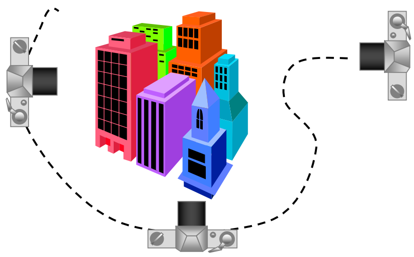
Structure from motion



Reconstruct

- Scene geometry
- Camera motion

Structure from motion

The SFM Problem

- Reconstruct scene geometry and camera motion from two or more images

Track 2D Features → **Estimate 3D** → **Optimize (Bundle Adjust)** → **Fit Surfaces**

SFM Pipeline

Structure from motion

Step 1: Track Features

- Detect good features
  - corners, line segments
- Find correspondences between frames
  - Lucas & Kanade-style motion estimation
  - window-based correlation

Structure from motion

$$\underbrace{\begin{bmatrix} \mathbf{I}_1 \\ \mathbf{I}_2 \\ \vdots \\ \mathbf{I}_f \end{bmatrix}}_{\text{Images}} = \underbrace{\begin{bmatrix} \mathbf{\Pi}_1 \\ \mathbf{\Pi}_2 \\ \vdots \\ \mathbf{\Pi}_f \end{bmatrix}}_{\text{Motion}} \underbrace{\begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_n \end{bmatrix}}_{\text{Structure}}$$

Step 2: Estimate Motion and Structure

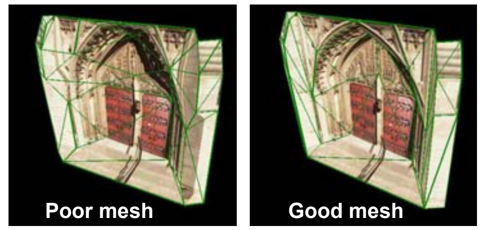
- Simplified projection model, e.g., [Tomasi 92]
- 2 or 3 views at a time [Hartley 00]

Structure from motion

Step 3: Refine Estimates

- "Bundle adjustment" in photogrammetry

Structure from motion



Morris and Kanade, 2000

Step 4: Recover Surfaces

- Image-based triangulation [Morris 00, Baillard 99]
- Silhouettes [Fitzgibbon 98]
- Stereo [Pollefeys 99]

Feature tracking

Problem

- Find correspondence between *n* features in *f* images

Issues

- What's a feature?
- What does it mean to "correspond"?
- How can correspondence be reliably computed?

Feature detection

What's a good feature?

Good features to track

Recall Lucas-Kanade equation:

$$\underbrace{\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}}_{A^T A} \begin{bmatrix} u \\ v \end{bmatrix} = - \underbrace{\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}}_{A^T b}$$

When is this solvable?

- **A^{T}A** should be invertible
- **A^{T}A** should not be too small due to noise
  - eigenvalues l_1 and l_2 of $\mathbf{A^T A}$ should not be too small
- **A^{T}A** should be well-conditioned
  - l_1 / l_2 should not be too large (l_1 = larger eigenvalue)

These conditions are satisfied when *min*(l_1, l_2) > c

Feature correspondence

Correspondence Problem

- Given feature patch F in frame *H*, find best match in frame *I*

Find displacement (u,v) that minimizes SSD error over feature region

$$\sum_{(x,y) \in F \subset J} [I(x+u, y+v) - H(x,y)]^2$$

Solution

- Small displacement: Lukas-Kanade

$$\underbrace{\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}}_{A^T A} \begin{bmatrix} u \\ v \end{bmatrix} = - \underbrace{\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}}_{A^T b}$$

- Large displacement: discrete search over (u,v)
  - Choose match that minimizes SSD (or normalized correlation)

Feature distortion

Feature may change shape over time

- Need a distortion model to really make this work


Find displacement (u,v) that minimizes SSD error over feature region

$$\sum_{(x,y) \in F \subset J} [I(W_x(x,y), W_y(x,y)) - J(x,y)]^2$$

Minimize with respect to W_x and W_y

- Affine model is common choice [Shi & Tomasi 94]

$$\begin{aligned} W_x(x,y) &= ax + by + c \\ W_y(x,y) &= ex + fy + g \end{aligned}$$

