Non-Linear Least Squares and Sparse Matrix Techniques: Applications

Richard Szeliski
Microsoft Research

UW-MSR Course on
Vision Algorithms

CSE/EE 577, 590CV, Spring 2004

Readings

- R.Szeliski. *Fast surface interpolation using hierarchical basis functions*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(6):513-528, June 1990.
- R. Szeliski and S. B. Kang. *Recovering 3D shape and motion from image streams using nonlinear least squares*. J. Vis. Comm. Image Representation, 5(1):10-28, March 1994. (Also available as CRL-93-3.)
- A. Levin and R. Szeliski, *Visual Odometry and Map Correlation*, CVPR'2004.
- P. Pérez, M. Gangnet, and A. Blake. *Poisson Image Editing*. SIGGRAPH'2003.
- L. Zhang, G. Dugas-Phocion, J.-S. Samson, and S. M. Seitz, *Single view modeling of free-form scenes*, J. Vis. Comp. Anim., 2002, vol. 13, no. 4, pp. 225-235.
- A. Agarwala, A. Hertzmann, D. H. Salesin, S. M. Seitz, *Keyframe-Based Tracking for Rotoscoping and Animation*, SIGGRAPH'2004.

Outline

Preconditioning

- diagonal scaling, partial Cholesky factorization
- hierarchical basis functions (wavelets)
- quadtree splines
- 2D application:
  height and normal interpolation
  (Single View Modeling)

Outline

Structure from motion

- problems with size, linearity, conditioning
- alternative parameterizations
- uncertainty modeling

Outline

Visual Odometry and Map Correlation

- fast visual localization
- visual odometry↔map correlation
- open problems

Preconditioned Conjugate Gradient

(See Shewchuk's TR)

Conditioning

Conjugate gradient's performance depends on the *condition number* of the Hessian

Suppose we wish to perform enough iterations to reduce the norm of the error by a factor of ε; that is, $\|e_{(i)}\| \le \varepsilon \|e_{(0)}\|$. Equation 28 can be used to show that the maximum number of iterations required to achieve this bound using Steepest Descent is

$$i \le \left\lceil \frac{1}{2}\kappa \ln\left(\frac{1}{\varepsilon}\right)\right\rceil,$$

whereas Equation 52 suggests that the maximum number of iterations CG requires is

$$i \le \left\lceil \frac{1}{2}\sqrt{\kappa} \ln\left(\frac{2}{\varepsilon}\right)\right\rceil.$$

Transformation of variables

Replace original unknowns x with transformed variables $\hat{x} = E^T x$

The system $Ax = b$ can be transformed into the problem

$$E^{-1}AE^{-T}\hat{x} = E^{-1}b, \qquad \hat{x} = E^T x,$$

The new condition number depends on $\kappa(M^{-1}A)$ with $M = EE^T$

Untransformed Precond. CG

Don't perform transform of variables explicitly

Instead, "rotate" the residual vector, $M^{-1}r$

$$r_{(0)} = b - Ax_{(0)},$$
$$d_{(0)} = M^{-1}r_{(0)},$$
$$\alpha_{(i)} = \frac{r_{(i)}^T \boxed{M^{-1}r_{(i)}}}{d_{(i)}^T A d_{(i)}},$$
$$x_{(i+1)} = x_{(i)} + \alpha_{(i)}d_{(i)},$$
$$r_{(i+1)} = r_{(i)} - \alpha_{(i)}Ad_{(i)},$$
$$\beta_{(i+1)} = \frac{r_{(i+1)}^T \boxed{M^{-1}r_{(i+1)}}}{r_{(i)}^T \boxed{M^{-1}r_{(i)}}},$$
$$d_{(i+1)} = \boxed{M^{-1}r_{(i+1)}} + \beta_{(i+1)}d_{(i)}.$$

Diagonal and Cholesky precond.

Simple preconditioner (almost for free) is *diagonal preconditioning*, $M = \mathrm{diag}(A)$

Better preconditioner is *incomplete Cholesky factorization*, $M = LL^T$, $A \approx LL^T$

- minimize fill-in (non-zero entries)
- not always stable (Shewchuk)

Hierarchical bases [Szeliski, PAMI'90]

Hierarchical Basis Functions

- Proposed by Yserentant [1986]
- Use a *multi-level* (pyramid-like) basis for system

$$x = Sy, \; S = S_1 \ldots S_{L-1}$$

$$S_1 = \begin{pmatrix} 1 & & & & \\ \frac{1}{2} & 1 & \frac{1}{2} & & \\ & & 1 & & \\ & & \frac{1}{2} & 1 & \frac{1}{2} \\ & & & & 1 \end{pmatrix}.$$

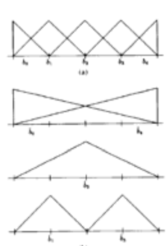


Fig. 4. Nodal and hierarchical basis functions. (a) Nodal. (b) Hierarchical.

