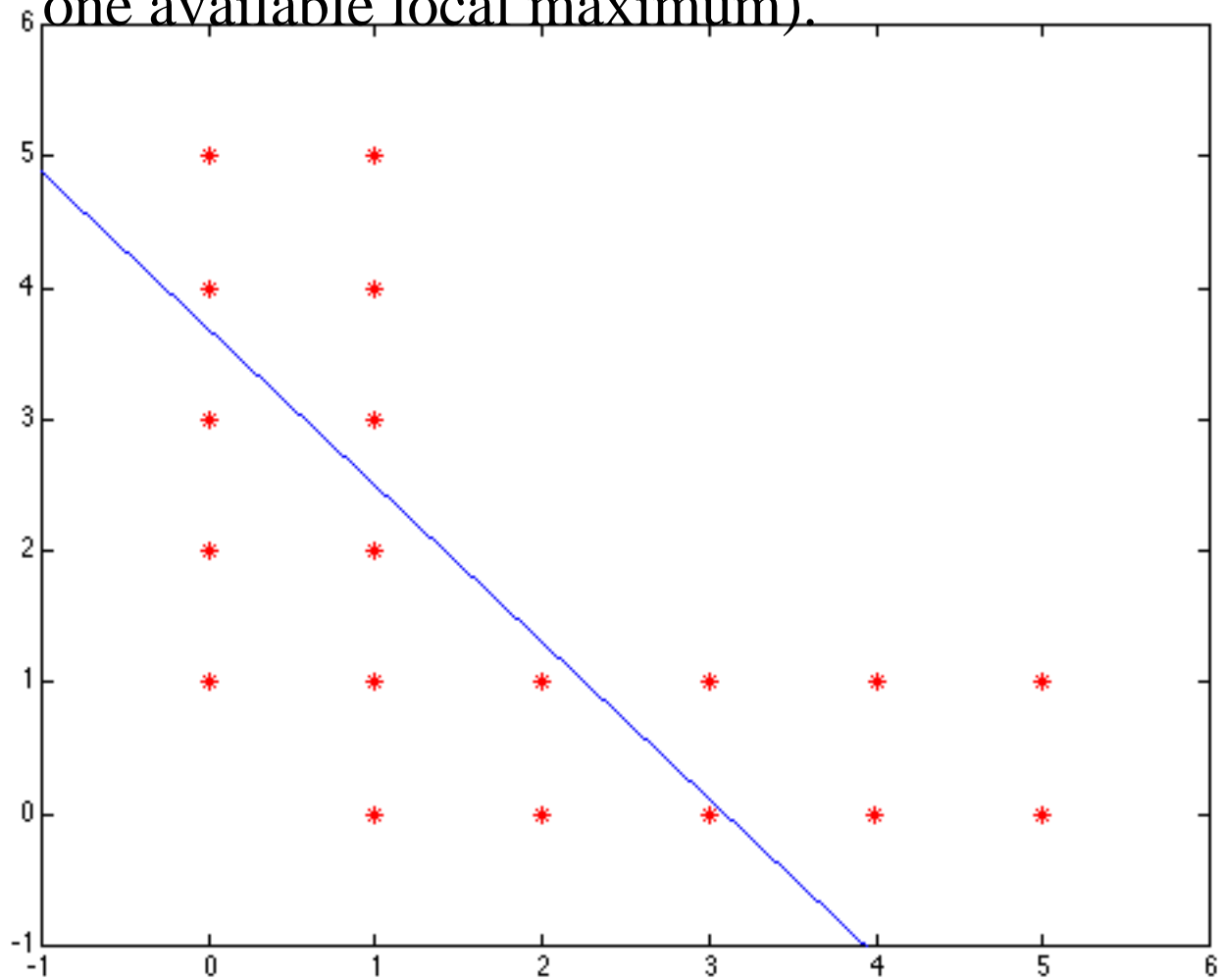
and the deltas for this maximum



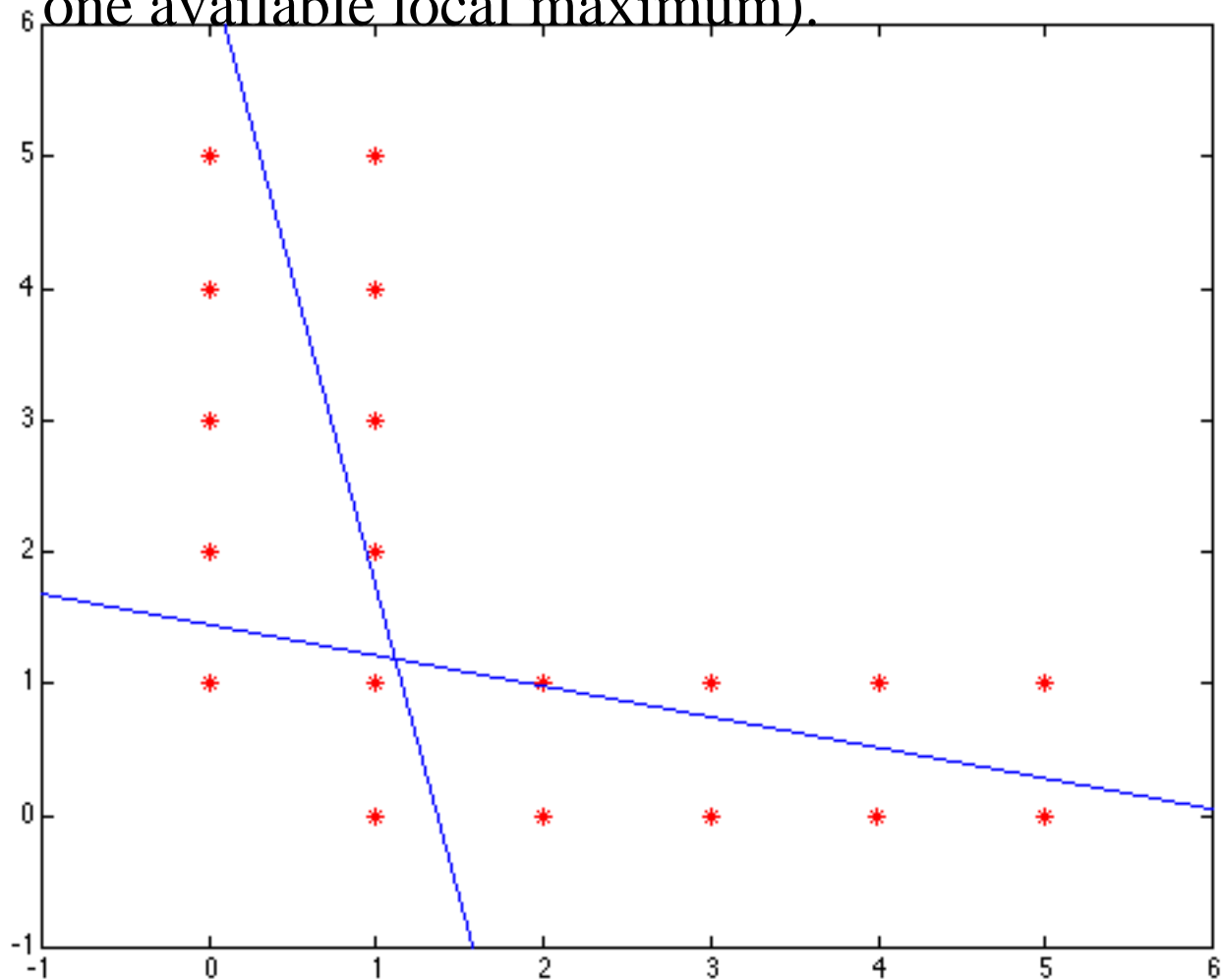
A dataset that is well fitted by four lines



