Visual Odometry

Introduction

Some slides were adapted/taken from various sources, including 3D Computer Vision of Prof. Hee, NUS, Air Lab Summer School, The Robotic Institute, CMU, Computer Vision of Prof. Mubarak Shah, UCF, Computer Vision of Prof. William Hoff, Colorado School of Mines, Coursera Visual Odometry, Robotics: Perception, University of Pennsylvania and many more. We thankfully acknowledge them. Students are requested to use this material for their study only and **NOT** to distribute it.

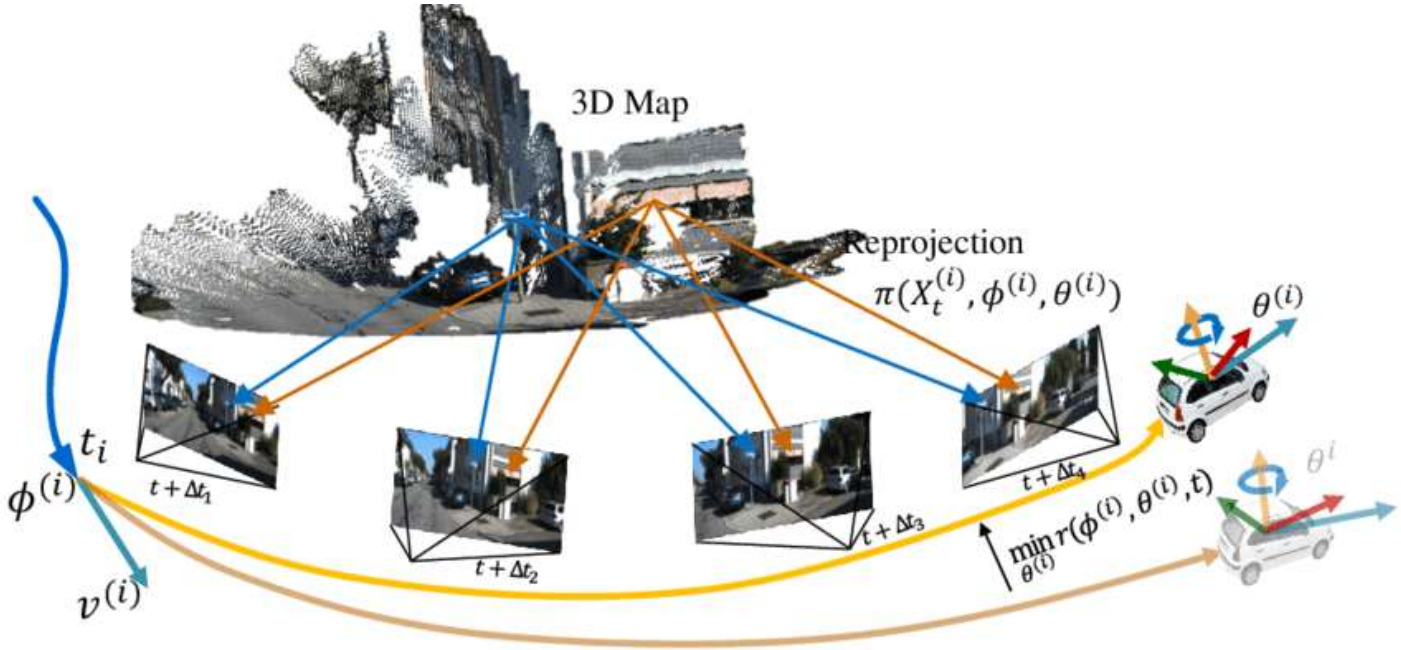
Visual Odometry

- What is odometry?
  - Measuring how far you go by counting wheel rations or steps.
  - Known as "path integration" in biological perception.
  - More general, integration of velocity or acceleration measurements: inertial odometry.

- What is visual odometry?
  - The process of incrementally estimating your position and orientation with respect with respect to an initial reference frames by tracking visual features.

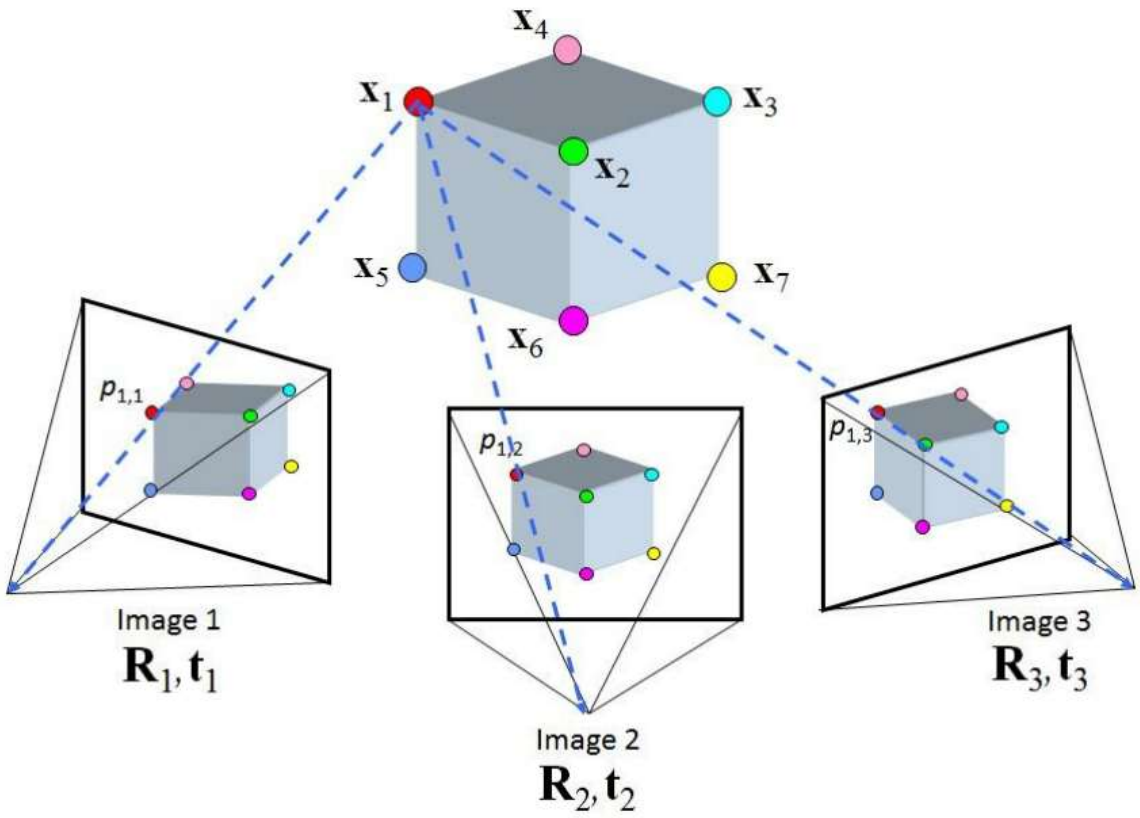
Visual Odometry

VO is defined as the process of estimating the robot's motion (translation and rotation with respect to a reference frame) by observing a sequence of images of its environment.



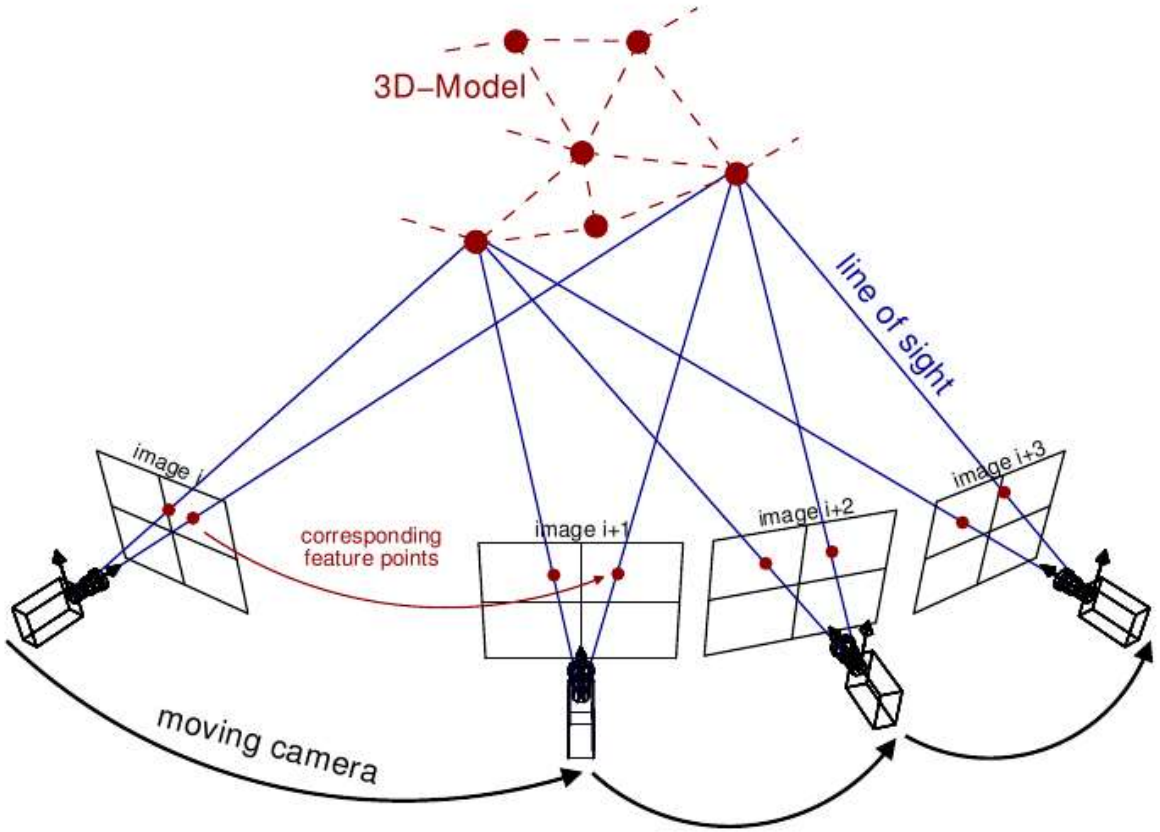
Structure from motion

VO is a particular case of a technique known as Structure From Motion (SFM) that tackles the problem of 3D reconstruction of both the structure of the environment and camera poses from sequentially ordered or unordered image sets [101].



