6.891

Computer Vision and Applications

Prof. Trevor. Darrell

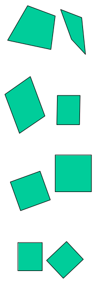
Lecture 10: Projective SFM
- Projective spaces
- Cross ratio
- Factorization algorithm
- Euclidean upgrade

Readings: F&P 13.0, 13.1, 13.4, 13.5

1

Transformations

Projective 8dof	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$		Concurrency, collinearity, order of contact (intersection, tangency, inflection, etc.), cross ratio
Affine 6dof	$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Parallellism, ratio of areas, ratio of lengths on parallel lines (e.g midpoints), linear combinations of vectors (centroids).
Similarity 4dof	$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Ratios of lengths, angles.
Euclidean 3dof	$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Lengths, areas.



2

[www.cs.unc.edu/~marc/mvg]

Last Time

Affine SFM
- Geometric Approach
- Algebraic Approach
- Tomasi/Kanade Factorization

3

Parallel projection ⟶ "Affine camera"

"Affine geometry is, roughly speaking, what is left after all ability to measure lengths, areas, angles, etc. has been removed from Euclidean geometry. The concept of parallelism remains, however, as well as the ability to measure the ratio of distances between collinear points."
[Snapper and Troyer, 1989]

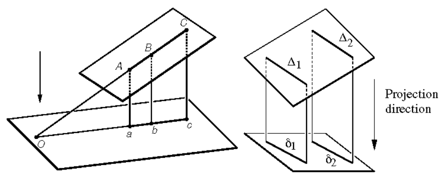


FIGURE 13.2: Parallel projection preserves: (left) the ratio of signed distances between collinear points and (right) the parallelism of lines.

[F&P]

Affine Coordinates

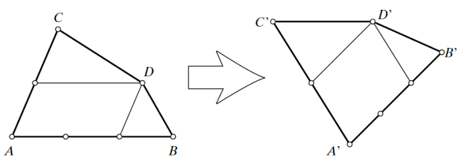
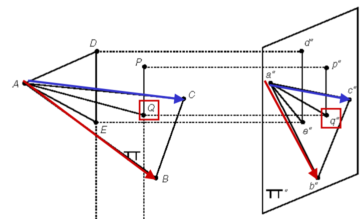
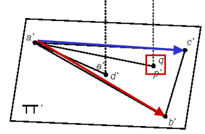


Figure 12-1

An affine transformation of the plane. The points *A*, *B*, *C*, and *D* are transformed into the points *A'*, *B'*, *C'*, and *D'*. The affine coordinates of *D* in the basis of the plane formed by *A*, *B*, and *C* are the same as those of *D'* in the basis formed by *A'*, *B'*, and *C'*—namely 2/3 and 1/2.

[F&P]

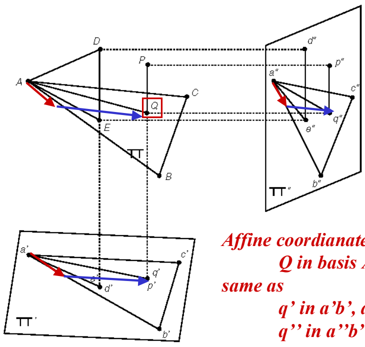




Can directly measure affine coordinate in plane

6

[F&P]



Affine coordianate of Q in basis AB,AC same as q' in a'b', a'c' q'' in a''b'', a''c''

[F&P]

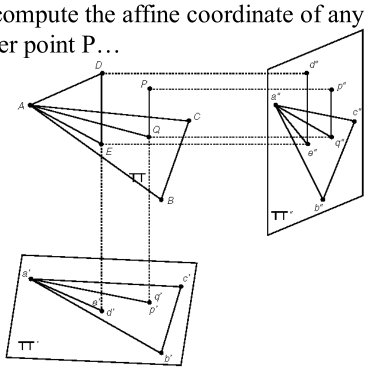
Affine Structure from Motion Theorem

Two affine views of four non co-planar points are sufficient to compute the affine coordinate of any other point P.

[Koenderink and Van Doorn, 1990]

8

Two affine views of four points are sufficient to compute the affine coordinate of any other point P…

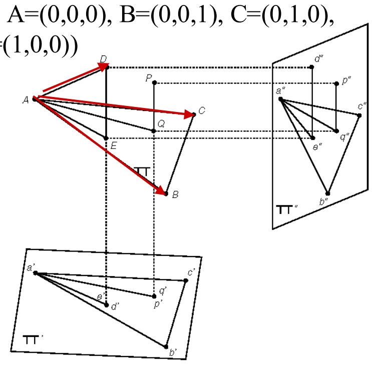


9

[F&P]

Given Affine Basis (A,B,C,D)

(e.g., A=(0,0,0), B=(0,0,1), C=(0,1,0), D=(1,0,0))



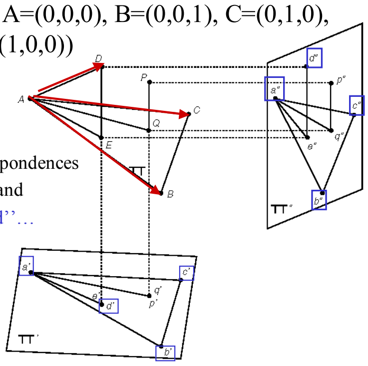
10

[F&P]

Given Affine Basis (A,B,C,D)

(e.g., A=(0,0,0), B=(0,0,1), C=(0,1,0), D=(1,0,0))

And correspondences a',b',c',d' and a'',b'',c'',d''…



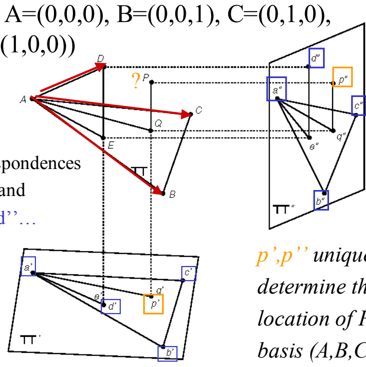
11

[F&P]