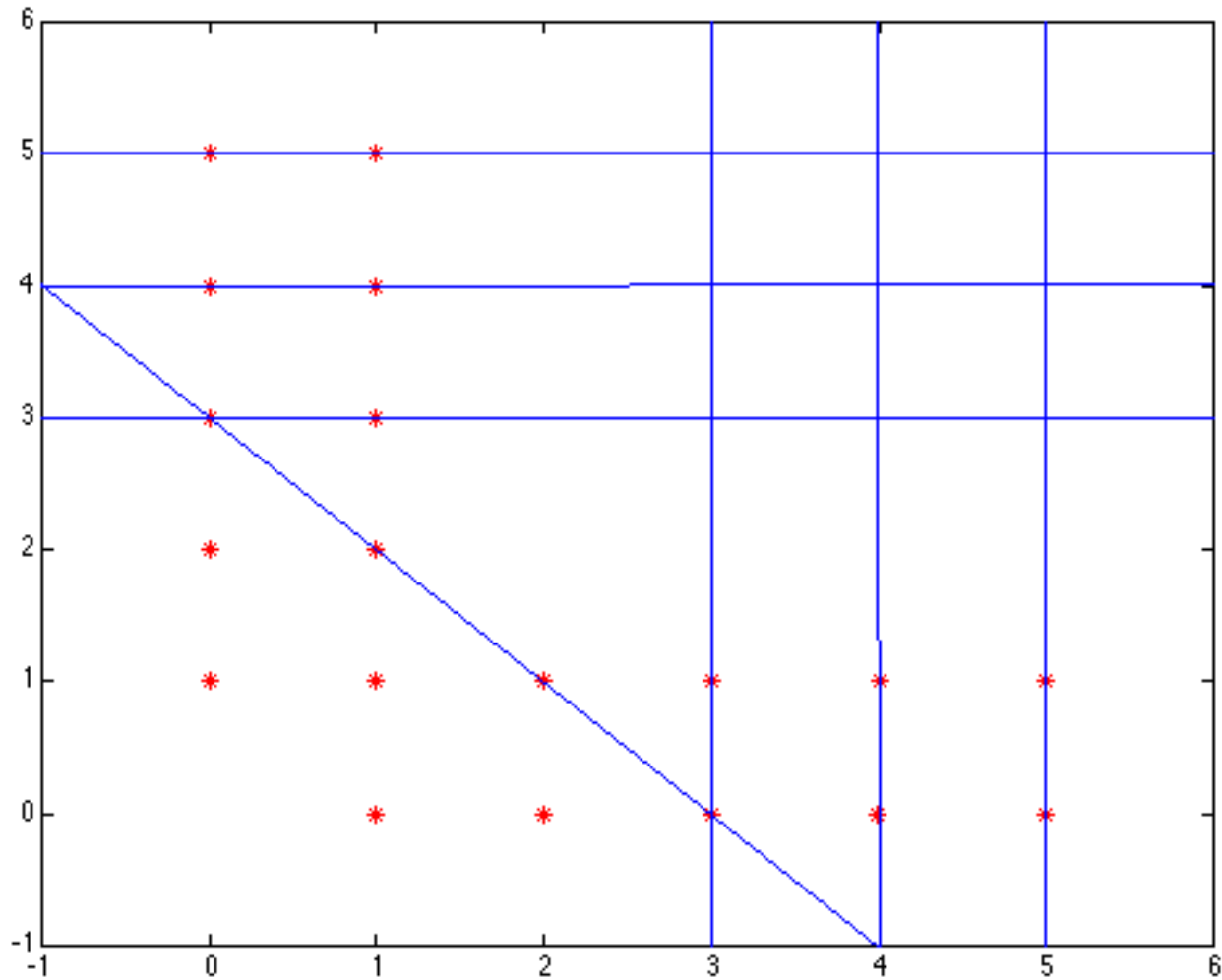
Result of EM fitting, with one line (or at least, one available local maximum).



