Affine Linear Transformations

Prof. Graeme Bailey

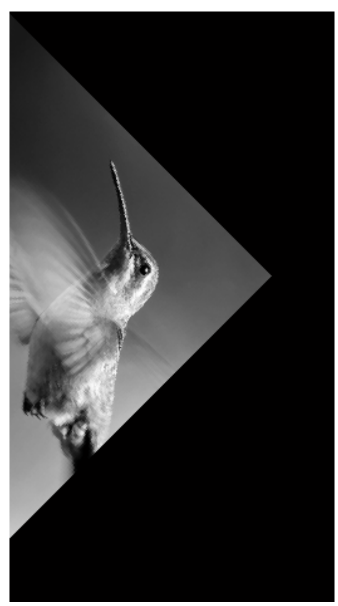
http://cs1114.cs.cornell.edu

(notes modified from Noah Snavely, Spring 2009)

Cornell University
Computer Science

More image transformations

- Recall from last time … the default rotation is around the point (0, 0) – the upper-left corner of the image



- This probably isn't really what we want…

Translation

- We really want to rotate around the *center* of the image

- Nice trick: move the center of the image to the origin, apply the default rotation, then move the center back. If T is the translation, R the desired rotation, and D the default rotation, then R = T^{-1}DT. We often call this *conjugation* of D by T … it's a very useful trick in a great many different contexts!

- But technically, translation isn't a linear function - check the two definition properties for **f**(**v**) = **v** + **w** (for some constant vector **w**)…
  - **f**(**v** + **u**) =? **f**(**v**) + **f**(**u**) *LHS* = **v** + **u** + **w** *RHS* = **v** + **u** + 2**w**
  - **f**(a**v**) =? a **f**(**v**) *LHS* = a**v** + **w** *RHS* = a**v** + 2a**w**

- Formally we call this an *affine* linear function

Reminder: shear from lecture 15

- Shear

$$H = \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix}$$

Remember again that for images, the positive y *direction is* ***downwards****!!*

Detour: Homogeneous coordinates

- We want a way to use MATLAB's matrix routines to compute translations, but matrices only do linear functions…
- Ans: add a dimension! Add a 1 to the end of our 2D point (x, y) → (x, y, 1) to give a point on the plane parallel to the 2D coordinate plane at height 1 = "homogeneous" 2D points
- Essentially each line through the origin in 3D 'corresponds' to a point in 2D (what about the lines actually in the xy-plane?)
- We can represent transformations on 2D homogeneous coordinates as 3D matrices
- 'Shear' in 3D is linear, and 'is' translation in (homog) 2D !!!!

Translation via homogeneous coordinates

$$T = \begin{bmatrix} 1 & 0 & s \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} s \\ t \\ 1 \end{pmatrix}$$

Shear parallel to the xy-plane by adding s to the x direction and t to the y direction

- Other transformations just add an extra row and column with [0 0 1]

