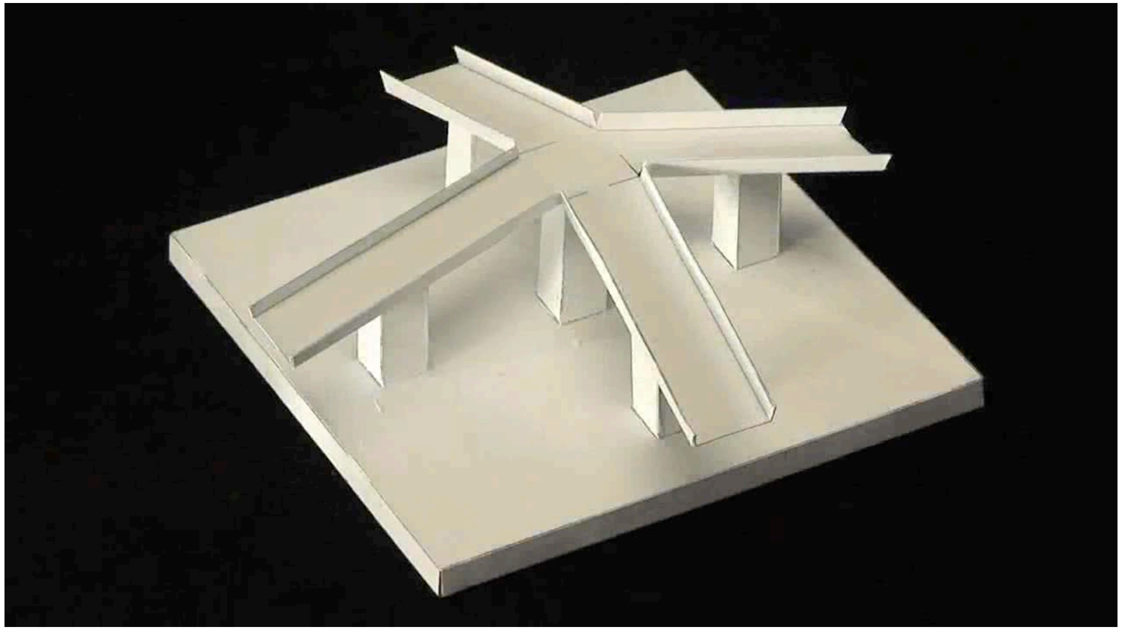
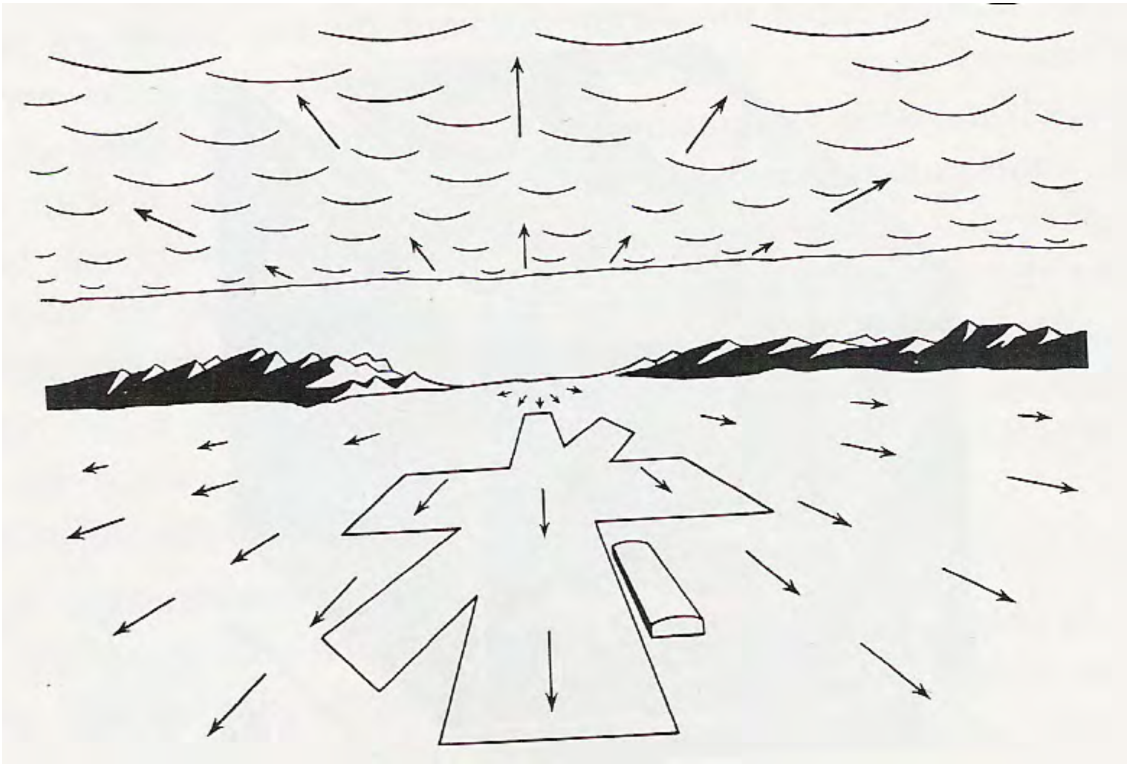
Shape from X

- One image:
  - Texture
  - Shading
- Two images or more:
  - Stereo
  - Contours
  - **Motion**



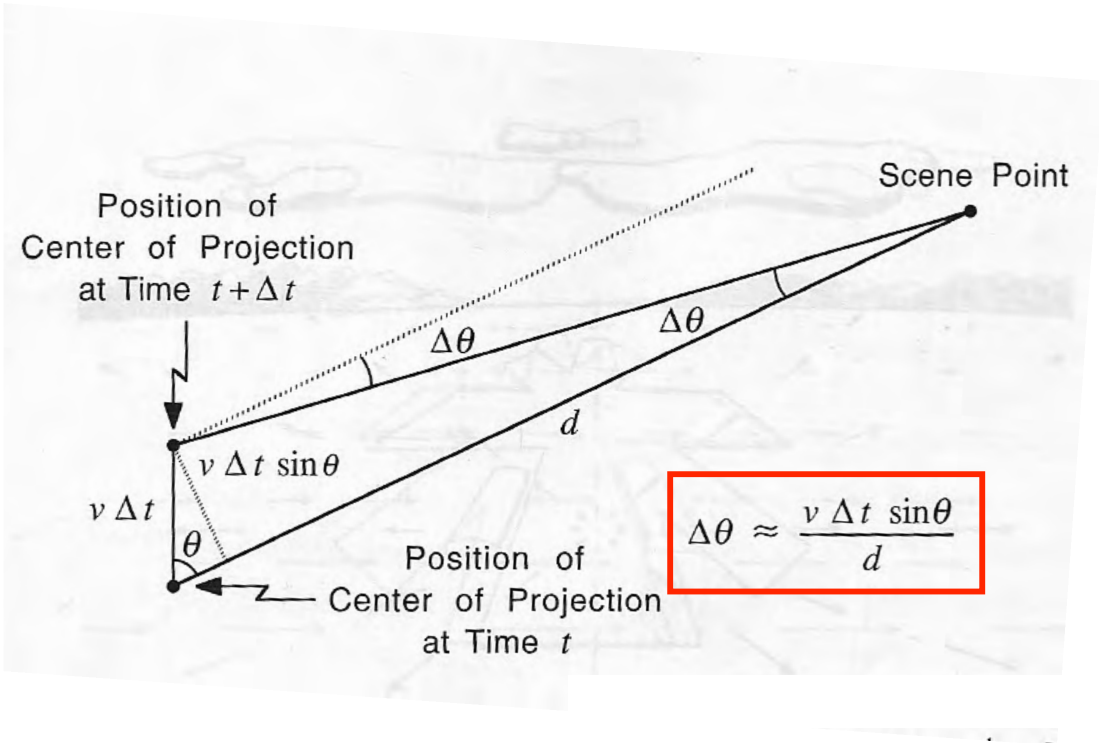
Motion



When objects move at equal speed, those more remote seem to move more slowly.

Euclid, 300 BC

Velocity vs Distance



Apparent velocity is:

- Inversely proportional to the distance of the point to the observer.
- Proportional to the sine of the angle between the line of sight and the direction of translation.

