Vision 3D artificielle
Session 3: Disparity maps, multi-view geometry

Pascal Monasse monasse@imagine.enpc.fr

IMAGINE, École des Ponts ParisTech

Contents

Contents

Triangulation

- Let us write again the binocular formulae:

$$\lambda x = K(RX + T) \quad \lambda' x' = K' X$$

- Write $Y^T = \begin{pmatrix} X^T & \lambda & \lambda' \end{pmatrix}$:

$$\begin{pmatrix} KR & -x & 0_3 \\ K' & 0_3 & -x' \end{pmatrix} Y = \begin{pmatrix} KT \\ 0_3 \end{pmatrix}$$

  (6 equations↔5 unknowns+1 epipolar constraint)
- We can then recover X.
- Special case: $R = Id$, $T = Be_1$
- We get:

$$z(x - KK'^{-1}x') = \begin{pmatrix} Bf & 0 & 0 \end{pmatrix}^T$$

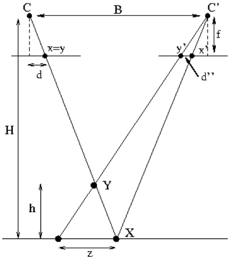
- If also $K = K'$,

$$z = fB/[(x - x') \cdot e_1] = fB/d$$

- d is the disparity

Triangulation

Fundamental principle of stereo vision



$$h \backsimeq \frac{z}{B/H}, \quad z = d'' \frac{H}{f}.$$

f focal length.

H distance optical center-ground.

B distance between optical centers (baseline).

Goal

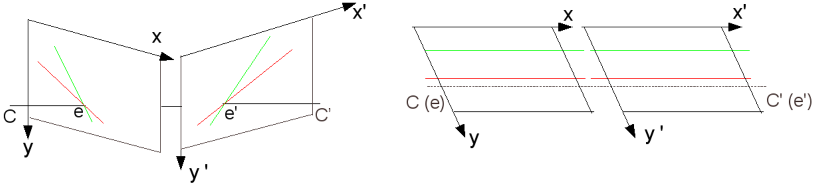
Given two rectified images, point correspondences and computation of their apparent shift (disparity) gives information about relative depth of the scene.

Reminder: Epipolar rectification

- It is convenient to get to a situation where epipolar lines are parallel and at same ordinate in both images.
- As a consequence, epipoles are at horizontal infinity:

$$e = e' = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

- It is always possible to get to that situation by virtual rotation of cameras (application of homography)



- Image planes coincide and are parallel to baseline.

Reminder: Epipolar rectification

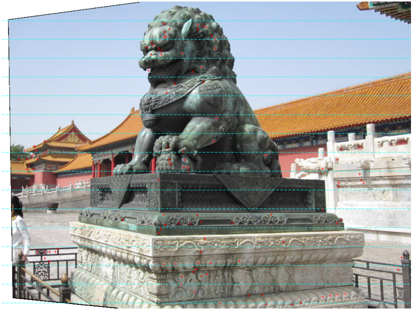
Image 1

Reminder: Epipolar rectification

Image 2

Reminder: Epipolar rectification

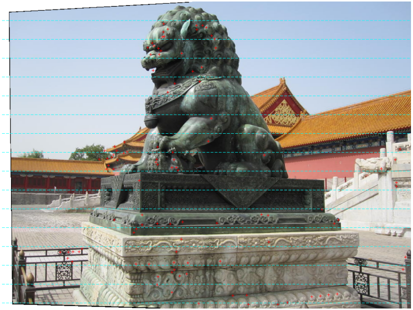
Image 1



Rectified image 1

Reminder: Epipolar rectification

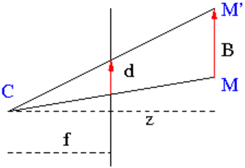
Image 2



Rectified image 2

Contents

Disparity map



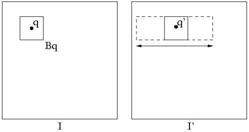
$$z = \frac{fB}{d}$$

Depth z is inversely proportional to disparity d (apparent motion, in pixels).

