Computer Vision: Algorithms and Applications

Stereo Correspondence

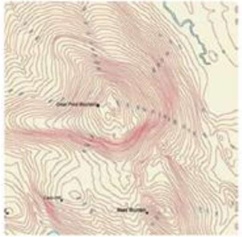
Jing Luo | Megvii Tech Talk | Apr 2018

Reference: R. Szeliski. *Computer Vision: Algorithms and Applications*. 2010. 1.

1.
Introduction to Stereo Vision

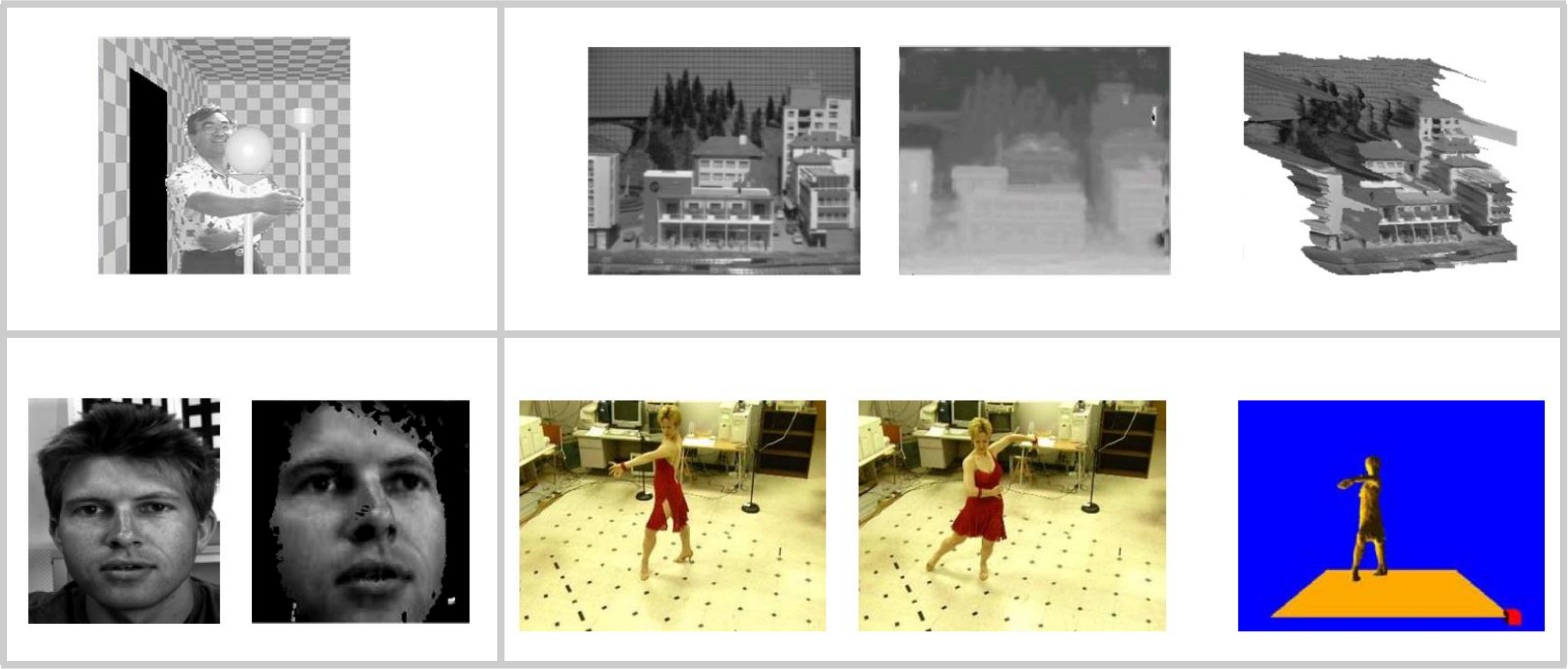
Introduction

- What is stereo vision?
  - The word "stereo" comes from the Greek for "solid"
  - Stereo vision: how we perceive solid shape
- Stereo matching
  - Take two or more images and estimate a 3D model of the scene by finding matching pixels in the images and converting their 2D positions into 3D depths.
- Application
  - Photogrammetric matching of aerial images
  - Modeling of the human visual system
  - Robotic navigation and manipulation
  - View interpolation and image-based rendering
  - 3D model building



Introduction

Introduction



2.
Epipolar Geometry

Two-frame structure from motion

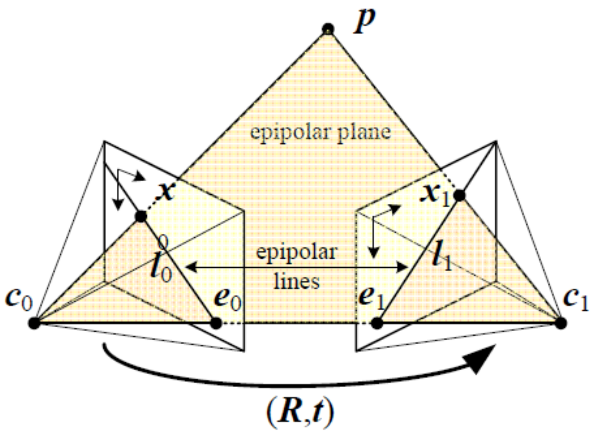
- **3D rotation**
  - Also known as 3D rigid body motion or the 3D Euclidean transformation, it can be written as

    $$x' = Rx + t \quad \text{or} \quad x' = [R\ t]\bar{x}$$

    R is a 3×3 orthonormal rotation matrix with $RR^T = I$ and $|R| = 1$.
- **Epipolar geometry**