Given Affine Basis (A,B,C,D)

(e.g., A=(0,0,0), B=(0,0,1), C=(0,1,0), D=(1,0,0))

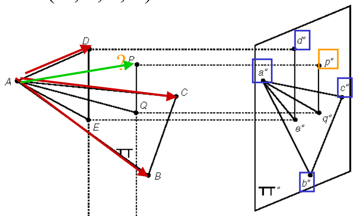
And correspondences a',b',c',d' and a'',b'',c'',d''…

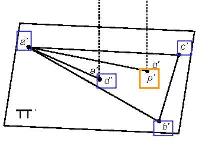


p',p'' uniquely determine the location of P in the basis (A,B,C,D)! 12

[F&P]

p',p'' uniquely determine the location of P in the basis (A,B,C,D)…





P=αAB+βAC+λAD

Find α,β,λ ?

13

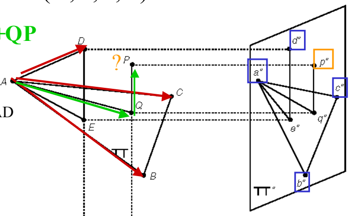
[F&P]

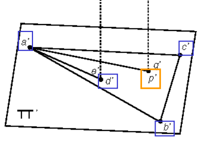
p',p'' uniquely determine the location of P in the basis (A,B,C,D)…

AP=AQ+QP

P=αAB+βAC+λAD

Find α,β,λ ?





$$\overrightarrow{AP} = \overrightarrow{AQ} + \overrightarrow{QP}$$

14

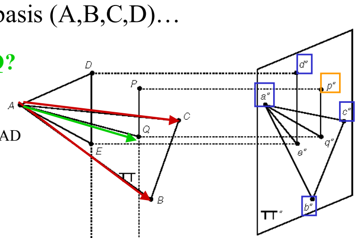
[F&P]

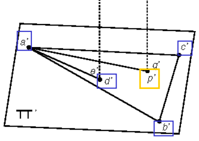
p',p'' uniquely determine the location of P in the basis (A,B,C,D)…

AQ?

P=αAB+βAC+λAD

Find α,β,λ ?





Find E and Q using basis A,B,C

15

[F&P]

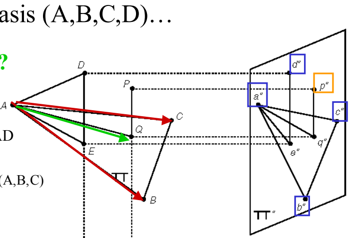
p',p'' uniquely determine the location of P in the basis (A,B,C,D)…

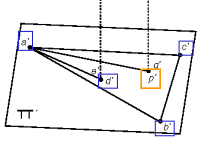
AQ?

P=αAB+βAC+λAD

Find α,β,λ ?

Find E and Q using (A,B,C)





E,Q = e',q' = d',p' =

$$(\alpha_{d'}, \beta_{d'})$$

$$(\alpha_{p'}, \beta_{p'})$$

[F&P]

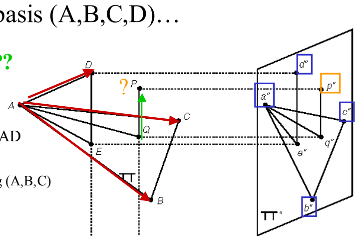
p',p'' uniquely determine the location of P in the basis (A,B,C,D)…

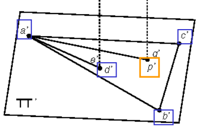
QP?

P=αAB+βAC+λAD

Find α,β,λ ?

Find E and Q using (A,B,C)





17

[F&P]

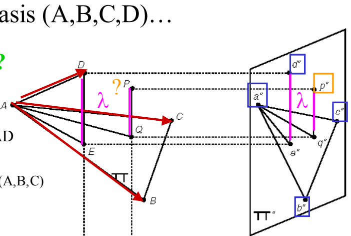
p',p'' uniquely determine the location of P in the basis (A,B,C,D)…

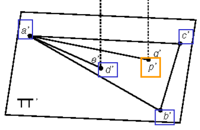
QP?

P=αAB+βAC+λAD

Find α,β,λ ?

Find E and Q using (A,B,C)





compute

λ=QP/ED=q''p''/e''d''

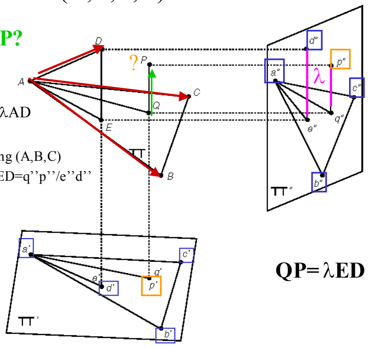
18

[F&P]

p’,p’’ uniquely determine the location of P in the basis (A,B,C,D)…

QP?