Epipolar Plane Analysis

Image sequence

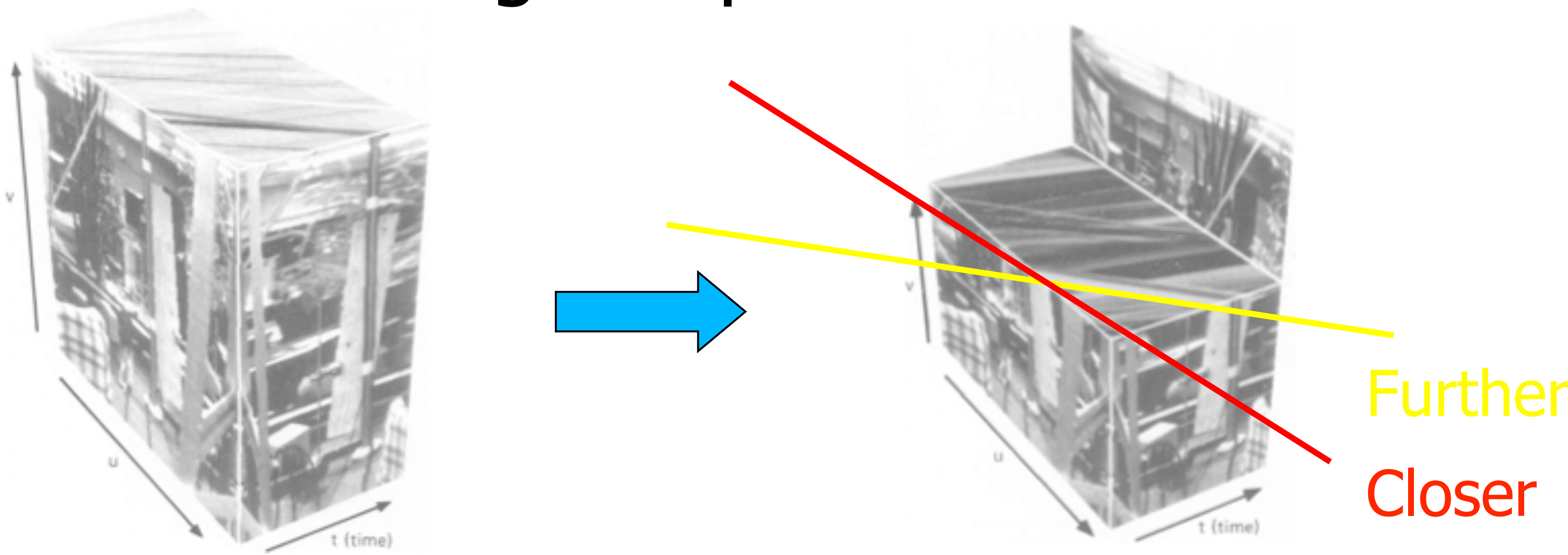
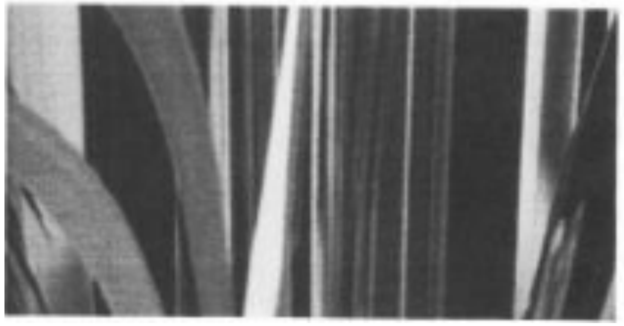


Image cube

Generalized Motion

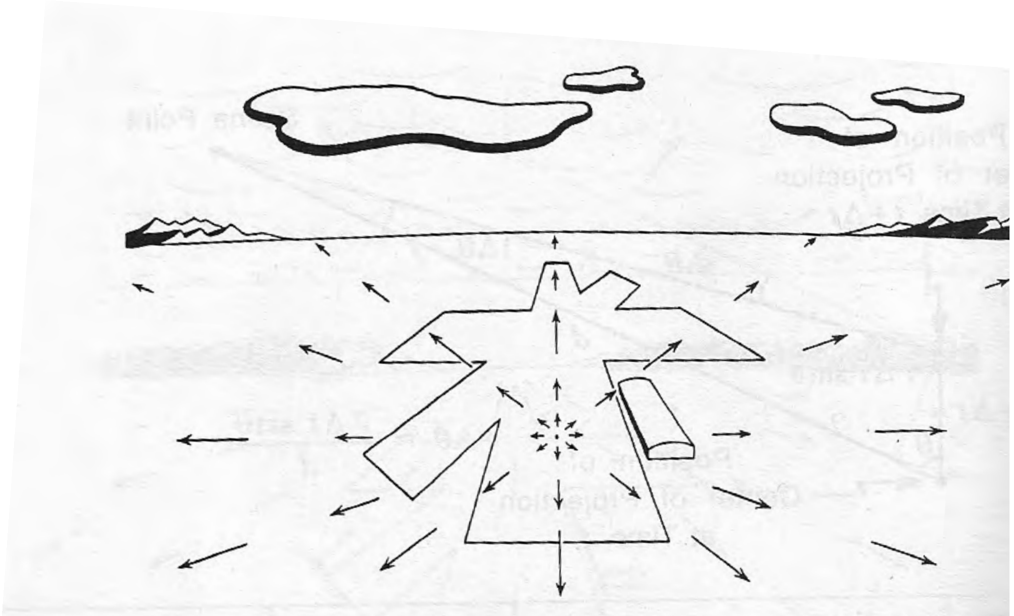
Orthogonal viewing



Non-orthogonal viewing

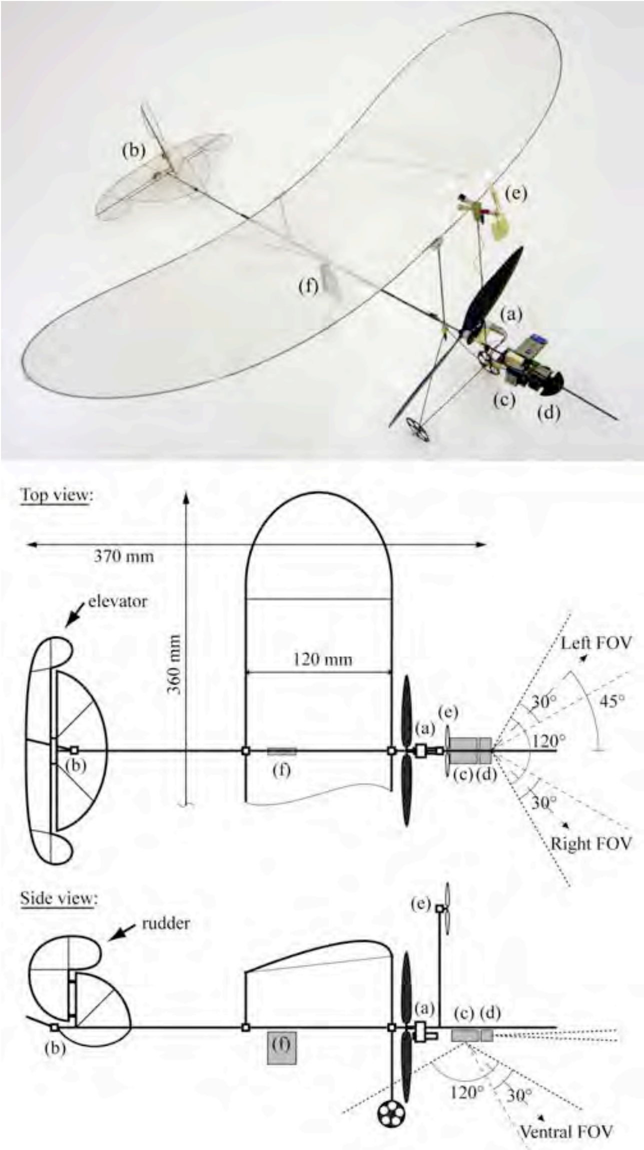
View direction varying

Focus of Expansion



For a translational motion of the camera, all the **motion-field** vectors converge or diverge from a single point: The focus of expansion (FOE) or contraction (FOC).

Microflyer



The plane detects POEs and uses them to avoid collisions.

Motion Field Estimation

Approaches can be classified with respect to the assumptions they make about the scene:

- Images properties remain invariant under relative motion between the camera and the scene.

- Feature points can be tracked across frames.