Result of EM fitting, with two lines (or at least, one available local maximum).



Seven lines can produce a rather logical answer



Motivating Multiple Models

Multiple Motions

Independent Object Motion



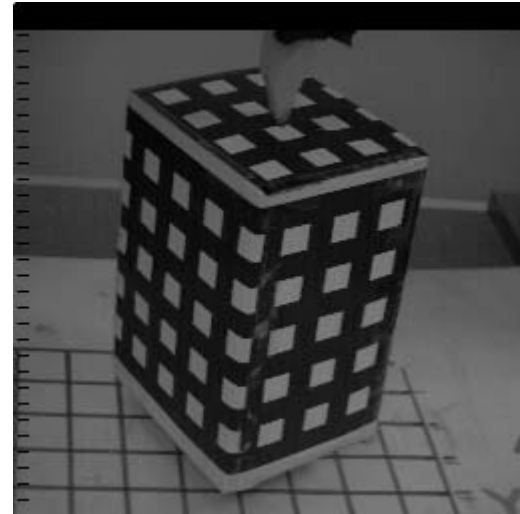
**Objects are the Focus
Camera is more or less steady**

Independent Object Motion with Camera Pan



**Most common scenario
for
capturing performances**

Multiple Motions may not be due only to independent object motions but due to different surfaces
Or
“ Motion Layers ”

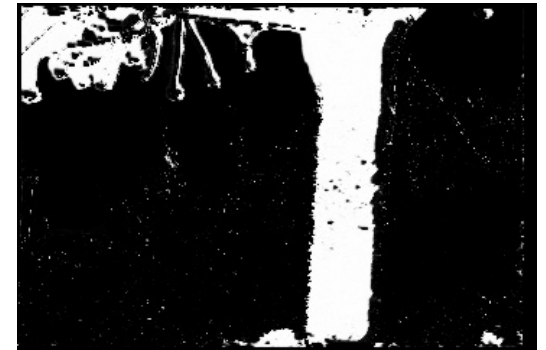


Multiple Motions as a Segmentation & Estimation Problem

- If we know which pixels go with what motion, can apply the now well-known methods of motion estimation to compute the motions
- Alternatively, given the motion parameters, potentially can label pixels corresponding to each of the motions.

Represent Multiple Motions as Layers

Layers

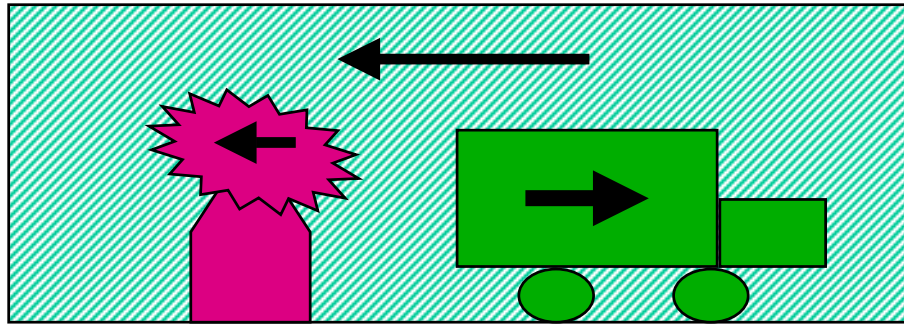


Input Sequence



Compact Video Representation

...motion and scene structure analysis...



Separate **coherent** & **significant** motion & structure components

- **Coherence** : Align images using 2D/3D models of motion and structure
Separate backgrounds and moving objects with layers
- **Significance** : Regions of support for various motion & structure components

MULTIPLE 2D PARAMETRIC MODEL ESTIMATION

... layered scene representation ...

THREE ISSUES

- How many models ?
- What are the model parameters ?
- What is the spatial support layer for each model ?

Competitive Multiple Model Estimation

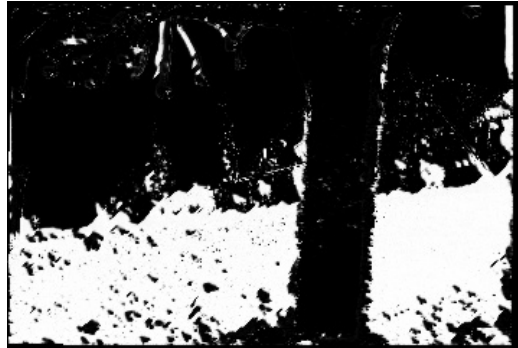
[Ayer, Sawhney '95 '96]

- Model image motion in terms of a **mixture of Gaussian** models
- **Layers of support** represented as ownership probabilities
- Robust Maximum-Likelihood estimation of mixture and layer parameters using the **Expectation-Maximization** algorithm
- **Minimum Description Length** (MDL) encoding to select adequate number of models

Automatic Layer Extraction : Intuition



Input Sequence



Layer



Motion

Assume that the segmentation of pixels into layers is known,
then estimating the motion is *easy*.

$$E_{\text{SSD}}(\mathbf{u}; \mathbf{A}_i) = \sum_{\mathbf{p} \in \mathbf{R}} w_i(\mathbf{p}) (\nabla I_1^T \mathbf{u}(\mathbf{p}; \mathbf{A}_i) + \delta I(\mathbf{p}))^2$$

Where do we get the weights from ?