Tracking over many frames

So far we've only considered two frames

Basic extension to *f* frames

1. Select features in first frame
2. Given feature in frame i, compute position/deformation in i+1
3. Select more features if needed
4. i = i + 1
5. If i < f, go to step 2

Issues

- Discrete search vs. Lucas Kanade?
  - depends on expected magnitude of motion
  - discrete search is more flexible
- How often to update feature template?
  - update often enough to compensate for distortion
  - updating too often causes drift
- How big should search window be?
  - too small: lost features. Too large: slow

Incorporating dynamics

Idea

- Can get better performance if we know something about the way points move
- Most approaches assume constant velocity

$$\begin{aligned}\dot{\mathbf{x}}_{i+1} &= \dot{\mathbf{x}}_i \\ \mathbf{x}_{i+1} &= 2\mathbf{x}_i - \mathbf{x}_{i-1}\end{aligned}$$

or constant acceleration

$$\begin{aligned}\ddot{\mathbf{x}}_{i+1} &= \ddot{\mathbf{x}}_i \\ \mathbf{x}_{i+1} &= 3\mathbf{x}_i - 3\mathbf{x}_{i-1} + \mathbf{x}_{i-2}\end{aligned}$$

- Use above to predict position in next frame, initialize search

Modeling uncertainty

Kalman Filtering (http://www.cs.unc.edu/~welch/kalman/)

- Updates feature state and Gaussian uncertainty model
- Get better prediction, confidence estimate

CONDENSATION

(http://www.dai.ed.ac.uk/CVonline/LOCAL_COPIES/ISARD1/condensation.html)

- Also known as “particle filtering”
- Updates probability distribution over all possible states
- Can cope with multiple hypotheses

Probabilistic Tracking

Treat tracking problem as a Markov process

- Estimate p($\mathbf{x}_t$ | $\mathbf{z}_t$, $\mathbf{x}_{t-1}$)
  - prob of being in state $\mathbf{x}_t$ given measurement $\mathbf{z}_t$ and previous state $\mathbf{x}_{t-1}$
- Combine Markov assumption with Bayes Rule

$$p(\mathbf{x}_t|\mathbf{z}_t, \mathbf{x}_{t-1}) \propto p(\mathbf{z}_t|\mathbf{x}_t)\ p(\mathbf{x}_t|\mathbf{x}_{t-1})$$

measurement likelihood (likelihood of seeing this measurement)

prediction (based on previous frame and motion model)

Approach

- Predict position at time *t*: $p(\mathbf{x}_t|\mathbf{x}_{t-1})$
- Measure (perform correlation search or Lukas-Kanade) and compute likelihood $p(\mathbf{z}_t|\mathbf{x}_t)$
- Combine to obtain (unnormalized) state probability

$$p(\mathbf{x}_t|\mathbf{z}_t, \mathbf{x}_{t-1})$$

Kalman filtering: assume p(x) is a Gaussian

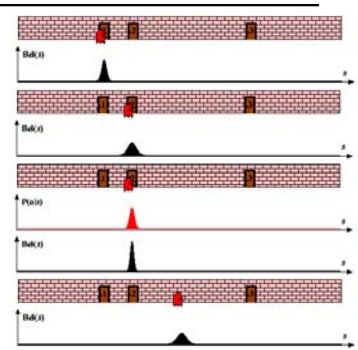
initial state

prediction

measurement

posterior

prediction



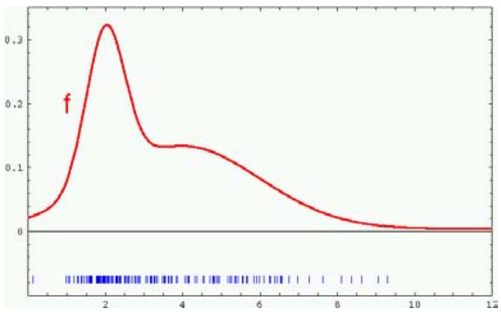
Key

- s = x (position)
- o = z (sensor)

[Schiele et al. 94], [Weiß et al. 94], [Borenstein 96], [Gutmann et al. 96, 98], [Arras 98]