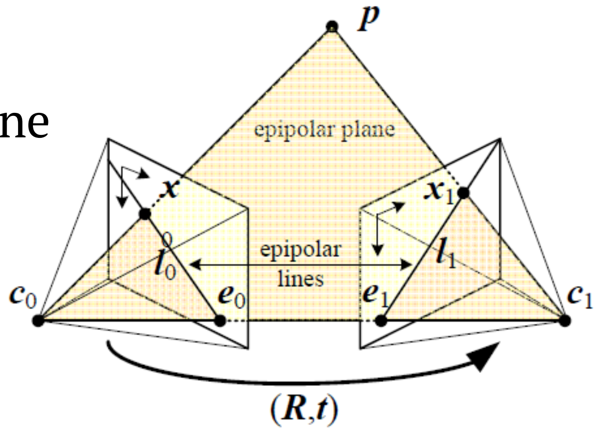
  $$d_1\hat{x}_1 = p_1 = Rp_0 + t = R(d_0\hat{x}_0) + t$$
  $$d_1[t]_\times\hat{x}_1 = d_0[t]_\times R\hat{x}_0$$
  $$d_0\hat{x}_1^{\mathrm{T}}[t]_\times R\hat{x}_0 = d_1\hat{x}_1^{\mathrm{T}}[t]_\times\hat{x}_1 = 0$$
- **Epipolar constraint**

  $\hat{x}_1^{\mathrm{T}}\mathrm{E}\hat{x}_0 = 0$, where $\mathrm{E} = [t]_\times R$ is the essential matrix.



Two-frame structure from motion

- Another perspective:
  - Epipolars: e_0 e_1
  - Epipolar plane: c_0 c_1 and p define a plane
  - Epipolar line: Intersections of epipolar plane with the image planes
  - Epipolar constraint: Corresponding points on conjugate epipolar lines



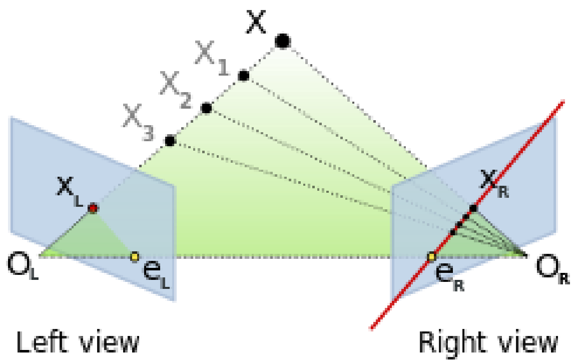
$$(\hat{\boldsymbol{x}}_0, \boldsymbol{R}^{-1}\hat{\boldsymbol{x}}_1, -\boldsymbol{R}^{-1}\boldsymbol{t}) = (\boldsymbol{R}\hat{\boldsymbol{x}}_0, \hat{\boldsymbol{x}}_1, -\boldsymbol{t}) = \hat{\boldsymbol{x}}_1 \cdot (\boldsymbol{t} \times \boldsymbol{R}\hat{\boldsymbol{x}}_0) = \hat{\boldsymbol{x}}_1^T([\boldsymbol{t}]_\times \boldsymbol{R})\hat{\boldsymbol{x}}_0 = 0$$

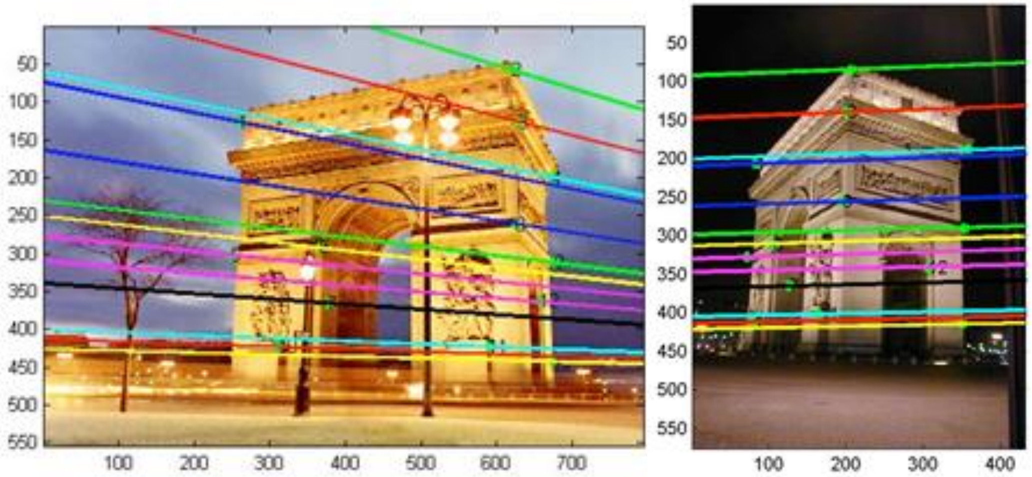
$$\hat{\boldsymbol{x}}_1^T \boldsymbol{l}_1 = 0$$