2-D Hierarchical basis

Like a pyramid, but *complete* (no extra variables)

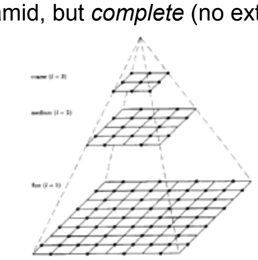


Fig. 5. Multiresolution pyramid. The circles indicate the nodes in the hierarchical basis.

Surface interpolation example

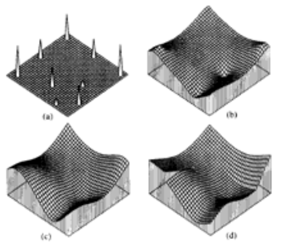


Fig. 1. Sample data points and interpolated solutions: (a) sample data points, (b) membrane interpolant, (c) thin plate interpolant, (d) controlled continuity spline (thin plate with discontinuities and creases).

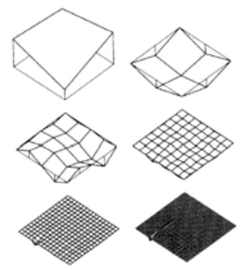


Fig. 6. Hierarchical basis representation of Fig. 1(d).

Surface interpolation - performance

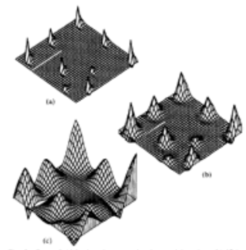


Fig. 2. Gauss-Seidel relaxation example after (a) 1 iteration, (b) 10 iterations, (c) 100 iterations.

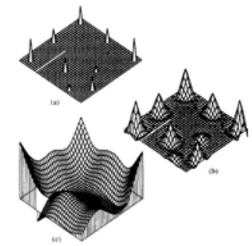


Fig. 3. Conjugate gradient relaxation example after (a) 1 iteration, (b) 10 iterations, (c) 100 iterations.

Surface interpolation - performance

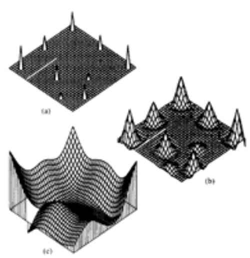


Fig. 3. Conjugate gradient relaxation example after (a) 1 iteration, (b) 10 iterations, (c) 100 iterations.

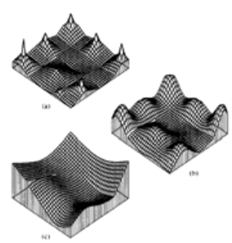


Fig. 9. Hierarchical conjugate gradient ($L = 4$) relaxation example after (a) 1 iteration, (b) 10 iterations, (c) 100 iterations.

Surface interpolation - performance

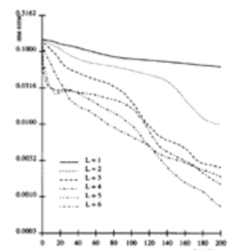


Fig. 8. Algorithm convergence as a function of L: controlled-continuity thin plate, bilinear interpolator.

Eigenvalue analysis

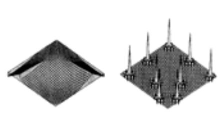


Fig. 20. Eigenvectors for original stiffness A: (a) $\lambda_1 = 0.000827$, (b) $\lambda_{max} = 77.52$.

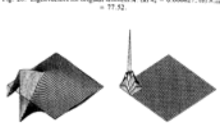


Fig. 21. Eigenvectors for preconditioned stiffness $\hat{A}$ with $L = 3$: (a) $\lambda_1 = 0.003979$, (b) $\lambda_{max} = 121.44$.

Comparison with multigrid

Solve a series of nested problems at different resolutions, using *injection* and *prolongation* to move between levels [Briggs, 1987]

(Personal experience): does not seem to work well on inhomogeneous problems (as opposed to very fine grid refinement)

Open issues

How many levels to use?

Which order interpolant?

Adaptation to data density and continuity

- intuition: when data terms dominate, want *nodal* basis; when smoothness, want *hierarchical*

Quadtree splines

Only populate (estimate) a subset of mesh

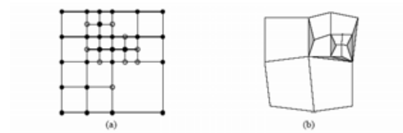


Figure 6: Quadtree associated with spline function, and potential cracks in quadtree spline (a) the nodes with filled circles (●) are free variables in the associated hierarchical basis, whereas the open circles (○) (and also the nodes not drawn) must be zero (in the nodal basis, these nodes are interpolated from their ancestors); (b) potential cracks in a simpler quadtree spline are shown as shaded areas.

[Szeliski & Shum, PAMI'96]

Quadtree spline

Equivalent to *sparse* hierarchical basis

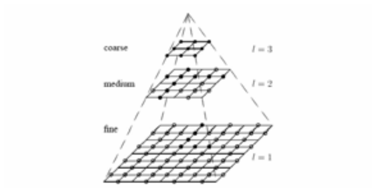
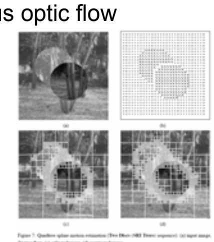


Figure 3: Multiresolution pyramid

The multiple resolution levels are a schematic representation of the hierarchical basis spline. The circles indicate the nodes in the hierarchical basis. Filled circles (●) are free variables in the quadtree spline (Section 6), while open circles (○) must be zero (see Figure 6).

