Multi-view 3D reconstruction

- Problem formulation
- Projective ambiguity
- Rectification
- Autocalibration
- Feature points and their matching

Problem

- Given m images of n scene points captured from different viewpoints, we want to estimate the 3D coordinates of the n points and the camera matrices of the m views

Geometric model

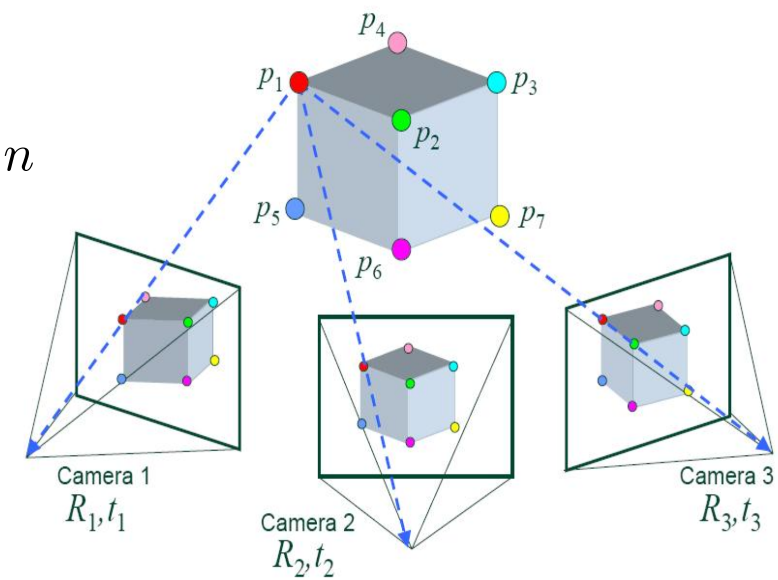
$$\mathbf{x}_j^{(i)} \propto \mathtt{P}_i \mathbf{X}_j$$

Input

$$\mathbf{x}_j^{(i)} = \begin{bmatrix} x_j^{(i)} & y_j^{(i)} & 1 \end{bmatrix}^\top$$

$$i = 1, \ldots, m, \quad j = 1, \ldots, n$$

$$2mn$$



Output

$$\mathbf{X}_j = \begin{bmatrix} X_j & Y_j & Z_j & 1 \end{bmatrix}^\top$$

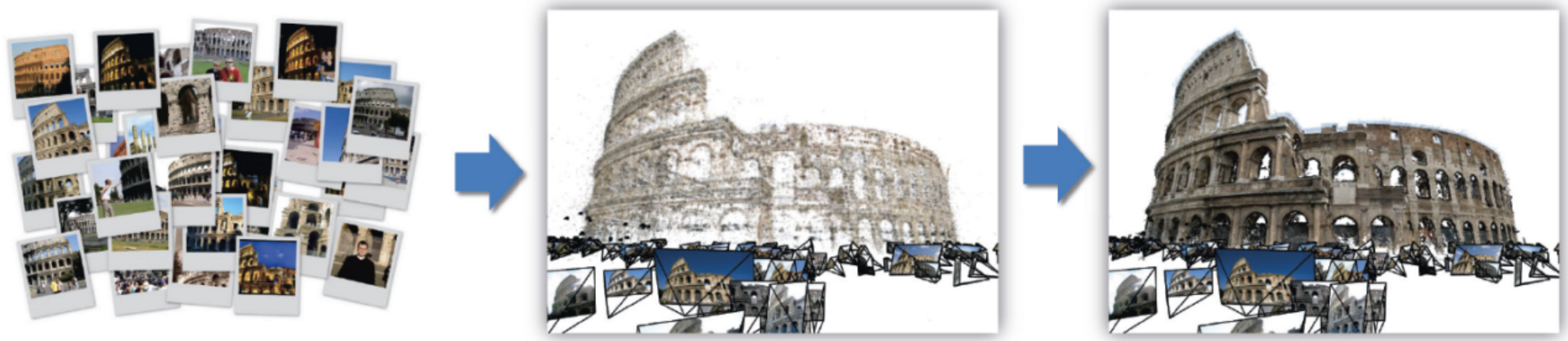
$$3n$$

$$\mathtt{P}_i = \mathtt{K}_i \begin{bmatrix} \mathtt{R}_i & \mathbf{t}_i \end{bmatrix}$$