Input Sequence

Layer

Motion

Model each pixel as potentially belonging to N layers each with its own motion model.

Assume that we know the motion model, but not the pixel ownership to the model

$$E_{\text{SSD}}(\mathbf{u}; \mathbf{A}_i) = \sum_{\mathbf{p} \in \mathbf{R}} w_i(\mathbf{p}) (\nabla I_1^T \mathbf{u}(\mathbf{p}; \mathbf{A}_i) + \delta I(\mathbf{p}))^2$$

Where do we get the weights from ?



Input Sequence

Layer

Motion

Each pixel has a likelihood associated with a motion model and the two images

$$L(I_2(p) | I_1(p), A_i) = \frac{1}{\sqrt{2\pi} \sigma} \exp\left(-\frac{(\nabla I_1^T u(p; A_i) + \delta I(p))^2}{2\sigma^2}\right)$$


$$w_i(p) = \frac{L(I_2(p) | I_1(p), A_i)}{\sum_i L(I_2(p) | I_1(p), A_i)}$$

Multiple Models : Mixture Models

- Model an image as a density function created using a mixture of Gaussian models conditioned on the adjacent images :

$$f(I(\mathbf{x},t) | I(\mathbf{x},t-1), \Phi) = \sum_{i=1}^g \pi_i p(I(\mathbf{x},t) | I(\mathbf{x} - \mathbf{u}(\mathbf{x}, \Theta_i), t-1), \sigma_i)$$

No. of models



The mixture model is parameterized by

$$\Phi = [\Pi, \Sigma, \Theta]$$

Θ : Ensemble of **motion parameters** for the g models

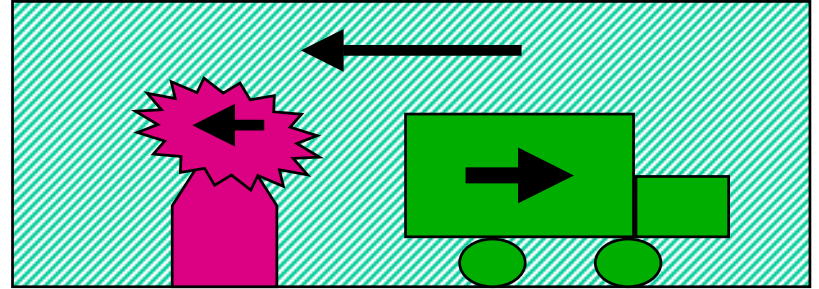
Σ : Ensemble of the **Gaussian distribution parameters** for the models

Π : Ensemble of the **ownership layers** for the models

Mixture Models

... represent layer ownerships as binary hidden variables ...

- $\mathbf{Z} = \{z_i(p_j), i = 1..g, j = 1..N\}$



is a set of binary indicator variables representing the model labels.

- The stochastic model for the complete data, measurements and hidden variables is:

$$\mathbf{f}(\mathbf{I}, \mathbf{Z} | \Phi) = \underbrace{\mathbf{f}(\mathbf{I} | \mathbf{Z}, \Phi)}_{\text{Observation Likelihood}} \underbrace{p(\mathbf{Z} | \Phi)}_{\text{Prior on the labels}}$$

Observation Likelihood

Prior on the labels

Mixture Model Estimation

The Expectation-Maximization (EM) Algorithm

- Maximize the **negative log-likelihood** of the parameters given the observations :

$$L(\Phi | I, Z) = -\log(f(I, Z | \Phi))$$

- Define the **expectation** of the likelihood :

$$Q(\Phi | \hat{\Phi}^{(k)}) = E[L(\Phi | I, Z) | I, \hat{\Phi}^{(k)}]$$

The diagram illustrates the decomposition of the EM Q-function into two main components: Mixing Proportions and Ownership. The equation is as follows:

$$Q(\Phi | \hat{\Phi}^{(k)}) = \sum_{i=1}^g \sum_x \log(\pi_i) p(i | x, \hat{\Phi}^{(k)}) + \sum_{i=1}^g \sum_x \log(p_i(I(x) | \hat{\Phi}^{(k)})) p(i | x, \hat{\Phi}^{(k)})$$

The first term, $\sum_{i=1}^g \sum_x \log(\pi_i) p(i | x, \hat{\Phi}^{(k)})$, is labeled "Data Likelihood" and "Mixing Proportions". The second term, $\sum_{i=1}^g \sum_x \log(p_i(I(x) | \hat{\Phi}^{(k)})) p(i | x, \hat{\Phi}^{(k)})$, is labeled "Ownership".

The EM Algorithm

... iterate between layer and motion estimation ...