$\hat{x}_0$ in image 0 $\xrightarrow{l_1 = E\hat{x}_0}$ l_1 in image 1

Two-frame structure from motion

- $\hat{x}_0$ in image 0 $\xrightarrow{l_1 = E\hat{x}_0}$ l_1 in image 1





Two-frame structure from motion

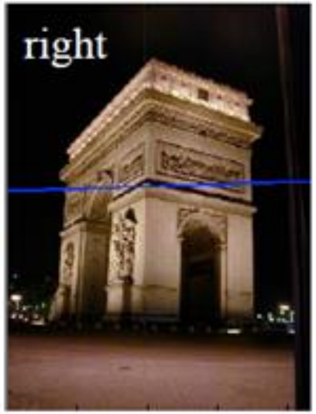
$$F = \begin{pmatrix} -0.00310695 & -0.0025646 & 2.96584 \\ -0.028094 & -0.00771621 & 56.3813 \\ 13.1905 & -29.2007 & -9999.79 \end{pmatrix} \begin{pmatrix} 343.53 \\ 221.70 \\ 1.0 \end{pmatrix}$$

x = 343.5300 y = 221.7005

$$\begin{matrix} 0.0001 \\ 0.0045 \\ -1.1942 \end{matrix} \rightarrow \begin{matrix} 0.0295 \\ 0.9996 \\ -265.1531 \end{matrix}$$

normalize so sum of squares
of first two terms is 1 (optional)

$$\begin{matrix} 0.0295 \\ 0.9996 \\ -265.1531 \end{matrix}$$



Two-frame structure from motion

$$(\,205.5526\quad 80.5\quad 1.0)\begin{pmatrix} -0.00310695 & -0.0025646 & 2.96584 \\ -0.028094 & -0.00771621 & 56.3813 \\ 13.1905 & -29.2007 & -9999.79 \end{pmatrix}$$

L= (0.3211   -0.9470 -151.39)

L = (0.0010   -0.0030   -0.4851)

→ (0.3211   -0.9470 -151.39)

x = 205.5526   y = 80.5000

Two-frame structure from motion

- How to calculate essential matrix?

$$\hat{\boldsymbol{x}}_1^T \boldsymbol{E}\, \hat{\boldsymbol{x}}_0 = 0$$

$$\begin{array}{ccccccc} x_{i0}x_{i1}e_{00} & + & y_{i0}x_{i1}e_{01} & + & x_{i1}e_{02} & + & \\ x_{i0}y_{i1}e_{00} & + & y_{i0}y_{i1}e_{11} & + & y_{i1}e_{12} & + & \\ x_{i0}e_{20} & + & y_{i0}e_{21} & + & e_{22} & = & 0 \end{array}$$

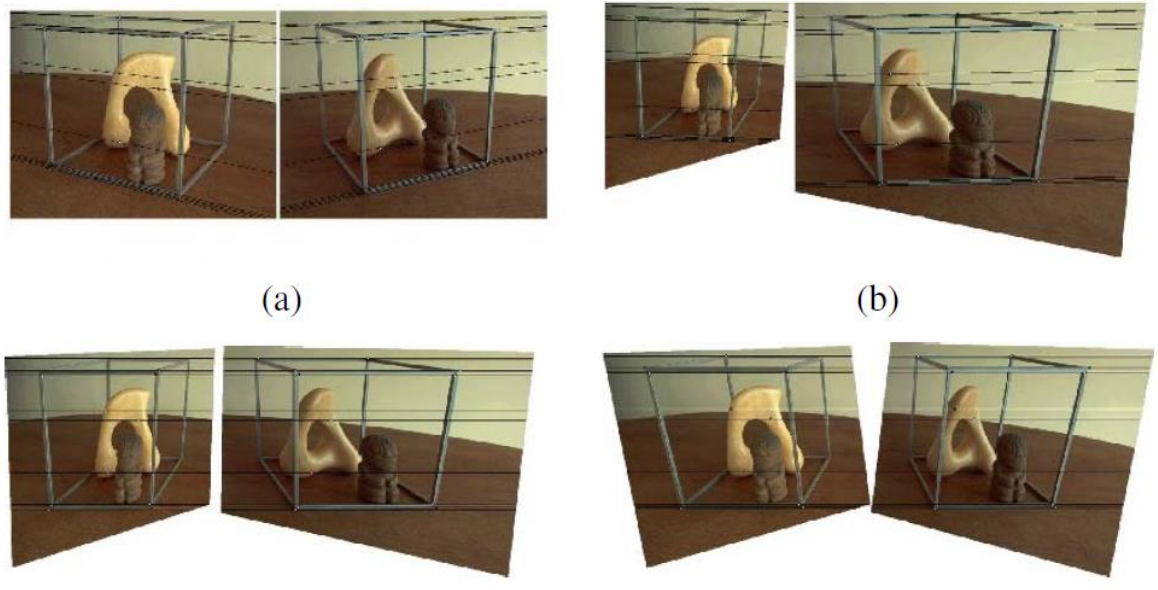
  - Method 1: SVD with more than eight equations
  - Method 2: make use of the condition that E is rank-deficient

$$\boldsymbol{E} = \alpha \boldsymbol{E}_0 + (1-\alpha)\boldsymbol{E}_1$$

$$\det |\alpha \boldsymbol{E}_0 + (1-\alpha)\boldsymbol{E}_1| = 0$$

Rectification

- Rectifying (i.e, warping) the input images so that corresponding horizontal scanlines are epipolar lines