- Disparity map: At each pixel, its apparent motion between left and right images.
- We already know disparity at feature points, this gives an idea about min and max motion, which makes the search for matching points less ambiguous and faster.

Local search

- At each pixel, we consider a context window and we look for the motion of this window.



- Distance between windows:

$$d(q) = \arg\min_{d} \sum_{p \in F} (I(q+p) - I'(q+de_1+p))^2$$

- Variants to be more robust to illumination changes:
  1. Translate intensities by the mean over the window.

$$I(q+p) \to I(q+p) - \sum_{r \in F} I(q+r)/\#F$$

  2. Normalize by mean and variance over window.

Distance between patches

Several distances or similarity measures are popular:

- SAD: Sum of Absolute Differences

$$d(q) = \arg\min_d \sum_{p\in F} |I(q+p) - I'(q+de_1+p)|$$

- SSD: Sum of Squared Differences

$$d(q) = \arg\min_d \sum_{p\in F} (I(q+p) - I'(q+de_1+p))^2$$

- CSSD: Centered Sum of Squared Differences

$$d(q) = \arg\min_d \sum_{p\in F} (I(q+p) - \bar{I}_F - I'(q+de_1+p) + \bar{I}'_F)^2$$

- NCC: Normalized Cross-Correlation

$$d(q) = \arg\max_d \frac{\sum_{p\in F}(I(q+p) - \bar{I}_F)(I'(q+de_1+p) - \bar{I}'_F)}{\sqrt{\sum(I(q+p) - \bar{I}_F)^2}\sqrt{\sum(I'(q+de_1+p) - \bar{I}'_F)^2}}$$

Another distance

- The following distance is more and more popular in recent articles:

$$\epsilon(p,q) = (1-\alpha)\min\left(\|I(p)-I'(q)\|_1, \tau_{\text{col}}\right) + \alpha \min\left(|\frac{\partial I}{\partial x}(p) - \frac{\partial I'}{\partial x}(q)|, \tau_{\text{grad}}\right)$$

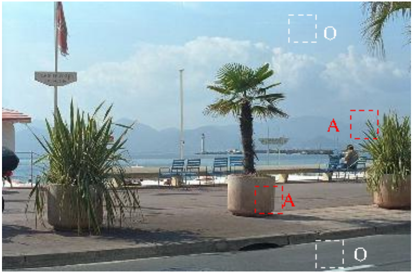
  with

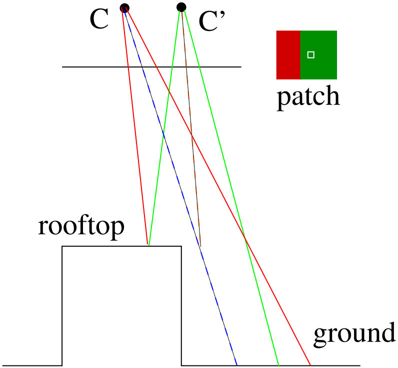
$$\|I(p)-I'(q)\|_1 = |I_r(p)-I_r(q)|+|I_g(p)-I_g(q)|+|I_b(p)-I_b(q)|$$

- Usual parameters:
  - $\alpha = 0.9$
  - $\tau_{\text{col}} = 30$ (not very sensitive if larger)
  - $\tau_{\text{grad}} = 2$ (not very sensitive if larger)
- Note that $\alpha = 0$ is similar to SAD.

Problems of local methods

- Implicit hypothesis: all points of window move with same motion, that is they are in a fronto-parallel plane.
- aperture problem: the context can be too small in certain regions, lack of information.
- adherence problem: intensity discontinuities influence strongly the estimated disparity and if it corresponds with a depth discontinuity, we have a tendency to dilate the front object.





- O: aperture problem
- A: adherence problem

Example: seeds expansion

- We rely on best found distances and we put them in a priority queue (seeds)
- We pop the best seed G from the queue, we compute for neighbors the best disparity between $d(G) - 1$, $d(G)$, and $d(G) + 1$ and we push them in the queue.

Right image

Example: seeds expansion

- We rely on best found distances and we put them in a priority queue (seeds)
- We pop the best seed G from the queue, we compute for neighbors the best disparity between $d(G)-1$, $d(G)$, and $d(G)+1$ and we push them in the queue.

