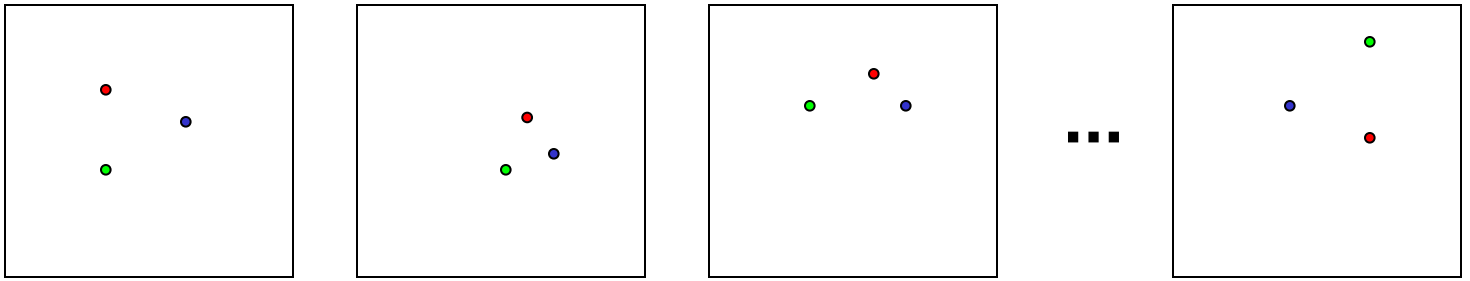
CS 664
Structure and Motion

Daniel Huttenlocher

Cornell University
Faculty of Computing and Information Science

Determining 3D Structure

- Consider set of 3D points X_j seen by set of cameras with projection matrices P_i
- Given only image coordinates x_{ij} of each point in each image, determine 3D coordinates X_j and camera matrices P_i
- Known correspondence between points and known form of projection matrix

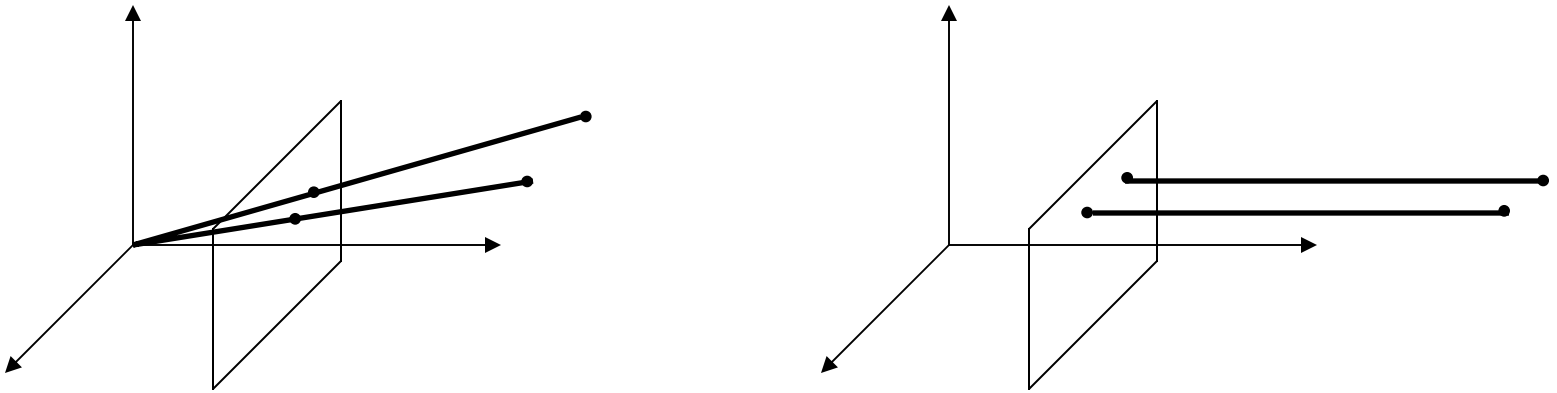


Structure From Motion

- Recover 3D coordinates and (relative) camera locations
- Issues
  - Point correspondences and visibility of points
  - Projection model
  - Calibrated vs. un-calibrated cameras
  - Non-rigid motions, multiple motions
  - Numerical stability of methods