Structure from motion

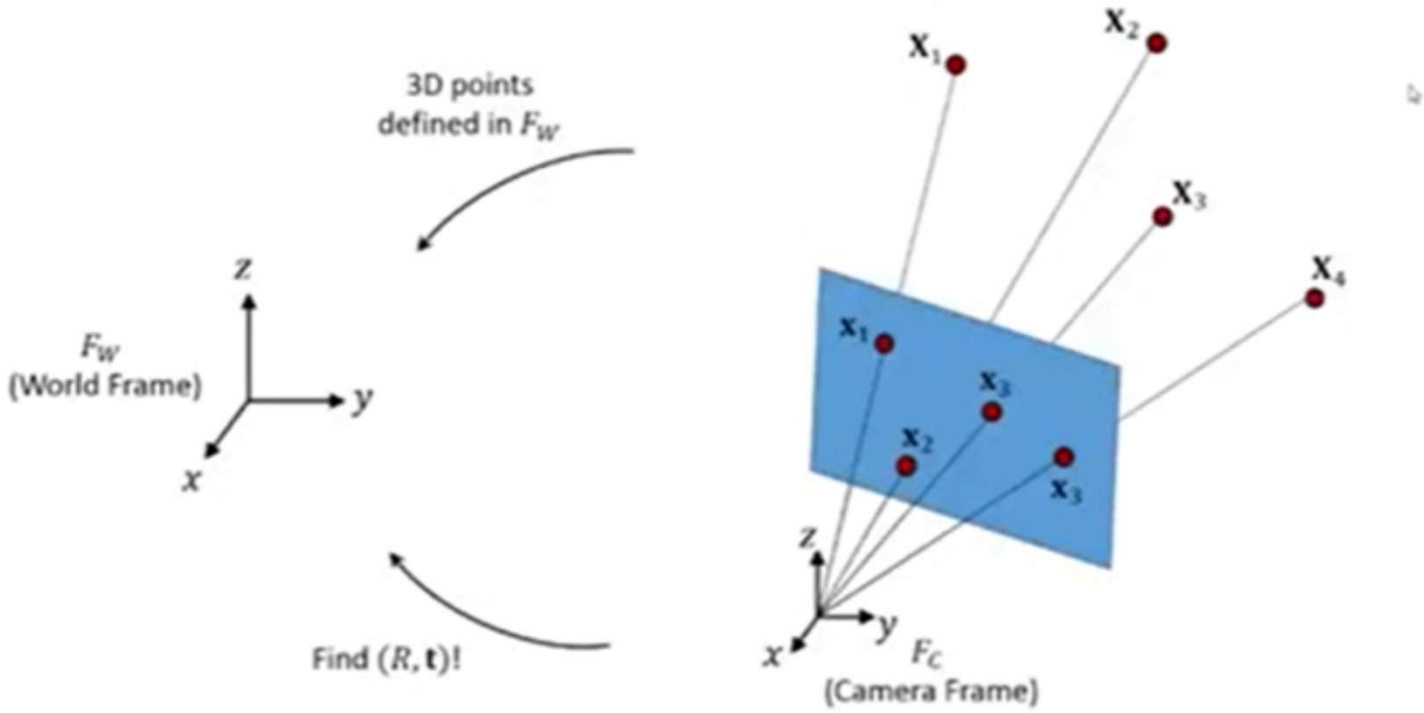
SFM's final refinement and global optimization step of both the camera poses and the structure is computationally expensive and usually performed off-line. However, the estimation of the camera poses in VO is required to be conducted in real-time.



Topics Covered

- Pose estimation
- Structure from motion
- Stereo Vision

Pose Estimation



Visual Odometry

- The term "Visual Odometry" was first introduced by Nister et al. [88] for it's similarity to the concept of wheel odometry.

- They proposed pioneering methods for obtaining camera motion from visual input in both monocular and stereo systems.

- They focused on the problem of estimating the camera motion in the presence of outliers (false feature matches) and proposed an outlier rejection scheme using RANSAC [44].

- Nister et al. were also the first to track features across all frames instead of matching features in consecutive frames. This has the benefit of avoiding feature drift during cross-correlation based tracking [101].

- They also proposed a RANSAC based motion estimation using the 3D to 2D re-projection error (see "Motion Estimation" section) instead of using the Euclidean distance error between 3D points. Using 3D to 2D re-projection errors were shown to give better estimates when compared to the 3D to 3D errors [56].

Topics to cover:

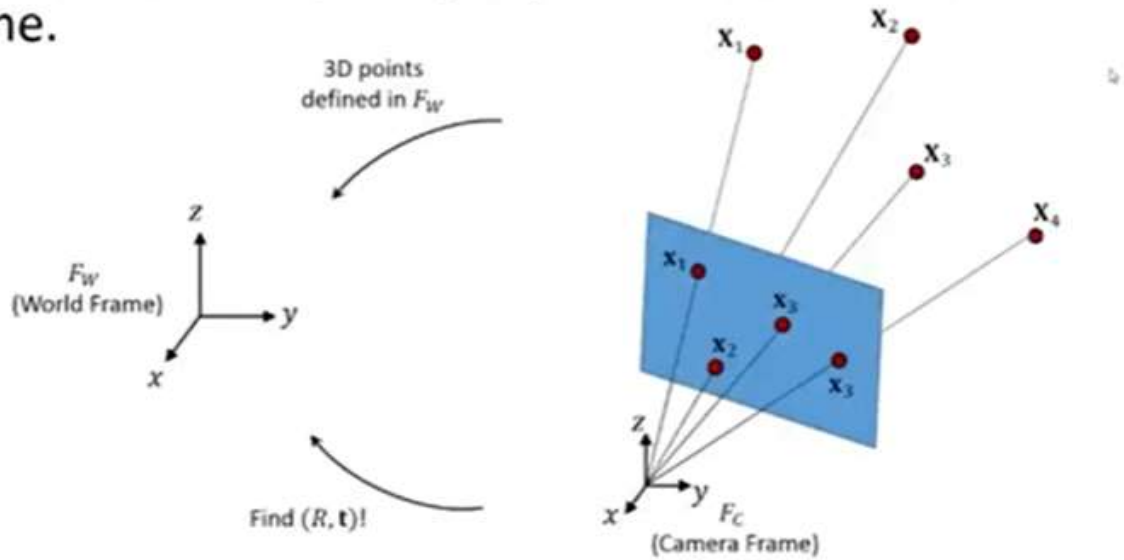
- Pose estimation
- Structure for motion
- Bundle Adjustment
- Visual SLAM

Pose Estimation

Perspective Pose Estimation Problem

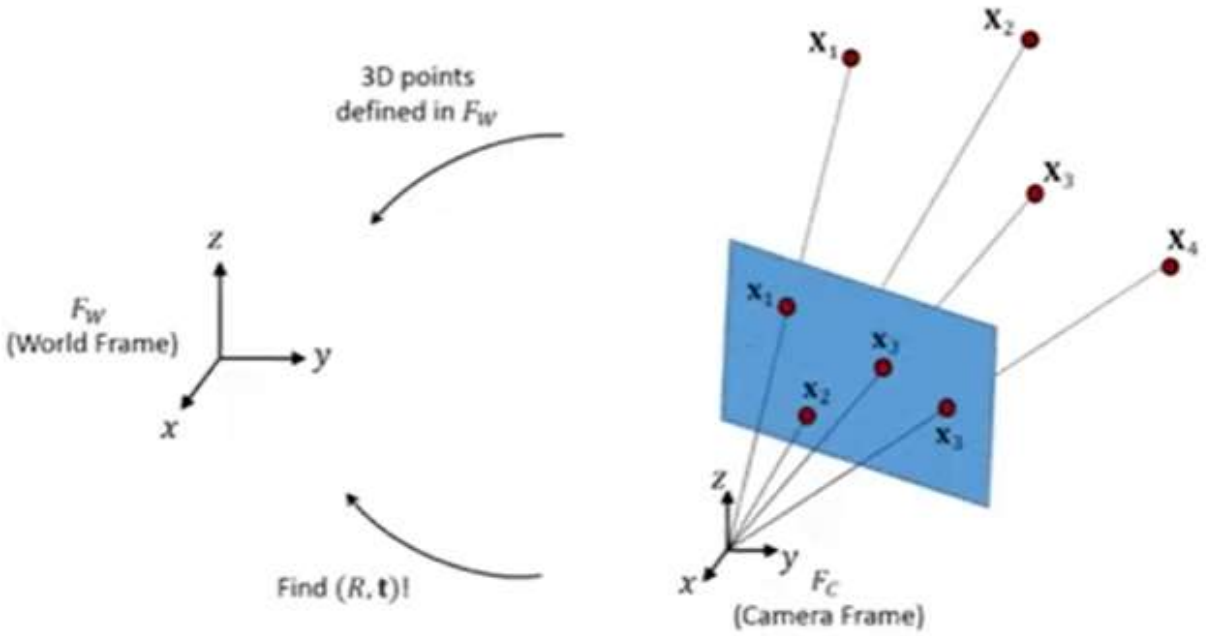
Given: a set of 3D points $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N\}$ defined in a world coordinate frame, and its corresponding 2D image points $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, i.e. $\{\mathbf{X}_i \leftrightarrow \mathbf{x}_i\}$.

Find: the camera pose $(R, \mathbf{t})$ in the world coordinate frame.



Perspective Pose Estimation Problem

- This problem is also known as the "Perspective-n-Point" or PnP problem.



Un-calibrated Camera: Unknown K

- We are required to find a camera matrix P, i.e. a 3×4 matrix such that $\gamma_i \mathbf{x}_i = \mathrm{P}\mathbf{X}_i$, from the correspondences $\{\mathbf{X}_\mathrm{i} \leftrightarrow \mathbf{x}_i\}, \forall i$.

- The camera projection can be written as cross-product to eliminate the unknown scale γ_i:

$$(\gamma_i \mathbf{x}_i) \times \mathrm{P}\mathbf{X}_i = 0 \quad \Rightarrow \quad \begin{bmatrix} \mathbf{0}^\mathsf{T} & -w_i \mathbf{X}_i^\mathsf{T} & y_i \mathbf{X}_i^\mathsf{T} \\ w_i \mathbf{X}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} & -x_i \mathbf{X}_i^\mathsf{T} \\ -y_i \mathbf{X}_i^\mathsf{T} & x_i \mathbf{X}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} \end{bmatrix} \begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{pmatrix} = 0.$$

- where each $\mathbf{P}^{iT}$ is a 4-vector, the i-th row of P.