(a) (b)

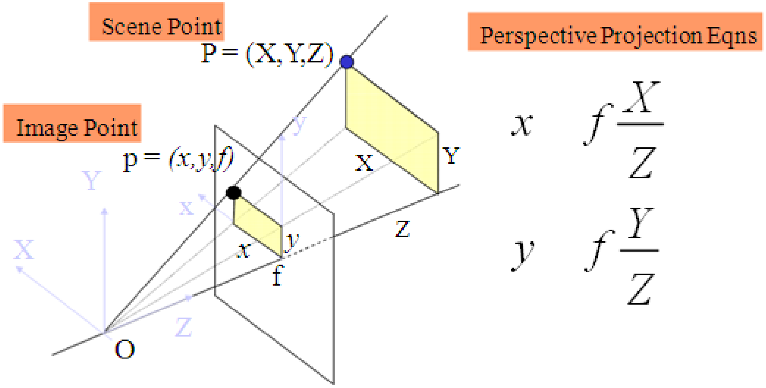
(c) (d)

Rectification

- After rectification:

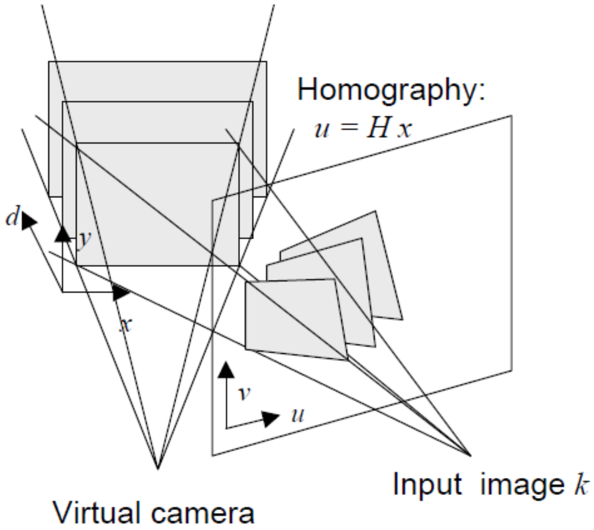
$$d = f\frac{B}{Z}$$

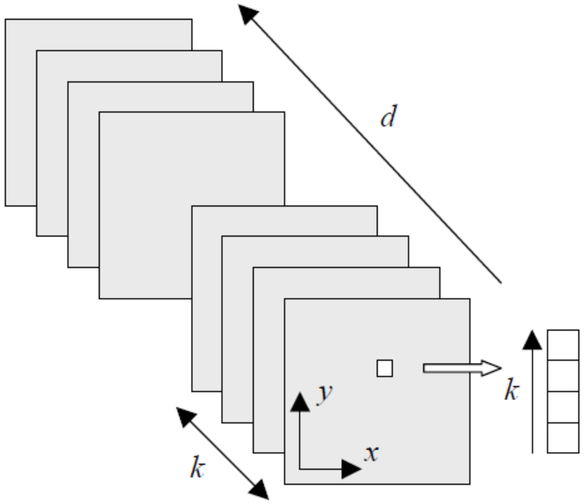
$$x' = x + d(x,y), \quad y' = y$$



Plane sweep

- Sweeping a set of planes through a scene:

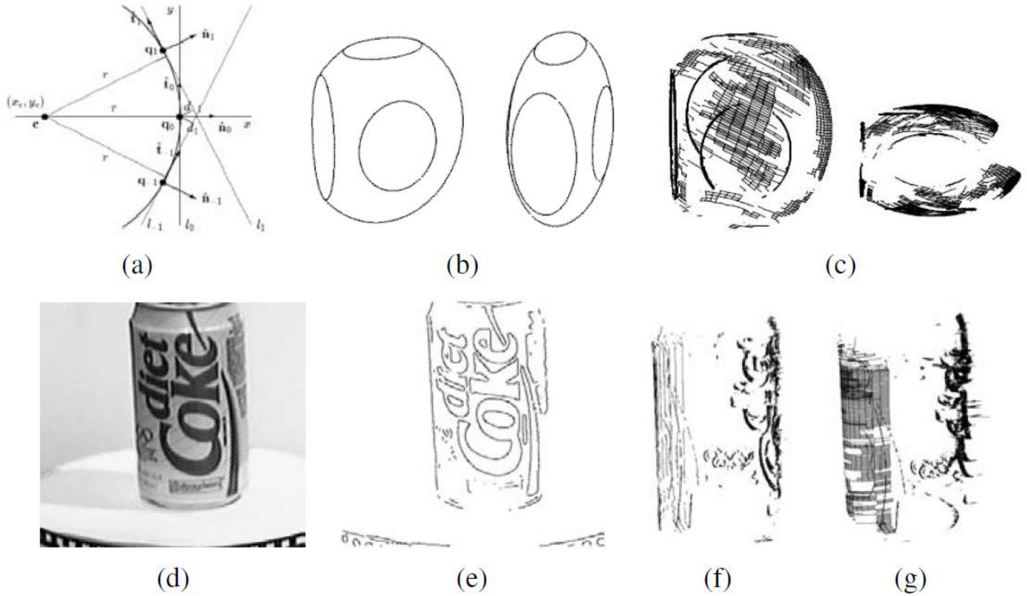




3.
Sparse Correspondence

3D curves and profiles

- Surface reconstruction from occluding contours



(a) (b) (c)