Quadtree splines - application

Discontinuous optic flow



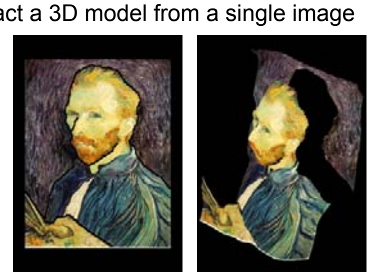
Discontinuous single view modeling (next)

Single view modeling of free-form scenes

L. Zhang, G. Dugas-Phocion, J.-S. Samson, and S. M. Seitz

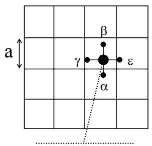
Problem

Extract a 3D model from a single image



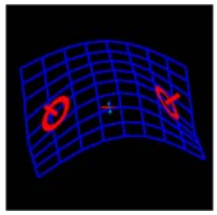
Normal constraints

Fix the local orientation of the surface

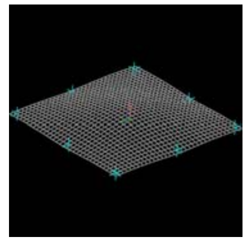


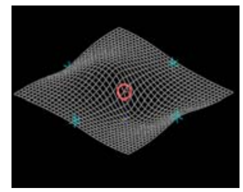
P : *point where we set the normal constraint*

- $D_y f(P) = \dfrac{f(\varepsilon) - f(\gamma)}{a}$
- $D_x f(P) = \dfrac{f(\beta) - f(\alpha)}{a}$



Sample deformations

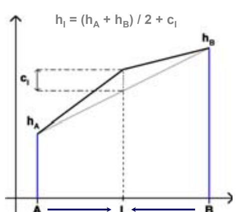




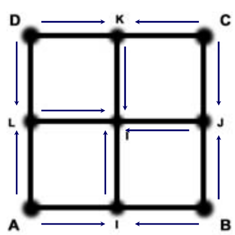
Speed up the solver

Wavelet transformation *(from coefficient to height)*

1D Convolution



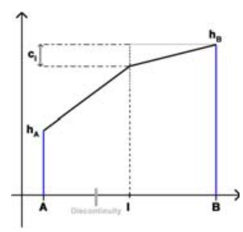
2D Convolution



Speed up the solver

What happens with discontinuities ?

1D Convolution



- Instead of :
  $h_I = (h_A + h_B) / 2 + c_I$
- We get :
  $h_I = h_B + c_I$

Advantage of discontinuities

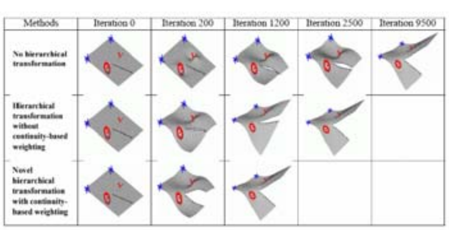
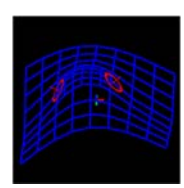


Figure 4: Performance comparison of solving Eq. 11 by using no hierarchical transformation, traditional transformation, and our novel transformation in terms of number of iterations. The model has approximately 1400 grid points, and 4 constraints.

Hierarchical structure

Ways of changing the resolution.

- User specified
- Along a curve
- High curvature



Final results

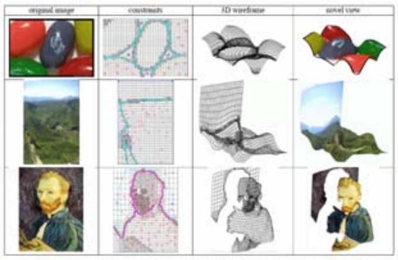


Figure 5: Examples of single view modeling on different scenes. From left to right, the columns show the original images, user-specified constraints on adaptive grids, 3D wireframe rendering, and textured rendering.

Structure from motion

Problem areas

Problems with size, linearity, conditioning

- reduce the number of variables [Shum, Ke, Zhang]
- partition the cameras / data [Steedly *et al.*]

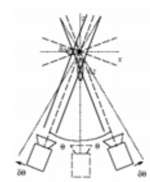
Uncertainty modeling

Alternative parameterizations

Uncertainty modeling

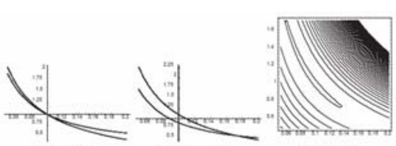
Bas-relief ambiguity:

- correlation between depth and motion/rotation



(b) antisymmetric cameras

Uncertainty modeling



(a) Noise-free (b) Noise = 0.01 (c) Contour plot

Figure 2: Constraint lines and energy surface for simple two-parameter example. The x-axis is the angle $\Delta\theta$ and the y-axis is the scale factor a.