Un-calibrated Camera: Unknown K

- Only the first two equations are independent:

$$\begin{bmatrix} \mathbf{0}^\mathsf{T} & -w_i\mathbf{X}_i^\mathsf{T} & y_i\mathbf{X}_i^\mathsf{T} \\ w_i\mathbf{X}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} & -x_i\mathbf{X}_i^\mathsf{T} \\ -y_i\mathbf{X}_i^\mathsf{T} & x_i\mathbf{X}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} \end{bmatrix} \begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{pmatrix} = 0 \quad \Rightarrow \quad \begin{bmatrix} \mathbf{0}^\mathsf{T} & -w_i\mathbf{X}_i^\mathsf{T} & y_i\mathbf{X}_i^\mathsf{T} \\ w_i\mathbf{X}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} & -x_i\mathbf{X}_i^\mathsf{T} \end{bmatrix} \begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{pmatrix} = 0$$

- From a set of n point correspondences, we obtain a $2n \times 12$ matrix A by stacking up the equations for each correspondence.

- P is computed by solving the set of equations $A\mathbf{p} = \mathbf{0}$, where $\mathbf{p}$ is the 12-vector containing the entries of the matrix P.

Un-calibrated Camera: Unknown K

Minimal solution:

- Since the matrix P has 11 dofs (12 entries – 1 dof for scale), a minimum of 5.5 correspondences are required.
- Effectively 6 point correspondences are needed, where only one of the equations is used from the sixth point.
- The solution is obtained by solving $\mathrm{A}\mathbf{p} = \mathbf{0}$, where A is an 11×12 matrix in this case.
- In general A will have rank 11, and the solution vector $\mathbf{p}$ is the 1-dimensional right null-space of A.

Un-calibrated Camera: Unknown K

Over-determined solution:

- No exact solution to $\mathrm{A}\mathbf{p} = \mathbf{0}$ when data is noisy and $n \geq 6$ point correspondences are needed.
- P may be obtained by minimizing an algebraic or geometric error.
- Minimize algebaric error $\|\mathrm{A}\mathbf{p}\|$ subject to a normalization constraint using SVD:

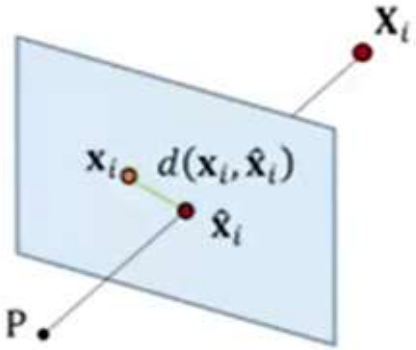
i. $\|\mathbf{p}\| = 1$;

ii. $\|\hat{\mathbf{p}}^3\| = 1$, where $\hat{\mathbf{p}}^3$ is the vector $(p_{31}, p_{32}, p_{33})^T$, i.e. the first three entries in the last row of P.

Un-calibrated Camera: Unknown K

- Minimize the geometric error:

$$\min_{\mathrm{P}} \sum_i d(\mathbf{x}_i, \mathrm{P}\mathbf{X}_i)^2$$



- where $\mathbf{x}_i$ is the measured point and $\hat{\mathbf{x}}_i$ is the point $\mathrm{P}\mathbf{X}_i$, i.e. the point which is the exact image of $\mathbf{X}_i$ under P.

- $d(\mathbf{x}, \mathbf{y})$ is the Euclidean distance between two points $\mathbf{x}, \mathbf{y}$.

- Minimization with Levenberg–Marquardt.

Un-calibrated Camera: Unknown K

Data normalization:

- The 2D points $\mathbf{x}_i$ should be translated for the centroid to be at the origin, and scaled for the RMS distance from the origin to be $\sqrt{2}$.

$$\mathrm{T}_{\mathrm{norm}} = \begin{bmatrix} s & 0 & -sc_x \\ 0 & s & -sc_y \\ 0 & 0 & 1 \end{bmatrix}$$

c: centroid of all 2D image points

$$s = \frac{\sqrt{2}}{\bar{d}}$$

where $\bar{d}$: mean distance of all points from centroid.

Un-calibrated Camera: Unknown K

Data normalization:

- Similarly, the 3D points $\mathbf{X}_i$ should be translated for the centroid to be at the origin, and scaled for the RMS distance from the origin to be $\sqrt{3}$.

$$\mathrm{U}_{\mathrm{norm}} = \begin{bmatrix} s & 0 & 0 & -sc_x \\ 0 & s & 0 & -sc_y \\ 0 & 0 & s & -sc_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

c: centroid of all 3D points

$$s = \frac{\sqrt{3}}{\bar{d}}$$

where $\bar{d}$: mean distance of all points from centroid.

Un-calibrated Camera: Unknown K

Objective

Given $n \geq 6$ world to image point correspondences $\{\mathbf{X}_i \leftrightarrow \mathbf{x}_i\}$, determine the Maximum Likelihood estimate of the camera projection matrix $\mathtt{P}$, i.e. the $\mathtt{P}$ which minimizes $\sum_i d(\mathbf{x}_i, \mathtt{P}\mathbf{X}_i)^2$.

Algorithm

(i) **Linear solution.** Compute an initial estimate of $\mathtt{P}$ using a linear method such as algorithm 4.2(p109):

  (a) **Normalization:** Use a similarity transformation $\mathtt{T}$ to normalize the image points, and a second similarity transformation $\mathtt{U}$ to normalize the space points. Suppose the normalized image points are $\tilde{\mathbf{x}}_i = \mathtt{T}\mathbf{x}_i$, and the normalized space points are $\tilde{\mathbf{X}}_i = \mathtt{U}\mathbf{X}_i$.

  (b) **DLT:** Form the $2n \times 12$ matrix $\mathtt{A}$ by stacking the equations (7.2) generated by each correspondence $\tilde{\mathbf{X}}_i \leftrightarrow \tilde{\mathbf{x}}_i$. Write $\mathbf{p}$ for the vector containing the entries of the matrix $\mathtt{P}$. A solution of $\mathtt{A}\mathbf{p} = \mathbf{0}$, subject to $\|\mathbf{p}\| = 1$, is obtained from the unit singular vector of $\mathtt{A}$ corresponding to the smallest singular value.

(ii) **Minimize geometric error.** Using the linear estimate as a starting point minimize the geometric error (7.4):

$$\sum_i d(\tilde{\mathbf{x}}_i, \tilde{\mathtt{P}}\tilde{\mathbf{X}}_i)^2$$

over $\tilde{\mathtt{P}}$, using an iterative algorithm such as Levenberg–Marquardt.

(iii) **Denormalization.** The camera matrix for the original (unnormalized) coordinates is obtained from $\tilde{\mathtt{P}}$ as

$$\mathtt{P} = \mathtt{T}^{-1}\tilde{\mathtt{P}}\mathtt{U}.$$

Table Source: R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision"

To continue…

Structure from Motion

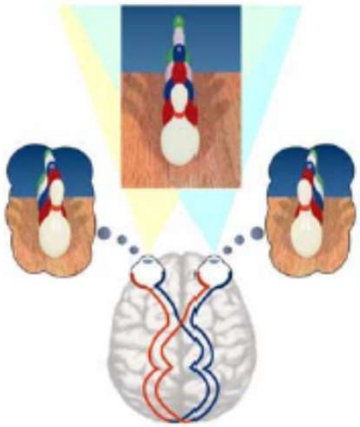
Shape from X

- Recovery of 3D (shape) from one or two (2D images).

Shape from X

- Stereo
- Motion
- Shading
- Photometric Stereo
- Texture
- Contours
- Silhouettes

Shape from stereo



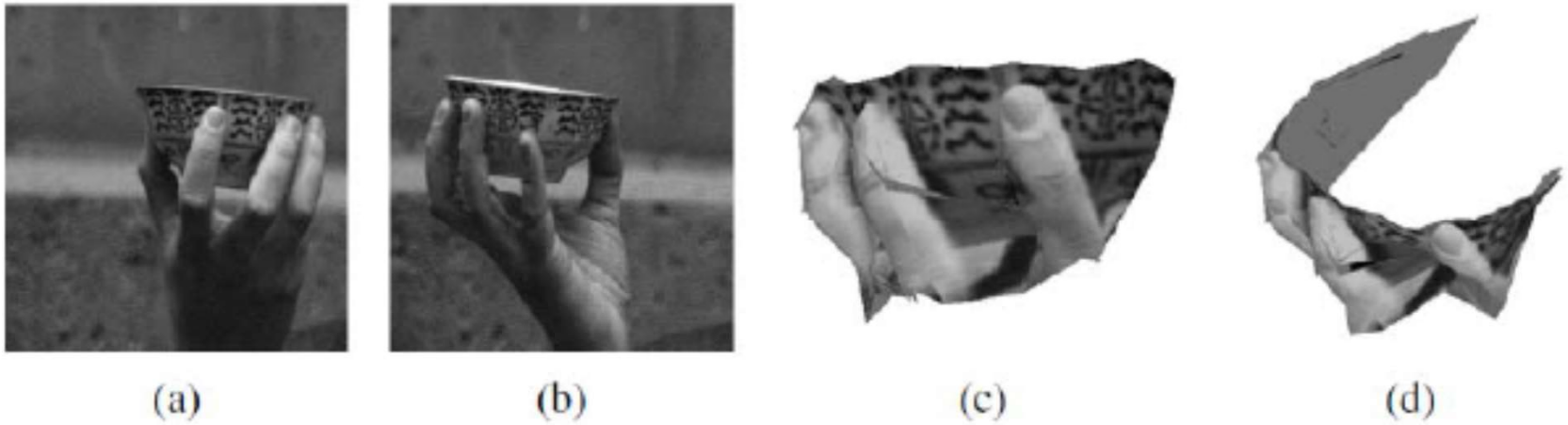
(a) (b) (c)

Shape from shading

Shape from motion

Moving light display

Shape from motion



(a) (b) (c) (d)

Problem

- Given optical flow or point correspondences, compute 3-D motion (translation and rotation) and shape (depth).