Robot figures courtesy of Dieter Fox

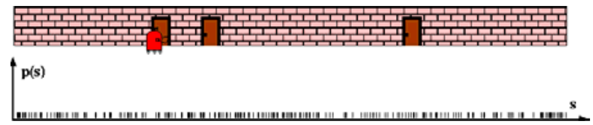
Modeling probabilities with samples



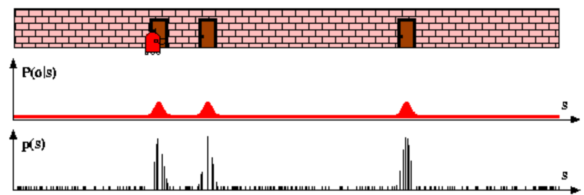
Allocate samples according to probability

- Higher probability—more samples

CONDENSATION [Isard & Blake]

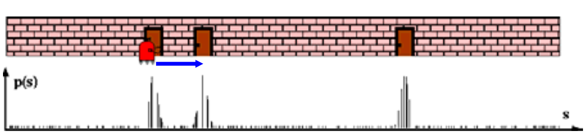


Initialization: unknown position (uniform)



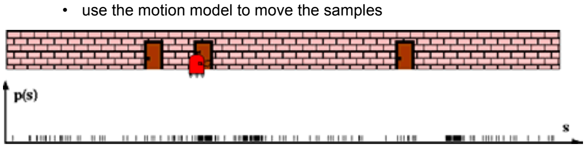
Measurement → posterior

CONDENSATION [Isard & Blake]

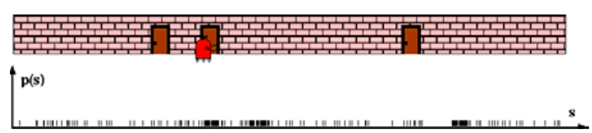


Prediction:

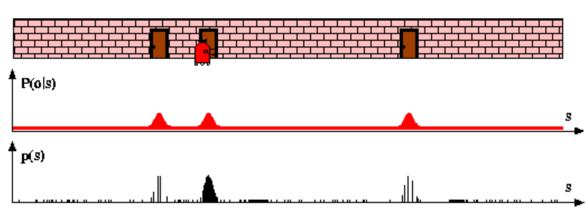
- draw new samples from the PDF
- use the motion model to move the samples



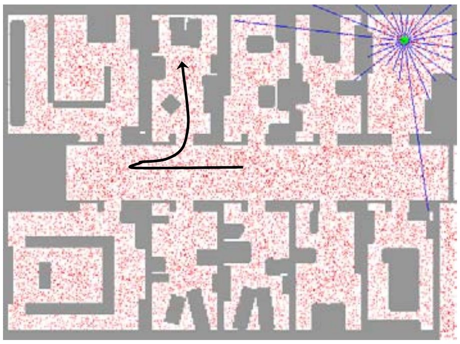
CONDENSATION [Isard & Blake]



Measurement → posterior



Monte Carlo robot localization



Particle Filters [Fox, Dellaert, Thrun and collaborators]

CONDENSATION Contour Tracking

Training a tracker

CONDENSATION Contour Tracking

Red: smooth drawing
Green: scribble
Blue: pause

Structure from motion

The SFM Problem

- Reconstruct scene geometry and camera positions from two or more images

Assume

- Pixel correspondence
  - via tracking
- Projection model
  - classic methods are orthographic
  - newer methods use perspective
  - practically any model is possible with bundle adjustment

SFM under orthographic projection

$$\underset{2\times1}{\mathbf{u}} = \underset{2\times3}{\Pi}\,\underset{3\times1}{\mathbf{X}} + \underset{2\times1}{\mathbf{t}}$$

image point — **projection matrix** — **scene point** — **image offset**

More generally: weak perspective, para-perspective, affine

Trick

- Choose scene origin to be centroid of 3D points
- Choose image origins to be centroid of 2D points
- Allows us to drop the camera translation:

$$\underset{2\times1}{\mathbf{u}} = \underset{2\times3}{\Pi}\,\underset{3\times1}{\mathbf{X}}$$