Projection Model

- Parallel (orthographic) Point X=(U,V,W) in space projects to x=(u,v) in image plane
  - Contrast with (fX/W,fY/W) in pinhole model
  - Light rays all parallel rather than through principal point
    - Similar when points at same depth, narrow FOV



Recovering 3D Structure

- With enough corresponding points and views can in principle determine 3D info
  - Redundant data
    - Each view changes only viewing parameters and not point locations
      - 3n unknowns for n points and d(k-1) unknowns for k views and d dof in transformation from one view to next
      - 2nk observed values
    - Overconstrained when $2nk \geq 3n+d(k-1)$
      - Optimization methods, for linear formulations generally least squares error minimization

Minimum Number of Measurements

- In principle can use small number of points and views
  - For instance, 5 points in two images for R,t
    - 5 dof + $3n$ point locations $\leq 4n$ point measurements when $n \geq 5$
- In 1980's many variants investigated
  - Different projection models
  - Correspondences of lines, points
  - Some nice geometric problems, but studied in absence of noise sensitivity/stability analysis

Sensitive to Measurement Noise

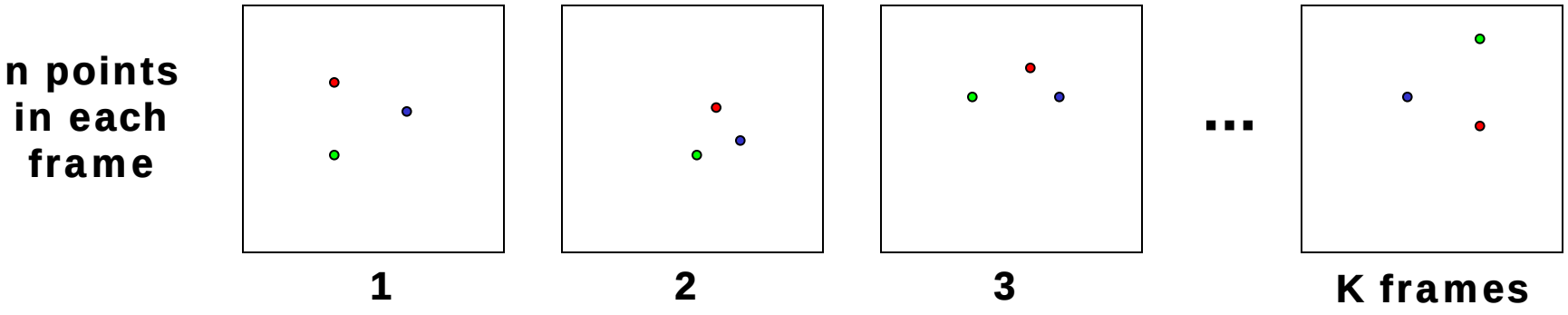
- Solutions based on a small number of points are not stable
  - Errors of the magnitude found in most images yield substantial differences in recovered 3D values
- Method that works in practice called factorization [Tomasi-Kanade 92]
  - Works on sequence of several frames
  - With correspondences of points
  - Consider case of factorization for orthographic projection, no outliers, can be extended

Input: Sequence of Tracked Points

- Point coordinates

$$w'_{ij} = (u'_{ij}, v'_{ij})$$

  - Where i denotes frame (camera) index and j denotes point index
  - Points tracked over frames
    - E.g., use corner trackers discussed previously