Structure from motion

- S. Ullman
- Hanson & Riseman
- Webb & Aggarwal
- T. Huang
- Heeger and Jepson
- Chellappa
- Faugeras
- Zisserman
- Kanade
- Pentland
- Van Gool
- Pollefeys
- Seitz & Szeliski
- Shahsua
- Irani
- Vidal & Yi Ma
- Medioni
- Fleet
- Tian & Shah

Tomasi and Kanade Factorization

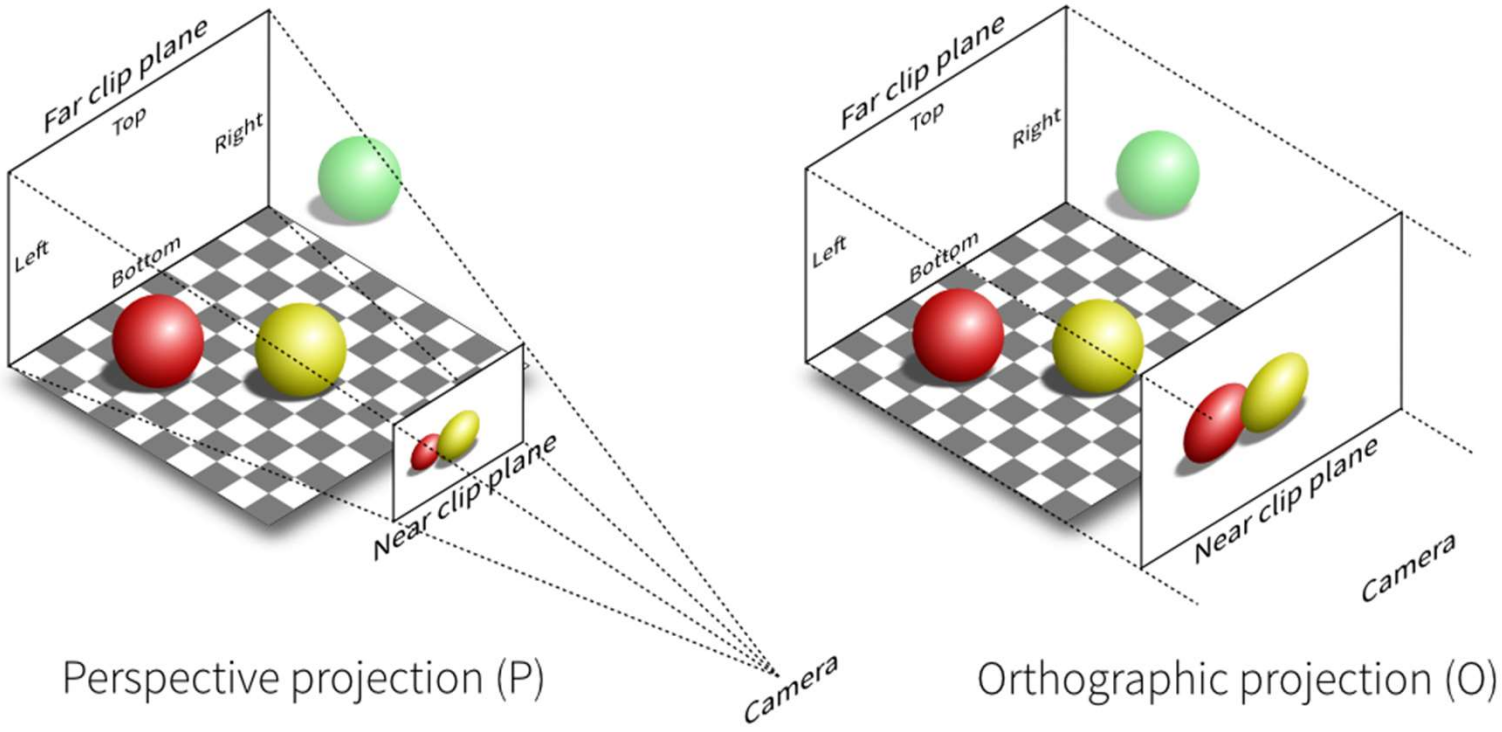
Orthographic Projection

Orthographic projection (camera)

- Orthographic projection is a means of representing three-dimensional objects in two dimensions. It is a form of parallel projection, in which all the projection lines are orthogonal to the projection plane, resulting in every plane of the scene appearing in affine transformation on the viewing surface.

- A perspective camera is how we see the real world. If we take a look at the things around us, they have depth and we can judge their distance. ... An **orthographic camera however removes this sense of perspective**. Objects are drawn without perspective distortion.

Orthographic projection (camera)

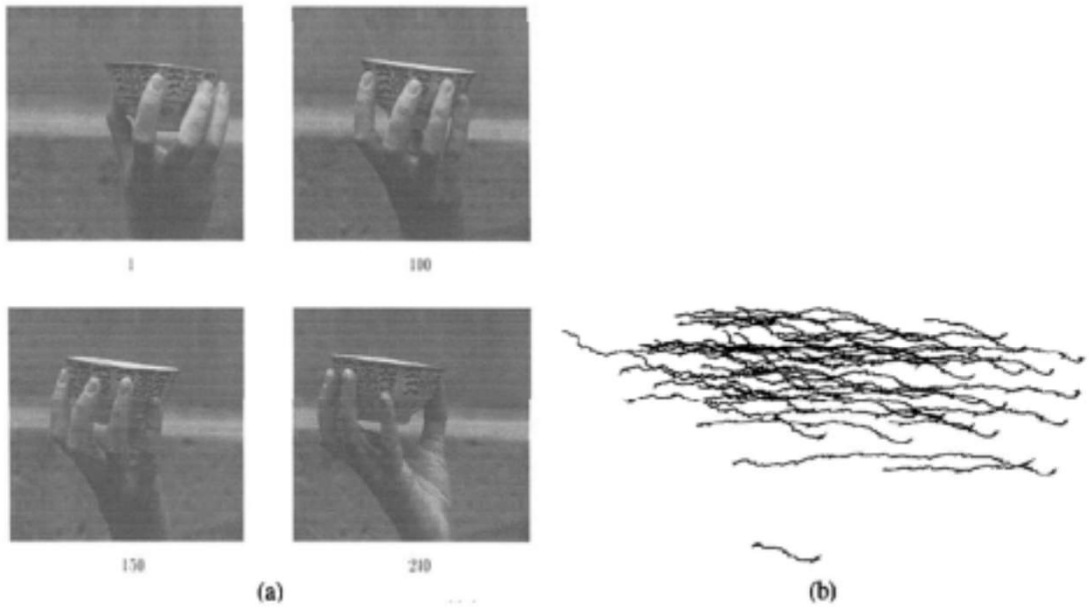


Assumptions

- The camera model is orthographic.
- The positions of “p” points in “F” frames (F>=3), which are not all coplanar, and have been tracked.
- The entire sequence has been acquired before starting (batch mode).
- Camera calibration not needed, if we accept 3D points up to a scale factor.

Input

KLT tracks gives the image points, the locations of the points in the different frames



Images KLT Tracks

Feature points

Image points $\left\{(u_{fp}, v_{fp}) \mid f = 1,\ldots,F,\ p = 1,\ldots,P\right\}$

$$W = \begin{bmatrix} u_{11}\ldots u_{1p} \\ \vdots \\ u_{F1}\ldots u_{FP} \\ v_{11}\ldots v_{1P} \\ \vdots \\ v_{F1}\ldots v_{FP} \end{bmatrix} \qquad W = \begin{bmatrix} U \\ - \\ V \end{bmatrix}$$

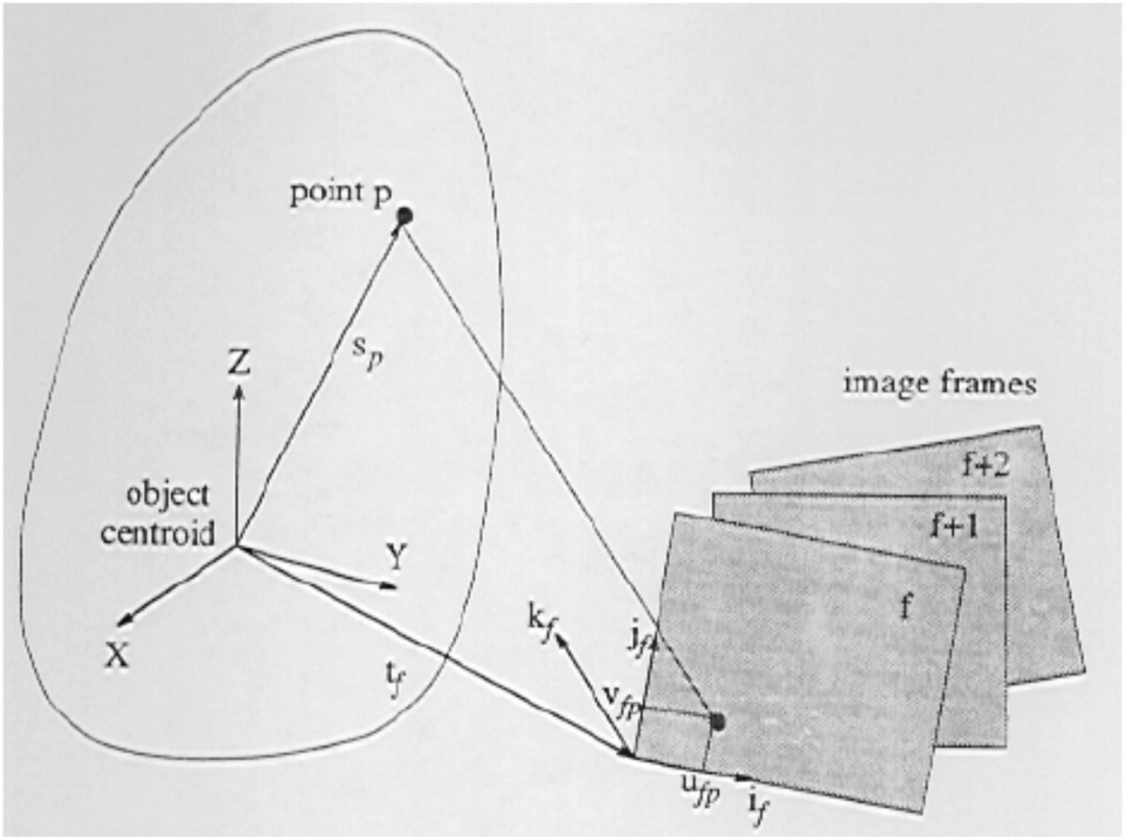
Mean normalized feature points

$$a_f = \frac{1}{P}\sum_{p=1}^{P} u_p \qquad b_f = \frac{1}{P}\sum_{p=1}^{P} v_p$$