P=αAB+βAC+λAD
Find α,β,λ ?
Find E and Q using (A,B,C)
Compute λ=QP/ED=q’’p’’/e’’d’’



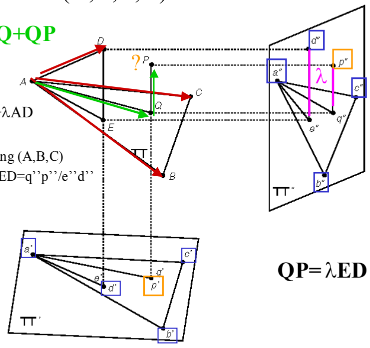
QP=λED

[F&P]

p’,p’’ uniquely determine the location of P in the basis (A,B,C,D)…

AP=AQ+QP

P=αAB+βAC+λAD
Find α,β,λ ?
Find E and Q using (A,B,C)
Compute λ=QP/ED=q’’p’’/e’’d’’

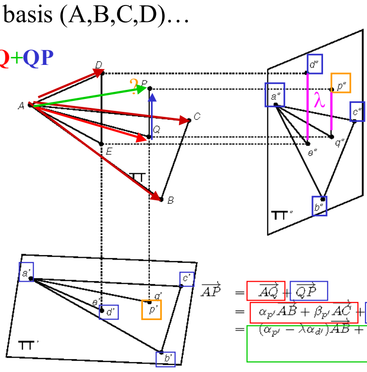


QP=λED

[F&P]

p’,p’’ uniquely determine the location of P in the basis (A,B,C,D)…

AP=AQ+QP



$$\begin{aligned}\overrightarrow{AP} &= \overrightarrow{AQ} + \overrightarrow{QP}\\ &= \alpha_{p'}\overrightarrow{AB} + \beta_{p'}\overrightarrow{AC} + \lambda\overrightarrow{ED}\\ &= (\alpha_{p'} - \lambda\alpha_{d'})\overrightarrow{AB} + (\beta_{p'} - \lambda\beta_{d'})\overrightarrow{AC}\\ &\quad + \lambda\overrightarrow{AD}.\end{aligned}$$

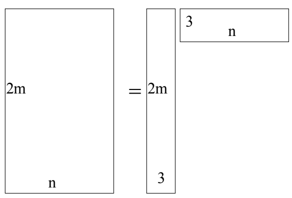
[F&P]

Review: Affine case

D = A P

Data-Matrix = Affine-Motions x 3-d-Points

(2m x n) = (2m x 3) x (3 x n)



D is rank 3 in affine case

Review: Factorization algorithm

Given a data matrix,

find Motion (A) and Shape (P) matrices that generate that data…

Tomasi and Kanade Factorization algorithm (1992):

Use Singular Value Decomposition to factor D into appropriately sized A and P.

Review: SVD

Technique: Singular Value Decomposition Let $\mathcal{A}$ be an $m \times n$ matrix, with $m \geq n$, then $\mathcal{A}$ can always be written as

$$\mathcal{A} = \mathcal{U}\mathcal{W}\mathcal{V}^T,$$

where:

- $\mathcal{U}$ is an $m \times n$ column-orthogonal matrix, i.e., $\mathcal{U}^T\mathcal{U} = \mathrm{Id}_m$,
- $\mathcal{W}$ is a diagonal matrix whose diagonal entries w_i $(i = 1, \ldots, n)$ are the singular values of $\mathcal{A}$ with $w_1 \geq w_2 \geq \ldots \geq w_n \geq 0$,
- and $\mathcal{V}$ is an $n \times n$ orthogonal matrix, i.e., $\mathcal{V}^T\mathcal{V} = \mathcal{V}\mathcal{V}^T = \mathrm{Id}_n$.

The SVD of a matrix can also be used to characterize matrices that are rank-deficient: suppose that $\mathcal{A}$ has rank $p < n$, then the matrices $\mathcal{U}$, $\mathcal{W}$, and $\mathcal{V}$ can be written as

$$\mathcal{U} = \begin{bmatrix}\mathcal{U}_p & \mathcal{U}_{n-p}\end{bmatrix} \quad \mathcal{W} = \begin{bmatrix}\mathcal{W}_p & 0\\ 0 & 0\end{bmatrix} \quad \text{and} \quad \mathcal{V}^T = \begin{bmatrix}\mathcal{V}_p^T\\ \mathcal{V}_{n-p}^T\end{bmatrix},$$

[F&P]

Review: Affine Factorization algorithm

1. Compute the singular value decomposition $\mathcal{D} = \mathcal{U}\mathcal{W}\mathcal{V}^T$.
2. Construct the matrices $\mathcal{U}_3$, $\mathcal{V}_3$, and $\mathcal{W}_3$ formed by the three leftmost columns of the matrices $\mathcal{U}$ and $\mathcal{V}$, and the corresponding 3×3 sub-matrix of $\mathcal{W}$.
3. Define

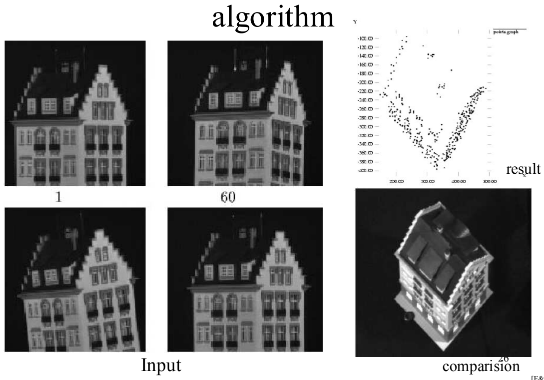
$$\mathcal{A}_0 = \mathcal{U}_3 \quad \text{and} \quad \mathcal{P}_0 = \mathcal{W}_3\mathcal{V}_3^T;$$

the $2m \times 3$ matrix $\mathcal{A}_0$ is an estimate of the camera motion, and the $3 \times n$ matrix $\mathcal{P}_0$ is an estimate of the scene structure.

25

[F&P]

Review: Affine Factorization algorithm



26

[F&P]

Today

Projective SFM
- Projective spaces
- Cross ratio
- Factorization algorithm
- Euclidean upgrade

27

Projective transformations

Definition:

A *projectivity* is an invertible mapping h from P^2 to itself such that three points x_1,x_2,x_3 lie on the same line if and only if $h(x_1),h(x_2),h(x_3)$ do.