Left image

Example: seeds expansion

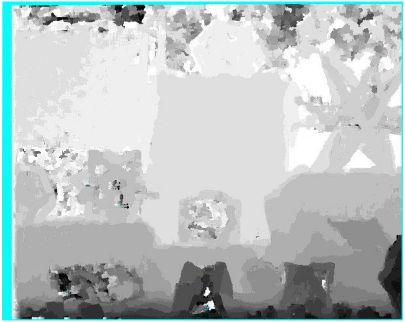
- We rely on best found distances and we put them in a priority queue (seeds)
- We pop the best seed G from the queue, we compute for neighbors the best disparity between $d(G) - 1$, $d(G)$, and $d(G) + 1$ and we push them in the queue.

Seeds

Example: seeds expansion

- We rely on best found distances and we put them in a priority queue (seeds)
- We pop the best seed G from the queue, we compute for neighbors the best disparity between $d(G) - 1$, $d(G)$, and $d(G) + 1$ and we push them in the queue.

Seeds expansion



Example: seeds expansion

- We rely on best found distances and we put them in a priority queue (seeds)
- We pop the best seed G from the queue, we compute for neighbors the best disparity between $d(G) - 1$, $d(G)$, and $d(G) + 1$ and we push them in the queue.

Left image

Adaptive neighborhoods

- To reduce adherence (aka fattening effect), an image patch should be on the same object, or even better at constant depth
- Heuristic inspired by bilateral filter [Yoon&Kweon 2006]:

$$\omega_I(p,p') = \exp\left(-\frac{\|p-p'\|_2}{\gamma_{\text{pos}}}\right) \cdot \exp\left(-\frac{\|I(p)-I(p')\|_1}{\gamma_{\text{col}}}\right)$$

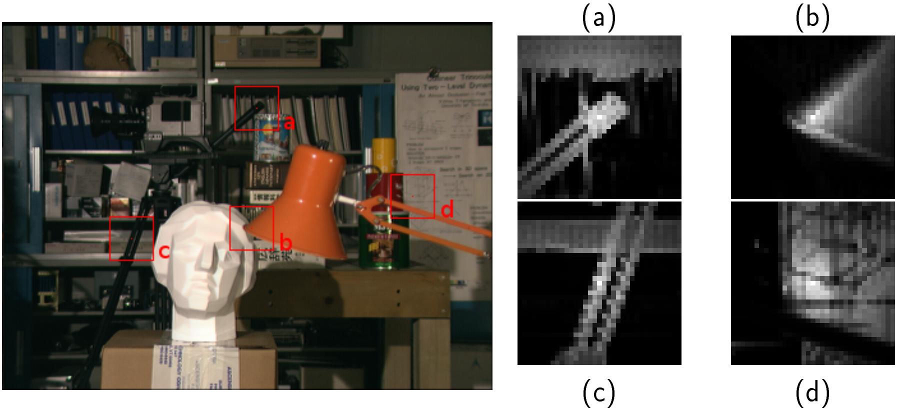
- Selected disparity:

$$d(p) = \arg\min_{d=q-p} E(p,q) \text{ with}$$

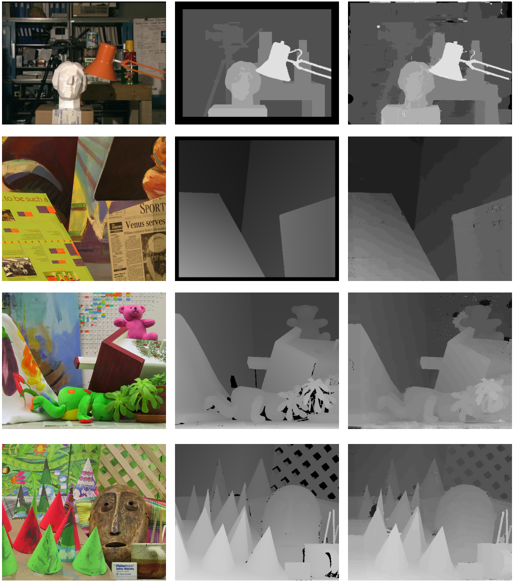
$$E(p,q) = \frac{\sum_{r\in F} \omega_I(p,p+r)\,\omega_{I'}(q,q+r)\,\epsilon(p+r,q+r)}{\sum_{r\in F} \omega_I(p,p+r)\,\omega_{I'}(q,q+r)}$$