Shape by factorization [Tomasi & Kanade, 92]

projection of *n* features in one image:

$$\underset{2\times n}{\begin{bmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n\end{bmatrix}} = \underset{2\times 3}{\Pi}\,\underset{3\times n}{\begin{bmatrix}\mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_n\end{bmatrix}}$$

projection of *n* features in *f* images

$$\underset{2f\times n}{\begin{bmatrix}\mathbf{u}_1^1 & \mathbf{u}_2^1 & \cdots & \mathbf{u}_n^1\\ \mathbf{u}_1^2 & \mathbf{u}_2^2 & \cdots & \mathbf{u}_n^2\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{u}_1^f & \mathbf{u}_2^f & \cdots & \mathbf{u}_n^f\end{bmatrix}} = \underset{2f\times 3}{\begin{bmatrix}\Pi^1\\ \Pi^2\\ \vdots\\ \Pi^f\end{bmatrix}}\underset{3\times n}{\begin{bmatrix}\mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_n\end{bmatrix}}$$

W measurement **M motion** **S shape**

Key Observation: *rank*(**W**) <= 3

Shape by factorization [Tomasi & Kanade, 92]

$$\underset{2f\times n}{\mathbf{W}} = \underset{2f\times 3}{\mathbf{M}}\;\underset{3\times n}{\mathbf{S}}$$

known (W) — **solve for** (M S)

Factorization Technique

- **W** is at most rank 3 (assuming no noise)
- We can use *singular value decomposition* to factor **W**:

$$\underset{2f\times n}{\mathbf{W}} = \underset{2f\times 3}{\mathbf{M}'}\;\underset{3\times n}{\mathbf{S}'}$$

Singular value decomposition (SVD)

SVD decomposes any mxn matrix **A** as

$$\underset{m\times n}{\mathbf{A}} = \underset{m\times m}{\mathbf{U}}\;\underset{m\times n}{\Sigma}\;\underset{n\times n}{\mathbf{V}^T}$$

Properties

- Σ is a diagonal matrix containing the eigenvalues of A^TA
  - known as "singular values" of A
  - diagonal entries are sorted from largest to smallest
- columns of U are eigenvectors of AA^T
- columns of V are eigenvectors of A^TA

If A is singular (e.g., has rank 3)

- only first 3 singular values are nonzero
- we can throw away all but first 3 columns of U and V

$$\underset{m\times n}{\mathbf{A}} = \underset{3\times m}{\mathbf{U}'}\;\underset{3\times 3}{\Sigma'}\;\underset{3\times n}{\mathbf{V}'^T}$$

- Choose M' = U', S' = $\Sigma' V'^T$

Shape by factorization [Tomasi & Kanade, 92]

known — $\underset{2f\times n}{\mathbf{W}} = \underset{2f\times 3}{\mathbf{M}}\ \underset{3\times n}{\mathbf{S}}$ — solve for

Factorization Technique

- **W** is at most rank 3 (assuming no noise)
- We can use *singular value decomposition* to factor **W**:

$$\underset{2f\times n}{\mathbf{W}} = \underset{2f\times 3}{\mathbf{M'}}\ \underset{3\times n}{\mathbf{S'}}$$

- **S'** differs from **S** by a linear transformation ***A***:

$$\mathbf{W} = \mathbf{M'S'} = (\mathbf{MA}^{-1})(\mathbf{AS})$$

- Solve for **A** by enforcing *metric* constraints on **M**

Metric constraints

Orthographic Camera

- Rows of Π are orthonormal: $\Pi\Pi^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

Weak Perspective Camera

- Rows of Π are orthogonal: $\Pi\Pi^T = \begin{bmatrix} * & 0 \\ 0 & * \end{bmatrix}$

Enforcing "Metric" Constraints

- Compute **A** such that rows of **M** have these properties

$$\mathbf{M'A} = \mathbf{M}$$

Trick (not in original Tomasi/Kanade paper, but in followup work)

- Constraints are linear in $\mathbf{AA}^T$:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \Pi\Pi^T = \Pi'\mathbf{A}\left(\mathbf{A}^T\Pi'^T\right) = \Pi'\mathbf{G}\Pi'^T \qquad \textit{where}\ \ \mathbf{G} = \mathbf{AA}^T$$