Theorem:

A mapping $h:P^2 \rightarrow P^2$ is a projectivity if and only if there exist a non-singular 3x3 matrix $\mathbf{H}$ such that for any point in P^2 reprented by a vector x it is true that $h(x)=\mathbf{H}x$

Definition: Projective transformation

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad \text{or} \quad x' = \mathbf{H}\,x$$

8DOF

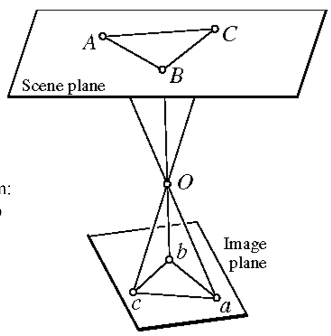
projectivity=collineation=projective transformation=homography

28

[www.cs.unc.edu/~marc/mvg]

Homography

Projective transform:
bijective linear map
a.k.a. *Homography*



[F&P]

Review: Perspective Projection

$$\boldsymbol{p} = \frac{1}{z}\mathcal{M}\boldsymbol{P}, \quad \text{where} \quad \mathcal{M} = \mathcal{K}(\mathcal{R} \quad \boldsymbol{t})$$

or

$$u = \frac{\boldsymbol{m}_1 \cdot \boldsymbol{P}}{\boldsymbol{m}_3 \cdot \boldsymbol{P}}$$
$$v = \frac{\boldsymbol{m}_2 \cdot \boldsymbol{P}}{\boldsymbol{m}_3 \cdot \boldsymbol{P}}$$

where $\boldsymbol{m}_{i1}^T$, $\boldsymbol{m}_{i2}^T$ and $\boldsymbol{m}_{i3}^T$ denote the rows of the 3×4 projection matrix M

30

(Projective) SFM

Goal: Estimate M and P from (u_{ij},v_{ij})…

$$u_{ij} = \frac{\boldsymbol{m}_{i1} \cdot \boldsymbol{P}_j}{\boldsymbol{m}_{i3} \cdot \boldsymbol{P}_j} \qquad v_{ij} = \frac{\boldsymbol{m}_{i2} \cdot \boldsymbol{P}_j}{\boldsymbol{m}_{i3} \cdot \boldsymbol{P}_j} \quad \text{for} \quad i = 1, \ldots, m \quad \text{and} \quad j = 1, \ldots, n,$$

31

Projective Ambiguity

if $\boldsymbol{P}_j$ and $\mathcal{M}_i$ are solutions to the SFM equations, then so are

$$\mathcal{M}_i' = \mathcal{M}_i \mathcal{Q}$$

$$\boldsymbol{P}_j' = \mathcal{Q}^{-1} \boldsymbol{P}_j$$

where $\mathcal{Q}$ is a projective transformation matrix (arbitrary nonsingular 4x4 matrix, defined up to scale)

32

Projective Geometry

The means of measurement available in projective geometry are even more primitive than those available in affine geometry

- no notions of lengths, areas and angles (Euclidean)
- no notions of ratios of lengths along parallel lines (Affine)
- no notion of parallelism (Affine)

The concepts of points, lines and planes remain (and incidence).

And a weaker scalar measure of the arrangement of collinear points, the cross-ratio…

33

The Cross-ratio

The non-homogeneous projective coordinates of a point can be defined geometrically in terms of cross-ratios.

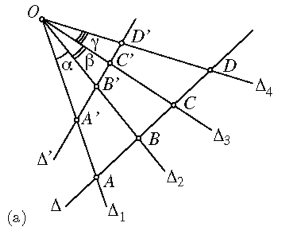
Given four collinear points A,B, C,D such that A, B and C are distinct, we define the cross-ratio of these points as:

$$\{A, B; C, D\} \stackrel{\text{def}}{=} \frac{\overline{CA}}{\overline{CB}} \times \frac{\overline{DB}}{\overline{DA}}$$

34

$$\{A, B; C, D\} \stackrel{\text{def}}{=} \frac{\overline{CA}}{\overline{CB}} \times \frac{\overline{DB}}{\overline{DA}}$$

The value of this cross ratio is independent of the intersecting line or plane:

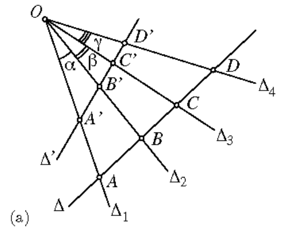


35

[F&P]

$$\{A, B; C, D\} \stackrel{\text{def}}{=} \frac{\overline{CA}}{\overline{CB}} \times \frac{\overline{DB}}{\overline{DA}}$$

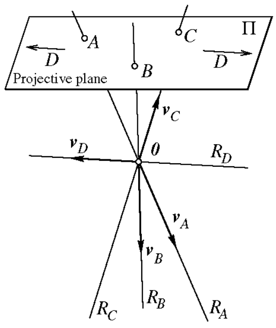
The value of this cross ratio is independent of the intersecting line or plane:



36

[F&P]

Projective Plane



an affine plane Π of $\mathbb{R}^3$

- Rays R_A, R_B and R_C associated with the vectors v_A, v_B and v_C can be mapped onto the points A,B,C
- The vectors v_A, v_B and v_C are linearly independent, and thus so are the points A,B,C
- As a ray becomes close to parallel to Π the point where it intersects Π moves to infinity
- Projective plane can be modeled by adding set of *points at infinity* to 2-D Π

37

[F&P]

Projective Spaces: (Semi-Formal) Definition

$\vec{X}$	a vector space of dimension $n+1$
$\mathbb{R}\boldsymbol{v}$	the ray $\{k\boldsymbol{v},\ k \in \mathbb{R}\}$, where $\boldsymbol{v} \in \vec{X}$
$X = P(\vec{X})$	the set of rays $\{\mathbb{R}\boldsymbol{v},\ \boldsymbol{v} \in \vec{X}\backslash\mathbf{0}\}$

→ the projective space of dimension n associated with $\vec{X}$

→ the set of points $\{p(\boldsymbol{v}) = \mathbb{R}\boldsymbol{v},\ \boldsymbol{v} \in \vec{X}\backslash\mathbf{0}\}$

38

[Ponce]

Projective SFM approach

Ignoring at first the Euclidean constraints associated with calibrated cameras will linearize the recovery of scene structure and camera motion from point correspondences

Decompose motion analysis into two stages

1. recovery of the projective shape of the scene and the estimation of the corresponding projection matrices.
2. exploit the geometric constraints associated with (partially or fully) calibrated perspective cameras to upgrade the projective reconstruction to a Euclidean one.