(d) (e) (f) (g)

4.
Dense Correspondence

Dense correspondence algorithms

- 4 steps:
  - 1. matching cost computation;
  - 2. cost (support) aggregation;
  - 3. disparity computation and optimization;
  - 4. disparity refinement.
- Local algorithm
  - use a matching cost that is based on a support region
- Global algorithm
  - make explicit smoothness assumptions and then solve a global optimization problem

Similarity measures

- Sum-of-squared difference technique
  - SSD is the template matching method done by finding the lowest difference value between input and template. The differences are squared in order to remove the sign.

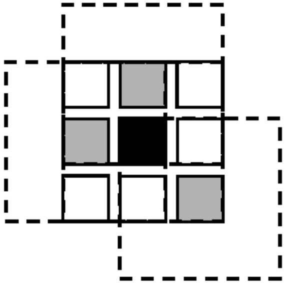
$$SSD(\vec{p},\vec{d}) = \sum_{j=-N/2}^{N/2} \sum_{i=-N/2}^{N/2} (I_1(x+i,y+j) - I_2(x+i,y+j))^2$$

- Other methods
  - Normalized correlation coefficients
  - Mutual information
  - Normalized gradient field

Local methods

- Local and window-based methods aggregate the matching cost by summing or averaging over a support region.
  - support region can be either two-dimensional at a fixed disparity (favoring fronto-parallel surfaces), or three-dimensional in x-y-d space (supporting slanted surfaces).
- Aggregation with a fixed support region can be performed using 2D or 3D convolution.

$$C(x, y, d) = w(x, y, d) * C_0(x, y, d)$$

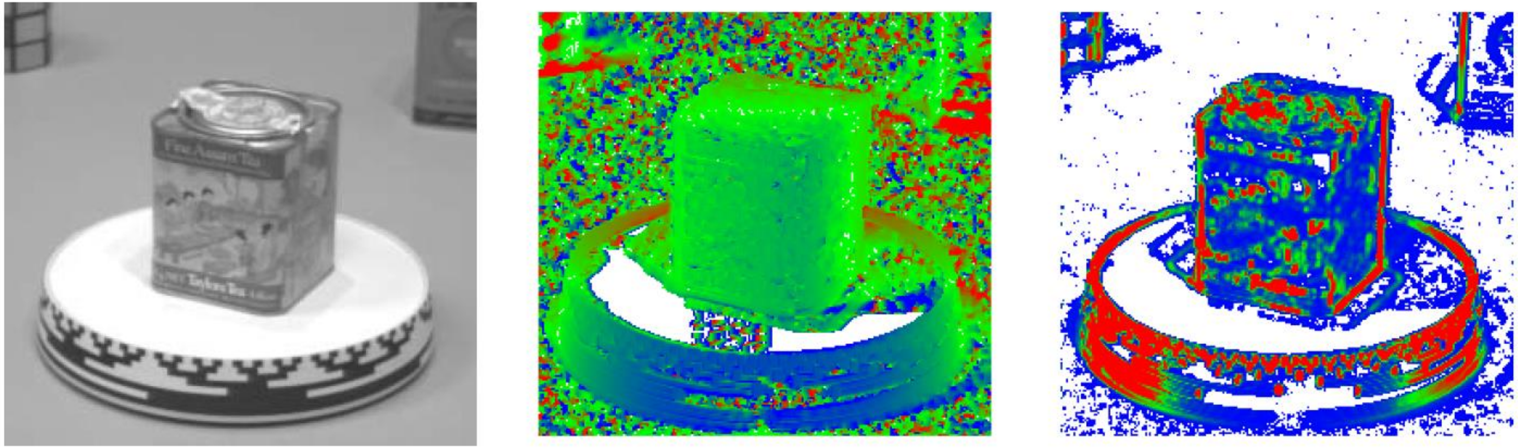


Local methods

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

Aggregation window sizes and weights adapted to image content (Tombari, Mattoccia, Di Stefano *et al.* 2008) © 2008 IEEE: (a) original image with selected evaluation points; (b) variable windows (Veksler 2003); (c) adaptive weights (Yoon and Kweon 2006); (d) segmentation-based (Tombari, Mattoccia, and Di Stefano 2007). Notice how the adaptive weights and segmentation-based techniques adapt their support to similarly colored pixels.

Local methods



(a) (b) (c)

Uncertainty in stereo depth estimation (Szeliski 1991b): (a) input image; (b) estimated depth map (blue is closer); (c) estimated confidence(red is higher). As you can see, more textured areas have higher confidence.