$$11m$$

Applications

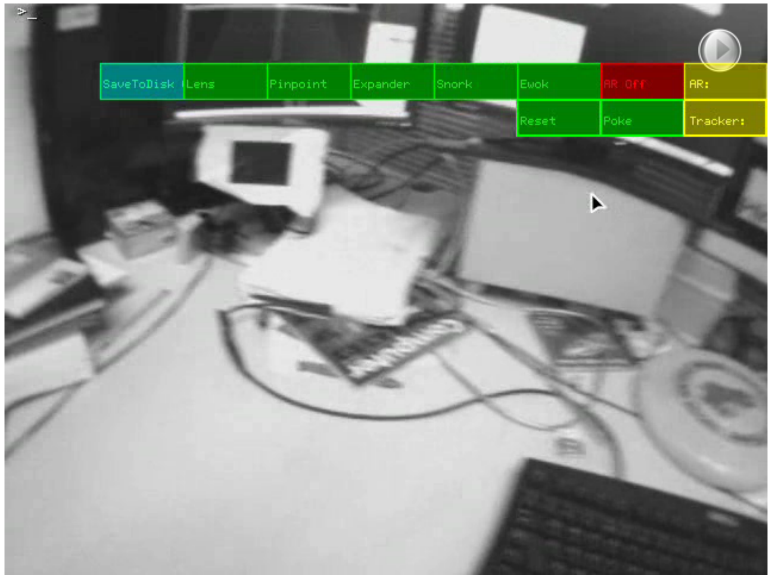
3D modeling from unsorted images [Snavely+04, Agarwal+10]



Input photos — Sparse reconstruction — Dense reconstruction

Autodesk 123D Catch

PTAM [Klein+07]