39

Projective SFM approach

$x_i \leftrightarrow x'_i$

Original scene X_i

Projective, affine, similarity reconstruction
= reconstruction that is identical to original up to projective, affine, similarity transformation

Literature: Metric and Euclidean reconstruction
= similarity reconstruction

40

[www.cs.unc.edu/~marc/mvg]

3D reconstruction of cameras and structure

<u>reconstruction problem:</u>

given $x_i \leftrightarrow x'_i$, compute P,P' and X_i

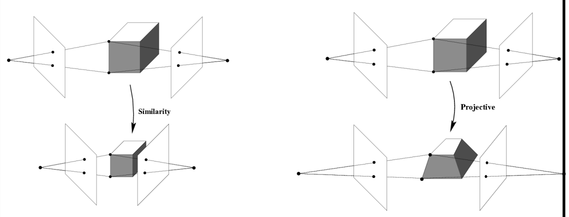
$$x_i = PX_i \quad x'_i = PX'_i \quad \text{for all } i$$

without additional information
possible up to projective ambiguity

41

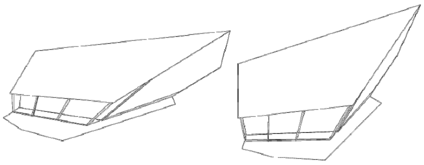
[www.cs.unc.edu/~marc/mvg]

Reconstruction ambiguities



42

[www.cs.unc.edu/~marc/mvg]



43

[www.cs.unc.edu/~marc/mvg]

Two-frame reconstruction

(i) Compute F from correspondences
(ii) Compute camera matrices from F
(iii) Compute 3D point for each pair of corresponding points

computation of F
use $x'_i F x_i = 0$ equations, linear in coeff. F
8 points (linear), 7 points (non-linear), 8+ (least-squares)
(more on this next class)

computation of camera matrices
Possible choice:

$$P = [I \mid 0] \quad P' = [[e']_{\times} F \mid e']$$

triangulation
compute intersection of two backprojected rays

44

[www.cs.unc.edu/~marc/mvg]

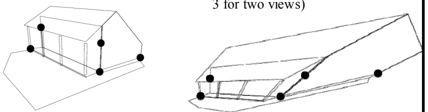
Direct reconstruction using ground truth

use control points X_{Ei} with known coordinates to go from projective to metric

$$X_{Ei} = HX_i$$

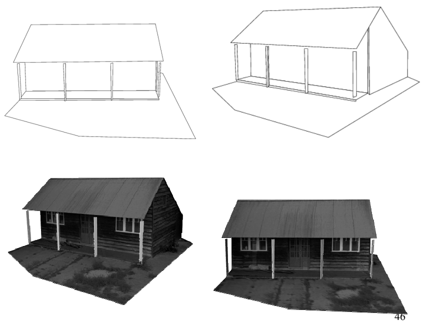
$$x_i = PH^{-1}X_{Ei}$$

(2 lin. eq. in H^{-1} per view, 3 for two views)



45

[www.cs.unc.edu/~marc/mvg]



46

[www.cs.unc.edu/~marc/mvg]

Factorization approach to Projective SFM

Use multiple frame sequence….
Generalize Tomasi-Kanade to the projective case…

47

Perspective factorization

The camera equations

$$\lambda_{ij} m_{ij} = \mathbf{P}_i M_j, i = 1,...,m, j = 1,...,m$$

for a fixed image *i* can be written in matrix form as

$$\mathbf{m}_i \Lambda_i = \mathbf{P}_i \mathbf{M}$$

where

$$\mathbf{m}_i = [m_{i1}, m_{i2},..., m_{im}], \ \mathbf{M} = [M_1, M_2,..., M_m]$$
$$\Lambda_i = \mathrm{diag}(\lambda_{i1}, \lambda_{i2},..., \lambda_{im})$$

48

[www.cs.unc.edu/~marc/mvg]

Perspective factorization

All equations can be collected for all i as

$$\mathbf{m} = \mathbf{PM}$$

where

$$\mathbf{m} = \begin{bmatrix} \mathbf{m}_1\Lambda_1 \\ \mathbf{m}_2\Lambda_2 \\ \dots \\ \mathbf{m}_n\Lambda_n \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \dots \\ \mathbf{P}_m \end{bmatrix}$$

In these formulas m are known, but Λ_i, **P** and **M** are unknown

Observe that **PM** is a product of a $3m$x4 matrix and a 4xn matrix, i.e. it is a rank 4 matrix

49

[www.cs.unc.edu/~marc/mvg]

Perspective factorization algorithm

Assume that L_i are known, then **PM** is known.

Use the singular value decomposition

PM=US V^T

In the noise-free case

$S=diag(s_1,s_2,s_3,s_4,0, \dots ,0)$

and a reconstruction can be obtained by setting:

P=the first four columns of US.
M=the first four rows of V.