Assumption 1: Brightness Constancy

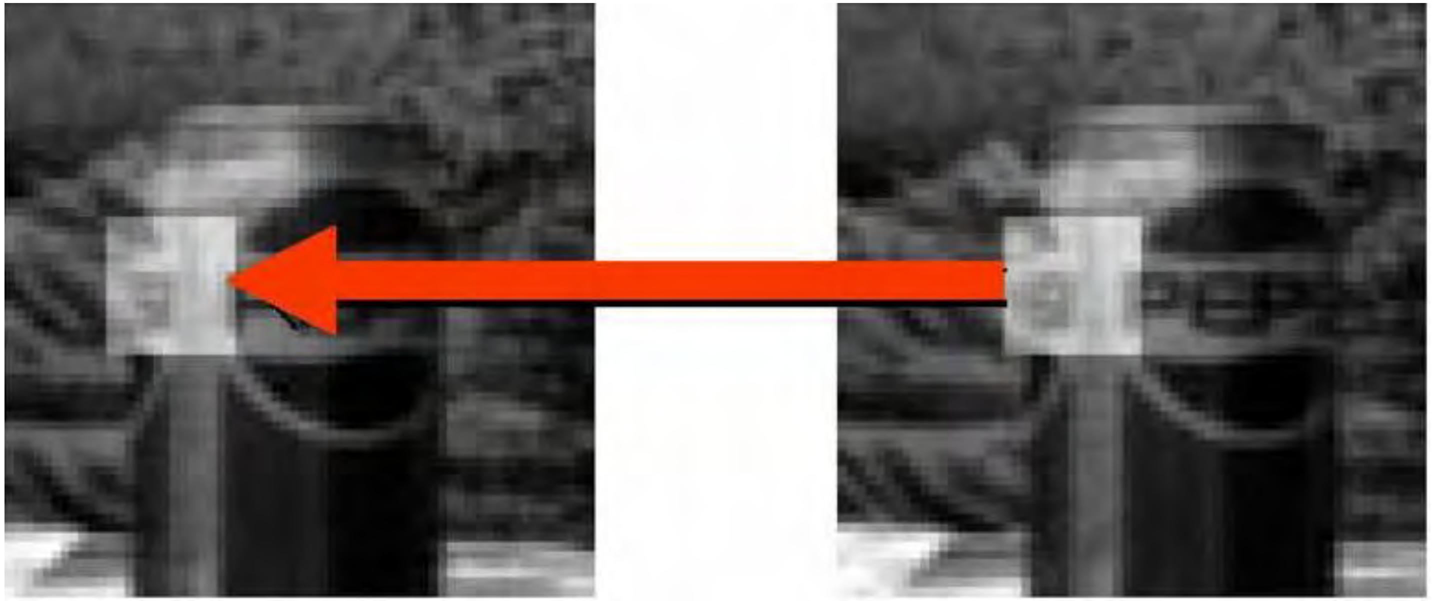
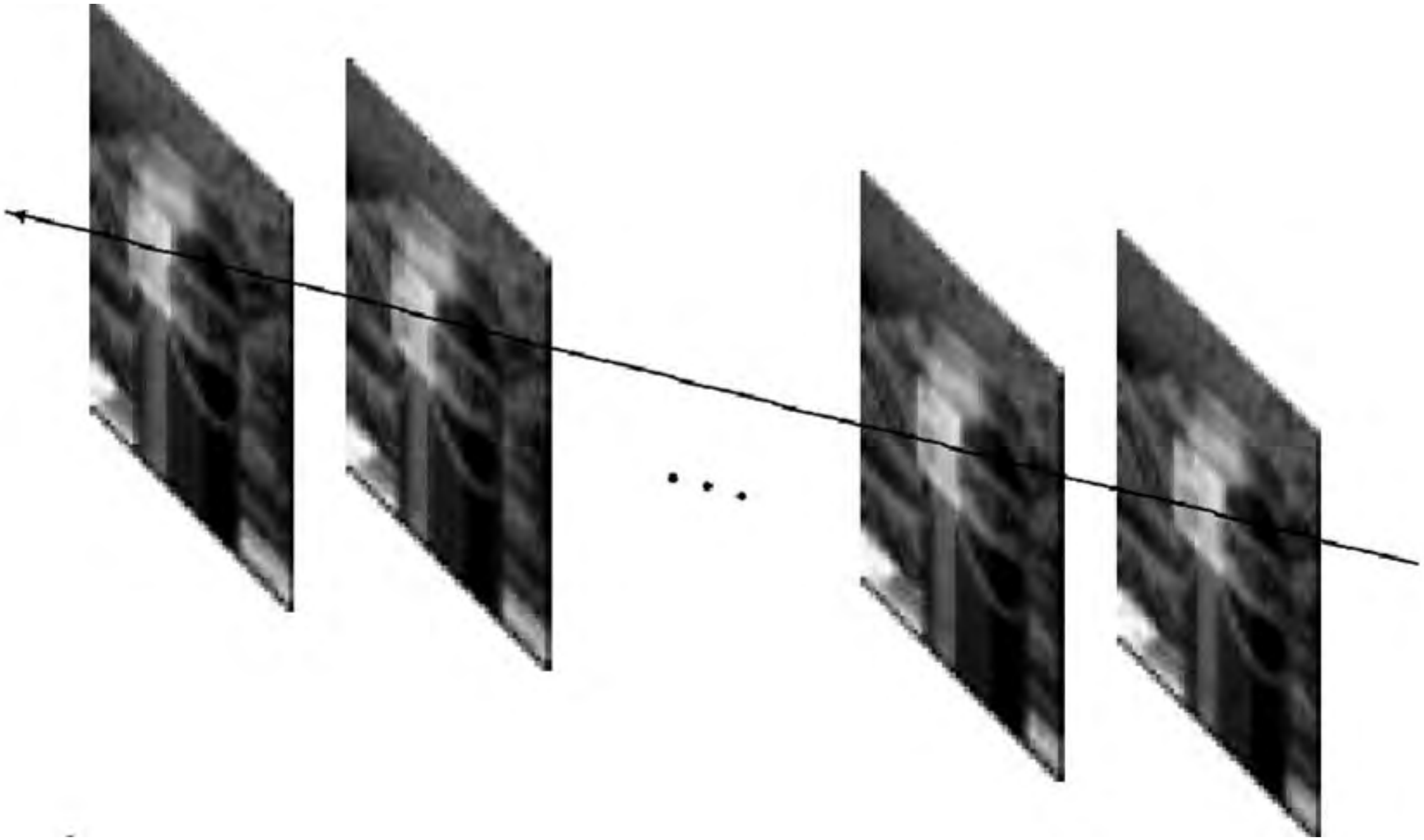


Image measurements (e.g. brightness) in a small region remain the same although its location may change.

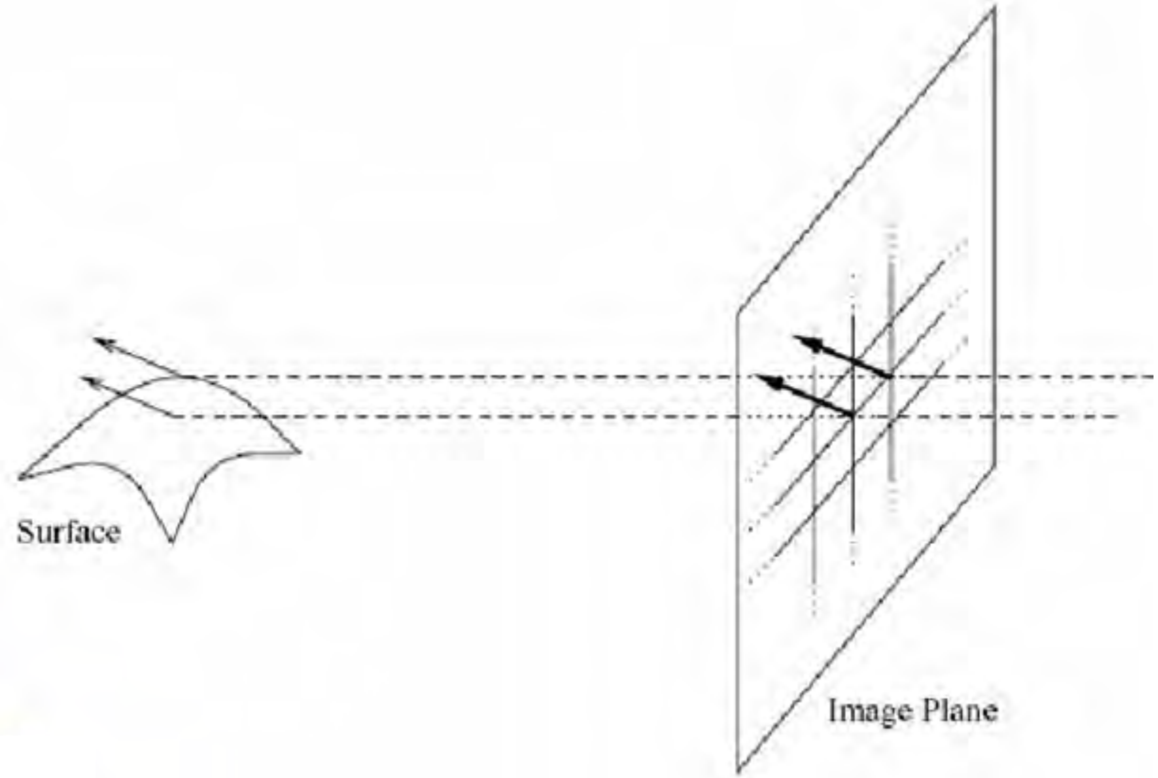
$$I(x + dx, y + dy, t + dt) = I(x, y, t)$$

Assumption 2: Temporal Consistency



The image speed of a surface patch only changes gradually over time.

Assumption 3: Spatial Consistency



- Neighboring points in the scene typically belong to the same surface and hence have similar motions.
- Since they also project to nearby image locations, we expect spatial coherence of the flow.

Spatio Temporal Derivatives

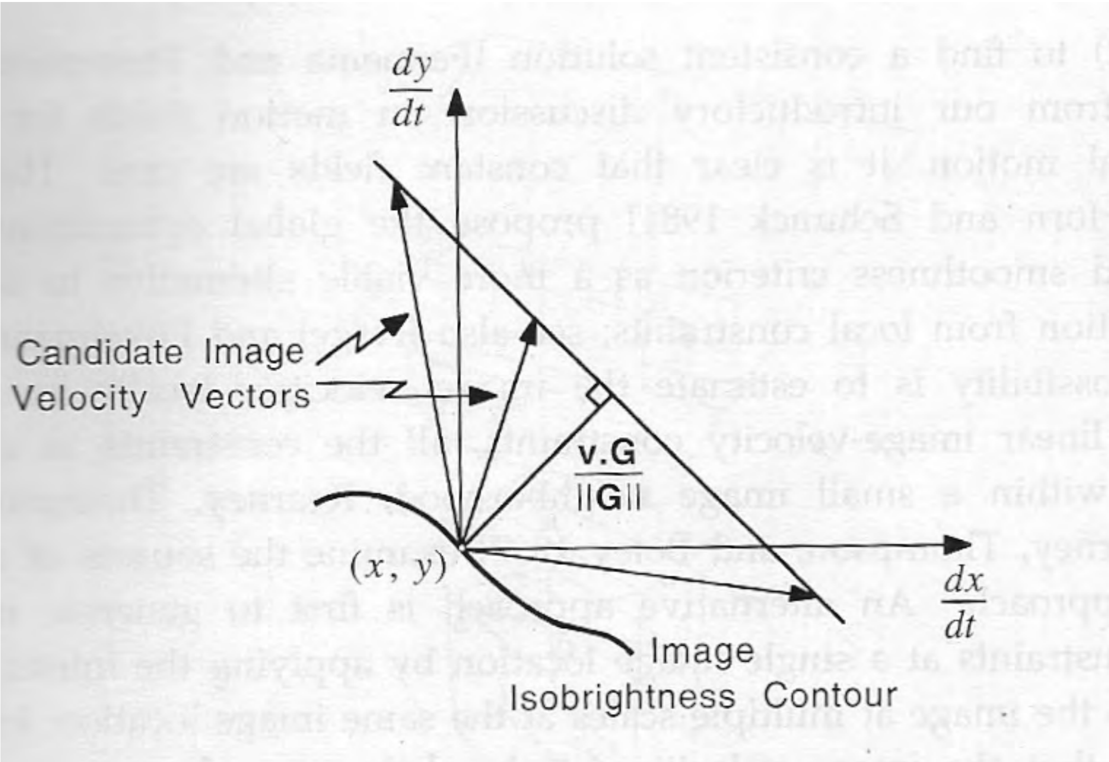
Under the assumptions of

- Brightness constancy,
- Temporal consistency,

we write:

$$\begin{aligned} \text{cst} &= \overbrace{I(x(t), y(t)}^{\text{Image projection at time t}}, t) \\ \Rightarrow 0 &= \frac{\delta I}{\delta x}\frac{dx}{dt} + \frac{\delta I}{\delta y}\frac{dy}{dt} + \frac{\delta I}{\delta t} \end{aligned}$$