Alternative projection

Use *object-centered* representation

The standard perspective projection equation used in computer vision is

$$\begin{pmatrix} u \\ v \end{pmatrix} = \mathcal{P}_1 \begin{pmatrix} x \\ y \\ z \end{pmatrix} \equiv \begin{pmatrix} f\frac{x}{z} \\ f\frac{y}{z} \end{pmatrix} \quad (5)$$

where f is a product of the focal length of the camera and the pixel array scale factor (we assume that pixels are square, since this has been verified experimentally for our camera).

An alternative formulation which we use in this paper is

$$\begin{pmatrix} u \\ v \end{pmatrix} = \mathcal{P}_2 \begin{pmatrix} x \\ y \\ z \end{pmatrix} \equiv \begin{pmatrix} s\frac{x}{1+\eta z} \\ s\frac{y}{1+\eta z} \end{pmatrix} \quad (6)$$

Video-Based Rendering

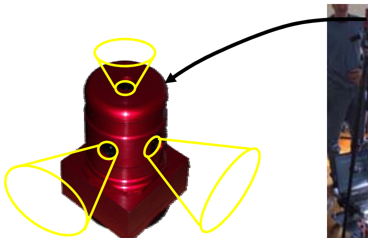
Video-Based Virtual Tours

Move camera along a rail (“dolly track”) and play back a 360° video

Applications:

- Homes and architecture
- Outdoor locations (tourist destinations)

360° video camera



OmniCam (six-camera head)

OmniCam

Built by Point Grey Research (*Ladybug*)

Six camera head

Portable hard drives, fiber-optic link

Resolution per image: 1024 x 768

FOV: ~100º x ~80º

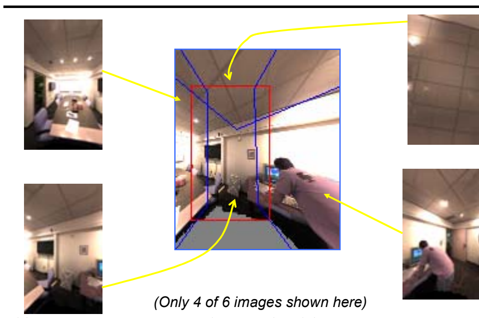
Acquisition speed: 15 fps uncompressed

Acquisition platforms

Robotic cart

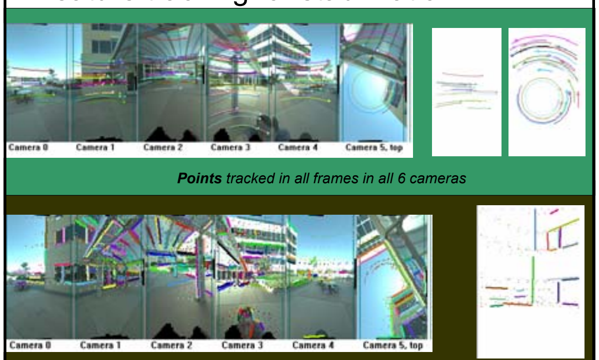
Wearable

Stitching



(Only 4 of 6 images shown here)

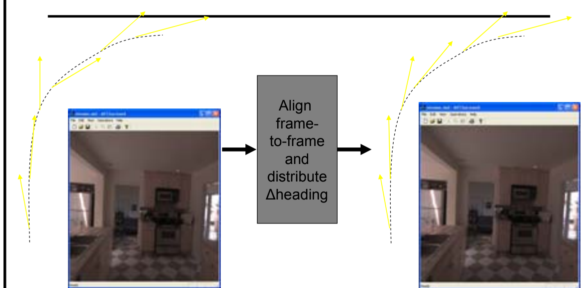
Feature tracking for stabilization



Points tracked in all frames in all 6 cameras

Edges tracked in all frames in all 6 cameras

Stabilization



Before motion stabilization

After motion stabilization

Outdoor Garden Tour

Head-mounted outdoor scene acquisition

Map control

Localized audio for a richer experience

Complex path navigation (greater freedom of motion)

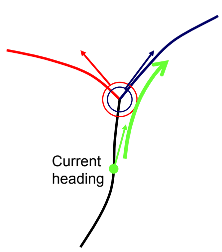
Map control

- **The garden map**
- **Drawing the acquisition path**
- **Mapping video frames positions on map**
- **Placing sound sources (background and dynamic)**
- **Output is a descriptive XML file**

Bifurcation handling

Hypothesize, align, choose best pair as bifurcation

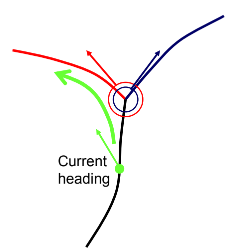
Choice of path depends on current heading and bifurcation point orientation