50

[www.cs.unc.edu/~marc/mvg]

Iterative perspective factorization

When L_i are unknown the following algorithm can be used:

1. Set l_{ij}=1 (affine approximation).

2. Factorize **PM** and obtain an estimate of **P** and **M**. If s_5 is sufficiently small then STOP.

3. Use **m**, **P** and **M** to estimate L_i from the camera equations (linearly) $\mathbf{m}_i L_i = \mathbf{P}_i\mathbf{M}$

4. Goto 2.

In general the algorithm minimizes the *proximity measure* $P(L,\mathbf{P},\mathbf{M})=s_5$

Structure and motion recovered up to an arbitrary projective transformation

51

[www.cs.unc.edu/~marc/mvg]

Bundle adjustment

Given initial estimates for the matrices Mi (i = 1, . . .,m) and vectors Pj (j = 1, . . ., n), we can refine these estimates by using non-linear least squares to minimize the global error measure

$$E = \frac{1}{mn}\sum_{i,j}\left[\left(u_{ij} - \frac{\boldsymbol{m}_{i1}\cdot\boldsymbol{P}_j}{\boldsymbol{m}_{i3}\cdot\boldsymbol{P}_j}\right)^2 + \left(v_{ij} - \frac{\boldsymbol{m}_{i2}\cdot\boldsymbol{P}_j}{\boldsymbol{m}_{i3}\cdot\boldsymbol{P}_j}\right)^2\right].$$

52

[F&P]

Euclidean upgrade

Given a camera with known intrinsic parameters, we can take the calibration matrix to be the identity and write the perspective projection equation in some Euclidean world coordinate system as

$$\boldsymbol{p} = \frac{1}{z}\begin{pmatrix}\mathcal{R} & \boldsymbol{t}\end{pmatrix}\begin{pmatrix}\boldsymbol{P}\\1\end{pmatrix} = \frac{1}{\lambda z}\begin{pmatrix}\mathcal{R} & \lambda\boldsymbol{t}\end{pmatrix}\begin{pmatrix}\lambda\boldsymbol{P}\\1\end{pmatrix}$$

for any non-zero scale factor λ. If $\mathcal{M}_i$ and $\mathbf{P}_j$ denote the shape and motion parameters measured in some Euclidean coordinate system, there must exist a 4 ×4 matrix Q such that

$$\hat{\mathcal{M}}_i = \mathcal{M}_i\mathcal{Q} \text{ and } \hat{\boldsymbol{P}}_j = \mathcal{Q}^{-1}\boldsymbol{P}_j.$$

53

[F&P]

Euclidean upgrade

$$\hat{\mathcal{M}}_i = \rho_i\mathcal{K}_i\begin{pmatrix}\mathcal{R}_i & \boldsymbol{t}_i\end{pmatrix},$$

where ρ_i accounts for the unknown scale of $\mathcal{M}_i$, and $\mathcal{K}_i$ is a calibration matrix

$$\mathcal{M}_i\mathcal{Q}_3 = \rho_i\mathcal{K}_i\mathcal{R}_i.$$

the 3×3 matrices M_iQ_3 are in this case scaled rotation matrices.

$$\begin{aligned}
&\boldsymbol{m}_{i1}^T\mathcal{Q}_3\mathcal{Q}_3^T\boldsymbol{m}_{i2} = 0,\\
&\boldsymbol{m}_{i2}^T\mathcal{Q}_3\mathcal{Q}_3^T\boldsymbol{m}_{i3} = 0,\\
&\boldsymbol{m}_{i3}^T\mathcal{Q}_3\mathcal{Q}_3^T\boldsymbol{m}_{i1} = 0,\\
&\boldsymbol{m}_{i1}^T\mathcal{Q}_3\mathcal{Q}_3^T\boldsymbol{m}_{i1} - \boldsymbol{m}_{i2}^T\mathcal{Q}_3\mathcal{Q}_3^T\boldsymbol{m}_{i2} = 0,\\
&\boldsymbol{m}_{i2}^T\mathcal{Q}_3\mathcal{Q}_3^T\boldsymbol{m}_{i2} - \boldsymbol{m}_{i3}^T\mathcal{Q}_3\mathcal{Q}_3^T\boldsymbol{m}_{i3} = 0.
\end{aligned}$$

54

[F&P]

Euclidean upgrade

FIGURE 14.6: A synthetic texture-mapped image of a castle constructed via projective motion analysis followed by a Euclidean upgrade. The principal point is assumed to be known. Reprinted from [Pollefeys, 1999], Figure 6.13.

55

[F&P]

Further Factorization work

Factorization with uncertainty

(Irani & Anandan, IJCV'02)

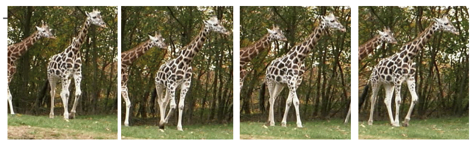
Factorization for dynamic scenes

(Costeira and Kanade '94)

(Bregler et al. 2000, Brand 2001)

56

3D Non-rigid Structure from Motion



Chris Bregler
Gene Alexander, Henning Biermann, Aaron Hertzmann, Lorenzo Torresani, Danny Yang

57

[www.cs.berkeley.edu/projects/vision/workshop01/ presentations/cbregler/cbregler.ppt]

3D Non-Rigid Structure from Motion

- **We want 3 things:**
  - **3D non-rigid shape model**
  - **for each frame:**
    - **3D Pose**
    - **non-rigid configuration (deformation)**

58

[www.cs.berkeley.edu/projects/vision/workshop01/ presentations/cbregler/cbregler.ppt]

Solution based on Factorization

- **We want 3 things:**
  - **3D non-rigid shape model**
  - **for each frame:**
    - **3D Pose**
    - **non-rigid configuration (deformation)**

-> **Tomasi-Kanade-92:**

Rank 3

$$W = P\,S$$

59

[www.cs.berkeley.edu/projects/vision/workshop01/ presentations/cbregler/cbregler.ppt]

Solution based on Factorization

- **We want 3 things:**
  - **3D non-rigid shape model**
  - **for each frame:**
    - **3D Pose**
    - **non-rigid configuration (deformation)**