Centroid Normalized Coordinates

- From observed coordinates $w'_{ij} = (u'_{ij}, v'_{ij})$

$$w_{fp} = (u'_{ij} - \bar{u}_{ij},\ v'_{ij} - \bar{v}_{ij})$$

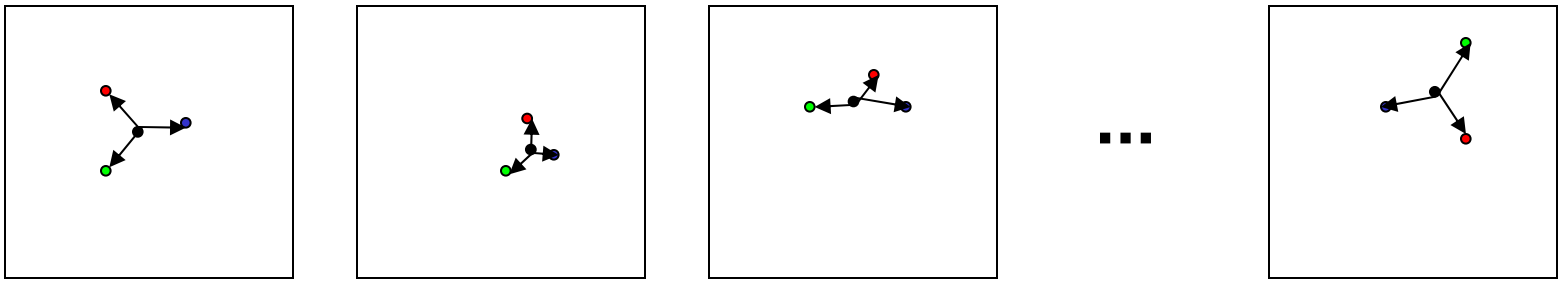
  - Where

$$\bar{u}_{ij} = (1/n) \sum_j u'_{ij}$$

and

$$\bar{v}_{ij} = (1/n) \sum_j v'_{ij}$$

Centroids in frame



Normalization

- Goal of separating out effects of camera translation from those of rotation
- Subtract out centroid to remove translation effects
  - Assume all points belong to object and present at all frames
  - Centroid preserved under projection
- Left to recover 3D coordinates (shape) of n points from k camera orientations

Measurement Matrix

- 2n by k – 2 rows per frame, one col per point
- In absence of senor noise this matrix is highly rank deficient
  - Under orthographic projection rank 3 or less

$$W = \begin{bmatrix} u_{11} & \cdots & u_{1n} \\ \vdots & & \vdots \\ u_{k1} & \cdots & u_{kn} \\ v_{11} & \cdots & v_{1n} \\ \vdots & & \vdots \\ v_{k1} & \cdots & v_{kn} \end{bmatrix}$$

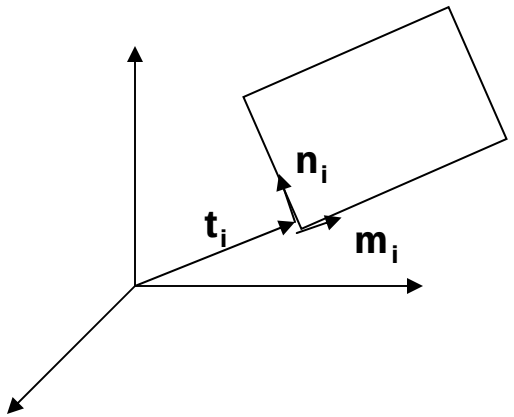
Structure of W

- World point $s_j' = (x_j', y_j', z_j')$ projects to image points

$$u'_{ij} = m_i^T (s_j' - t_i)$$

$$v'_{ij} = n_i^T (s_j' - t_i)$$

  - Where m_i, n_i are unit vectors defining orientation of image plane in world
  - And t_i is vector from world origin to image plane origin



Structure of W (Cont'd)

- Can rewrite in centroid normalized coordinates
  - Since centroid preserved under projection
  - Projection of centroid is centroid of projection

$$u_{ij} = m_i^T s_j$$

$$v_{ij} = n_i^T s_j$$

  - Where

$$s_j = s_j' - \bar{s}$$

and

$$\bar{s} = (1/n) \sum_j s'_j$$

W Factors Into Simple Product

- W=MS where
  - M is 2kx3 matrix of camera locations
  - S is 3xn matrix of points in world
  - Product is 2kx3 matrix W
    - Clearly rank at most 3

$$M = \begin{bmatrix} m_1^T \\ \vdots \\ m_k^T \\ n_1^T \\ \vdots \\ n_k^T \end{bmatrix} \qquad S = \begin{bmatrix} s_1 & \cdots & s_n \end{bmatrix}$$

Factoring W

- Don't know M,S only measurements W
- Given noise or errors in measurements seek least squares approximation
  - Note assuming no outliers (bad data)

$$\text{argmin}_{M,S} \| W\text{-}MS \|^2$$

- Several methods for solving linear least squares problems
  - Here highly rank deficient, use SVD

Singular Value Decomposition

- Seek M,S from the SVD of $W=U\Sigma V$
  - Where U and V are orthogonal and Σ is diagonal matrix
- Know from structure of problem that rank is at most 3
  - Consider only 3 largest singular values, let Σ' denote matrix with other singular values set to zero
- Then estimate