- Solve for **G** first by writing equations for every Π_i in **M**
- Then $\mathbf{G} = \mathbf{AA}^T$ by SVD (since **U** = **V**)

Factorization with noisy data

$$\underset{2f\times n}{\mathbf{W}} = \underset{2f\times 3}{\mathbf{M}}\ \underset{3\times n}{\mathbf{S}} + \underset{2f\times n}{\mathbf{E}}$$

Once again: use SVD of W

- Set all but the first three singular values to 0
- Yields new matrix **W'**
- **W'** is optimal rank 3 approximation of **W**

$$\underset{2f\times n}{\mathbf{W}} = \underset{2f\times n}{\mathbf{W'}} + \underset{2f\times n}{\mathbf{E}}$$

Approach

- Estimate **W'**, then use noise-free factorization of **W'** as before
- Result minimizes the SSD between positions of image features and projection of the reconstruction

Many extensions

Independently Moving Objects
Perspective Projection
Outlier Rejection
Subspace Constraints
SFM Without Correspondence

Extending factorization to perspective

Several Recent Approaches

- [Christy 96]; [Triggs 96]; [Han 00]; [Mahamud 01]
- Initialize with ortho/weak perspective model then iterate

Christy & Horaud

- Derive expression for weak perspective as a perspective projection plus a correction term:

$$\mathbf{u}_w = (1+\varepsilon)\mathbf{u}_p$$

$$\text{where} \quad \varepsilon = \frac{\mathbf{k}\cdot\mathbf{X}}{t_z}$$

$$\text{and} \quad [\mathbf{k} \quad t_z] \text{ is third row of projection matrix}$$

- Basic procedure:
  - Run Tomasi-Kanade with weak perspective
  - Solve for ε_i (different for each row of M)
  - Add correction term to W, solve again (until convergence)

Bundle adjustment

3D → 2D mapping

- a function of intrinsics **K**, extrinsics **R** & **t**
- measurement affected by noise

$$\begin{aligned} u_i &= f(\mathbf{K},\mathbf{R},\mathbf{t},\mathbf{x}_i) + n_i = \hat{u}_i + n_i, \quad n_i \sim N(0,\sigma) \\ v_i &= g(\mathbf{K},\mathbf{R},\mathbf{t},\mathbf{x}_i) + m_i = \hat{v}_i + m_i, \quad m_i \sim N(0,\sigma) \end{aligned}$$

Log likelihood of ***K,R,t*** given $\{(u_i,v_i)\}$

$$C = -\log L = \sum_i (u_i - \hat{u}_i)^2/\sigma_i^2 + (v_i - \hat{v}_i)^2/\sigma_i^2$$

Minimized via nonlinear least squares regression

- called "Bundle Adjustment"
- e.g., Levenberg-Marquardt
  - described in Press et al., Numerical Recipes

Match Move

Film industry is a heavy consumer

- composite live footage with 3D graphics
- known as "match move"

Commercial products

- 2D3
  - http://www.2d3.com/
- RealVis
  - http://www.realviz.com/

Show video

Closing the loop

Problem

- requires good tracked features as input

Can we use SFM to help track points?

- basic idea: recall form of Lucas-Kanade equation:

$$\begin{bmatrix} a_i & b_i \\ b_i & c_i \end{bmatrix}\begin{bmatrix} u_{ij} \\ v_{ij} \end{bmatrix} = \begin{bmatrix} g_{ij} \\ h_{ij} \end{bmatrix}$$

- with n points in f frames, we can stack into a big matrix

$$\underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B} & \mathbf{C} \end{bmatrix}}_{\mathbf{2n\times 2n}}\underbrace{\begin{bmatrix} \mathbf{U} \\ \mathbf{V} \end{bmatrix}}_{\mathbf{2n\times f}} = \underbrace{\begin{bmatrix} \mathbf{G} \\ \mathbf{H} \end{bmatrix}}_{\mathbf{2n\times f}}$$

Matrix on RHS has rank <= 3 !!