-> **PCA-based representations:**

Rank K

$$W = P_{non\text{-}rigid}\,S$$

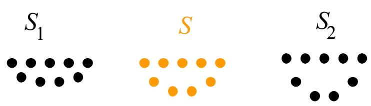
60

[www.cs.berkeley.edu/projects/vision/workshop01/ presentations/cbregler/cbregler.ppt]

3D Shape Model

Linear Interpolation between 3D Key-Shapes:

$$S = \sum_{i=1}^{K} l_i \cdot S_i \qquad S, S_i \in \mathbb{R}^{3\times P}, l_i \in \mathbb{R} \qquad (1)$$



[www.cs.berkeley.edu/projects/vision/workshop01/ presentations/cbregler/cbregler.ppt]

Basis Shape Factorization

$$\begin{bmatrix} u_1 & u_2 & \dots & u_P \\ v_1 & v_2 & \dots & v_P \end{bmatrix} = R \cdot \left(\sum_{i=1}^{K} l_i \cdot S_i \right) + T \qquad (2)$$

$$\downarrow$$

$$\begin{bmatrix} u_1 & \dots & u_P \\ v_1 & \dots & v_P \end{bmatrix} = \begin{bmatrix} l_1 R & \dots & l_K R \end{bmatrix} \cdot \begin{bmatrix} S_1 \\ S_2 \\ \dots \\ S_K \end{bmatrix} \qquad (4)$$

[www.cs.berkeley.edu/projects/vision/workshop01/ presentations/cbregler/cbregler.ppt]

Basis Shape Factorization

$$W = \begin{bmatrix} u_1^{(1)} & \dots & u_P^{(1)} \\ v_1^{(1)} & \dots & v_P^{(1)} \\ u_1^{(2)} & \dots & u_P^{(2)} \\ v_1^{(2)} & \dots & v_P^{(2)} \\ & \dots & \\ u_1^{(N)} & \dots & u_P^{(N)} \\ v_1^{(N)} & \dots & v_P^{(N)} \end{bmatrix} = \underbrace{\begin{bmatrix} l_1^{(1)} R^{(1)} & \dots & l_K^{(1)} R^{(1)} \\ l_1^{(2)} R^{(2)} & \dots & l_K^{(2)} R^{(2)} \\ & \dots & \\ l_1^{(N)} R^{(N)} & \dots & l_K^{(N)} R^{(N)} \end{bmatrix}}_{Q} \cdot \underbrace{\begin{bmatrix} S_1 \\ S_2 \\ \dots \\ S_K \end{bmatrix}}_{B} \qquad (5)$$

↑ Complete 2D Tracks or Flow

↑ Matrix-Rank <= 3*K

[www.cs.berkeley.edu/projects/vision/workshop01/ presentations/cbregler/cbregler.ppt]

Factoring twice

$$W = \underbrace{\begin{bmatrix} l_1^{(1)} R^{(1)} & \dots & l_K^{(1)} R^{(1)} \\ l_1^{(2)} R^{(2)} & \dots & l_K^{(2)} R^{(2)} \\ & \dots & \\ l_1^{(N)} R^{(N)} & \dots & l_K^{(N)} R^{(N)} \end{bmatrix}}_{Q} \cdot \underbrace{\begin{bmatrix} S_1 \\ S_2 \\ \dots \\ S_K \end{bmatrix}}_{B} \qquad (5)$$

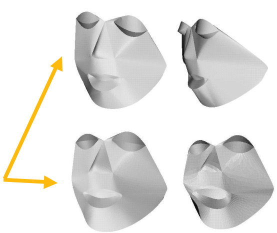
$$= \begin{bmatrix} l_1 r_1 & l_1 r_2 & l_1 r_3 & l_1 r_4 & l_1 r_5 & l_1 r_6 \\ l_2 r_1 & l_2 r_2 & l_2 r_3 & l_2 r_4 & l_2 r_5 & l_2 r_6 \\ & & & \dots & & \\ l_K r_1 & l_K r_2 & l_K r_3 & l_K r_4 & l_K r_5 & l_K r_6 \end{bmatrix}$$

$$= \begin{bmatrix} l_1 \\ l_2 \\ \dots \\ l_K \end{bmatrix} \cdot \begin{bmatrix} r_1 & r_2 & r_3 & r_4 & r_5 & r_6 \end{bmatrix}$$

[www.cs.berkeley.edu/projects/vision/workshop01/ presentations/cbregler/cbregler.ppt]

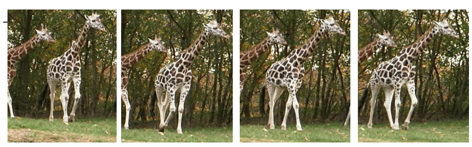
Nonrigid 3D Kinematics results

$$W = \underbrace{\begin{bmatrix} l_1^{(1)} R^{(1)} & \dots & l_K^{(1)} R^{(1)} \\ l_1^{(2)} R^{(2)} & \dots & l_K^{(2)} R^{(2)} \\ & \dots & \\ l_1^{(N)} R^{(N)} & \dots & l_K^{(N)} R^{(N)} \end{bmatrix}}_{Q} \cdot \underbrace{\begin{bmatrix} S_1 \\ S_2 \\ \dots \\ S_K \end{bmatrix}}_{B} \qquad (5)$$



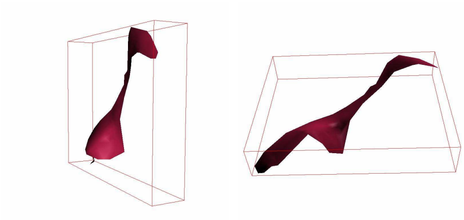
[www.cs.berkeley.edu/projects/vision/workshop01/ presentations/cbregler/cbregler.ppt]

Nonrigid 3D Kinematics results



[www.cs.berkeley.edu/projects/vision/workshop01/ presentations/cbregler/cbregler.ppt]