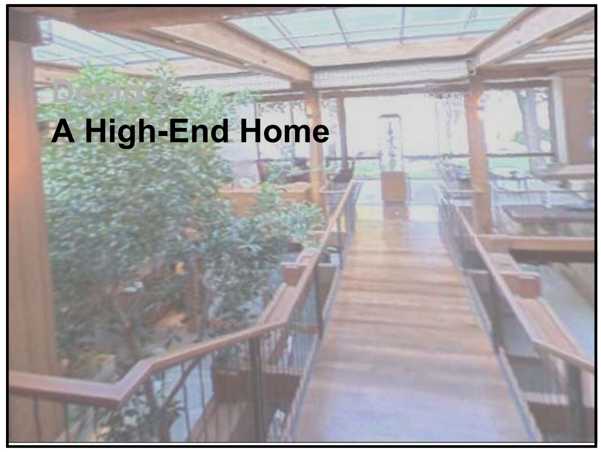


Bifurcation handling

Hypothesize, align, choose best pair as bifurcation

Choice of path depends on current heading and bifurcation point orientation





Future directions

Towards true 3D:

- better compression
- object interaction (e.g., vase on table)
- object compositing (video game)
- more general motion ("true" IBR)
- automate generation of bifurcation points and map control (UI)

Visual Odometry and Map Correlation

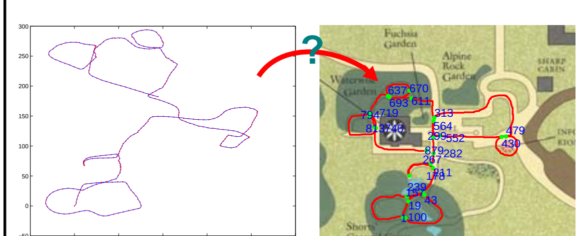
Anat Levin and Richard Szeliski
CVPR'2004

Map Correlation

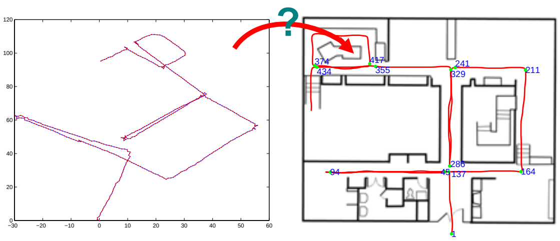
Four stages:

1. Overlap detection (path bifurcations, better egomotion estimation)
2. Map correlation: [temporally] align hand-drawn map with video-based ego-motion estimates
3. Align ego-motion to map
4. Optimize camera position from both visual and map data