- use SVD to compute a rank 3 approximation
- has effect of filtering optical flow values to be consistent
- [Irani 99]

From [Irani 99]

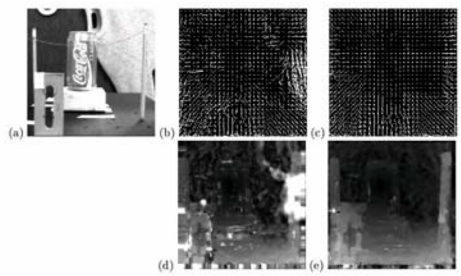


Figure 1: *Real image sequence (the NASA coke-can sequence).* **(a)** One frame from a 27-frame sequence of a forward moving camera in a 3D scene. **(b)** Flow field generated with the two-frame Lucas & Kanade algorithm. Note the errors in the right hand side, where there is depth discontinuity (pole in front of sweater), as well as the aperture problem. **(c)** The flow field for the corresponding frame generated by the multi-frame constrained algorithm. Note the good recovery of flow in those regions. **(d,e)** The flow magnitudes at every pixel. This display provides a higher resolution display of the error. Note the clear depth discontinuities in the multi-frame flow image. The flow values on the coke can are very small, because the camera FOE is in that area.

References

- C. Baillard & A. Zisserman, "*Automatic Reconstruction of Planar Models from Multiple Views*", Proc. Computer Vision and Pattern Recognition Conf. (CVPR 99) 1999, pp. 559-565.
- S. Christy & R. Horaud, "*Euclidean shape and motion from multiple perspective views by affine iterations*", IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(10):1098-1104, November 1996 (ftp://ftp.inrialpes.fr/pub/movi/publications/rec-affiter-long.ps.gz)
- A.W. Fitzgibbon, G. Cross, & A. Zisserman, "*Automatic 3D Model Construction for Turn-Table Sequences*", SMILE Workshop, 1998.
- M. Han & T. Kanade, *"Creating 3D Models with Uncalibrated Cameras"*, Proc. IEEE Computer Society Workshop on the Application of Computer Vision (WACV2000), 2000.
- R. Hartley & A. Zisserman, "*Multiple View Geometry*", Cambridge Univ. Press, 2000.
- R. Hartley, "*Euclidean Reconstruction from Uncalibrated Views*", In Applications of Invariance in Computer Vision, Springer-Verlag, 1994, pp. 237-256.
- M. Isard and A. Blake, *"CONDENSATION -- conditional density propagation for visual tracking"*, International Journal Computer Vision, 29, 1, 5--28, 1998. (ftp://ftp.robots.ox.ac.uk/pub/ox.papers/VisualDynamics/ijcv98.ps.gz)
- S. Mahamud, M. Hebert, Y. Omori and J. Ponce, "Provably-Convergent Iterative Methods for Projective Structure from Motion",Proc. Conf. on Computer Vision and Pattern Recognition, (CVPR 01), 2001. (http://www.cs.cmu.edu/~mahamud/cvpr-2001b.pdf)
- D. Morris & T. Kanade, *"Image-Consistent Surface Triangulation"*, Proc. Computer Vision and Pattern Recognition Conf. (CVPR 00), pp. 332-338.
- M. Pollefeys, R. Koch & L. Van Gool, *"Self-Calibration and Metric Reconstruction in spite of Varying and Unknown Internal Camera Parameters"*, Int. J. of Computer Vision, 32(1), 1999, pp. 7-25.
- J. Shi and C. Tomasi, *"Good Features to Track"*, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 94), 1994, pp. 593-600 (http://www.cs.washington.edu/education/courses/cse590ss/01wi/notes/good-features.pdf)
- C. Tomasi & T. Kanade, *"Shape and Motion from Image Streams Under Orthography: A Factorization Method"*, Int. Journal of Computer Vision, 9(2), 1992, pp. 137-154.
- B. Triggs, *"Factorization methods for projective structure and motion"*, Proc. Computer Vision and Pattern Recognition Conf. (CVPR 96), 1996, pages 845--51.
- M. Irani, *"Multi-Frame Optical Flow Estimation Using Subspace Constraints"*, IEEE International Conference on Computer Vision (ICCV), 1999 (http://www.wisdom.weizmann.ac.il/~irani/abstracts/flow_iccv99.html)