Normal Flow Equation



$$v \frac{G}{\|G\|} = -\frac{\dfrac{\partial I}{\partial t}}{\sqrt{\dfrac{\partial I}{\partial x}^2 + \dfrac{\partial I}{\partial y}^2}}$$

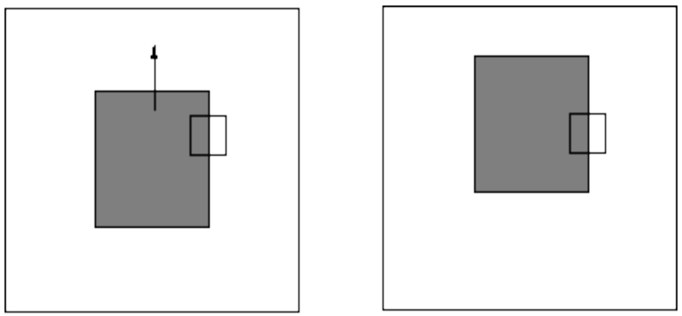
$$G = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right]$$

$$v = \left[\frac{dx}{dt}, \frac{dy}{dt} \right]$$

Ambiguities

- At each pixel, we have 1 equation and 2 unknowns.

- Only the flow component in the gradient direction can be determined locally.



The motion is parallel to the edge, and it cannot be determined.

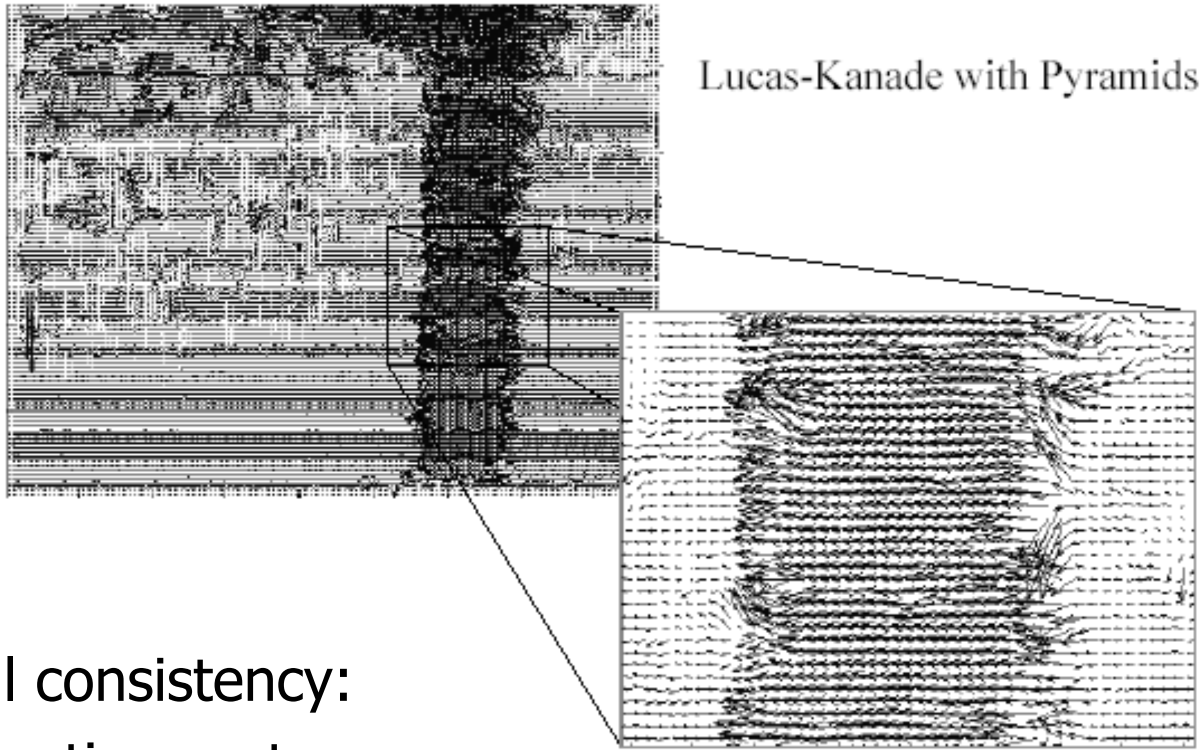
Local Constancy

Assume the flow to be constant is a 5x5 window:

$$\begin{bmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ I_x(\mathbf{p}_2) & I_y(\mathbf{p}_2) \\ \vdots & \vdots \\ I_x(\mathbf{p}_{25}) & I_y(\mathbf{p}_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p}_1) \\ I_t(\mathbf{p}_2) \\ \vdots \\ I_t(\mathbf{p}_{25}) \end{bmatrix}$$

--> 25 equations for 2 unknown, which can be solved in the least squares sense.

Enforcing Consistency



Lucas-Kanade with Pyramids

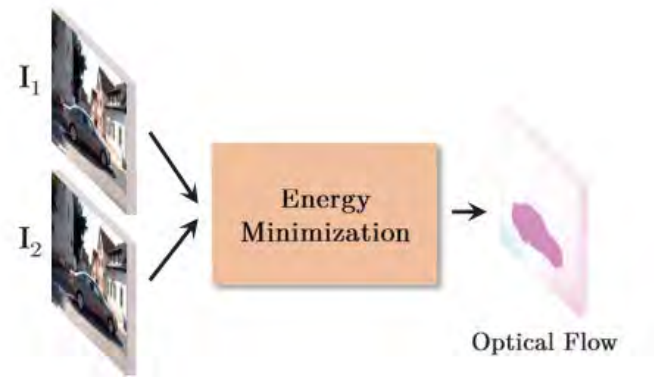
Under the assumption of spatial consistency:

- Hough Transform on the motion vectors.
- Regularization of the motion field.
- Multi scale approach.

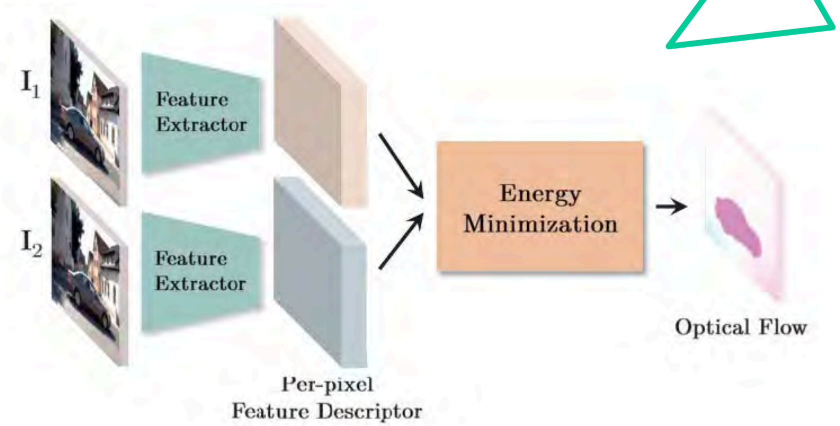
But, the world is neither Lambertian nor smooth.

→ These assumptions are rarely valid.

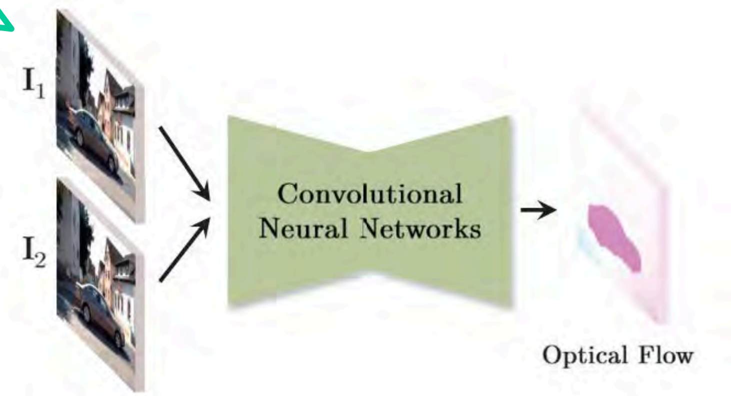
Deep Networks to the Rescue



$$\text{Minimize } E(\mathbf{U}) = \int \left(I_x u_x + I_y u_y + I_t \right)^2 + \alpha \|\nabla u_x\|^2 + \beta \|\nabla v_x\|^2 dxdy$$



- CNN is used as feature extractor.
- These features can be trained to be more invariant.



- Direct regression from images using and hour-glass shaped architecture reminiscent of U-Net.
- The best current techniques uses this approach but this could change.