  $M^* = U\,\Sigma'^{1/2}$

  $S^* = \Sigma'^{1/2} V$

Factorization Not Unique

- Any linear transformation of M,S possible

  $W = MS = M(LL^{-1})S = (ML)(L^{-1}S)$

- Often referred to as "affine shape"
  - Preserves parallelism/coplanarity
- Still haven't used a constraint on the form of M
  - Describes camera plane orientation at each frame

    m_i, n_i all unit vectors

    $m_i n_i = 0$

$$\begin{bmatrix} m_1^T \\ \vdots \\ m_k^T \\ n_1^T \\ \vdots \\ n_k^T \end{bmatrix}$$

Factorization Results

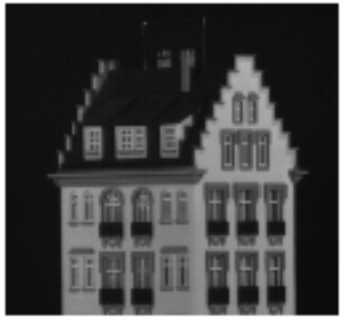
1

40

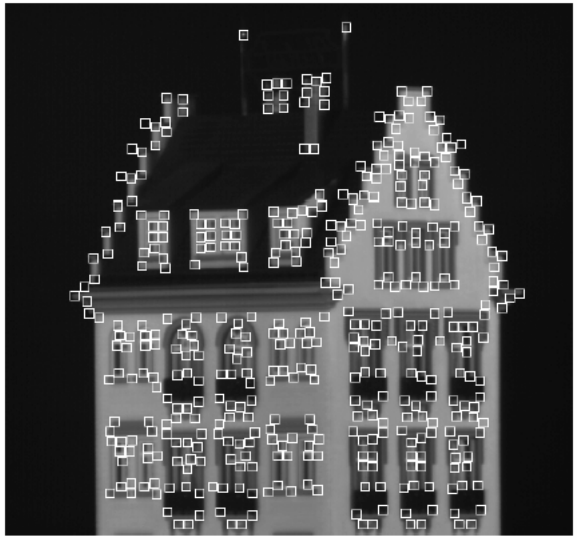
60

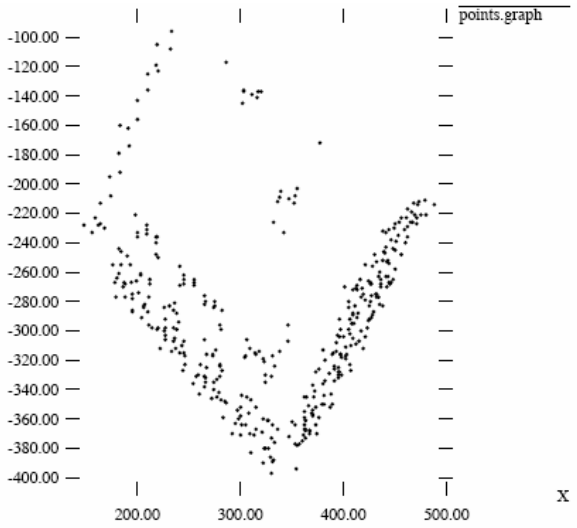
80

120



150





Extensions

- Paraperspective
  [Poelman & Kanade, PAMI 97]
- Sequential Factorization
  [Morita & Kanade, PAMI 97]
- Factorization under perspective
  [Christy & Horaud, PAMI 96]
  [Sturm & Triggs, ECCV 96]
- Factorization with Uncertainty
  [Anandan & Irani, IJCV 2002]