- We can take a large window F (e.g., 35×35)

Bilateral weights



Results



Tsukuba

Venus

Teddy

Cones

Left image | Ground truth | Results

Global methods

- There is another category of methods: global methods
- Minimization of an energy:

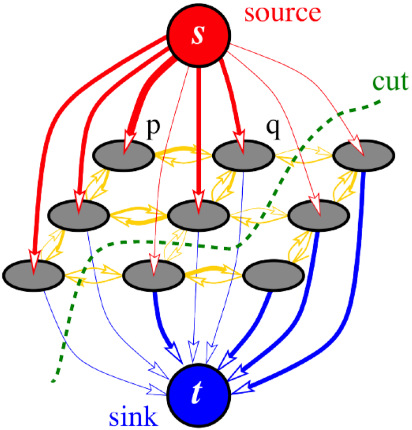
$$E(d(.)) = E_{\text{data}}(I_1(. + d), I_2(.)) + E_{\text{smooth}}(d(.))$$

- E_{data} related to data attachment: distance between colors or variant.
- E_{data} based on some *a priori* measure of regularity of the disparity map (e.g., total variation, or Potts measure)
- Difficulty: how to minimize such an energy (usually not convex...)
- See also optical flow methods: small displacement, recovery of dense apparent motion field.

Graph cuts

- Graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$
- Two special nodes s (source) and t (sink)
- Capacities $c : \mathcal{E} \to \mathbb{R}^+$
- Cut $C = (\mathcal{V}^s, \mathcal{V}^t)$ with $\mathcal{V} = \mathcal{V}^s \cup \mathcal{V}^t$ and $s \in \mathcal{V}^s$, $t \in \mathcal{V}^t$.
- Cost of C:

$$\sum_{e \in \mathcal{E} \cap (\mathcal{V}^s \times \mathcal{V}^t)} c(e)$$



from [Boykov&Veksler 2006]

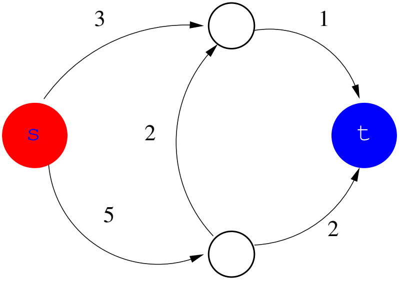
Maximum flow

- Flow: $f : \mathcal{E} \to \mathbb{R}^+$ such that

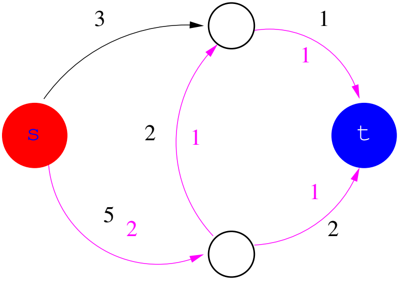
$$f \leq c \text{ and } \forall p \in \mathcal{V} \setminus \{s, t\}, \sum_{e=(q,p)\in\mathcal{E}} f(e) = \sum_{e=(p,q)\in\mathcal{E}} f(e)$$

- Value of flow:

$$\sum_{e=(s,p)\in\mathcal{E}} f(e) = \sum_{e=(p,t)\in\mathcal{E}} f(e)$$



(a)



(b)

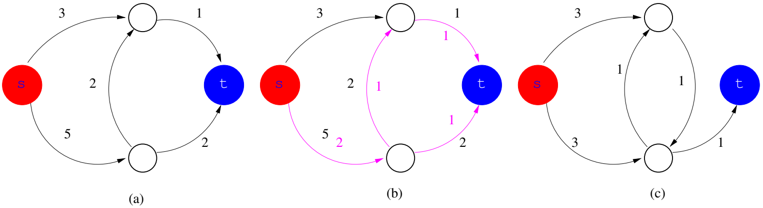
Min cut=max flow

Theorem

The minimum cut of a graph is equal to the value of its maximal flow

Proof: they are dual linear problems.
Algorithm (Ford-Fulkerson):