Hur and Roth, ArXiv'20

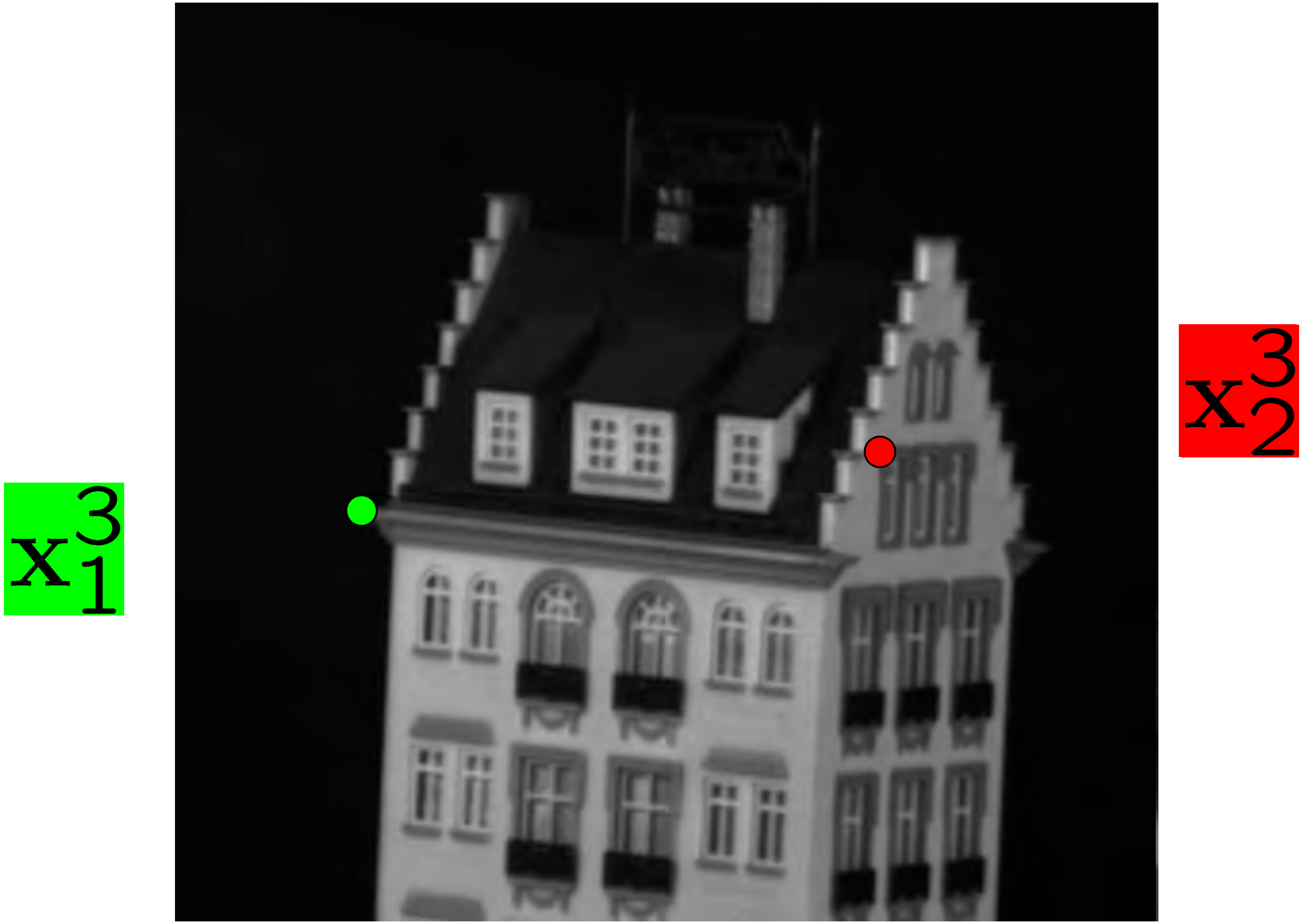
More Research Needed

Deep Network based techniques now outperform classical ones but:

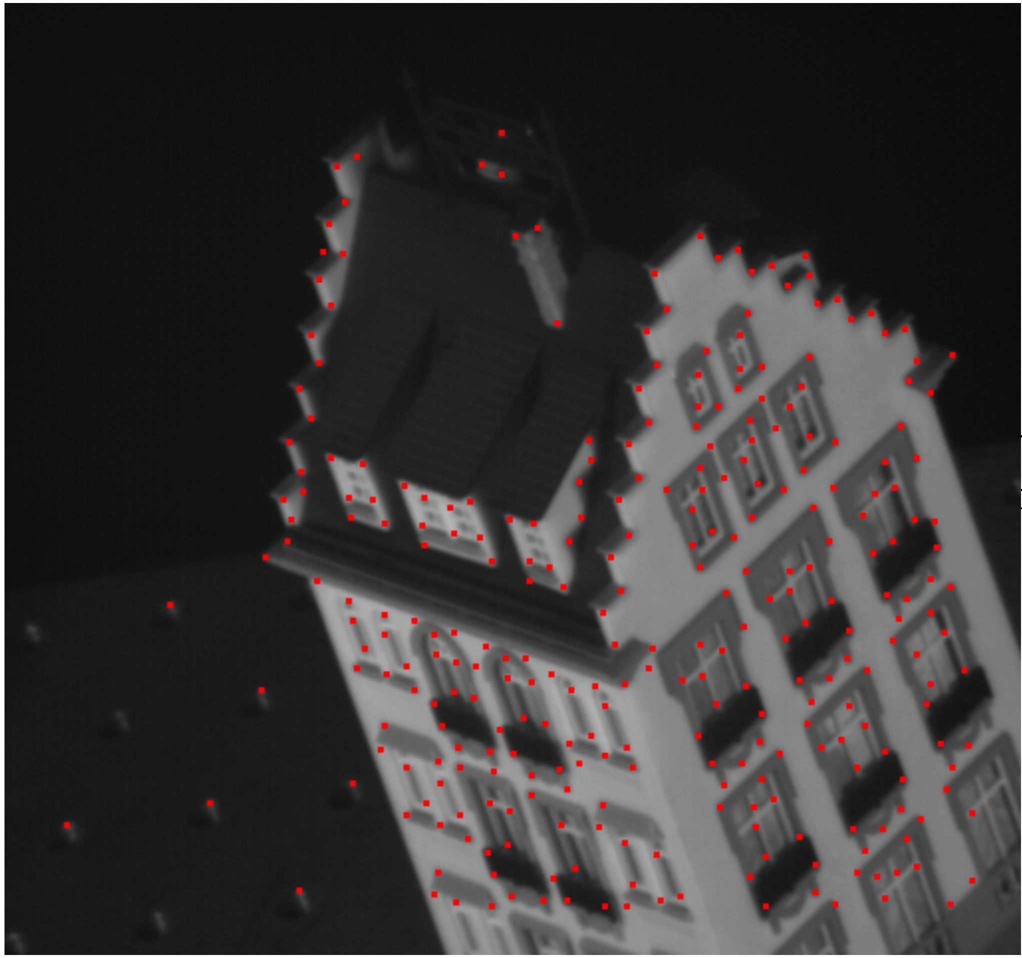
- Tendency to overfit to the training domain.
- Complex training schemes are required.

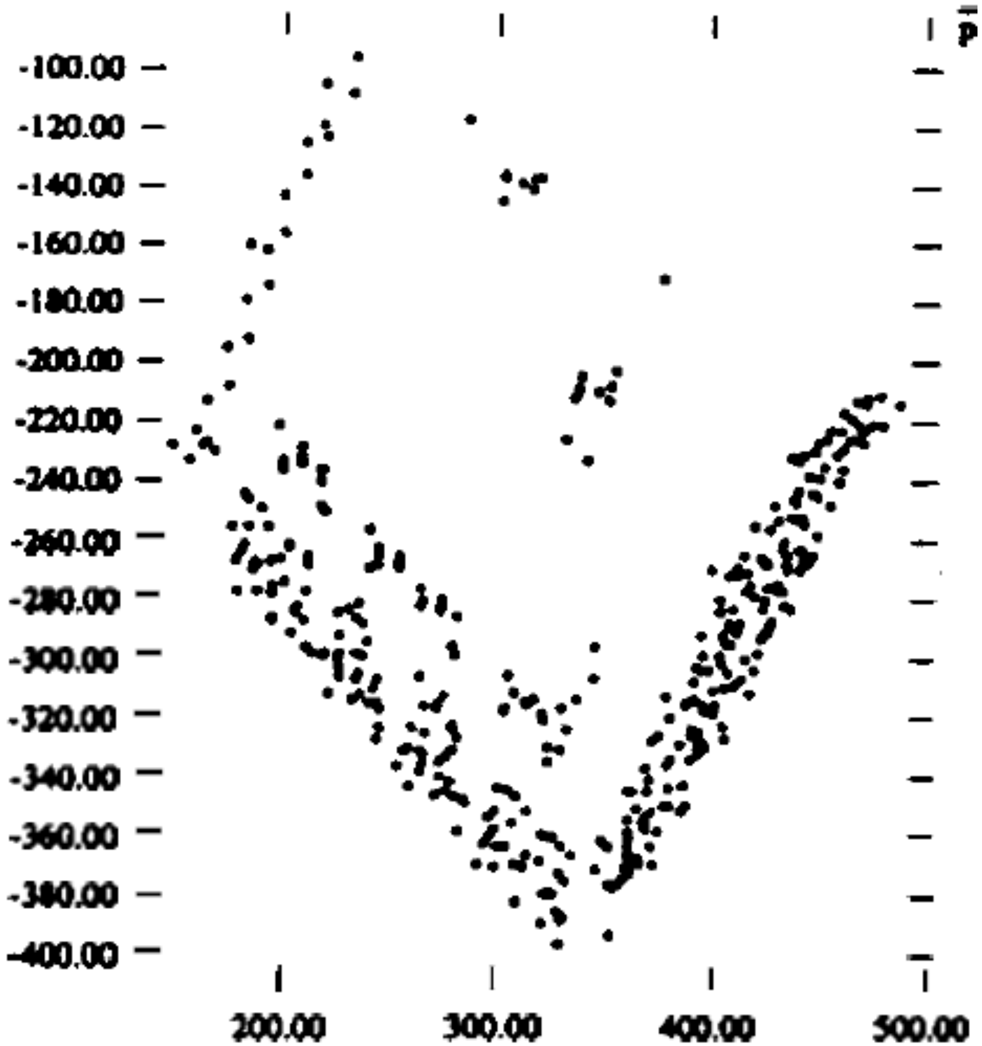
—> There still is work to do.