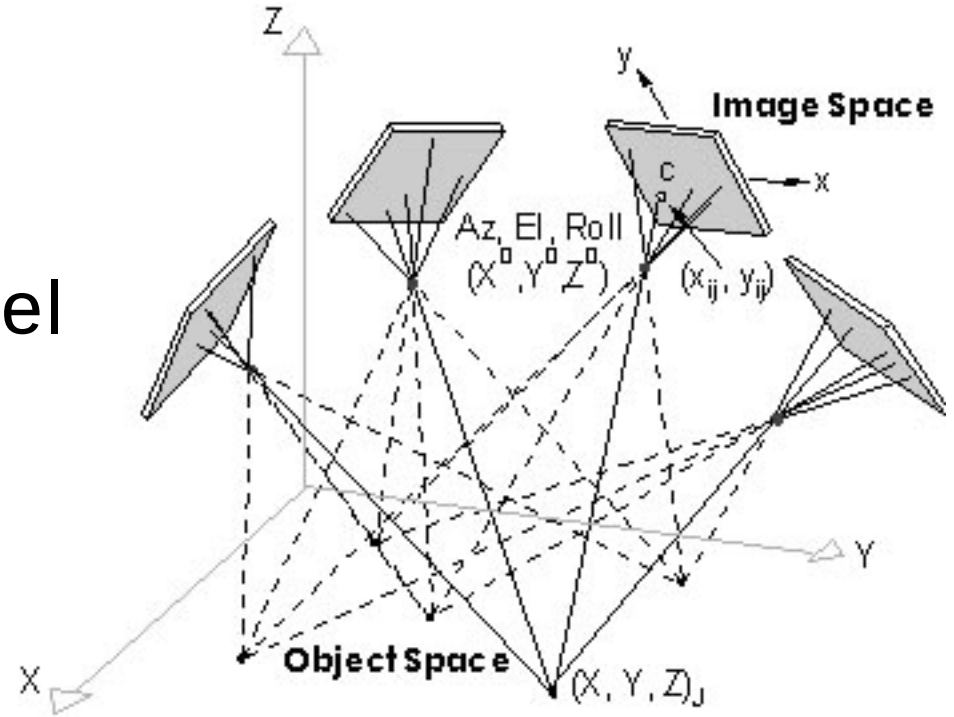
Bundle Adjustment

- More generally don't necessarily have linear least squares form of problem
- Technique from photogrammetry literature dating back many years
- Needs good initialization
- Estimate projection matrices and 3D points which minimize image distance d between re-projected and measured points

$$\min_{P_i^{'}, X_j^{'}} \sum_{i,j} d(P_i^{'} X_j^{'}, x_{ij})$$

Bundle Adjustment

- Involves adjusting bundle of rays between each camera center and set of 3D points
  - Or equivalently, each 3D point and set of camera centers
- Maximum likelihood estimate under Gaussian noise model
- X_j's depend on P_i's and vice versa
  - Solved iteratively



Iterative Minimization

- Local search from initial solution
  - Convergence depends on solution quality
- In full projective case each camera has 11 dof and each point 3 dof
  - Often use 12 parameter homogeneous P matrix, so 3n+12k
- Using Levenberg-Marquardt algorithm
  - Matrices of dimension (3n+12k) x (3n+12k) can be slow to factor/invert
- Various approaches

Addressing Computational Issues

- Solve smaller problems and merge…
- Interleave by alternately minimizing error by moving cameras for fixed point locations and vice versa
  - Limits matrices to 12x12 (or number of dof squared)
  - May have different convergence properties
- Sparse matrix methods
- Initial estimate can be obtained using factorization if (nearly) affine

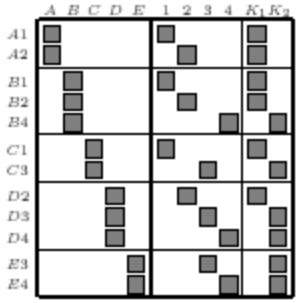
Sparsity

$$\begin{aligned} \hat{u}_{ij} &= f(\mathbf{K}, \mathbf{R}_j, \mathbf{t}_j, \mathbf{x}_i) \\ \hat{v}_{ij} &= g(\mathbf{K}, \mathbf{R}_j, \mathbf{t}_j, \mathbf{x}_i) \end{aligned}$$

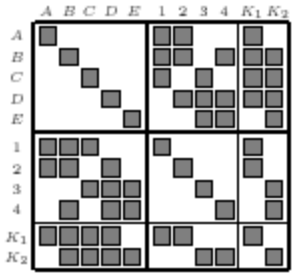
- Only a few entries in Jacobian are non-zero

$$\frac{\partial \hat{u}_{ij}}{\partial \mathbf{K}}, \quad \frac{\partial \hat{u}_{ij}}{\partial \mathbf{R}_j}, \quad \frac{\partial \hat{u}_{ij}}{\partial \mathbf{t}_j}, \quad \frac{\partial \hat{u}_{ij}}{\partial \mathbf{x}_i},$$