$$\tilde{u}_{fP} = u_{fP} - a_{fP} \qquad \text{(A)}$$

$$\tilde{v}_{fP} = v_{fP} - b_{fP}$$

Orthographic projection



Orthographic projection: Features points

$$s_p = (X_p, Y_P, Z_P)$$

3D world point

$$u_{fP} = i_f^T (s_P - t_f) \quad (C)$$

$$v_{fP} = j_f^T (s_P - t_f)$$

Orthographic projection

$$k_f = i_f \times j_f$$

i, j, k are unit vectors along X, Y, Z

Orthographic projection: Features points

$$u_{fP} = i_f^T (s_P - t_f)$$

For a world co-ordinate point (S_P), we subtract the translation (t_f) and then project to on x-axis of the camera coordinate system

$$v_{fP} = j_f^T (s_P - t_f)$$

For a world co-ordinate point (S_P), we subtract the translation (t_f) and then project to on y-axis of the camera coordinate system

$$k_f = i_f \times j_f$$

Then we do the cross product to get the Z-axis.

i, j, k are unit vectors along X, Y, Z

Orthographic projection: Mean normalized Features points

$$\tilde{u}_{fp} = u_{fP} - a_f \qquad a_f = \frac{1}{P}\sum_{n=1}^{P} u_p$$

$$= i_f^T (s_p - t_f) -$$

$$= i_f^T \left[s_P - \frac{1}{P}\sum_{q=1}^{P} s_q \right]$$

$$= i_f^T s_P$$

If the origin of the world coordinate system is placed as the centroid of the object points, the 2nd term become zero.

If Origin of world is at the centroid of object points

Orthographic projection: Mean normalized Features points

$$\tilde{u}_{fp} = u_{fP} - a_f \qquad a_f = \frac{1}{P}\sum_{p=1}^{P} u_p$$

$$= i_f^T (s_p - t_f) - \frac{1}{P}\sum_{q=1}^{P} i_f^T (s_q - t_f)$$

$$= i_f^T \left[s_P - \frac{1}{P}\sum_{q=1}^{P} s_q \right]$$

If Origin of world is at the centroid of object points

Registered measurement matrix

$$\tilde{u}_{fP} = i_f^T S_P$$

$$\tilde{v}_{fP} = j_f^T S_P$$

$$\tilde{W} = \begin{bmatrix} \tilde{U} \\ - \\ \tilde{V} \end{bmatrix}$$

Registered measurement matrix

$$\tilde{u}_{fP} = i_f^T s_P \qquad \text{(B)}$$

$$\tilde{v}_{fP} = j_f^T s_P$$

$$\tilde{W} = \begin{bmatrix} \tilde{U} \\ - \\ \tilde{V} \end{bmatrix}$$

$$\tilde{W} = \begin{bmatrix} \tilde{u}_{11} \dots \tilde{u}_{1p} \\ \vdots \\ \tilde{u}_{F1} \dots \tilde{u}_{FP} \\ \tilde{v}_{11} \dots \tilde{v}_{1P} \\ \vdots \\ \tilde{v}_{F1} \dots \tilde{v}_{FP} \end{bmatrix}$$

F no. of frames and P no. of points in each frame

$$\tilde{W} = \begin{bmatrix} i_1^T \\ \vdots \\ i_F^T \\ j_1^T \\ \vdots \\ j_F^T \end{bmatrix} \begin{bmatrix} s_1 & \dots & s_P \end{bmatrix} = RS$$

Motion

Shape

2FX3

3XP

Rank of S is 3, because points in 3D space are not Co-planar

Rank Theorem

Without noise, the registered measurement matrix $\tilde{W}$ is at most of rank three.

$$\tilde{W} = \begin{bmatrix} i_1^T \\ \vdots \\ i_F^T \\ j_1^T \\ \vdots \\ j_F^T \end{bmatrix} \begin{bmatrix} s_1 & \dots & s_P \end{bmatrix} = RS$$

Rank of S is 3 as it is 3D points in world frame

Linear Independence

A finite subset of n vectors, $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$, from the vector space V, is ***linearly dependent*** if and only if there exists a set of n scalars, $a_1, a_2, \ldots, a_n$, not all zero, such that

$$a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \cdots + a_n\mathbf{v}_n = \mathbf{0}.$$

Rank of a matrix

- The **column rank** of a matrix A is the maximum number of linearly independent column vectors of A.
- The **row rank** of a matrix A is the maximum number of linearly independent row vectors of A.
- The column rank of A is the dimension of the column space of A
- The row rank of A is the dimension of the row space of A.

How to find translation

$$\tilde{u}_{fp} = u_{fP} - a_f \qquad \text{From (A)}$$

$$u_{fp} = \tilde{u}_{fP} + a_f \quad \tilde{u}_{fp} = i_f^T s_P$$

From (B)

$$u_{fp} = i_f s_p + a_f \quad u_{fp} = i_f^T (s_p - t_f)$$

From (C)

$$a_f = -t_f^T i_f \qquad \text{(D)}$$

a_f is projection of camera translation along x-axis

Registered measurement matrix in terms of motion and shape

$$\tilde{u}_{fP} = i_f^T s_P \qquad \text{(B)}$$

$$\tilde{v}_{fP} = j_f^T s_P$$

$$\tilde{W} = \begin{bmatrix} \tilde{U} \\ - \\ \tilde{V} \end{bmatrix}$$

$$\tilde{W} = \begin{bmatrix} i_1^T \\ \vdots \\ i_F^T \\ j_1^T \\ \vdots \\ j_F^T \end{bmatrix}_{\text{2FX3}} \underbrace{\begin{bmatrix} s_1 & \dots & s_P \end{bmatrix}}_{\text{3XP}} = RS$$

Rank of S is 3, because points in 3D space are not Co-planar

Registered measurement matrix in terms of motion and shape

$$u_{fp} = i_f^T (s_p - t_f) \quad \text{From (C)} \qquad a_f = -t_f i_f^T \quad \text{(D)}$$

$$u_{fp} = i_f s_p + a_f \qquad v_{fp} = j_f s_p + b_f$$

$$\underset{2FXP}{\mathbf{W}} = \underset{2FX3}{\mathbf{R}} \underset{3XP}{\mathbf{S}} + \underset{2FX1}{\mathbf{t}} \underset{1XP}{\mathbf{e}_p^T}$$

$$\mathbf{t} = (a_1, \ldots, a_f, b_1, \ldots, b_f)^T$$

$$\mathbf{e}_p^T = (1, \ldots 1)$$

Registered measurement matrix in terms of motion and shape

$$u_{fp} = i_f s_p + a_f \quad v_{fp} = j_f s_p + b_f$$

$$\mathbf{W} = \mathbf{RS} + \mathbf{te}_{\mathbf{p}}^{\mathbf{T}}$$

2FXP 2FX3 3XP 2FX1 1XP

$$W = \begin{bmatrix} u_{11} \dots u_{1P} \\ \vdots \\ u_{F1} \dots u_{FP} \\ v_{11} \dots v_{1P} \\ \vdots \\ v_{F1} \dots v_{FP} \end{bmatrix} = \begin{bmatrix} i_1 & j_1 & k_1 \\ \vdots & & \\ & & \\ & & \\ \vdots & & \\ i_F & j_F & k_F \end{bmatrix} \begin{bmatrix} X_1 & \cdots & X_P \\ Y_1 & \cdots & Y_P \\ Z_1 & \cdots & Z_P \end{bmatrix} + \begin{bmatrix} a_1 \\ \vdots \\ a_F \\ b_1 \\ \vdots \\ b_F \end{bmatrix} \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}$$

Translation

Projected camera translation can be computed:

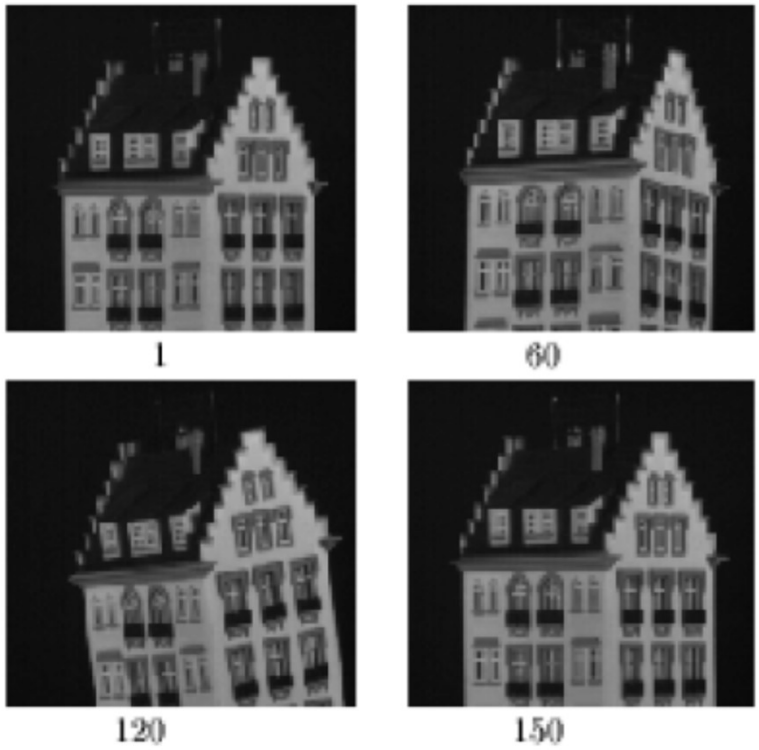
$$-i_f^T t_f = a_f = \frac{1}{P}\sum_{p=1}^{P} u_p$$