Starting with an initial estimate $\hat{\Phi}^{(0)}$ repeat :

- **E-step** : Compute the function $Q(\Phi | \hat{\Phi}^{(k)})$

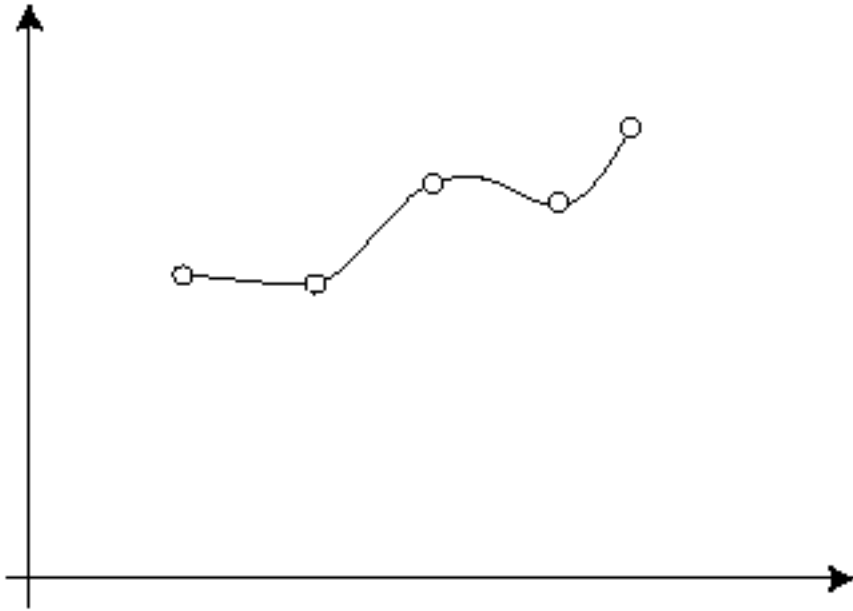
Given the current estimate of the alignment parameters, compute the layer ownerships.

- **M-step** : Compute $\hat{\Phi}^{(k+1)} = \arg \max_{\Phi} Q(\Phi | \hat{\Phi}^{(k)})$

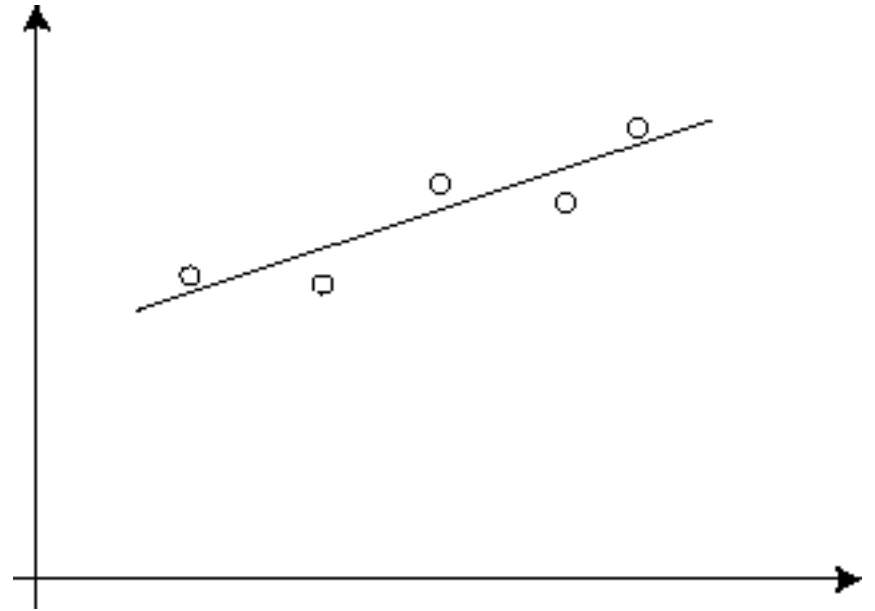
Given the layer ownerships compute the alignment parameters.