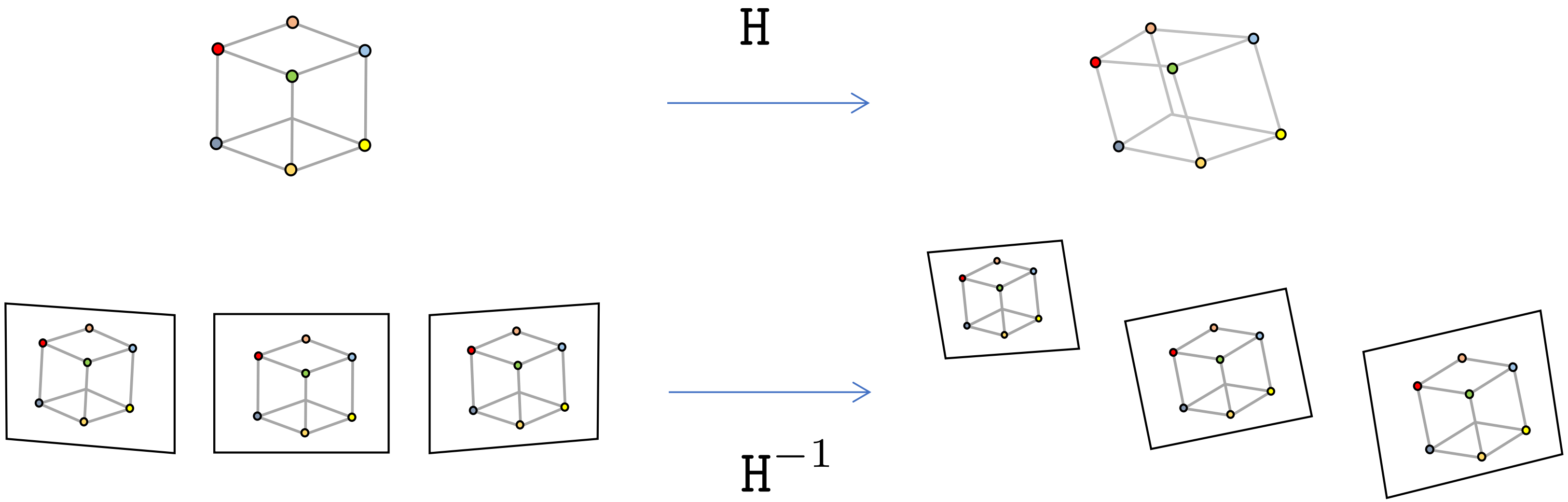


Projective ambiguity

- Solutions are ambiguous
  - Images alone cannot resolve this ambiguity

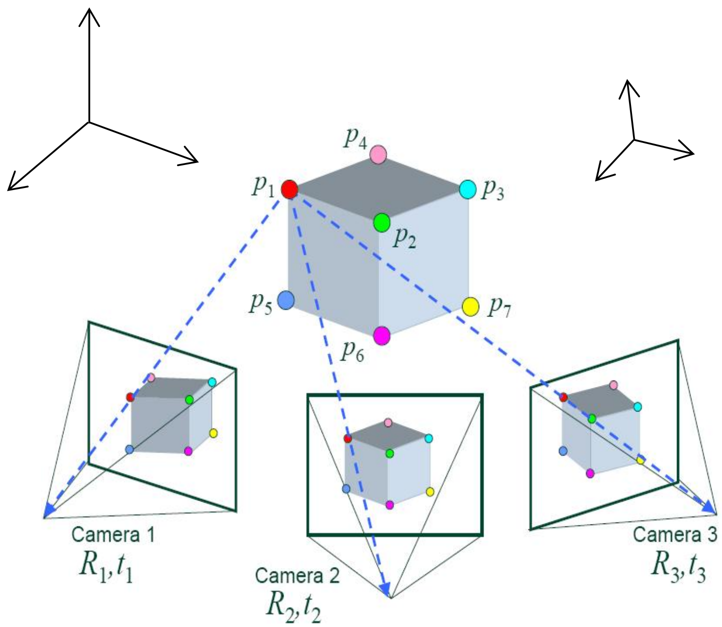
$$\mathbf{x}_j^{(i)} \propto \mathtt{P}_i\mathbf{X}_j = \mathtt{P}_i\mathtt{H}^{-1}\mathtt{H}\mathbf{X}_j = (\mathtt{P}_i\mathtt{H}^{-1})(\mathtt{H}\mathbf{X}_j) = \mathtt{P}'_i\mathtt{X}'_j$$

  - There is ambiguity of 15 DoFs, which corresponds to 3D projective transformation

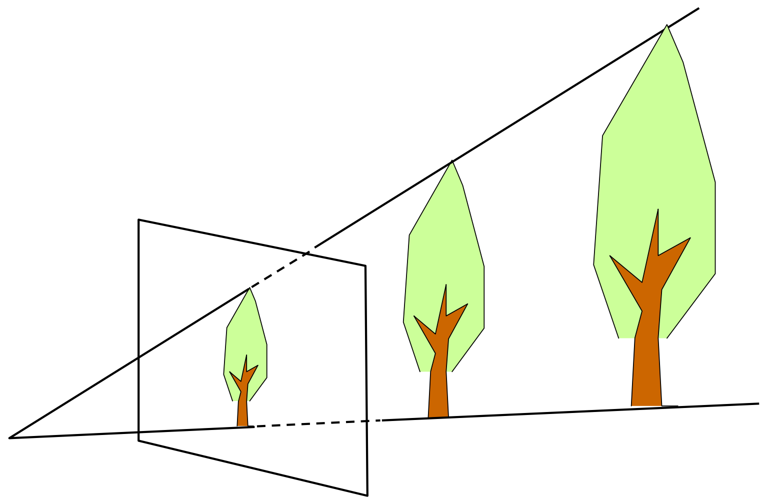


Fundamental ambiguity

- Projective ambiguity contains more fundamental ambiguity, which we usually leave as it is; 7 out of 15 DoFs
  - Equal to a similarity trans.
  - Choice of the world coordinates + scaling ambiguity



How do we choose the world coordinates?



Absolute scale cannot be determined from image(s)

Rectification of 3D reconstruction

- 3D reconstruction up to projective ambiguity is called projective reconstruction
  - There are also affine reconstruction and similarity reconstruction

- Given a projective reconstruction of a scene:

$$(\mathtt{P}_i, \mathbf{X}_j) \;\; (i = 1, \ldots, m, j = 1, \ldots, n)$$

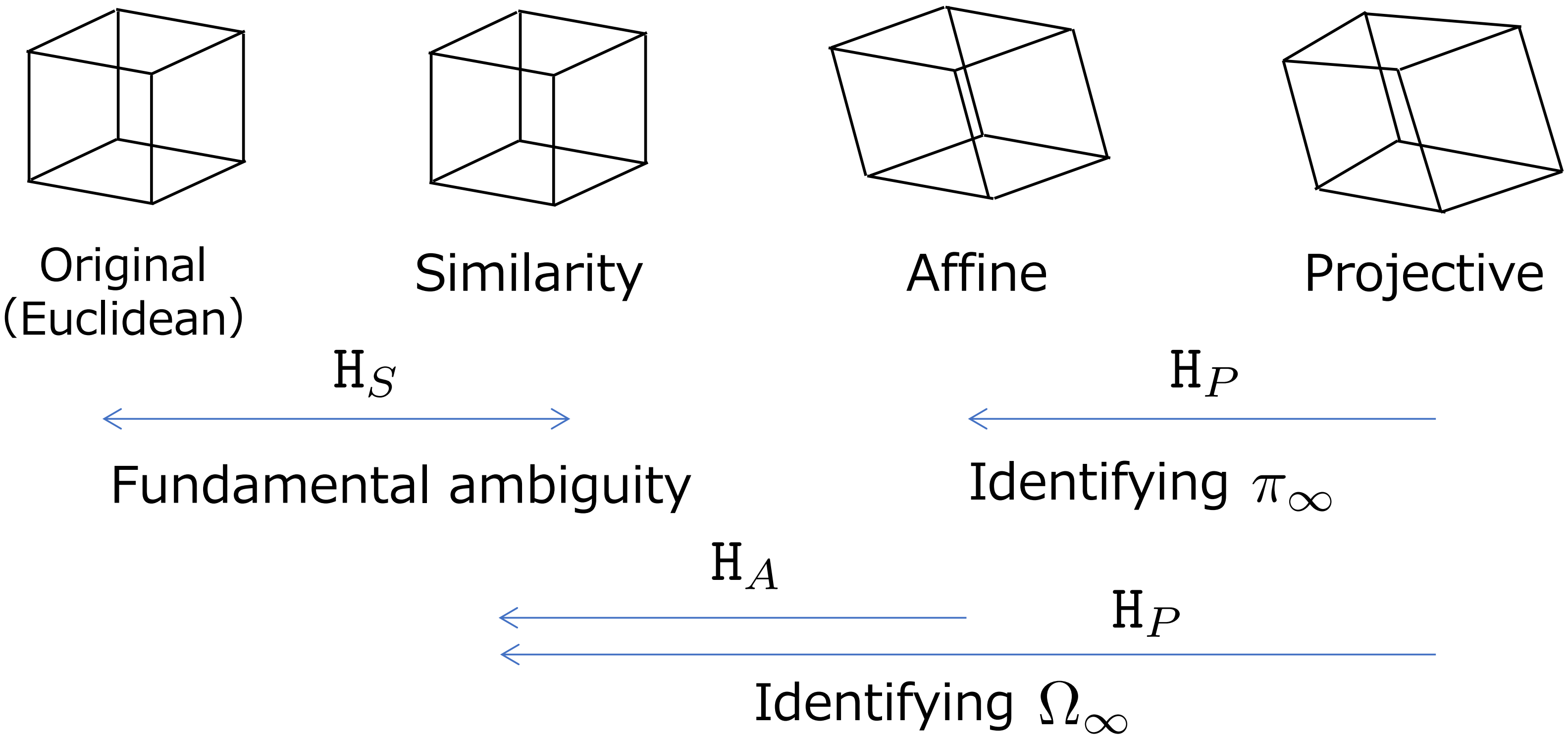
we wish to find H such that the transformed reconstruction

$$\mathtt{P}_i' = \mathtt{P}_i \mathtt{H} \qquad \mathbf{X}_j' = \mathtt{H}^{-1} \mathbf{X}_j$$

gives a similarity (or an affine) reconstruction of the scene

Rectification of 3D reconstruction

- 3D reconstruction up to projective ambiguity is called projective reconstruction
  - There are also affine reconstruction and similarity reconstruction
- Rectification: Given a projective reconstruction of a scene, we wish to rectify it to its affine or similarity reconstruction