$$-j_f^T t_f = b_f = \frac{1}{P}\sum_{p=1}^{P} v_p$$

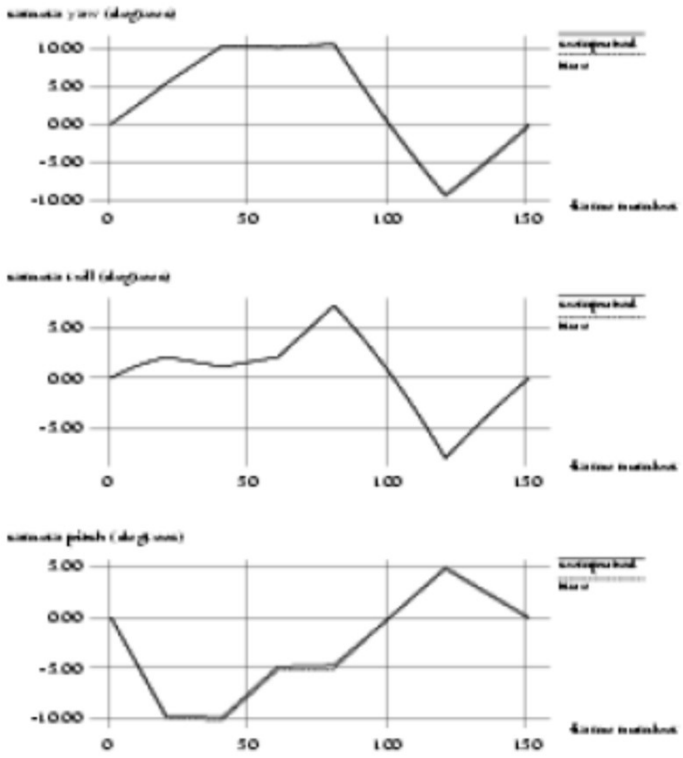
Algorithm

- Compute SVD of $\tilde{W} = O_1 \Sigma O_2$
- define $\hat{R} = O_1' [\Sigma']^{1/2}$ $\hat{S} = [\Sigma']^{1/2} O_2'$
- Compute Q
  - Compute $R = \hat{R} Q$ $S = Q^{-1} \hat{S}$

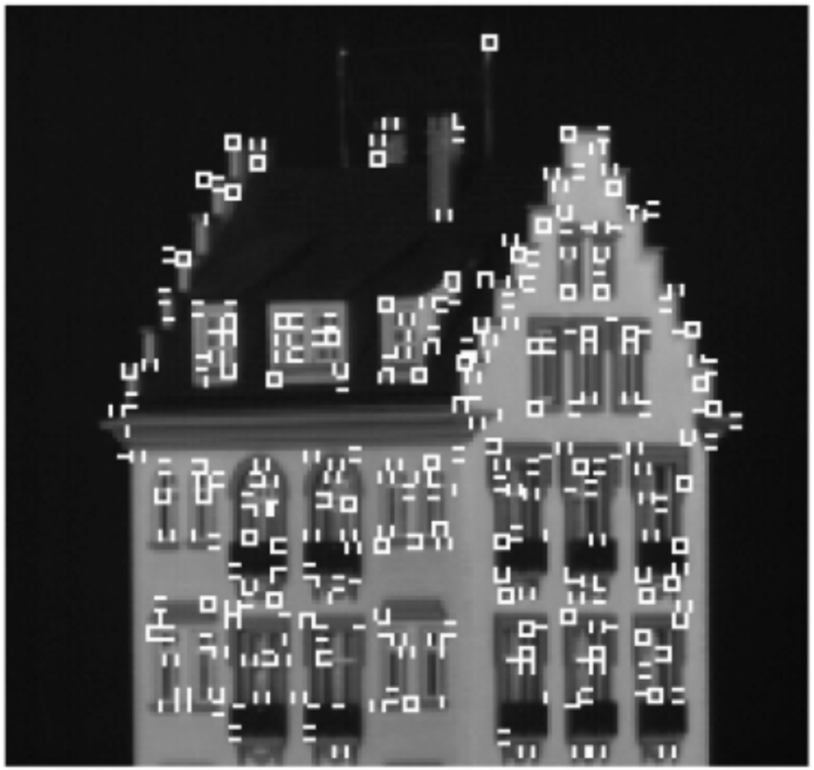
Hotel sequence



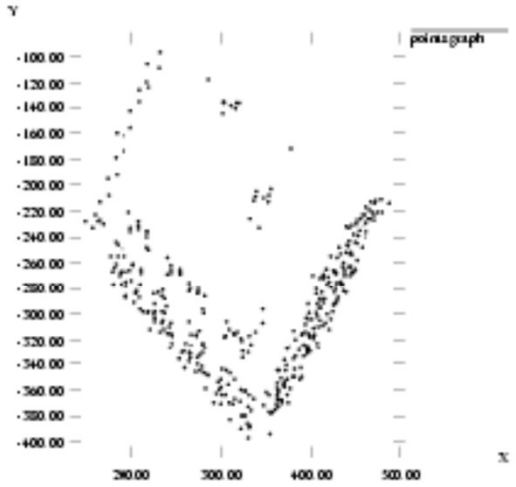
Results

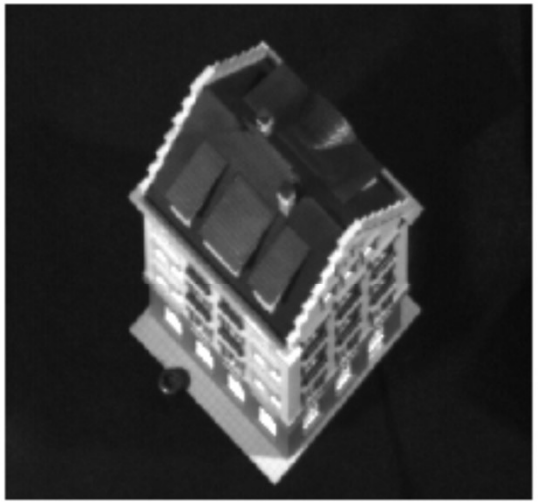

10.00
5.00
0.00
-5.00
-10.00
0
50
100
150
5.00
0.00
-5.00
0
50
100
150
5.00
0.00
-5.00
-10.00
0
50
100
150

Selected features

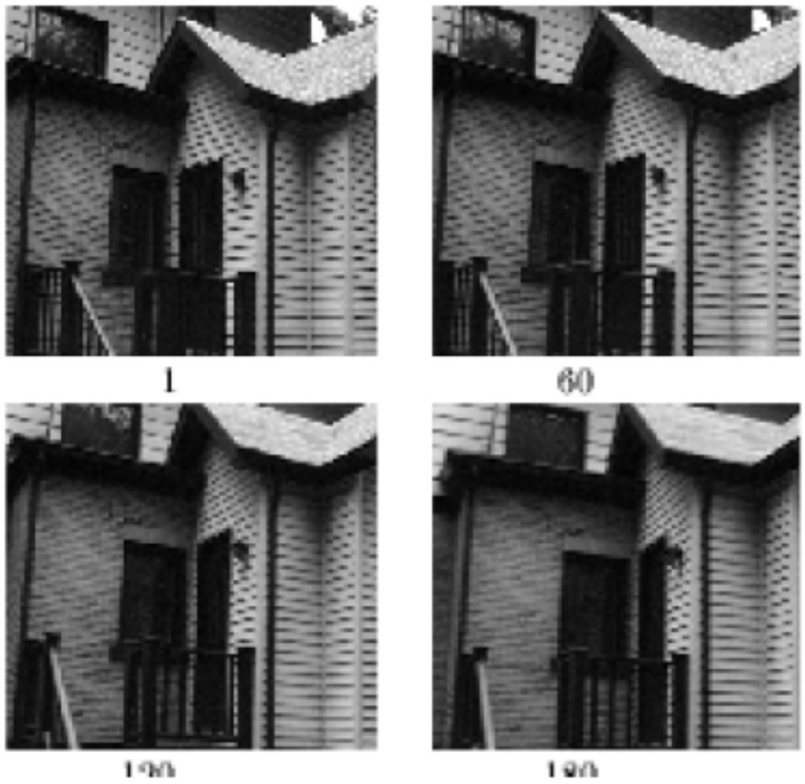


Reconstructed shape

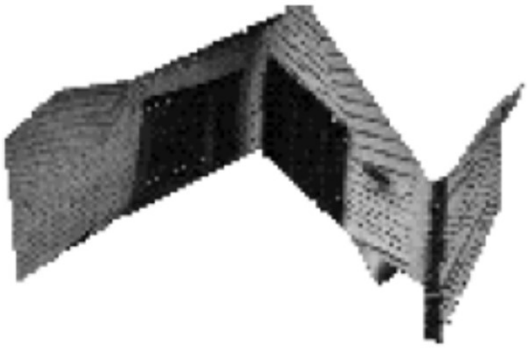




House sequence



Reconstructed walls



Paper

- C. Tomasi and T. Kanade. Shape and motion from image streams under orthography---a factorization method. *International Journal on Computer Vision*, 9(2):137-154, November 1992. (2357 citations)

To continue…

Stereo Vision

Visual Odometry

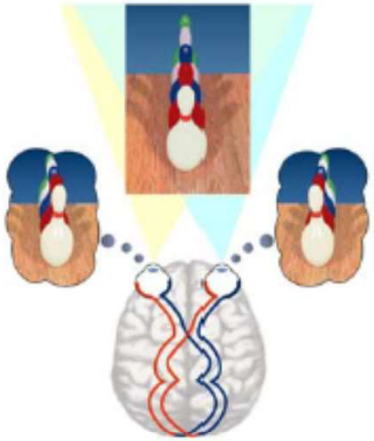
Shape from X

- Recovery of 3D (shape) from one or two (2D images).