Map Correlation

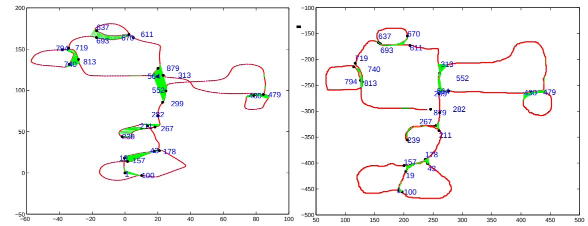


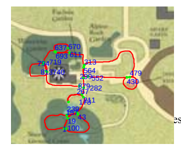
Map Correlation



5/4/2004 NLS and Sparse Matrix Techniques 55

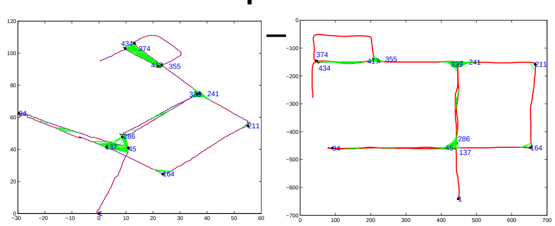
1. Overlap detection

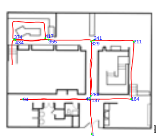




5/4/2004 56

Overlap detection

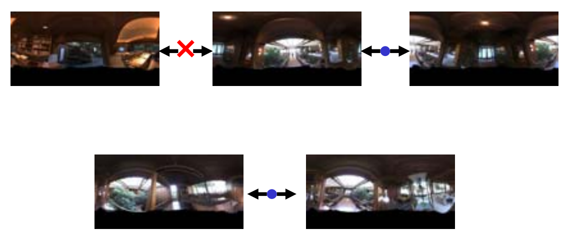




5/4/2004 57

Overlap detection

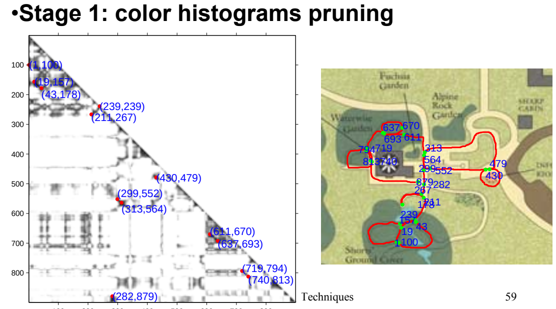
•Stage 1: color histograms pruning



5/4/2004 NLS and Sparse Matrix Techniques 58

Overlap detection

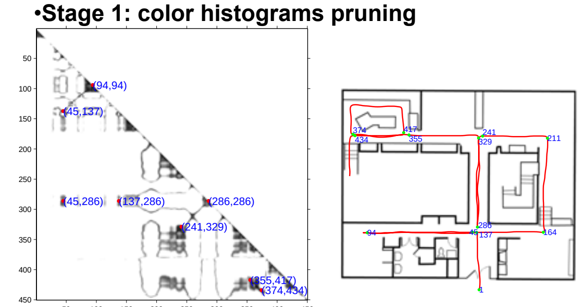
•Stage 1: color histograms pruning



Techniques 59

Overlap detection

•Stage 1: color histograms pruning



Overlap detection

•Stage 2: moment matching

$$I(u) \leftrightarrow I(Ru)$$

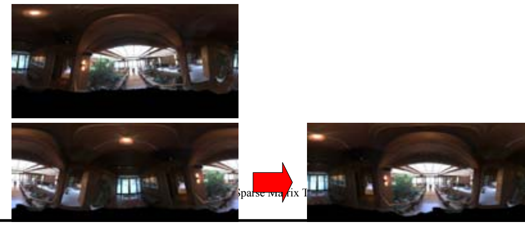
$$\int u I(Ru)du \leftrightarrow R^T \int v I(v)dv$$

use different Image functions (color histogram bins)

Overlap detection

Stage 2: For each $(I_1(u),I_2(u))$ with similar color histograms:

RANSAC computation of R based on 1st order moments



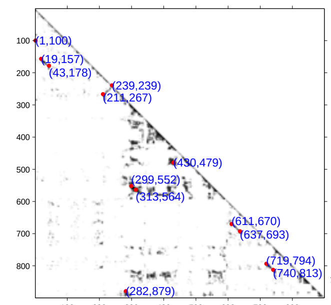
Overlap detection

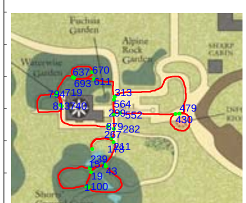
Stage 2: For each $(I1(u),I2(u))$ with similar color histograms:

RANSAC computation of R based on 1st order moments

Overlap detection

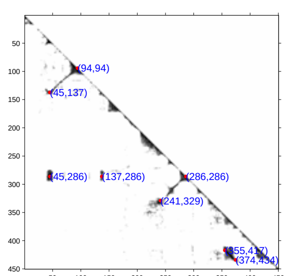
•Stage 2: moments matching

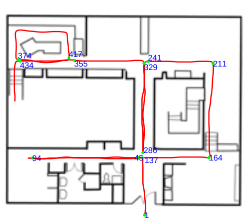




Overlap detection

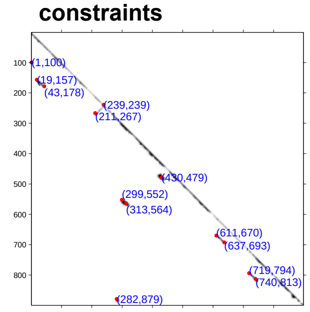
•Stage 2: moments matching

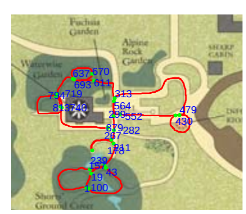




Overlap detection

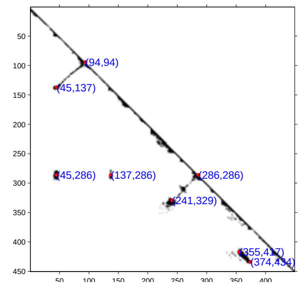
•Stage 3:Features matching & epipolar constraints

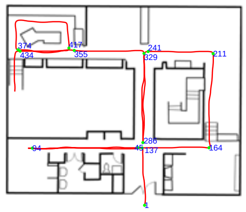




Overlap detection

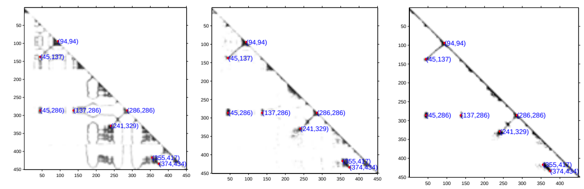
•Stage 3:Feature matching & epipolar constraints

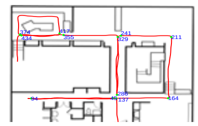




Overlap detection

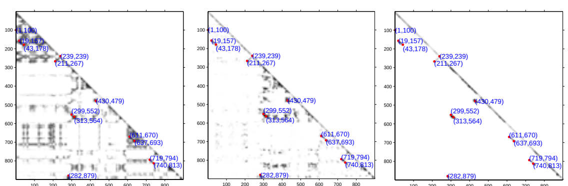
1 2 3

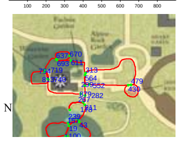




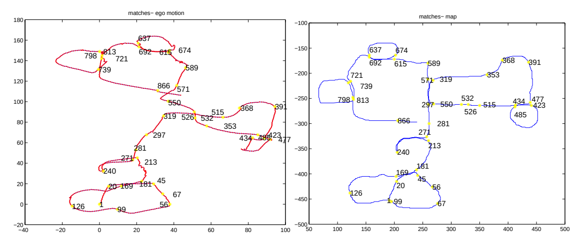
Overlap detection

1 2 3





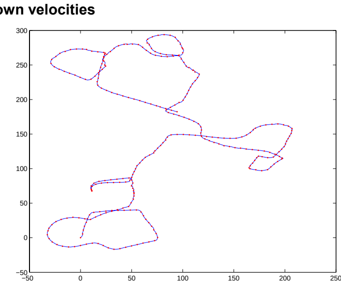
2. Map correlation



Ego-motion challenges

•Tracks visible only other a short time window-> ego motion estimation tends to drift over time

•Unknown velocities



Ego motion <-> map matching

Find:

$$F:\{1,\ldots,m\} \to \{1,\ldots,n\}$$

Map points

Sequence frames

Any ideas?