Global optimization

- Many global methods are formulated in an energy-minimization framework.
  - the objective is to find a solution d that minimizes a global energy

$$E(d) = E_d(d) + \lambda E_s(d)$$

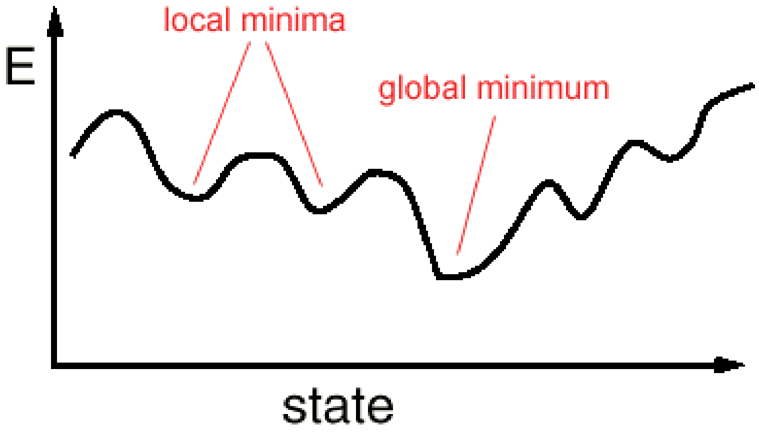
$$E_d(d) = \sum_{(x,y)} C(x, y, d(x, y))$$

$$E_s(d) = \sum_{(x,y)} \rho(d(x, y) - d(x+1, y)) + \rho(d(x, y) - d(x, y+1))$$

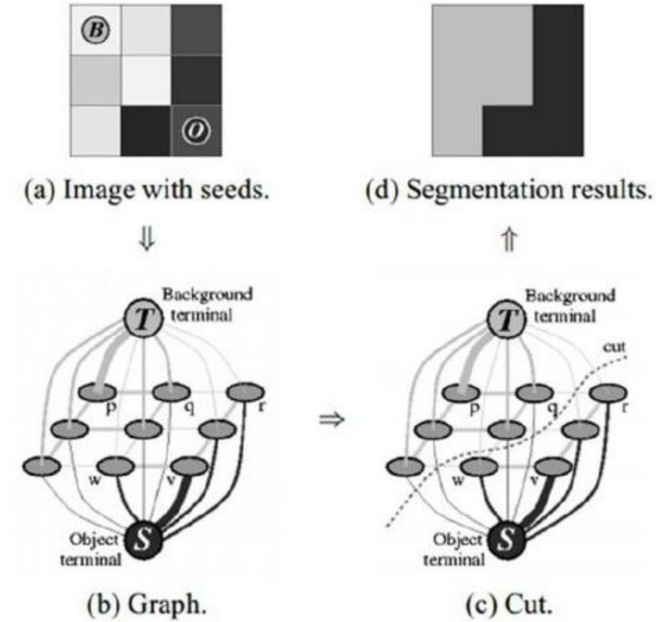
$$\rho_d(d(x, y) - d(x+1, y)) \cdot \rho_I(\|I(x, y) - I(x+1, y)\|)$$