Different ways

- Stereo
- Motion
- Shading
- Photometric Stereo
- Texture
- Contours
- Silhouettes

Shape from stereo



(a) (b) (c)

Stereo

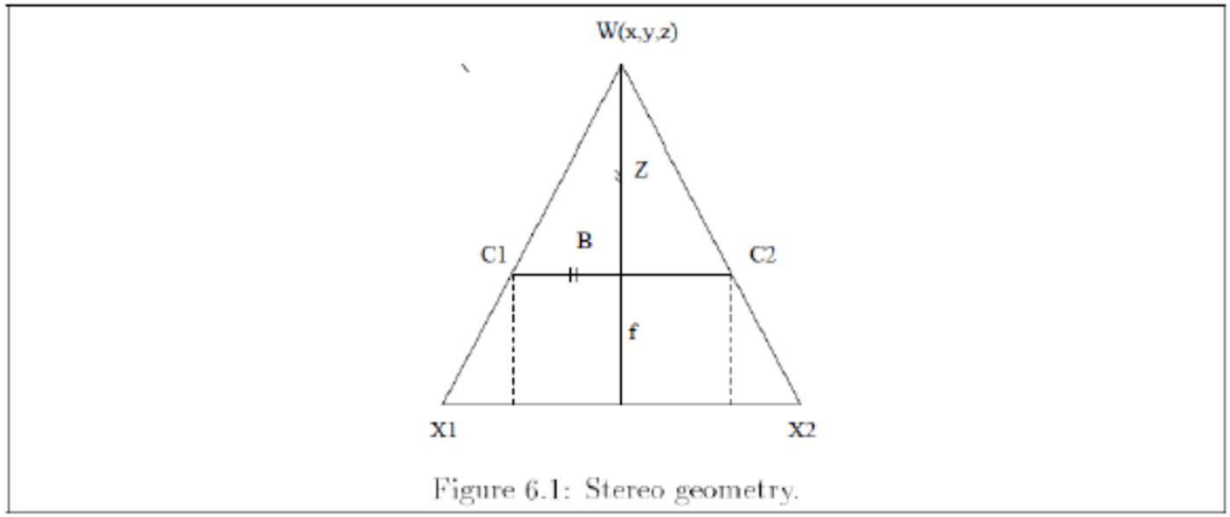


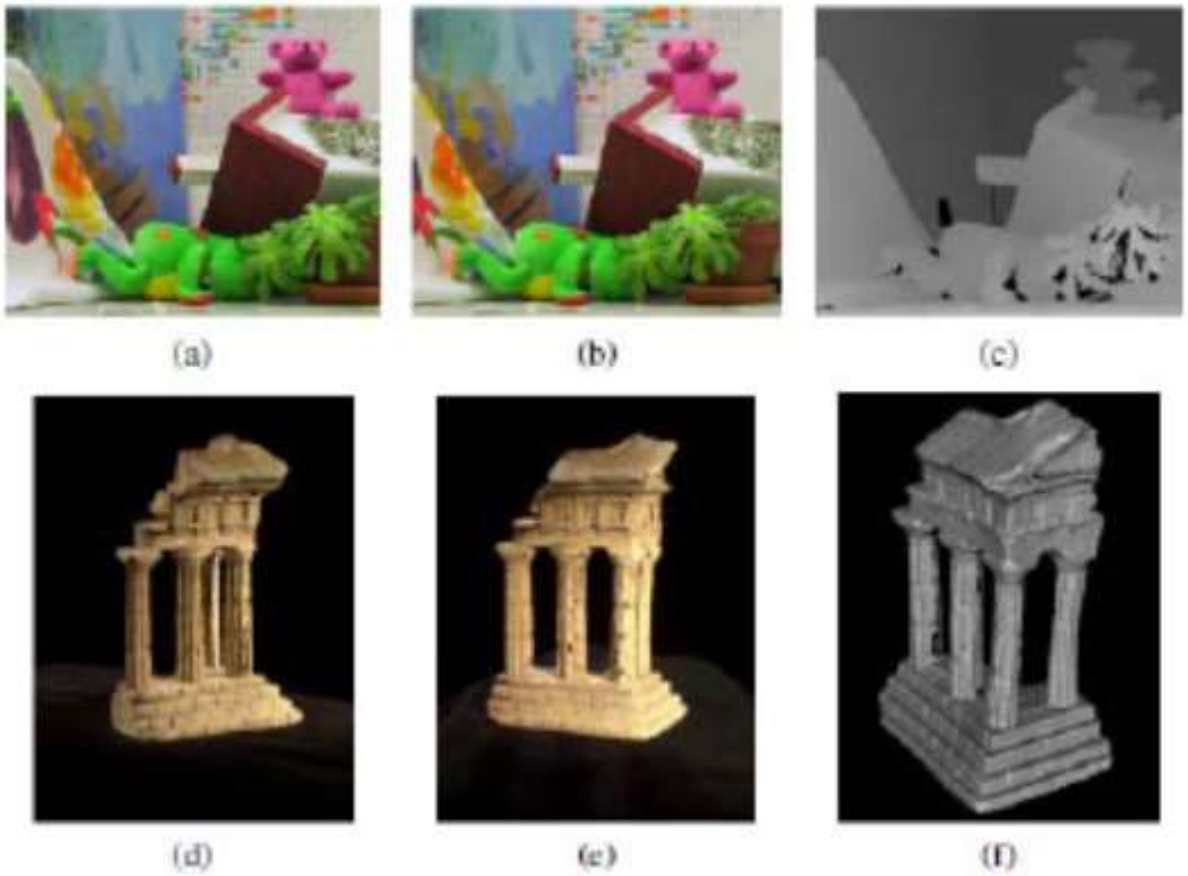
Figure 6.1: Stereo geometry.

$$\frac{Z+f}{Z} = \frac{x_1 + x_2 + B}{B}, \qquad Z = \frac{fB}{x_1 + x_2},$$

B=Baseline
f=focal length
C1 and C2=Camera Centers
X1, X2=Image location in left and right cameras

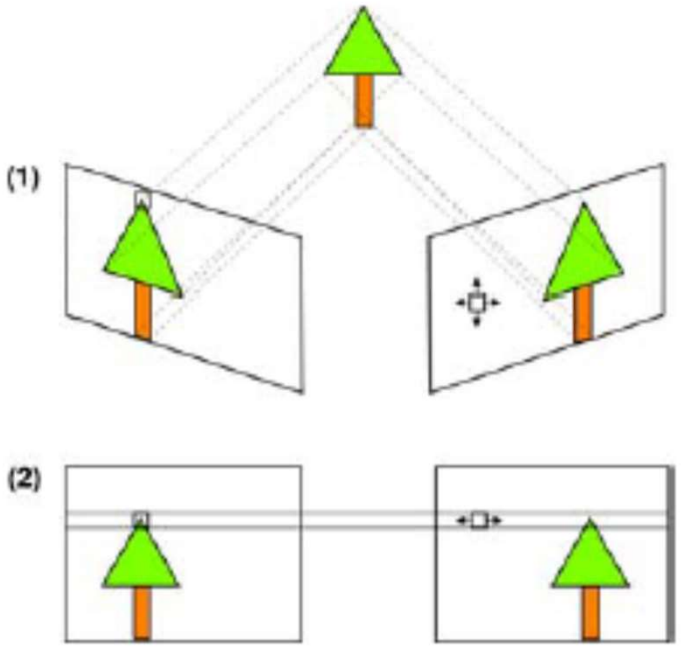
$x_1 + x_2 = \text{disparity} = d$

Stereo pairs and depth maps

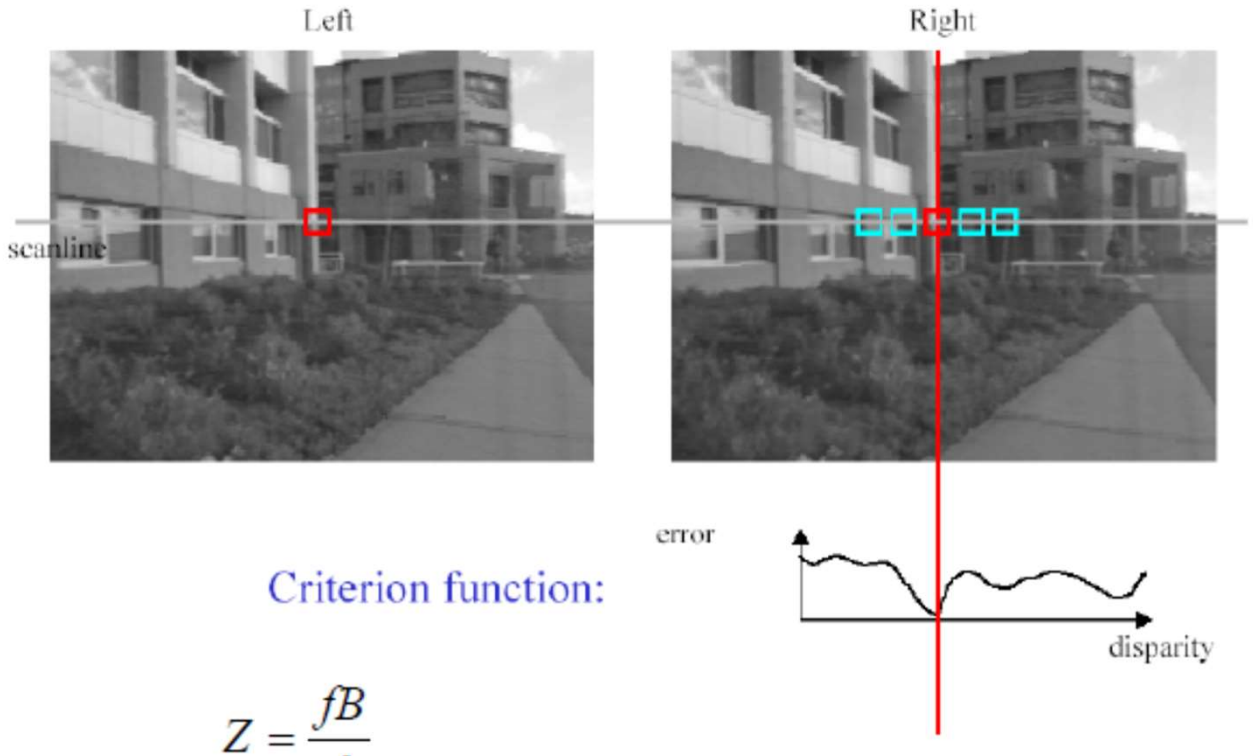


Courtesy: Book by Szeliski

Rectification



Correspondence using search



Criterion function:

$$Z = \frac{fB}{d}$$

Correlation based stereo methods

- Disparity map can be constructed based on a correlation measure

$$SSD = \sum\sum \left(I_{left} - I_{right}\right)^2$$ Sum of squares difference

$$AD = \sum\sum \left|\left(I_{left} - I_{right}\right)\right|$$ Absolute difference

$$CC = \sum\sum I_{left} I_{right}$$ Cross correlation

$$NC = \frac{\sum\sum \left(I_{left} \cdot I_{right}\right)}{\sqrt{\sum\sum I_{left} \cdot I_{right}}}$$