- Find path from s to t with positive weights in residual graph



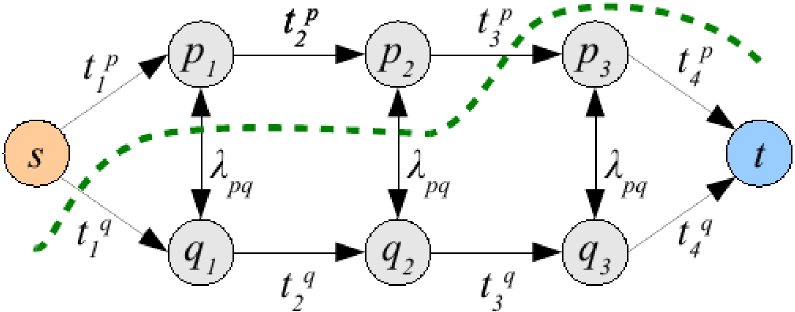
- If it exists, push the flow that saturates one edge of the path
- Iterate until no more path.

Disparity map estimation

- Put labels to pixels: $f : \mathcal{P} \to \mathcal{L} = \{1, \cdots, k\}$
- Minimize energy

$$E = \sum_{p \in \mathcal{P}} D_p(f(p)) + \sum_{(p,q) \in \mathcal{N}} \lambda_{pq} |f(p) - f(q)|$$

- $k - 1$ nodes for pixel p: $p_1 \cdots p_{k-1}$, $p_0 = s$, $p_k = t$
- $c(p_{i-1}, p_i) = D_p(i) + L$ with L large
- $c(p_i, q_i) = \lambda_{pq}$ if $(p, q) \in \mathcal{N}$
- Exact minimization [Boykov&Veksler&Zabih 1998]



Other smoothness terms

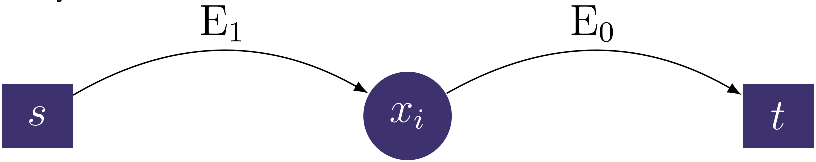
- Only approximate minimizations for other smoothness terms
- For example: $V_{pq} = \mathbf{1}(f(p) = f(q))$
- Algorithm: α-expansions
- f' α-expansion move of f iff

$$\forall p,\ f'(p) \in \{f(p), \alpha\}$$

Which binary energies can be minimized?

- $E(x_1, \ldots, x_n) = \sum E^i(x_i) + \sum_{ij} E^{ij}(x_i, x_j)$
- Only *some* of these energies can be minimized by graph cuts
- Constraints on the binary terms E^{ij}

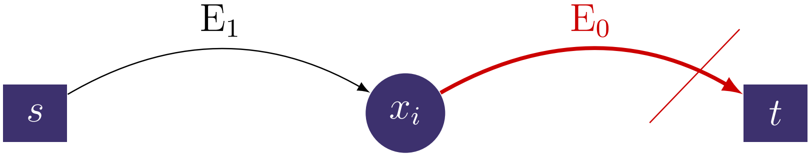
Unary terms:



Which binary energies can be minimized?

- $E(x_1, \ldots, x_n) = \sum E^i(x_i) + \sum_{ij} E^{ij}(x_i, x_j)$
- Only *some* of these energies can be minimized by graph cuts
- Constraints on the binary terms E^{ij}

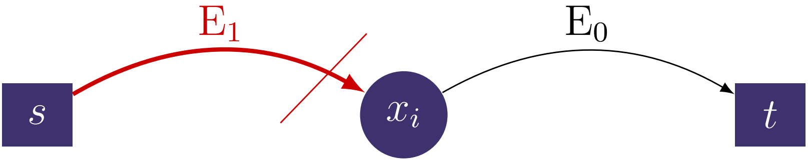
Unary terms:



Which binary energies can be minimized?