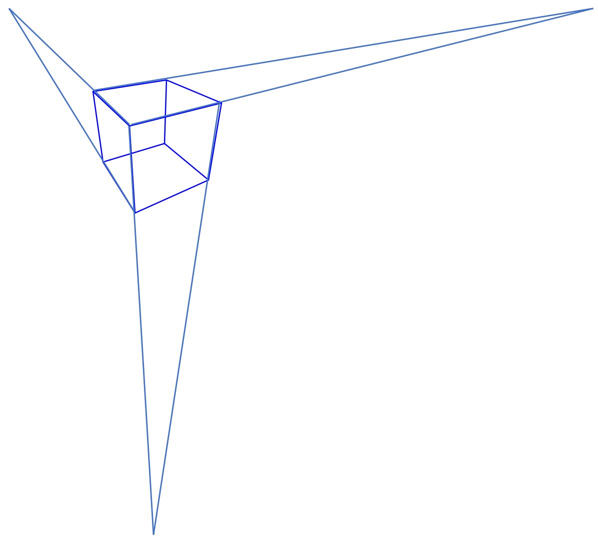
Affine rectification

- Suppose that we can identify the projection π of π_∞ in a projective reconstruction
- Find a projective trans. from the projective to an affine reconstruction
  - Trans. mapping π to π_∞

$$\pi_\infty (= \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}) \propto \mathtt{H}_P^{-\top} \pi$$

$$\mathtt{H}_P = \begin{bmatrix} \mathtt{I} & \mathbf{0} \\ \pi^\top & \end{bmatrix} \quad \Leftarrow \quad \mathtt{H}_P^\top \pi_\infty \propto \pi$$

- How can we identify the image of π_∞ ?
  - E.g., Find three pairs of parallel lines in the scene, from which we can find the projections of three points at infinity



Similarity rectification

- Consider rectifying a projective reconstruction to a similarity reconstruction by $\mathtt{P}'_i = \mathtt{P}_i\mathtt{H}$, $\mathbf{X}'_j = \mathtt{H}^{-1}\mathbf{X}_j$
- Similarity reconstruction is a reconstruction obtained by applying a similarity transform to the true reconstruction (i.e., the fundamental ambiguity); thus, the transformed camera matrix should be

$$\mathtt{P}'_i = \mathtt{P}_i^{(true)}\mathtt{H}_S = \mathtt{K}_i\begin{bmatrix}\mathtt{R}_i & \mathbf{t}_i\end{bmatrix}\mathtt{H}_S = \mathtt{K}_i\begin{bmatrix}\mathtt{R}_i\mathtt{R} & \mathtt{R}_i\mathbf{t} + \mathbf{t}_i\end{bmatrix} \quad \mathtt{H}_S = \begin{bmatrix} s\mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$$

- In short, we wish to find $\mathtt{H}$ such that $\mathtt{P}'_i = \mathtt{P}_i\mathtt{H} = \mathtt{K}_i\begin{bmatrix}\mathtt{R}'_i & \mathbf{t}'_i\end{bmatrix}$
- Remark:
  - Similarity rectification is equivalent to knowing $\mathtt{K}_i$
  - Any $\mathtt{P}_i$ can be decomposed into the form of $\mathtt{K}_i\begin{bmatrix}\mathtt{R}_i & \mathbf{t}_i\end{bmatrix}$; we need additional info. about the scene or the cameras

Autocalibration (self-calibration)

- Suppose we do not have knowledge about the scene
- If we have no knowledge about the cameras, either, then projective reconstruction is the maximum we can hope for
- If we have at least partial knowledge about the internal parameters $\mathtt{K}_i$ of the camera(s), then we can fully calibrate the camera(s) and obtain a similarity reconstruction
  - Usual setting: only focal lengths are unknown; or plus image centers; and plus lens distortion; all others are known
  - We may assume that usually skew = 0 and aspect = 1; sometimes image center is merely the center of image; however, focal length is difficult to know beforehand, due to focusing and zooming
  - There is the minimum number of images necessary for each setting → Details are given in the section 'Bundle Adjustment'

Problem

- Given m images of n scene points captured from different viewpoints, we want to estimate the 3D coordinates of the n points and the camera matrices of the m views