Tracking Points across Images



3D Shape Reconstruction





Multi-View Projection

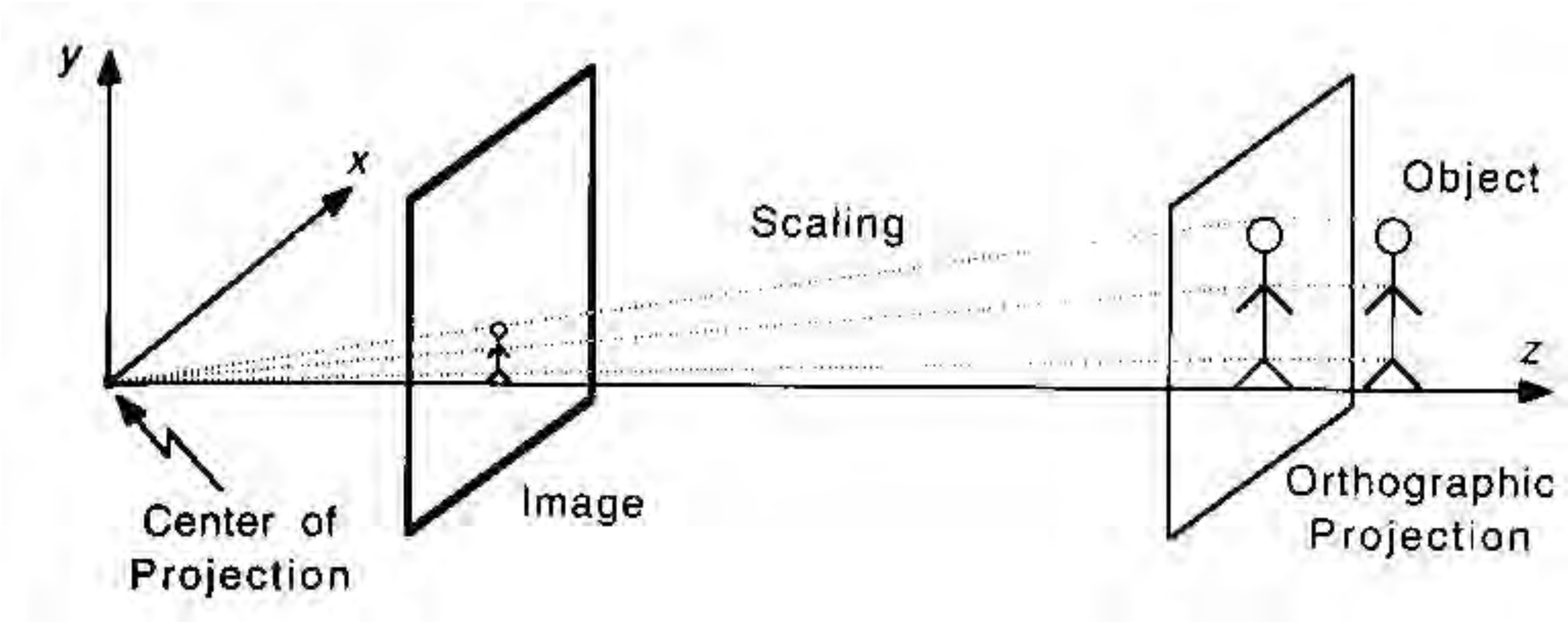
- n image points are projected from 3-D scene points over m views via

$$\mathbf{x}_j^i = \mathbf{P}^i \mathbf{X}_j$$

where i = 1, … , m and j = 1, … , n.

- Here each $\mathbf{P}^i$ is a 3 x 4 matrix and each $\mathbf{X}_j$ is a homogeneous 4-vector.

Orthographic Projection



$$u = sx$$

$$v = sy$$

Multi-View Orthographic Projection

- The last row of each $\mathbf{P}^i$ is $(0, 0, 0, 1)$ for affine cameras, so we can "ignore" it and write the orthographic projection as:

$$\mathbf{x}_j^i = \mathrm{M}^i \mathrm{X}_j + \mathbf{t}^i$$

where each $\mathbf{X}_j$ is now an inhomogeneous 3-vector.

- Here, each $\mathbf{M}^i$ a 2 x 3 matrix, and each $\mathbf{t}^i$ a 2-vector.

Reconstruction Problem

- Estimate affine cameras $\mathbf{M}^i$, translations $\mathbf{t}^i$, and 3-D points $\mathbf{X}_j$ that minimize the geometric error in image coordinates:

$$\min_{\mathbf{M}^i, \mathbf{t}^i, \mathbf{X}_j} \sum_{i,j} \left(\mathbf{x}_j^i - (\mathbf{M}^i \mathbf{X}_j + \mathbf{t}^i) \right)^2$$

Simplifying the Problem

- Normalization: We can eliminate the translation vectors $\mathbf{t}^i$ by choosing the centroid of the image points in each image as the coordinate system origin

$$\mathbf{x}_j^i \leftarrow \mathbf{x}_j^i - \frac{1}{n}\sum_j \mathbf{x}_j^i$$

- Working in "centered coordinates", the minimization problem becomes:

$$\min_{\mathbf{M}^i, \mathbf{X}_j} \sum_{i,j} \left(\mathbf{x}_j^i - \mathbf{M}^i \mathbf{X}_j\right)^2$$

- This works because the centroid of the 3-D points is preserved under affine transformations

Matrix Formulation

- Let the measurement matrix be:

$$\mathbf{W} = \begin{pmatrix} \mathbf{x}_1^1 & \mathbf{x}_2^1 & \cdots & \mathbf{x}_n^1 \\ \mathbf{x}_1^2 & \mathbf{x}_2^2 & \cdots & \mathbf{x}_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_1^m & \mathbf{x}_2^m & \cdots & \mathbf{x}_n^m \end{pmatrix}$$

- Since $\mathbf{x}_j^i = \mathbf{M}^i \mathbf{X}_j$, this means solving

$$\mathbf{W} = \underbrace{\begin{bmatrix} \mathbf{M}^1 \\ \vdots \\ \mathbf{M}^m \end{bmatrix}}_{2m \times 3} \underbrace{[\mathbf{X}_1, \ldots, \mathbf{X}_n]}_{3 \times n}$$

in the least squares sense.

Solving with SVD

- There will be no exact solution with noisy points, so we want the nearest **W'** to **W** that is an exact solution
  - **W'** is rank 3 since it's the product of a 2m x 3 motion matrix **M'** and a 3 x n structure matrix **X'**

- Use singular value decomposition to get rank 3 matrix **W'** closest to **W**
  - Let SVD of $\mathbf{W} = \mathbf{UDV}^T$
  - Then $\mathbf{W}' = \mathbf{U}_{2mx3}\mathbf{D}_{3x3}\mathbf{V}_{nx3}^T$, where
    - $\mathbf{U}_{2mx3}$ is the first 3 columns of **U**, $\mathbf{D}_{3x3}$ is an upper-left 3 x 3 submatrix of **D**,
    - $\mathbf{V}_{nx3}^T$ is first three columns of **V**.

Structure and Motion

- Set stacked camera matrix as

$$\mathbf{M}' = \mathbf{U}_{2m\times 3}\,\mathrm{sqrt}(\mathbf{D}_{3\times 3})$$

- Set stacked 3-D structure matrix as

$$\mathbf{X}' = \mathrm{sqrt}(\mathbf{D}_{3\times 3})\mathbf{V}_{n\times 3}^{T}$$

so that $\mathbf{W}' = \mathbf{M}'\mathbf{X}'$

Metric Upgrade

- There is an affine ambiguity since an arbitrary 3 x 3 rank 3 matrix **A** can be inserted as:

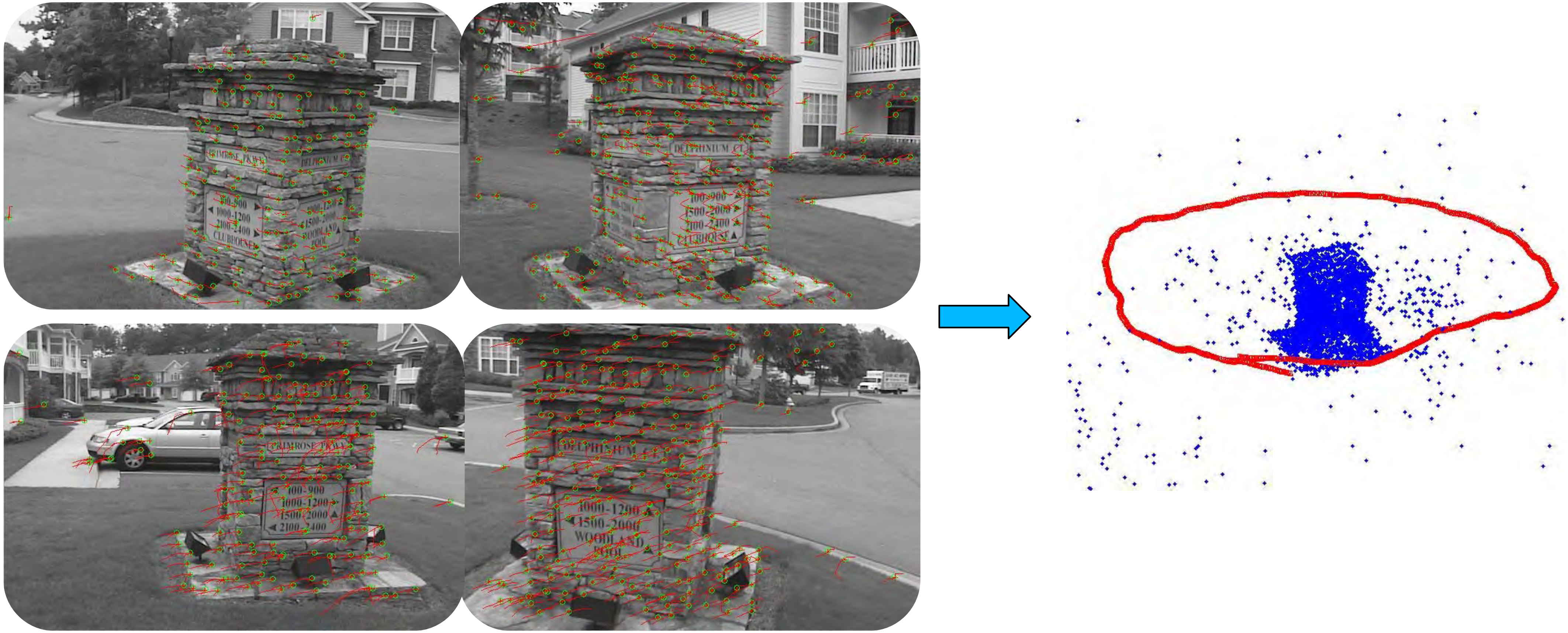
$$\mathbf{W}' = (\mathbf{M}'\mathbf{A})(\mathbf{A}^{-1}\mathbf{X}')$$

- Get rid of ambiguity by finding **A** that performs "metric rectification"

- Affine camera provides orthonormality constraints on **A**:
  - Rows of **M=M'A** are unit vectors: $\mathbf{m}_i \cdot \mathbf{m}_i = 1$.
  - Rows of **M=M'A** are orthogonal: $\mathbf{m}_i \cdot \mathbf{m}_j = 0$.

- Everything relies on linear algebra but is limited to orthographic cameras.

Simultaneous Localization And Mapping

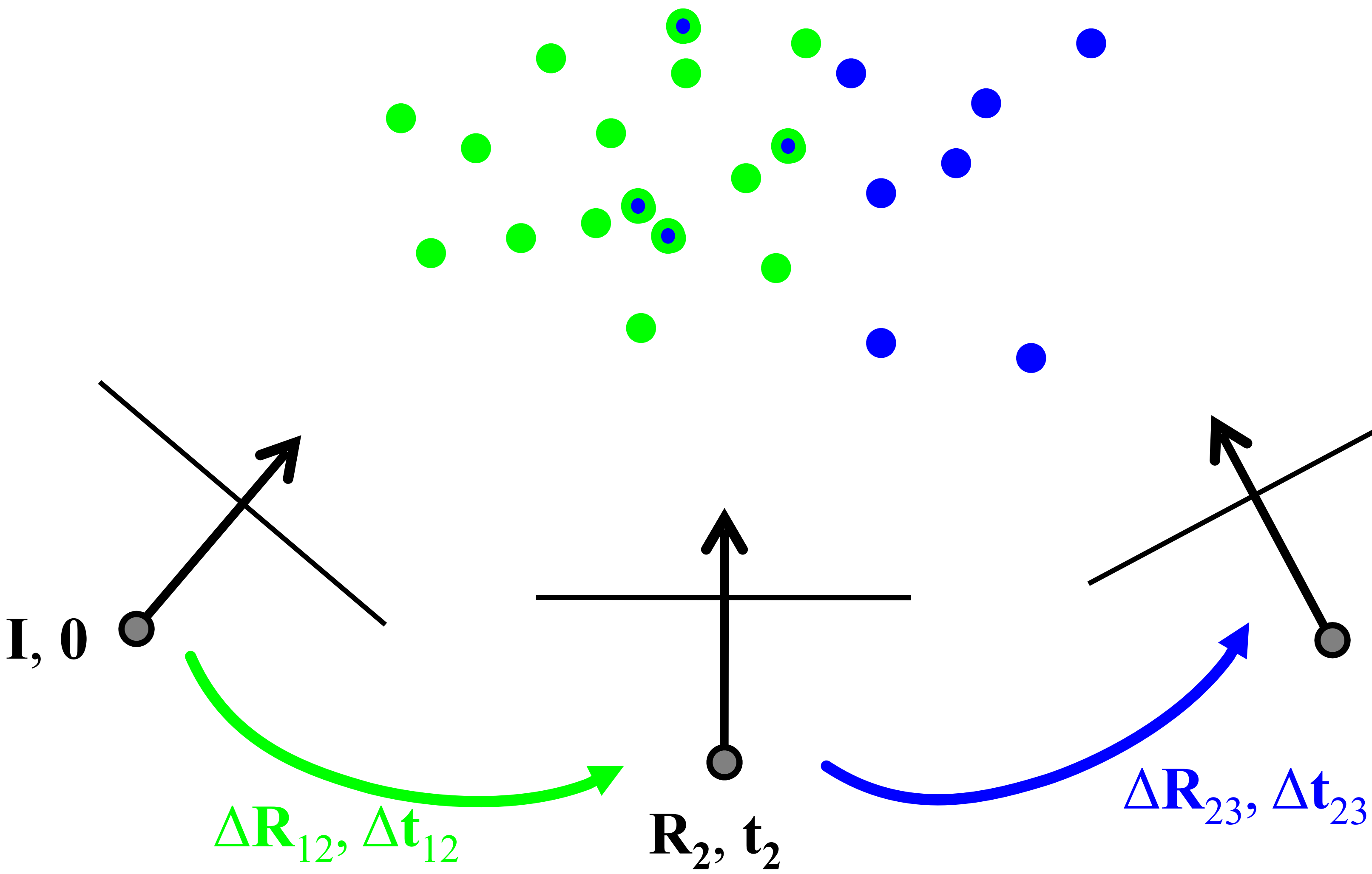


- Compute point tracks.
- Infer both camera motion and 3D structure.

Steedly et al., ICCV'03

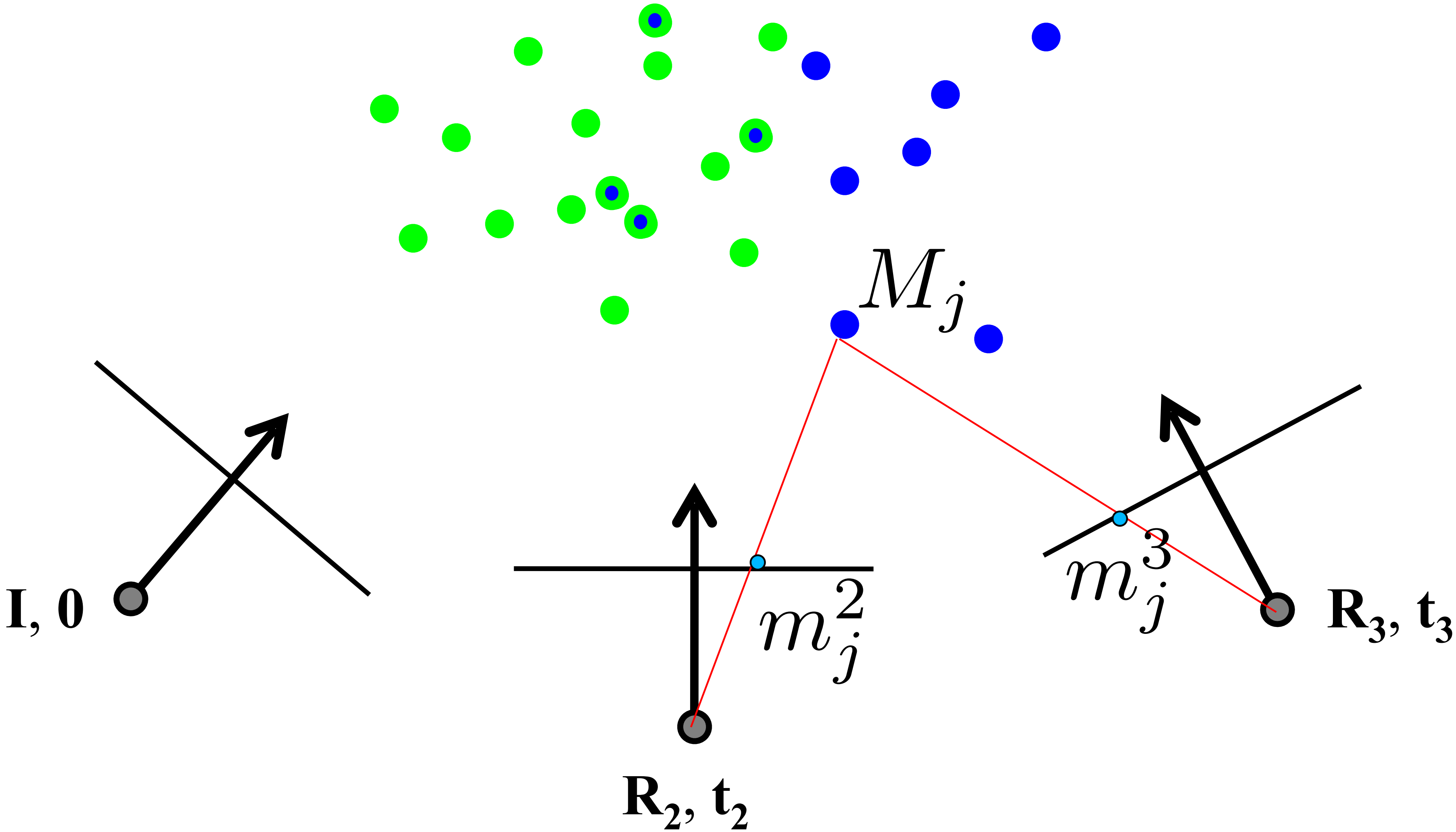
Archeological Reconstruction

Sequential Structure from Motion



-> Trajectory and 3D points defined up to a Euclidean motion and scale

Bundle Adjustment



$$\mathrm{argmin}_{R_i, t_i, M_j} \sum_i \sum_j \|\mathrm{proj}(R_i, t_i, M_j) - m_j^i\|^2$$