Ego motion ↔ map matching

Find: $F:\{1,\ldots,m\} \rightarrow \{1,\ldots,n\}$

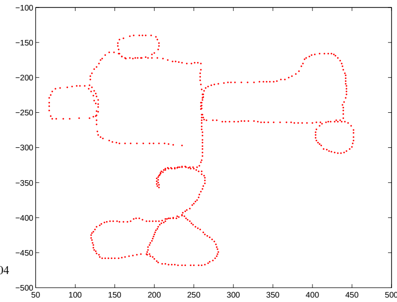
A "good" F:

- **Monotonic and smooth**
- **Minimize local orientation differences**
- (m_i, m_j) **close on the map** $\rightarrow$ (F_i, F_j) **Identified as "same location" by the previous stage**

Graphical model representation

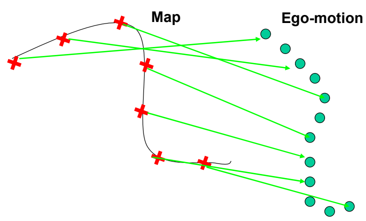
Nodes- discretized map points

Mapped into sequence frames



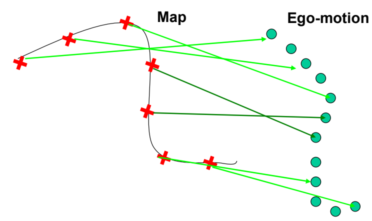
Graphical model representation

- **F Monotonic and smooth**



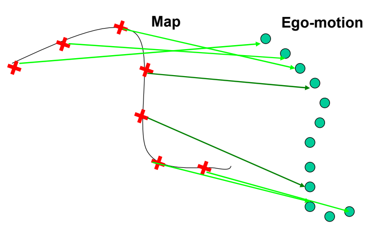
Graphical model representation

- **F Monotonic and smooth**



Graphical model representation

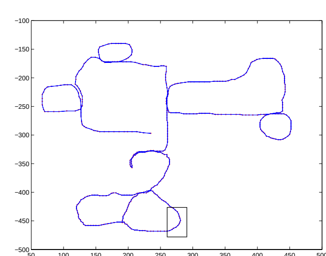
- **F Monotonic and smooth**

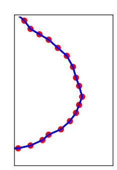


Graphical model representation

- **F Monotonic and smooth**

$$\psi^1_{i,i+1} = \exp(-\|F_i - F_{i+1}\|)[F_i \le F_{i+1}]$$

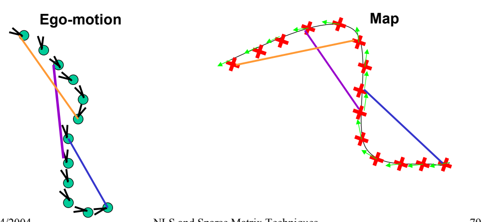




Graphical model representation

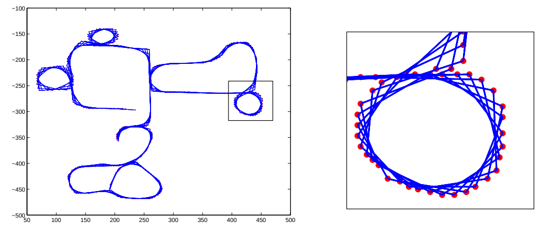
•F Minimize local orientation differences

$$\psi^2_{i,i+\rho} = \exp(-\| [\alpha(i)-\alpha(i+\rho) \|]- [\alpha(F_i)-\alpha(F_{i+\rho}) \|]$$



Graphical model representation

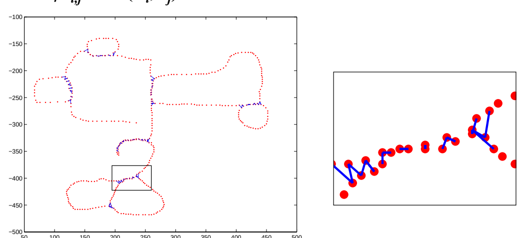
•F Minimize local orientation differences



Graphical model representation

• (m_i,m_j) close on the map → (F_i,F_j)