Geometric model

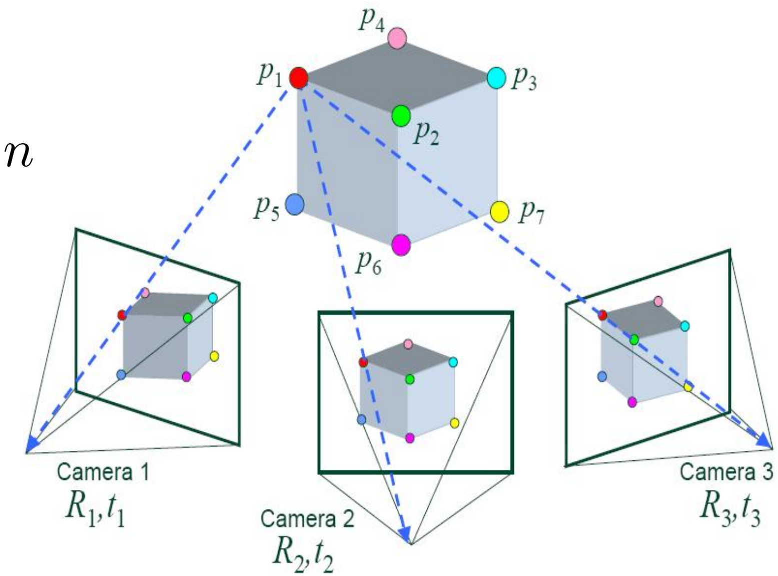
$$\mathbf{x}_j^{(i)} \propto \mathtt{P}_i \mathbf{X}_j$$

Input

$$\mathbf{x}_j^{(i)} = \begin{bmatrix} x_j^{(i)} & y_j^{(i)} & 1 \end{bmatrix}^\top$$

$$i = 1, \ldots, m, \quad j = 1, \ldots, n$$

$$2mn$$



Output

$$\mathbf{X}_j = \begin{bmatrix} X_j & Y_j & Z_j & 1 \end{bmatrix}^\top$$

$$3n$$

$$\mathtt{P}_i = \mathtt{K}_i \begin{bmatrix} \mathtt{R}_i & \mathbf{t}_i \end{bmatrix}$$

$$11m$$

Key points

- What are *good key points*?
- Points are good if we can determine their positions in images as precisely as possible

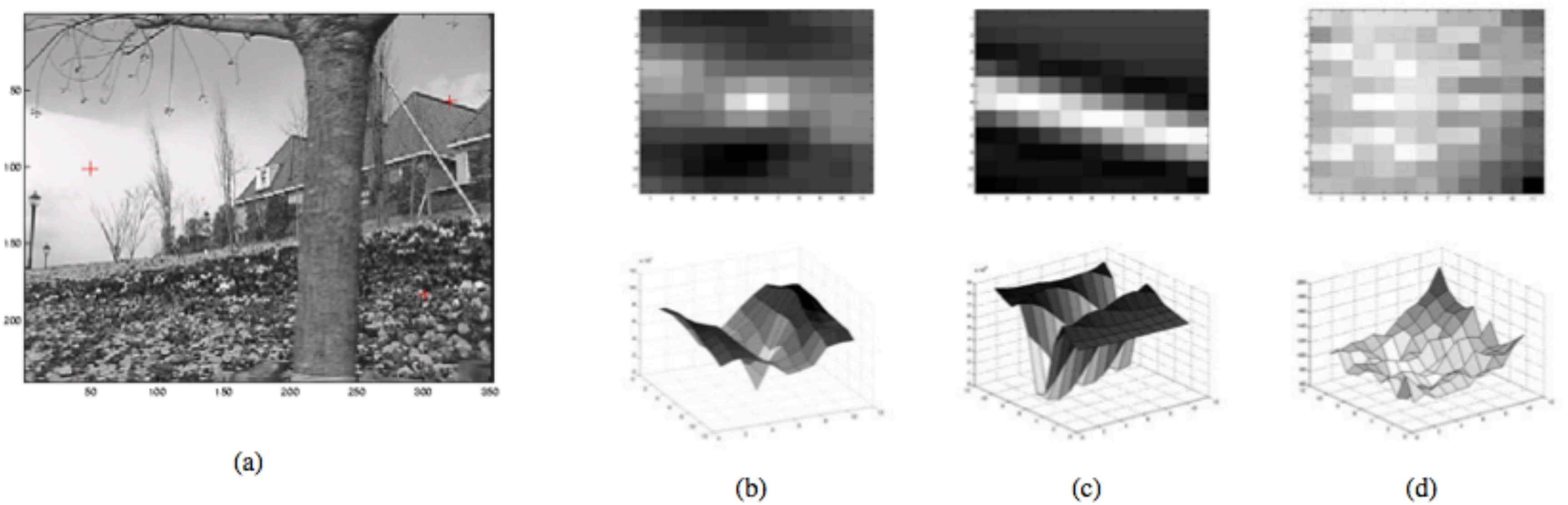
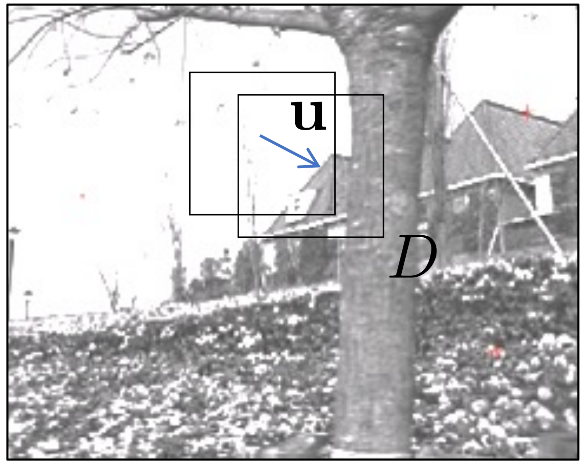