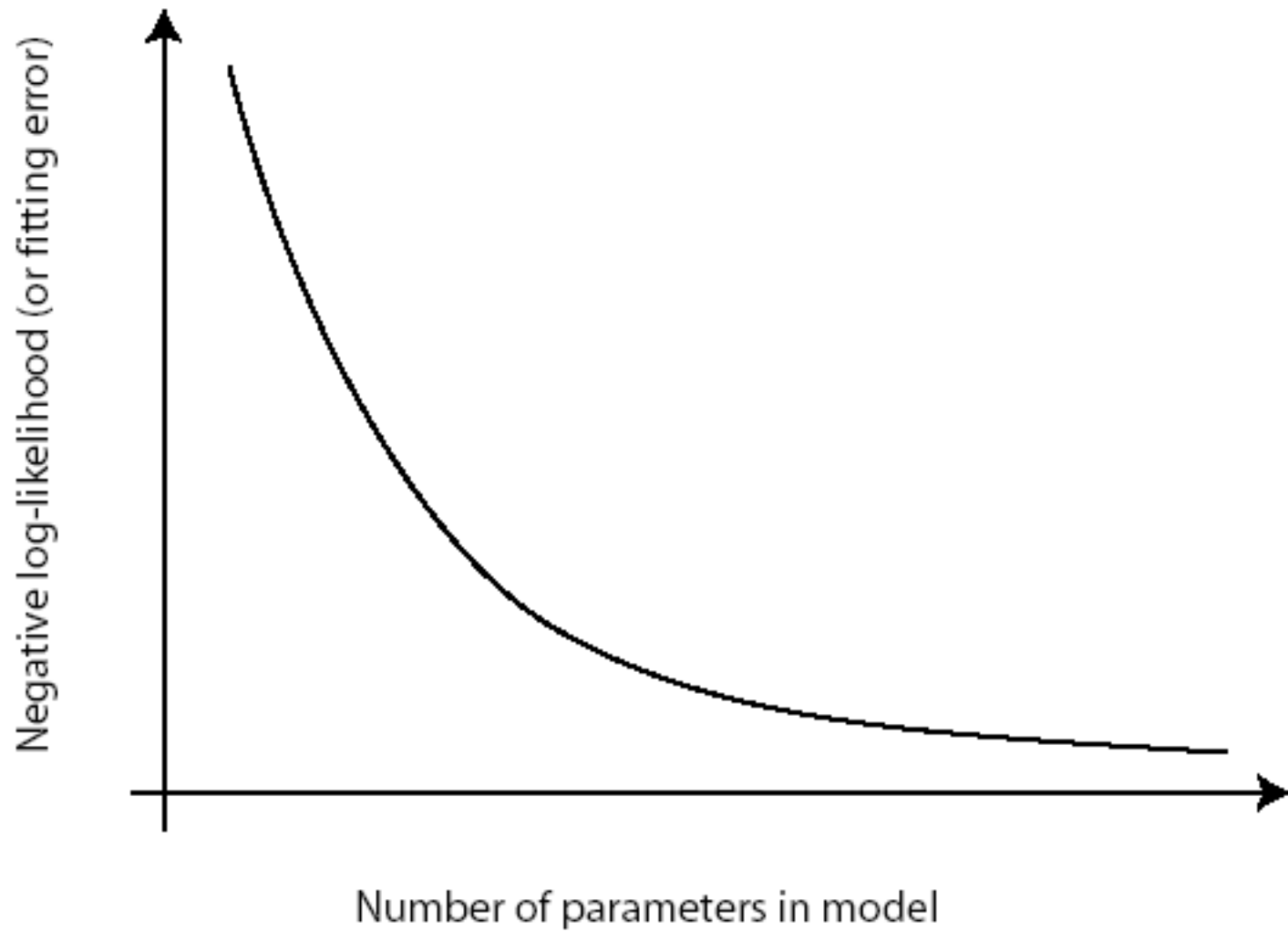
Model Selection

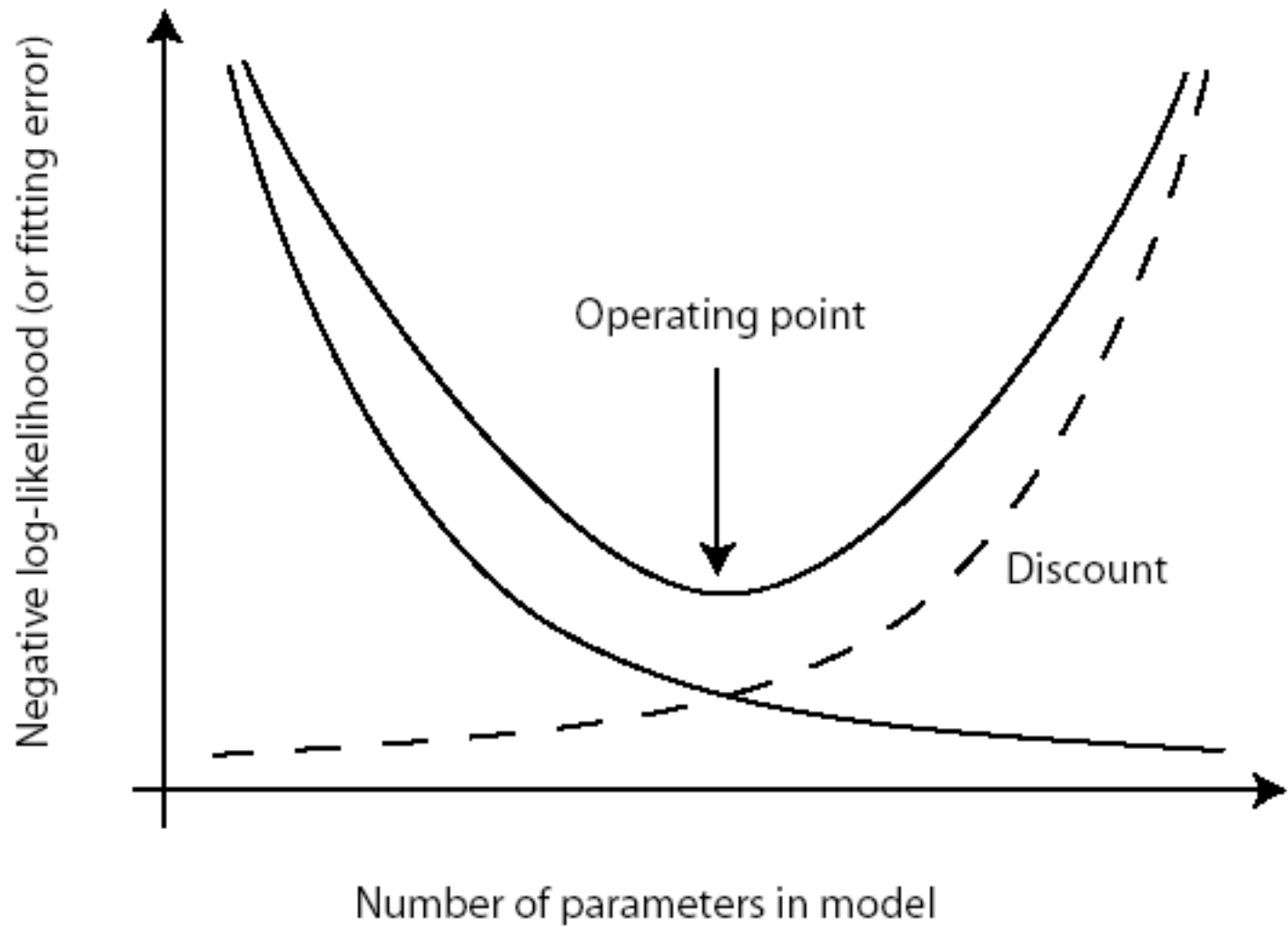
- We wish to choose a model to fit to data
 - e.g. is it a line or a circle?
 - e.g. is this a perspective or orthographic camera?
 - e.g. is there an aeroplane there or is it noise?
- Issue
 - In general, models with more parameters will fit a dataset better, but are poorer at prediction
 - This means we can't simply look at the negative log-likelihood (or fitting error)