Global Non-Linear Optimization

$$\operatorname{argmin}_{R_i,t_i,M_j} \sum_i \sum_j \|\operatorname{proj}(R_i, t_i, M_j) - m_j^i\|^2$$

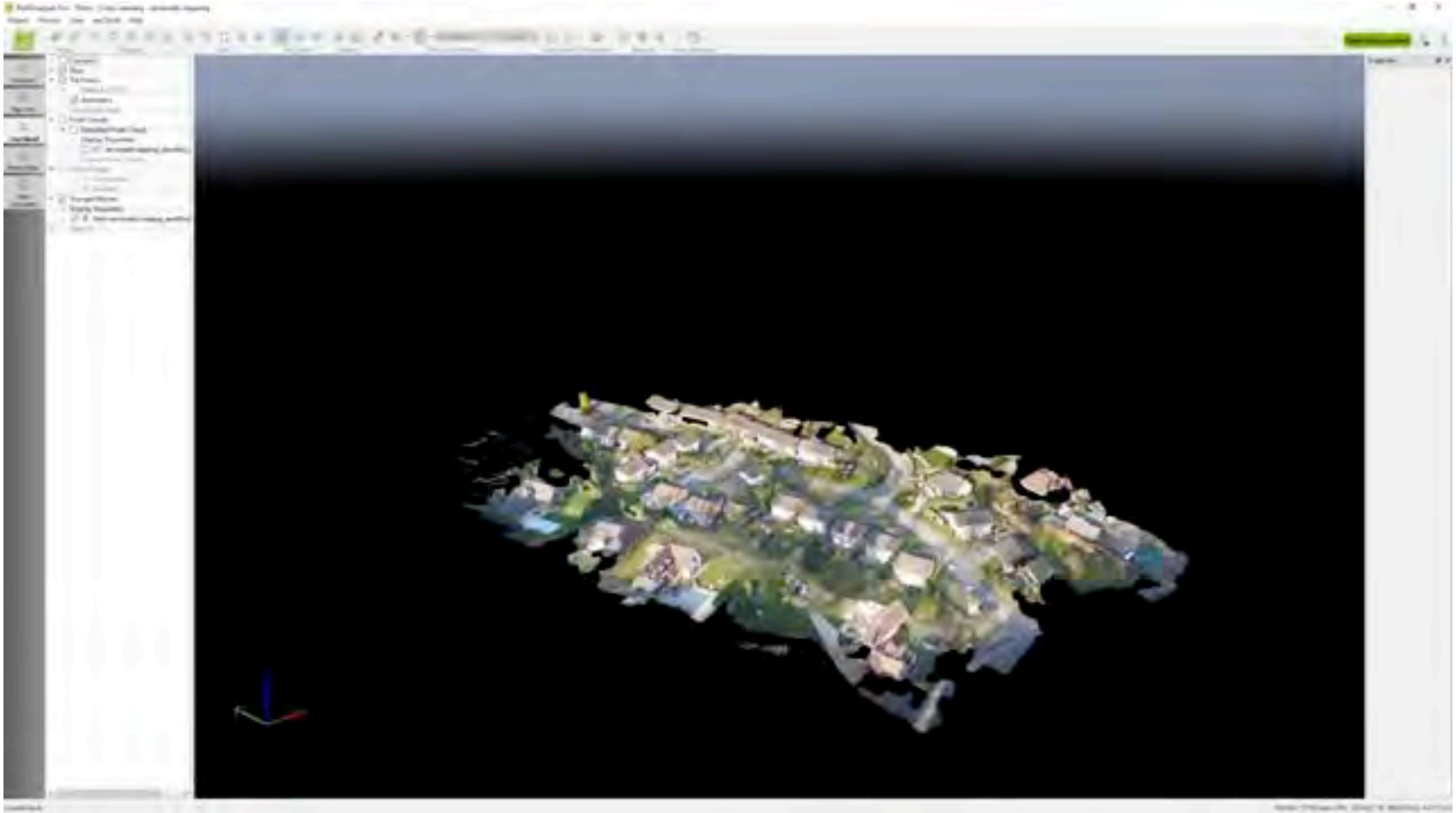
- Often performed using the Levenberg-Marquardt algorithm.
- Many parameters to estimate, but sparse Jacobian matrix.
- Initial estimates computed using the eight point algorithm:

  - Given 8 point correspondences between a pair of images, ΔR and ΔT can be estimated in closed form by solving an SVD.

From Images to Houses (1)

- Pick an area on your phone.
- The system will define a flight plan for your drone.
- It will fly it and bring back images.

From Images to Houses (2)



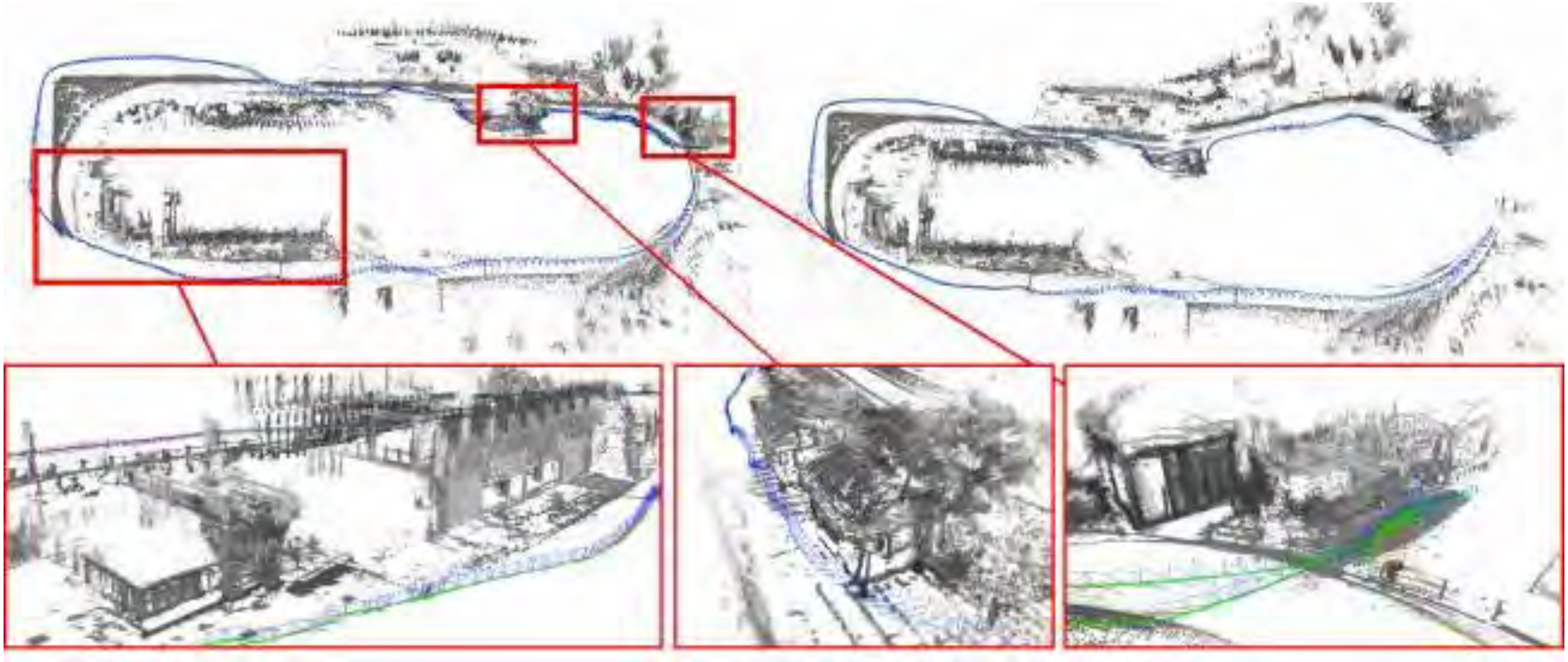
- Download the images on your computer.
- Get a full model without further human intervention.

Virtual Matterhorn

Real Time Augmented Reality

Klein and Murray, ISMAR'07

Simultaneous Localization And Mapping



A robot can reconstruct its environment and position itself at the same time.

Fusing Depth Maps

- Both the depth camera and the person are moving.
- Use a deformable model to combine the data over time.
- Real-time implementation.

Virtual Reality Headsets

Microsoft Hololens

Magic Leap

… and one of them is being worked on in Zurich!

Strengths And Limitations

Strengths:

- Combine information from many images.

Limitations:

- Requires multiple views.
- Requires either texture or a depth camera.