- $E(x_1, \ldots, x_n) = \sum E^i(x_i) + \sum_{ij} E^{ij}(x_i, x_j)$
- Only *some* of these energies can be minimized by graph cuts
- Constraints on the binary terms E^{ij}

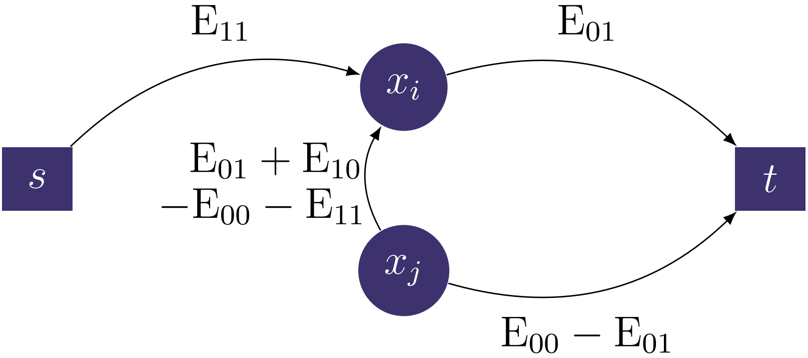
Unary terms:



Which binary energies can be minimized?

- $E(x_1, \ldots, x_n) = \sum E^i(x_i) + \sum_{ij} E^{ij}(x_i, x_j)$
- Only *some* of these energies can be minimized by graph cuts
- Constraints on the binary terms E^{ij}

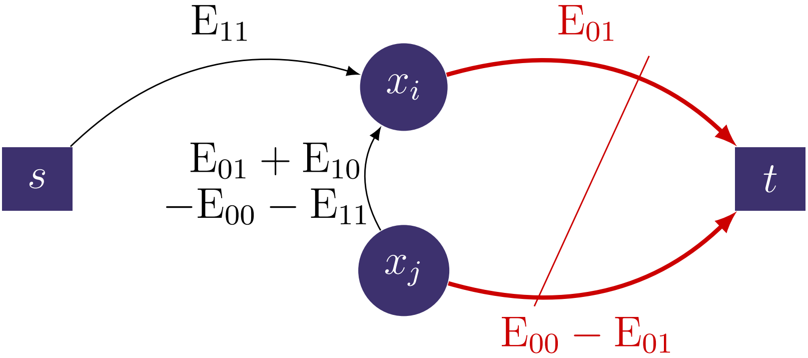
Binary terms:



Which binary energies can be minimized?

- $E(x_1, \ldots, x_n) = \sum E^i(x_i) + \sum_{ij} E^{ij}(x_i, x_j)$
- Only *some* of these energies can be minimized by graph cuts
- Constraints on the binary terms E^{ij}

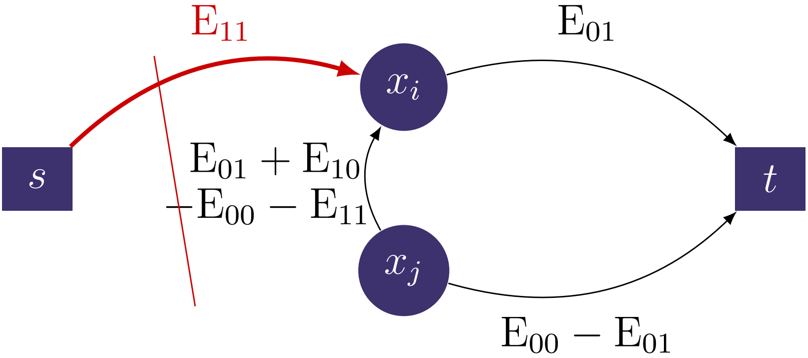
Binary terms:



Which binary energies can be minimized?