Top is not necessarily a better fit than bottom
(actually, almost always worse)







We can discount the fitting error with some term in the number of parameters in the model.

How Many Models Are Adequate ?

*Minimum Description Length (MDL) encoding
for
Optimizing Modeling Complexity*

- Define model complexity as the total number of bits needed to encode the data and the models:

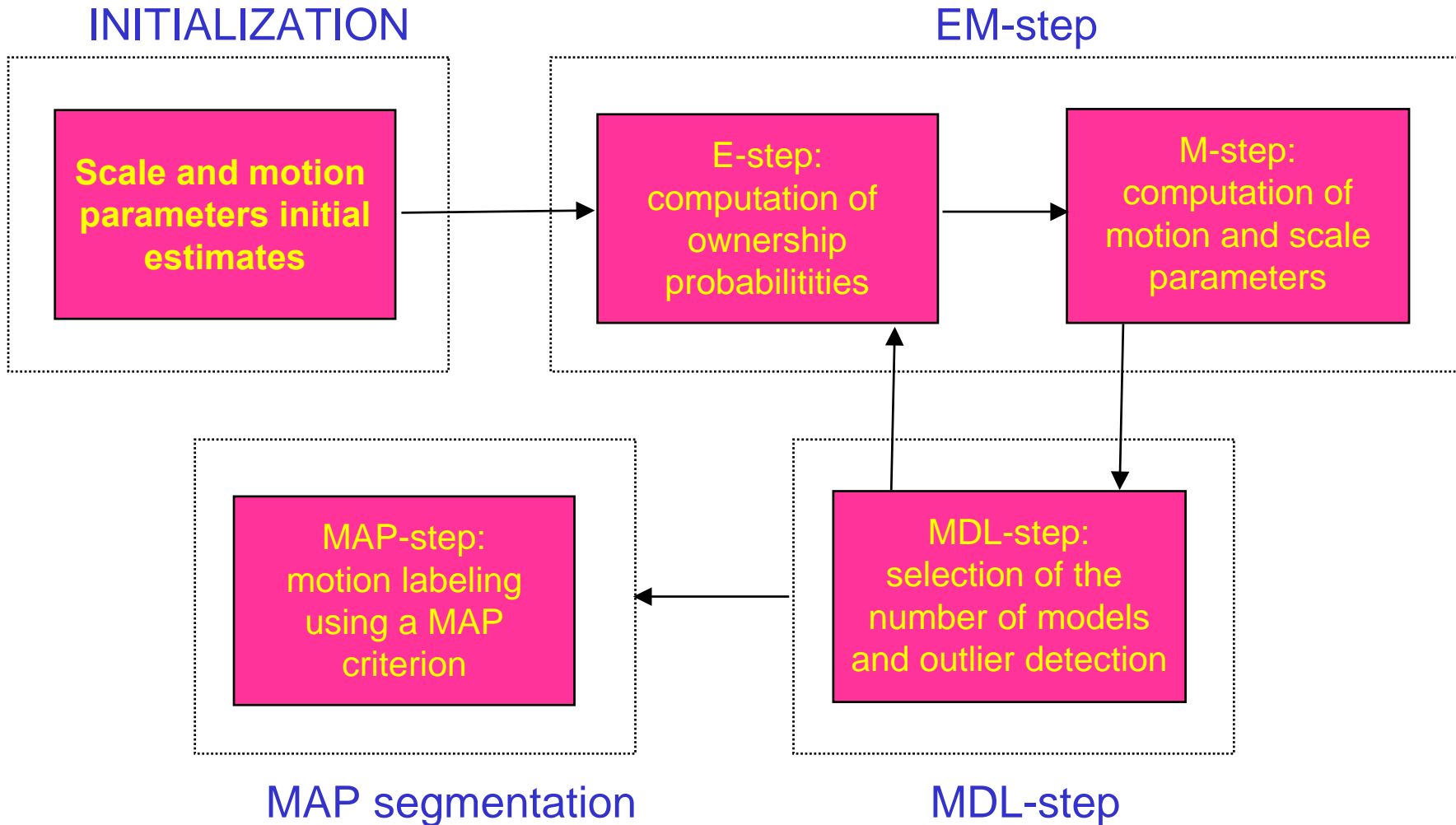
$$L(\{I(p_j), \Phi\}) = \underbrace{L_M(\Phi)}_{\text{Model Encoding Length}} + \underbrace{L_D(\{I(p_j)\} | \Phi)}_{\text{Data Encoding Length}}$$

Model Encoding Length
(including pixel ownerships)

Data Encoding Length
(residuals after alignment)

- Find the optimum number of models and the model parameters that minimize the total encoding complexity.