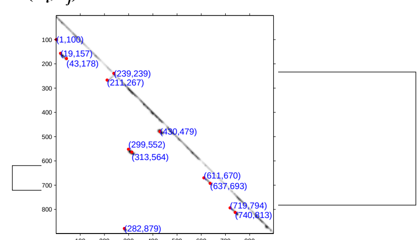
$\psi^1_{i,j}$ = iff (F_i,F_j) identified as "same location"



Graphical model representation

• (m_i,m_j) close on the map → (F_i,F_j)

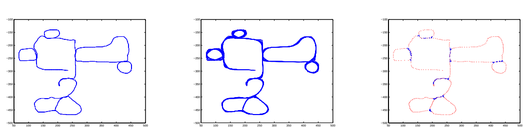
$\psi^1_{i,j}$ = iff (F_i,F_j) identified as "same location"



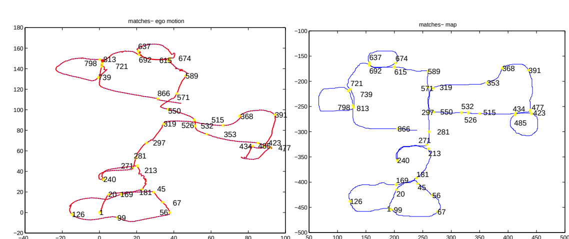
Graphical model representation

How do we find the best assignment?
Use belief propagation to find
assignment maximizing:

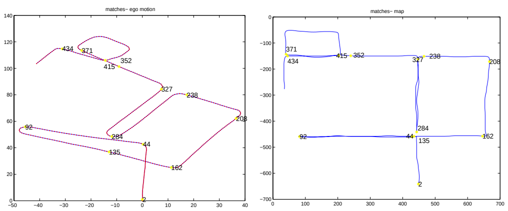
$$\prod_{<i,j>\in E_1} \Psi^1_{ij} \prod_{<i,j>\in E_1} \Psi^2_{ij} \prod_{<i,j>\in E_3} \Psi^3_{ij}$$



Ego motion ↔ map matching

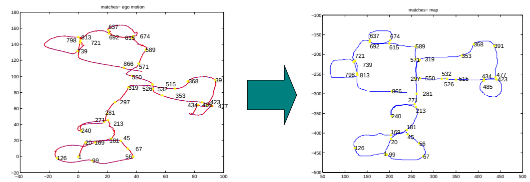


Ego motion ↔ map matching

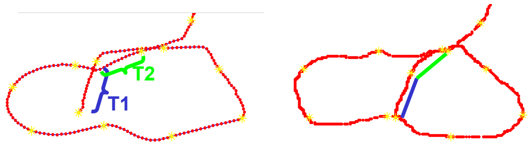


Transforming ego motion → map

Obtain initial guess for motion parameters-minimizing distance to map while preserving ego-motion high frequencies.



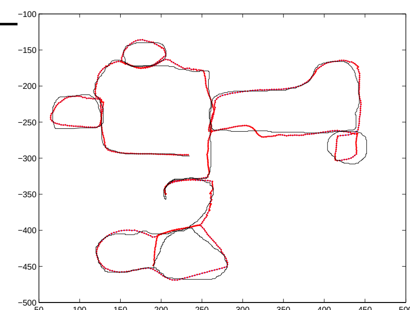
Transforming ego motion → map



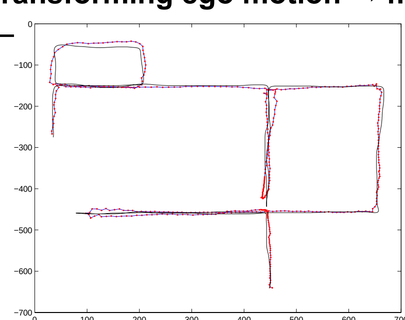
Local Euclidian transformations:

- **Minimizing distance to map**
- **Agree on concatenation points**
- **Respect overlapping constraints**

Transforming ego motion → map



Transforming ego motion → map

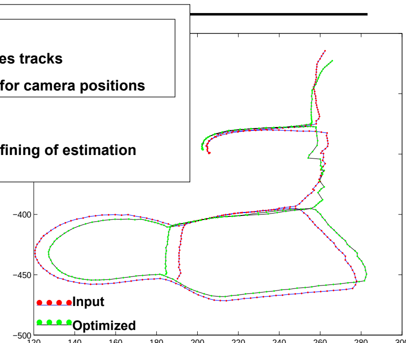


Bundle adjustment optimization

Input:

- **Visual features tracks**
- **Initial guess for camera positions**

Non linear refining of estimation



Bundle adjustment optimization

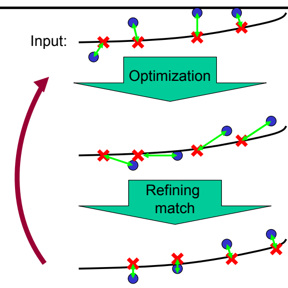
Lessons learned:

•poor linearization

•numerical conditioning

•may need alternate representation and/or optimization algorithm

Match Refining



Visual Odometry & Map Correlation

1. **Fast overlap detection**
2. **Rough ego-motion <-> map correlation based on rotations (orientation)**
3. **Local non-linear optimization**

Questions?