- $E(x_1, \ldots, x_n) = \sum E^i(x_i) + \sum_{ij} E^{ij}(x_i, x_j)$
- Only *some* of these energies can be minimized by graph cuts
- Constraints on the binary terms E^{ij}

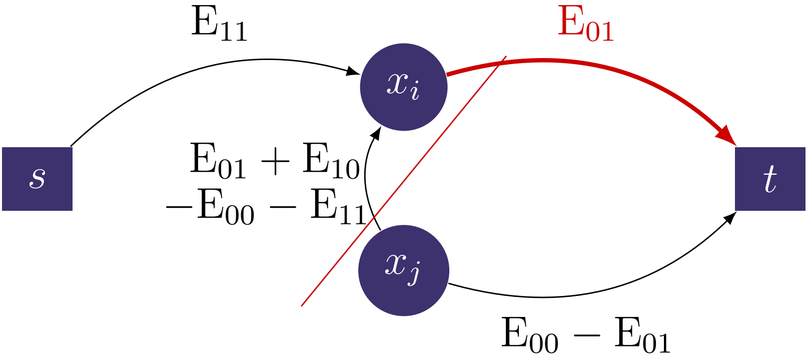
Binary terms:



Which binary energies can be minimized?

- $E(x_1, \ldots, x_n) = \sum E^i(x_i) + \sum_{ij} E^{ij}(x_i, x_j)$
- Only *some* of these energies can be minimized by graph cuts
- Constraints on the binary terms E^{ij}

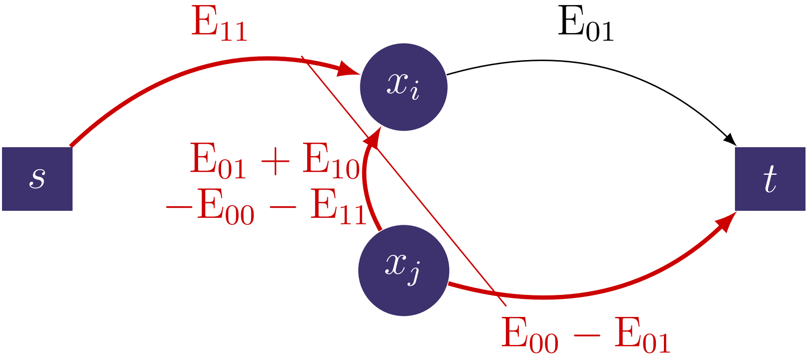
Binary terms:



Which binary energies can be minimized?

- $E(x_1, \ldots, x_n) = \sum E^i(x_i) + \sum_{ij} E^{ij}(x_i, x_j)$
- Only *some* of these energies can be minimized by graph cuts
- Constraints on the binary terms E^{ij}

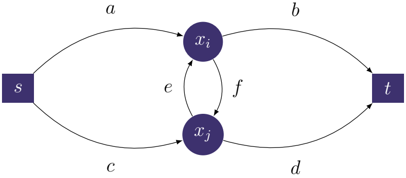
Binary terms:



Which binary energies can be minimized?

- $E(x_1, \ldots, x_n) = \sum E^i(x_i) + \sum_{ij} E^{ij}(x_i, x_j)$
- Only *some* of these energies can be minimized by graph cuts
- Constraints on the binary terms E^{ij}

Submodularity:



$$
\begin{aligned}
a + c + C &= \mathrm{E}^{ij}(1,1) \\
b + d + C &= \mathrm{E}^{ij}(0,0) \\
a + d + e + C &= \mathrm{E}^{ij}(1,0) \quad (1) \\
b + c + f + C &= \mathrm{E}^{ij}(0,1). \quad (2)
\end{aligned}
$$

Adding equalities (1) and (2) yields

$$
\begin{aligned}
\mathrm{E}^{ij}(0,1) + \mathrm{E}^{ij}(1,0) &= a + b + c + d + e + f + 2C \\
&= (a + c + C) + (b + d + C) + e + f \\
&= \mathrm{E}^{ij}(0,0) + \mathrm{E}^{ij}(1,1) + e + f \\
\mathrm{E}^{ij}(0,0) + \mathrm{E}^{ij}(1,1) &\leq \mathrm{E}^{ij}(0,1) + \mathrm{E}^{ij}(1,0),
\end{aligned}
$$

since $e + f$ is nonnegative.