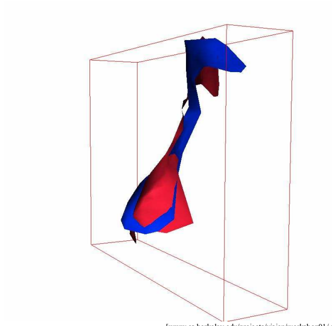
Nonrigid 3D Kinematics results



67

[www.cs.berkeley.edu/projects/vision/workshop01/ presentations/cbregler/cbregler.ppt]

Nonrigid 3D Kinematics results



68

[www.cs.berkeley.edu/projects/vision/workshop01/ presentations/cbregler/cbregler.ppt]

From Pixels to 3D Blend Shapes (Torresani et al 2001)

- No Point Tracks:
  Lucas-Kanade -> Irani -> Model-free Nonrigid:

$$[U|V] \cdot \begin{bmatrix} C & D \\ D & E \end{bmatrix} = [G|H]$$

$$[\hat{Q}_u \cdot \hat{B} | \hat{Q}_v \cdot \hat{B}] \cdot \begin{bmatrix} C & D \\ D & E \end{bmatrix} = [G|H]$$

- Region-Based
- Iterative Refinement
- Occlusion Prediction

69

[www.cs.berkeley.edu/projects/vision/workshop01/ presentations/cbregler/cbregler.ppt]

From Pixels to 3D Blend Shapes (Torresani et al 2001)

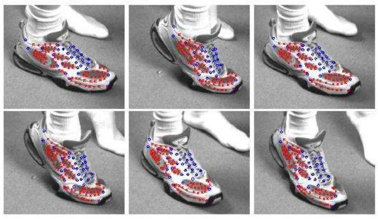


Figure 1. Example tracks of the shoe sequence. The blue circles are reliable points, the red crosses are features with 1D texture that have been recovered using rank constraints.

70

[www.cs.berkeley.edu/projects/vision/workshop01/ presentations/cbregler/cbregler.ppt]

From Pixels to 3D Blend Shapes (Torresani et al 2001)

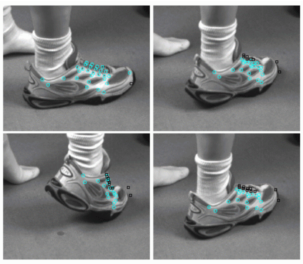


Figure 2. The black rectangular markers are predicted locations of disappearing features. The algorithm can recover the complete trajectories of the temporarily occluded points.

71

[www.cs.berkeley.edu/projects/vision/workshop01/ presentations/cbregler/cbregler.ppt]

From Pixels to 3D Blend Shapes (Torresani et al 2001)

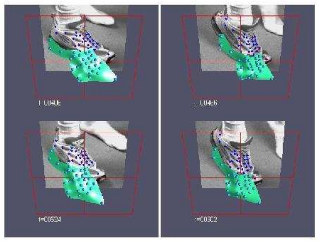
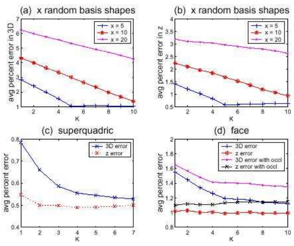


Figure 3. 3D reconstruction of corresponding 2D tracks from monocular video sequence. Please check video to see all details.

72

[www.cs.berkeley.edu/projects/vision/workshop01/ presentations/cbregler/cbregler.ppt]

From Pixels to 3D Blend Shapes (Torresani et al 2001)



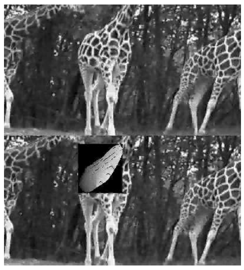
K	2	3	4	5	6	7	8
3D error	3.25	2.19	3.80	2.84	2.24	2.62	2.62
z error	2.33	1.69	1.91	2.49	2.02	2.34	2.34

73

[www.cs.berkeley.edu/projects/vision/workshop01/ presentations/cbregler/cbregler.ppt]

From Pixels to 3D Blend Shapes (Torresani et al 2001)

Synthesis
Modeling
Motion Capture

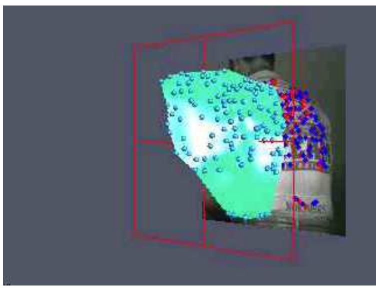


74

[www.cs.berkeley.edu/projects/vision/workshop01/ presentations/cbregler/cbregler.ppt]

From Pixels to 3D Blend Shapes (Torresani et al 2001)

Synthesis
Modeling
Motion Capture



75

[www.cs.berkeley.edu/projects/vision/workshop01/ presentations/cbregler/cbregler.ppt]

Next Lecture: Horn, *Perspective Projection Properly Models Image Formation*

Date: WEDNESDAY 3-10-2004 Time: 1:00 PM - 2:00 PM Location: NE43-814

Methods based on projective geometry have become popular in machine vision because they lead to elegant mathematics, and easy-to-solve linear equations.

It is often not realized that one pays a heavy price for this convenience. Such methods do not correctly model the physics of image formation, require more correspondences, and are considerably more sensitive to measurement error than methods based on true perspective projection.

In this talk we find that for the example of exterior orientation: (i) Methods based on projective geometry are fundamentally different from methods based on perspective projection; (ii) Methods based on projective geometry yield a transformation matrix T that in general does not correspond to a physical imaging situation that is, a rotation, translation and perspective projection; (iii) Optimization methods based on the real physical imaging equations (true perspective projection) produce considerably more accurate results.

76

Date: WEDNESDAY 3-10-2004 Time: 1:00 PM - 2:00 PM Location: NE43-814

Projective Geometry Considered Harmful

Berthold K.P. Horn

77