Figure 4.5 Three auto-correlation surfaces $E_{\mathrm{AC}}(\Delta \boldsymbol{u})$ shown as both grayscale images and surface plots: (a) The original image is marked with three red crosses to denote where the auto-correlation surfaces were computed; (b) this patch is from the flower bed (good unique minimum); (c) this patch is from the roof edge (one-dimensional aperture problem); and (d) this patch is from the cloud (no good peak). Each grid point in figures b–d is one value of $\Delta \boldsymbol{u}$.

[Szeliski2010]

Key points

- How can we measure such goodness of points?
- Answer:
  - The brightness must have as sharp a peak as possible
  - Having a peak = the two eigenvalues of **A** are both large enough



$$f(\mathbf{u}) = \sum_{\mathbf{x}\in D} [I(\mathbf{x}+\mathbf{u}) - I(\mathbf{x})]^2$$

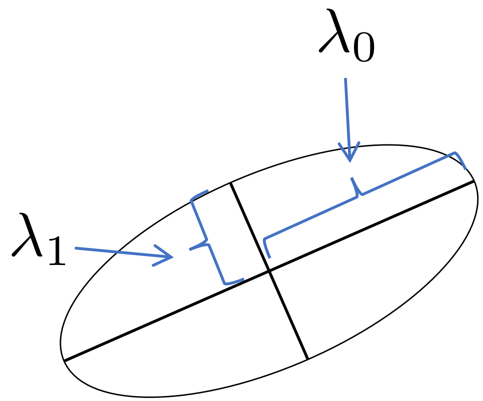
$$\approx \sum_{\mathbf{x}\in D} [I(\mathbf{x}) + \nabla I(\mathbf{x})^\top \mathbf{u} - I(\mathbf{x})]^2 \qquad \nabla I(\mathbf{x}) = \begin{bmatrix} I_x(\mathbf{x}) \\ I_y(\mathbf{x}) \end{bmatrix}$$

$$= \sum_{\mathbf{u}\in D} \mathbf{u}^\top \nabla I(\mathbf{x}) \nabla I(\mathbf{x})^\top \mathbf{u}$$

$$= \mathbf{u}^\top \mathbf{A} \mathbf{u} \qquad \mathbf{A} = \sum_{\mathbf{x}\in D} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = \lambda_0 \mathbf{e}_0 \mathbf{e}_0^\top + \lambda_1 \mathbf{e}_1 \mathbf{e}_1^\top$$

Harris-Stephens(1988): $\det(\mathbf{A}) - \alpha \mathrm{trace}(\mathbf{A})^2 = \lambda_0 \lambda_1 - \alpha(\lambda_0 + \lambda_1)^2$

Brown-Szeliski-Winder(2005): $\dfrac{\det(\mathbf{A})}{\mathrm{trace}(\mathbf{A})} = \dfrac{\lambda_0 \lambda_1}{\lambda_0 + \lambda_1}$



SIFT (Scale Invariant Feature Transform)