Global optimization

- Simulated annealing

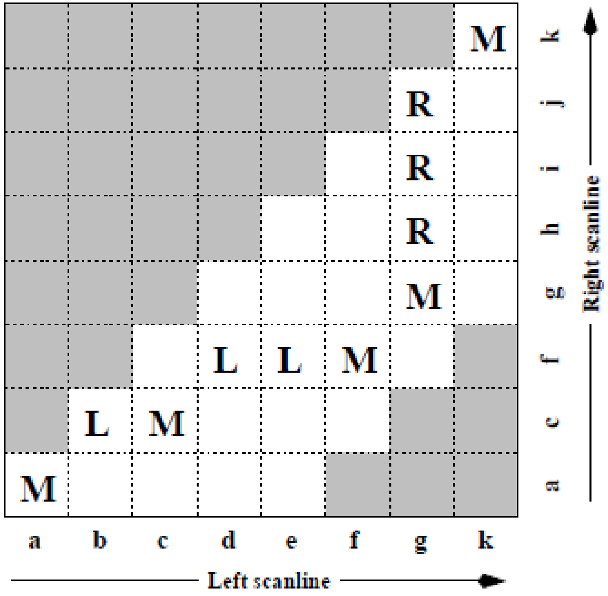


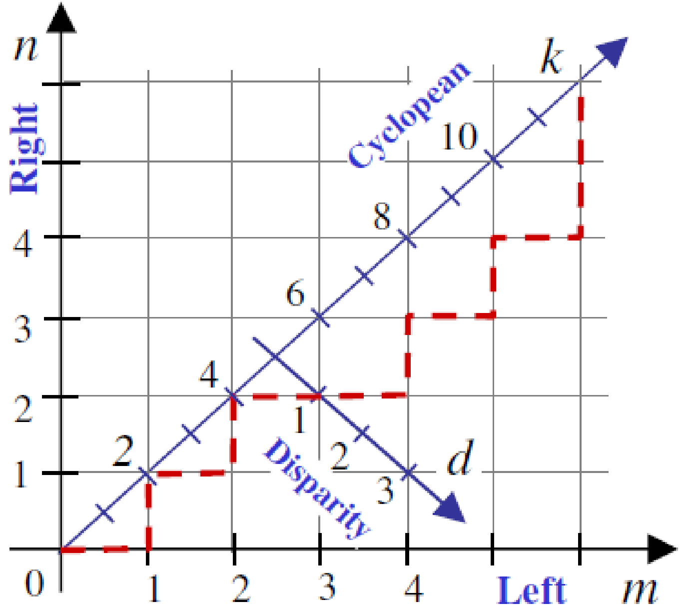
- Max-flow / Graph cut



Global optimization

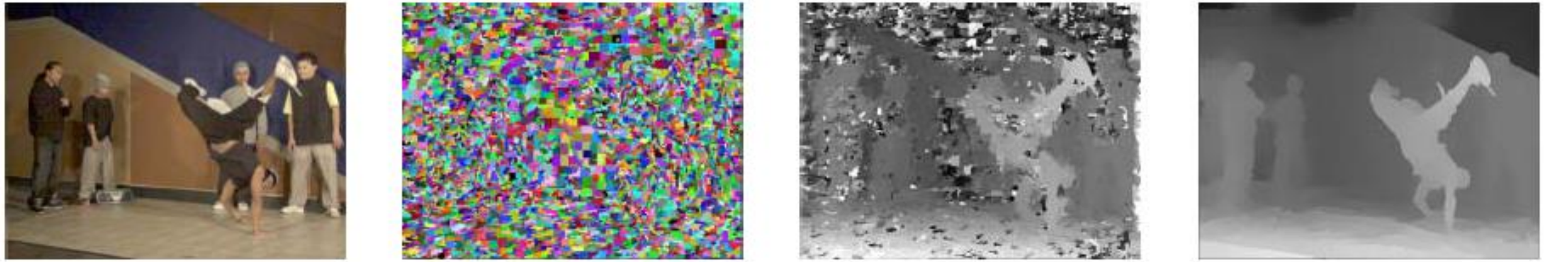
- Dynamic programming

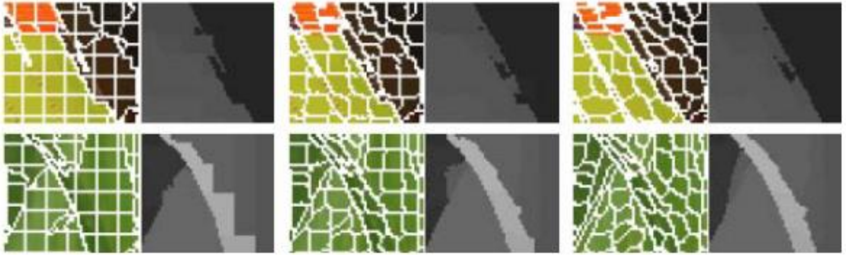




Global optimization

- Segmentation-based techniques





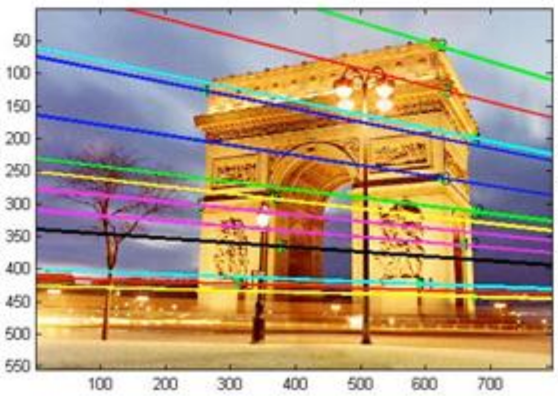
Global optimization