Normalized Correlation

$$MC = \frac{1}{64\sigma_{left}\sigma_{right}} \sum\sum \left(I_{left} - \mu_{left1}\right)\left(I_{right} - \mu_{right}\right)$$

Mutual Correlation

Mean

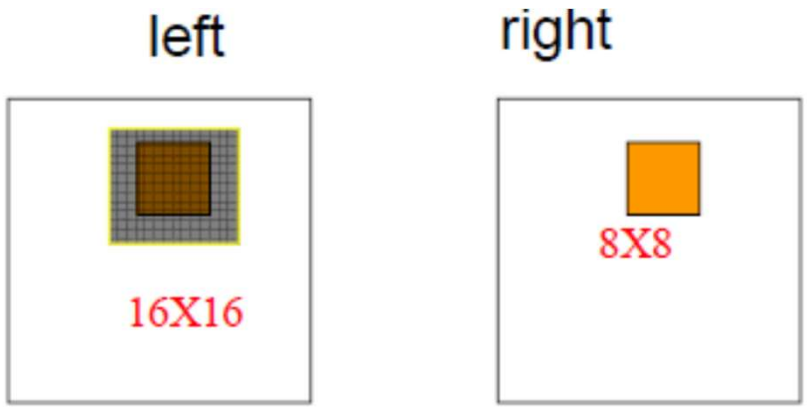
Standard Deviation

Alper Yilmaz, Mubarak shah, Fall 2011 UCF

Correlation

- Similarity/Dissimilarity Measures
  - Sum of Squares Difference (SSD)
  - Normalized Correlation
  - Mutual Correlation
  - Mutual information $I(x,y) = \sum\sum p(x,y) \log \frac{p(x,y)}{p_1(x)p_2(y)}$
- Use
  - Gray levels
  - Laplacian of Gaussian
  - Gradient magnitude

Block Matching



- Can be used for
  - Computing MPEG motion vectors
  - Optical flow
  - Stereo (displacement limited to only x-axis)
  - Image matching

Block Matching

- For each 8X8 block, centered around pixel (x,y) in right image, B_k
  - Obtain 16X16 block in left, centered around (x,y), B_{k-1}
  - Compute Sum of Squares Differences (SSD) between 8X8 block, B_k, and all possible 8X8 blocks in B_{k-1}
  - The 8X8 block in B_{k-1} centered around (x',y'), which gives the least SSD is the match
  - The displacement vector (disparity, optical flow) is given by $u=x-x'$; $v=y-y'$

Sum of square difference

$$(u(x,y),v(x,y)) = \arg\min_{u,v=-4\ldots4} \sum_{i=0}^{-7}\sum_{j=0}^{-7}\left(f_k(x+i,y+j) - f_{k-1}(x+i+u,y+j+v)\right)^2$$

Minimum absolute difference

$$(u(x,y),v(x,y)) = \arg\min_{u,v=-4\ldots4} \sum_{i=0}^{-7}\sum_{j=0}^{-7} \left| \left(f_k(x+i,y+j) - f_{k-1}(x+i+u,y+j+v)\right) \right|$$

Maximum matching pixel counts

$$T(x,y;u,v)=\left\{\begin{array}{cc} 1 & \text{if}\left|f_k(x,y)-f_{k-1}(x+u,y+v)\right|\le t \\ 0 & \text{otherwise} \end{array}\right\}$$

$$(u(x,y),v(x,y))=\arg\max_{u,v=-4\ldots4}\sum_{i=0}^{-7}\sum_{j=0}^{-7}T(x+i,y+j;u,v)$$

Cross Correlation

$$(u(x,y),v(x,y)) = \arg\max_{u,v=-4\ldots4} \sum_{i=0}^{-7} \sum_{j=0}^{-7} \left(f_k(x+i,y+j).f_{k-1}(x+i+u,y+j+v) \right)$$

Normalized Correlation

$$(u,v) = \arg\max_{u,v=-4\ldots4} \frac{\sum_{i=0}^{-7}\sum_{j=0}^{-7}\left((f_k(x+i,y+j)-\mu_1).(f_{k-1}(x+i+u,y+j+v)-\mu_2)\right)}{\sqrt{\left(\sum_{i=0}^{-7}\sum_{j=0}^{-7}(f_k(x+i,y+j)-\mu_1)^2\right)\left(\sum_{i=0}^{-7}\sum_{j=0}^{-7}(f_{k-1}(x+i+u,y+j+v)-\mu_2)^2\right)}}$$

and μ_2 are the means of patch-1 and patch-2 respectively.

Mutual Correlation

$$(u(x,y),v(x,y)) = \arg\max_{u,v=-4\ldots4} \frac{1}{64\sigma_1\sigma_2} \sum_{i=0}^{-7}\sum_{j=0}^{-7} \left(f_k(x+i,y+j)-\mu_1\right).f_{k-1}(x+i+u,y+j+v)-\mu_2\right)$$

Sigma and mu are standard deviation and mean of patch-1 and patch-2 respectively

Barnard's Stereo Method

- Similar intensity
  - Similar to brightness constraint
- Smoothness of disparity

$$E = \sum_{i=-1}^{1}\sum_{j=-1}^{1}\left\| I_{left}(x+i, y+j) - I_{right}(x+i+D_x(x,y), y+j)\right\| + \lambda\left\|\nabla D(x,y)\right\|$$

$$\nabla D(x,y) = \sum_{i=-1}^{1}\sum_{j=-1}^{1}\left|D(x+i, y+j) - D(x,y)\right|$$

Barnard's Stereo Method

- Energy can be minimized using brute force search
  - Let max allowed disparity is 10 pixels
  - For 128x128 image for 10 possible levels of disparity
    - There 10^{16384} possible disparity values
  - We can select any minimization technique
  - Barnard choose simulated annealing

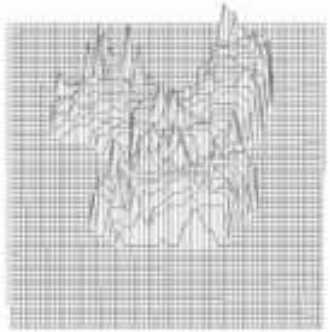
Simulated Annealing

- Select a random state S (disparities)
- Select a high temperature
  - Select random S'
  - Compute $\Delta E=E(S')-E(S)$
  - If $(\Delta E<0)$ $S \leftarrow S'$
  - Else
    - $P \leftarrow \exp(-\Delta E/T)$
    - $X \leftarrow$ random(0,1)
      - If $X<P$ then $S \leftarrow S'$
    - If no decrease in several iterations lower T

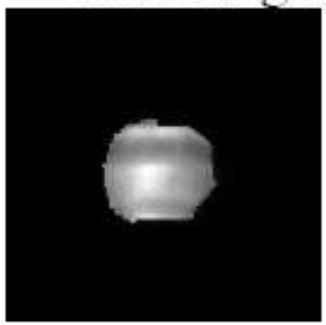
Example

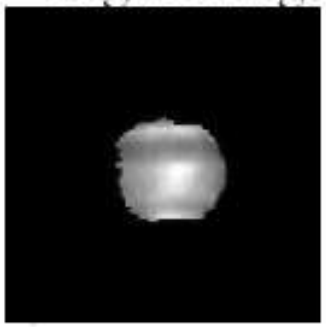
Left Image

Right Image



Depth Map





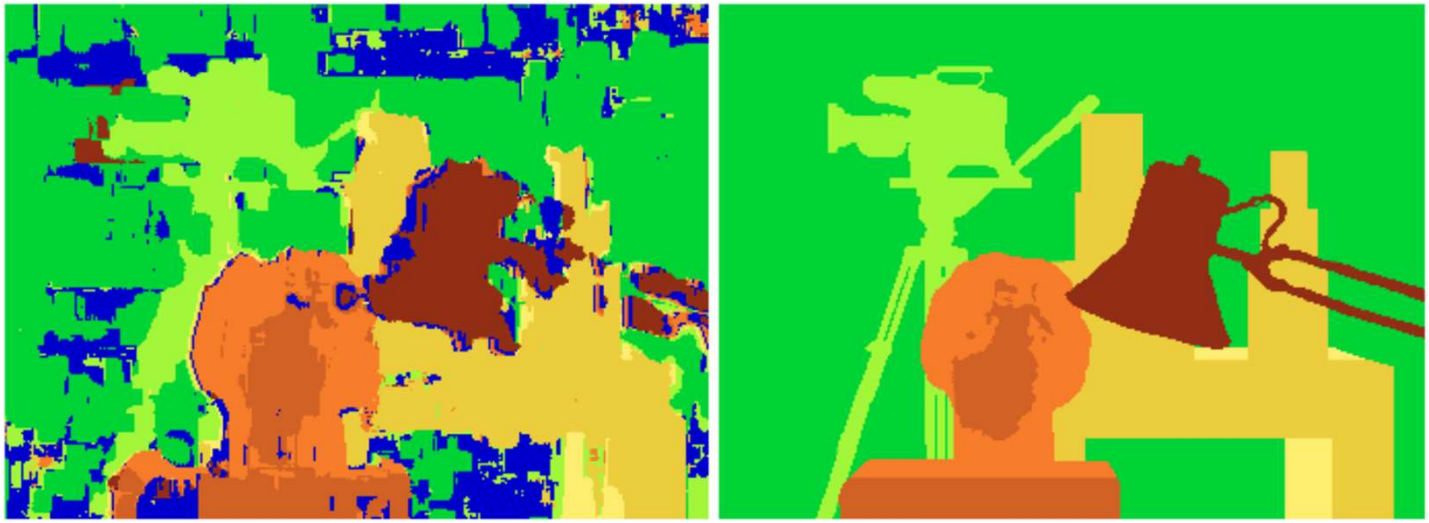
Stereo Results

– Data from University of Tsukuba

Scene

Ground truth

Results with window correlation



Window-based matching
(best window size)

Ground truth

Results with better method

State of the art method

Ground truth

Boykov et al., Fast Approximate Energy Minimization via Graph Cuts, International Conference on Computer Vision, September 1999.

To continue…