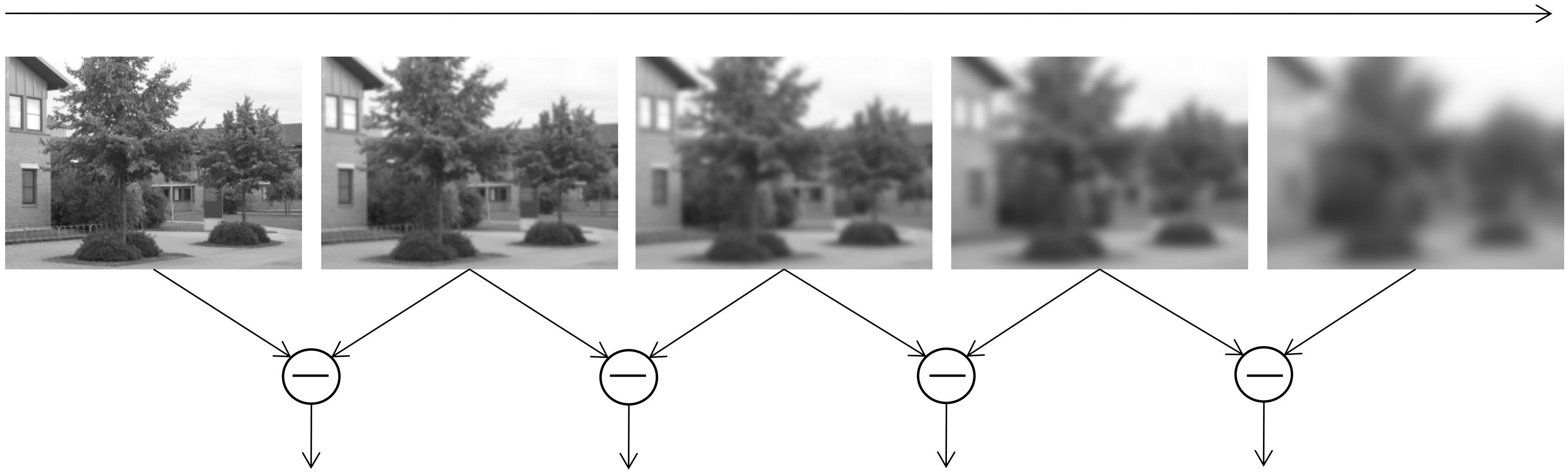
Lowe, Object recognition from local scale-invariant features, ICCV99

- We wish to match image points of an identical scene point on multi-view images

1. Key point
   - Invariant to scale and orientation
2. Descriptor
   - Image feature that is invariant to scale and orientation

SIFT: Keypoint

- Extrema of DoG in scale space are chosen as keypoints
  - DoG: Difference of Gaussian
  - Scale is automatically chosen, obtaining invariance to scale
- Scale space
  - A series of images that are blurred by Gaussian filters



DoG $f(x, y, s)$

SIFT: Descriptors

- Besides the scale, principal orientation is determined
  - The peak of orientation histogram (36 bins) is chosen
  - Enables rotation invariance
- A square is chosen in accordance with the chosen orientation and scale; then, it is divided into boxes, for each of them an orientation histogram is generated

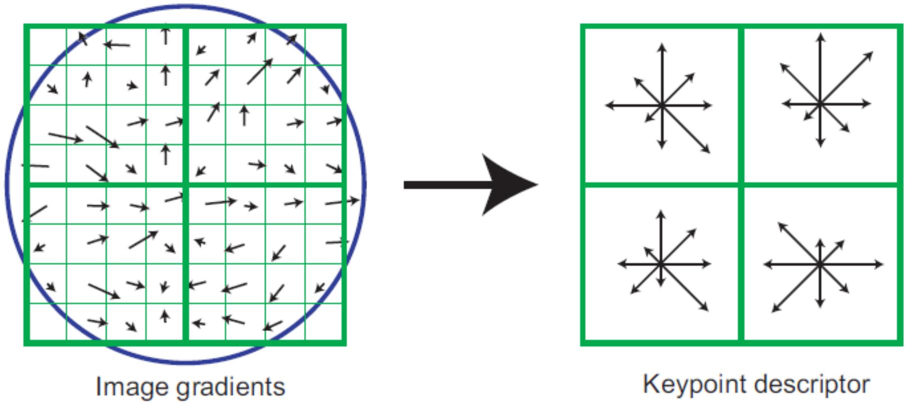


Figure 7: A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4x4 subregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. This figure shows a 2x2 descriptor array computed from an 8x8 set of samples, whereas the experiments in this paper use 4x4 descriptors computed from a 16x16 sample array.

Matching keypoints: nearest neighbor search

- Comparing distances between descriptors
- Keypoints and descriptors are invariant to scale and rotation