The Complete Algorithm



Automatic Extraction of 2D Layers

Layers

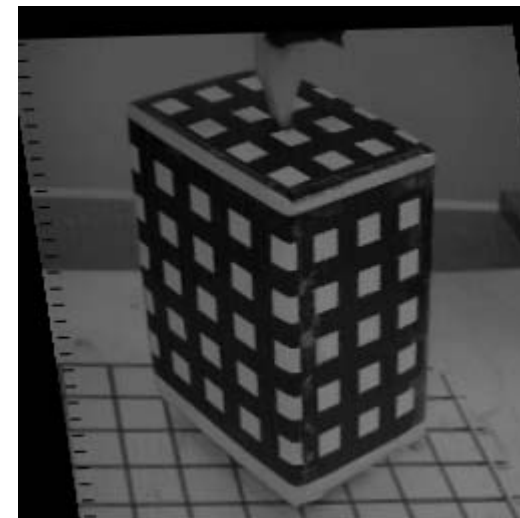
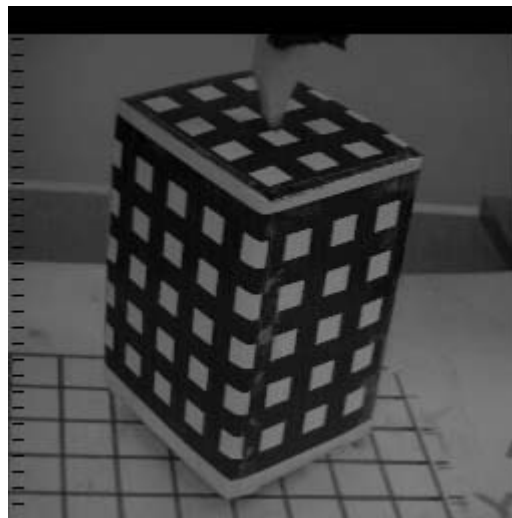
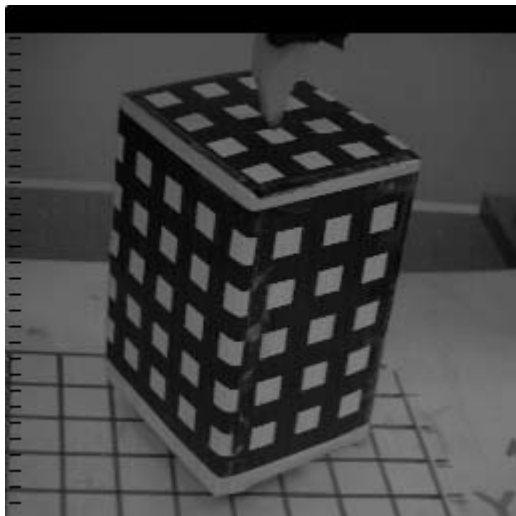


Input Sequence

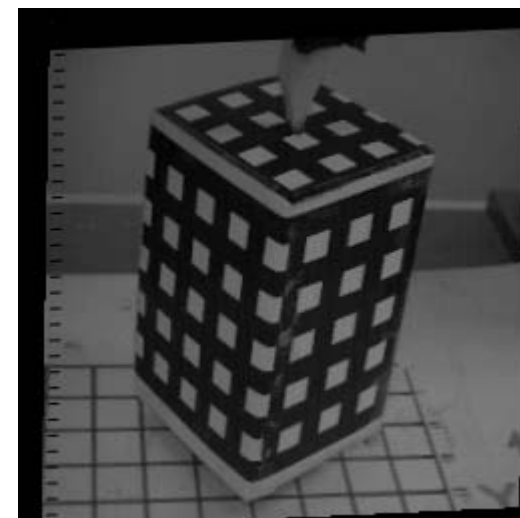
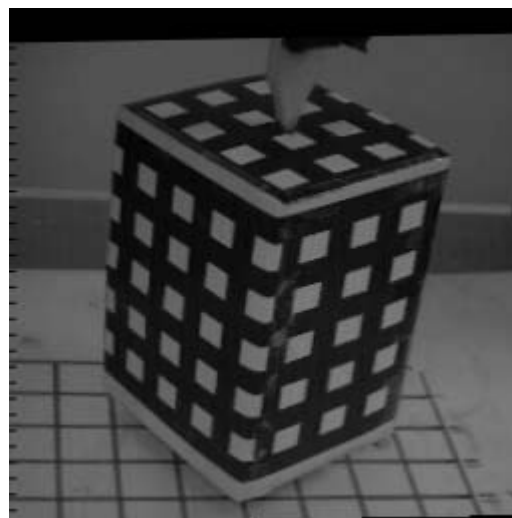
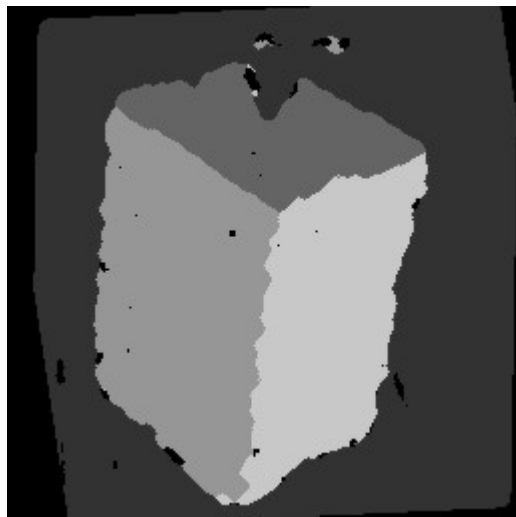


Automatic Extraction of 2D Layers

Layers



Input Sequence



The End