Some examples

With a fairly more complex method that ensures left-right consistency and detects occlusions:

V. Kolmogorov, P. Monasse, P. Tan (2014)
Kolmogorov and Zabih's Graph Cuts Stereo Matching Algorithm
preprint Image Processing On Line (IPOL)
`http://demo.ipol.im/demo/97/`

Contents

Multi-linear constraints

- Bilinear constraints: fundamental matrix $x^T F x' = 0$.
- There are trilinear constraints: $x_i'' = x^T T_i x'$, which are *not* combinations of bilinear contraints
- All constraints involving more than 3 views are combinations of 2- and/or 3-view constraints.

Trilinear constraints

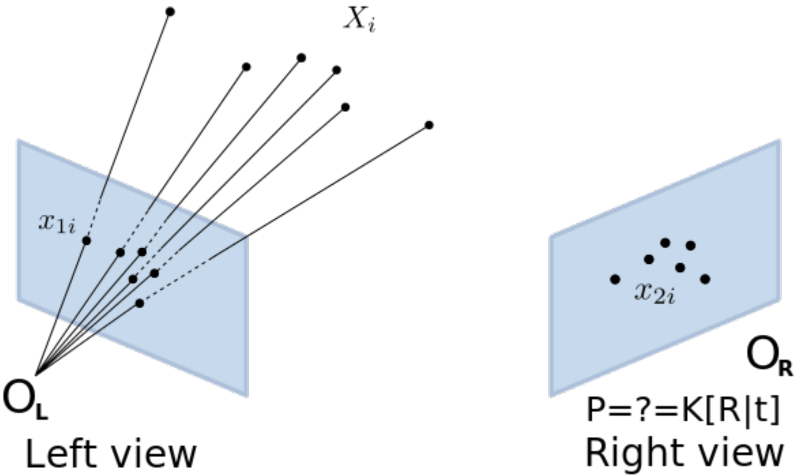
- Write $\lambda_i x_i = K_i \begin{pmatrix} R_i & T_i \end{pmatrix} X$
- Write as $AY = 0$ with $Y = \begin{pmatrix} X & \lambda_1 & \cdots & \lambda_n \end{pmatrix}^T$
- Look at the rank of A...

Incremental multi-view calibration

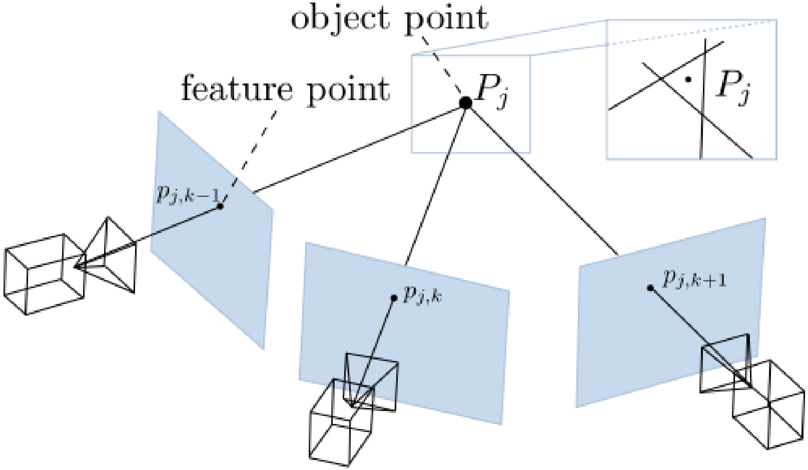
1. Compute two-view correspondences
2. Build tracks (multi-view correspondences)
3. Start from initial pair: compute F, deduce R, T and 3D points (known K)
4. Add image with common points.
5. Estimate pose (R, T)
6. Add new 3D points
7. Bundle adjustment
8. Go to 4

see open source software Bundler: SfM for Unordered Image Collections
`http://www.cs.cornell.edu/~snavely/bundler/`

Incremental multi-view calibration



Incremental multi-view calibration



Conclusion

- We can get back to the canonical situation by epipolar rectification. Limit: when epipoles are in the image, standard methods are not adapted.
- For disparity map computation, there are many choices:
  1. Size and shape of window?
  2. Which distance?
  3. Filtering of disparity map to reject uncertain disparities?
- Very active domain of research, >150 methods tested at `http://vision.middlebury.edu/stereo/`