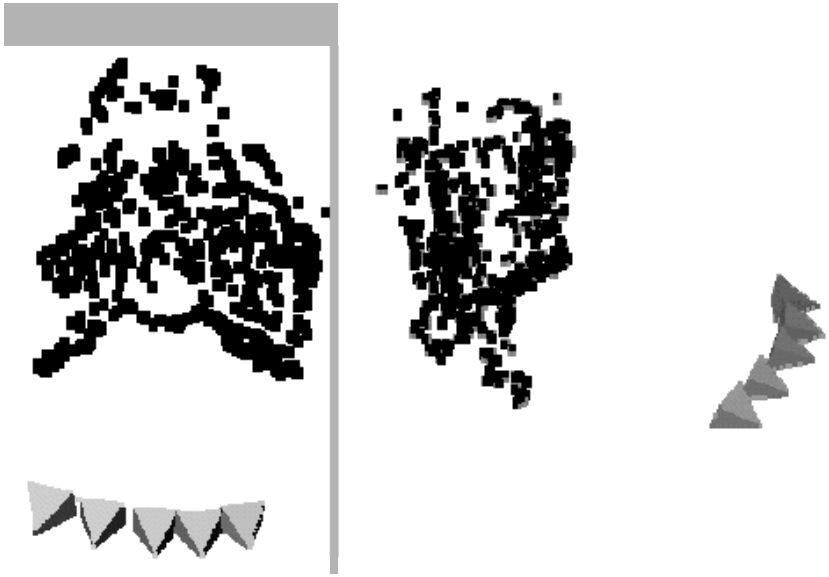
J =



H =

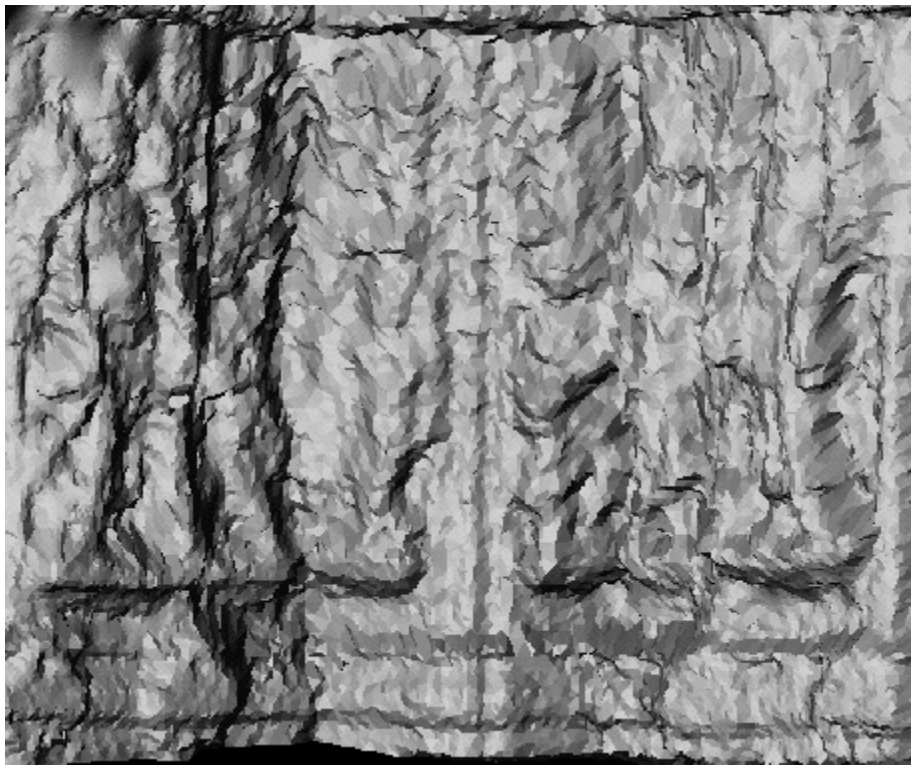


Example 3D Reconstruction



[Pollefeys 98-01]

Example Cont'd



Structure from Motion: Limitations

- Very difficult to reliably estimate *metric* structure and motion unless:
  - Large (*x* or *y*) rotation, or
  - Large field of view and depth variation
- Camera calibration important for Euclidean reconstructions
- Need good feature tracker