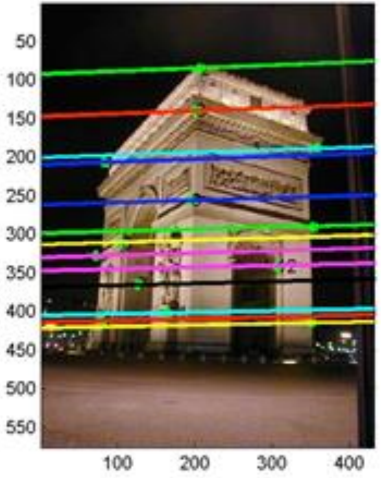
- Z-keying and background replacement

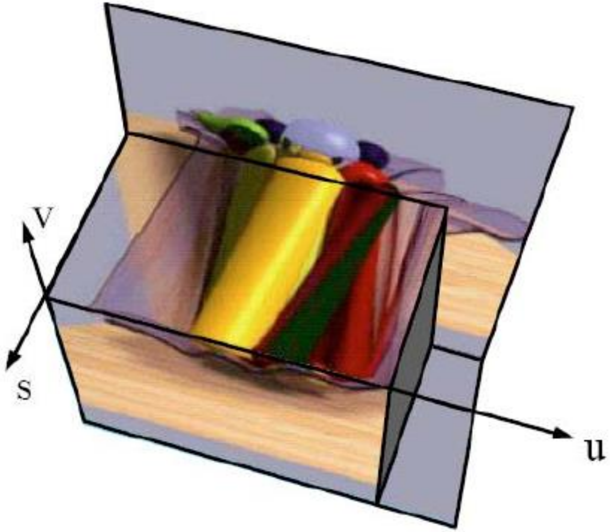
5.
Multi-view stereo

Epipolar plane

- Epipolar plane image

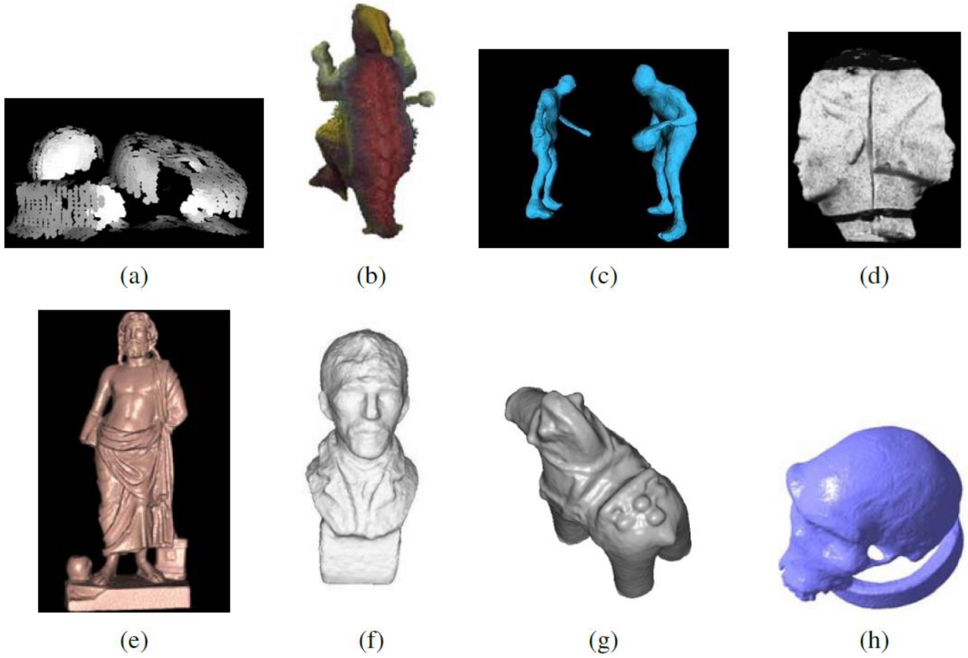






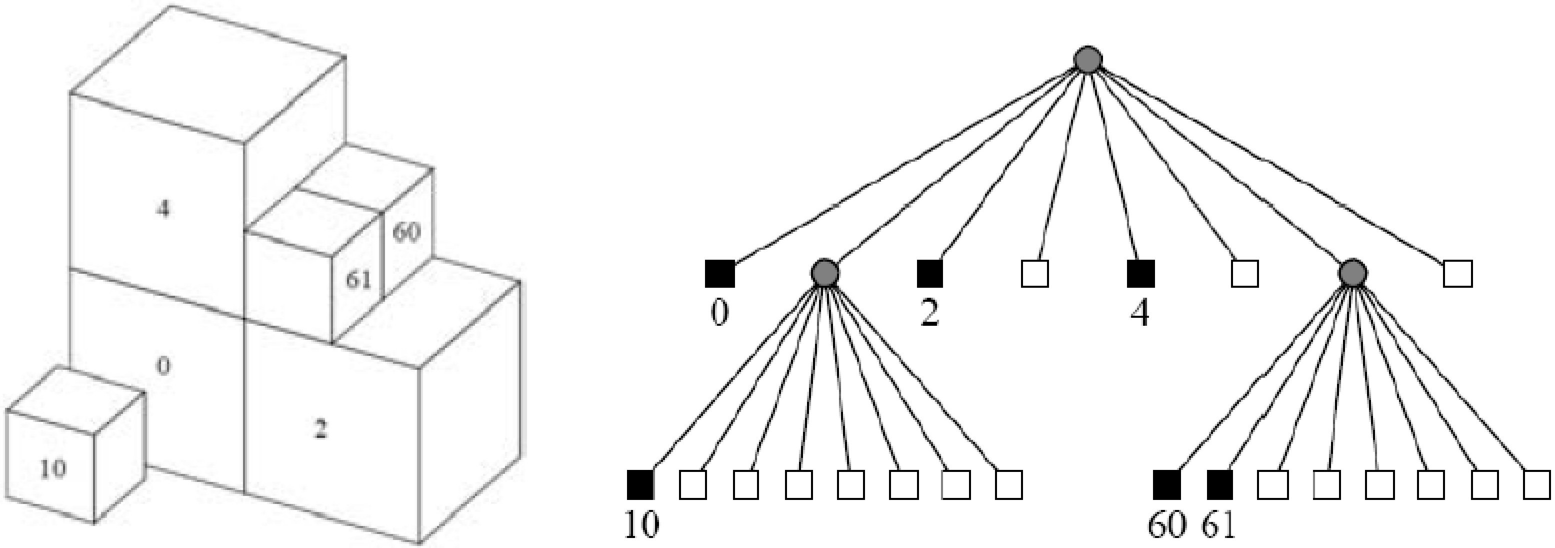
3D reconstruction

- Volumetric and 3D surface reconstruction



3D reconstruction

- Shape from silhouettes



3D reconstruction

- Shape from silhouettes