$$S = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

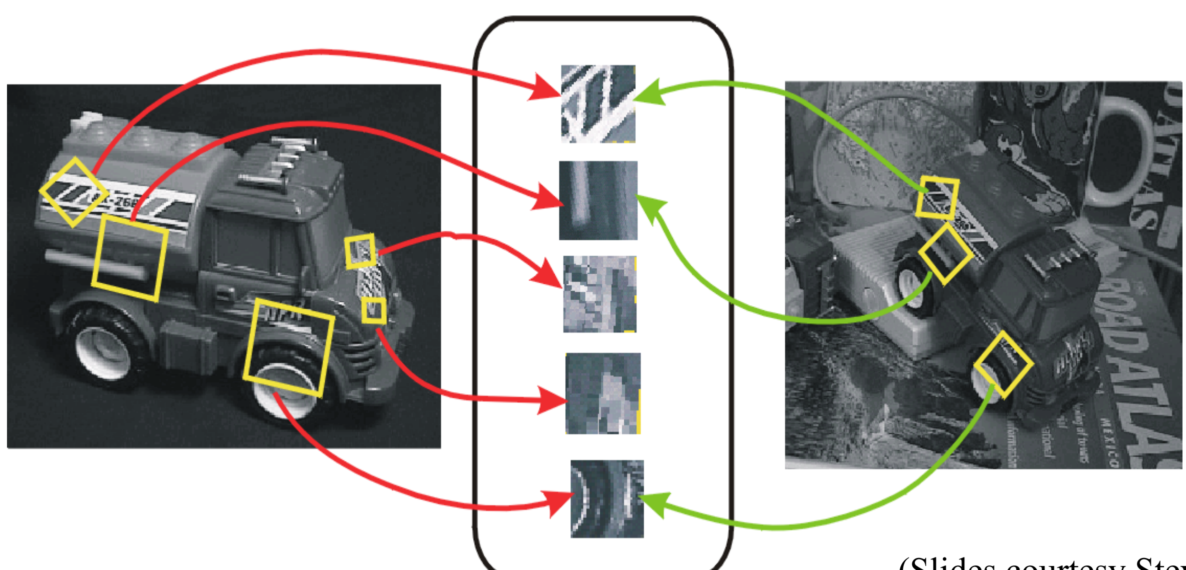
scale *rotation*

Rotation about image centre

- Translate center to origin $T = \begin{bmatrix} 1 & 0 & -w/2 \\ 0 & 1 & -h/2 \\ 0 & 0 & 1 \end{bmatrix}$
- Rotate $R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- Translate back to center $T^{-1} = \begin{bmatrix} 1 & 0 & w/2 \\ 0 & 1 & h/2 \\ 0 & 0 & 1 \end{bmatrix}$
- Combining these gives $T^{-1} R\ T$

Invariant local features

- Find features that are invariant to transformations
  - geometric invariance: translation, rotation, scale
  - photometric invariance: brightness, exposure, …



(Slides courtesy Steve Seitz)

Why local features?

- Locality
  - features are local, so robust to occlusion and clutter
- Distinctiveness:
  - can differentiate a large database of objects
- Quantity
  - hundreds or thousands in a single image
- Efficiency
  - real-time performance achievable

More motivation…

- Feature points are used for:
  - Image alignment (e.g., mosaics)
  - 3D reconstruction
  - Motion tracking
  - Object recognition
  - Robot navigation
  - …

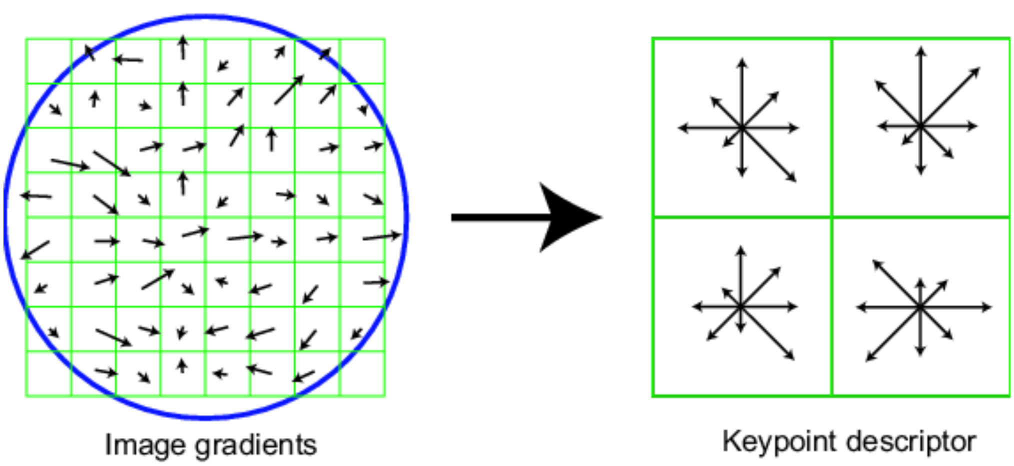
SIFT Features

- **S**cale-**I**nvariant **F**eature **T**ransform

Cornell University

SIFT descriptor

- Very complicated, but very powerful
- (The details aren't all that important for this class.)
- 128 dimensional descriptor

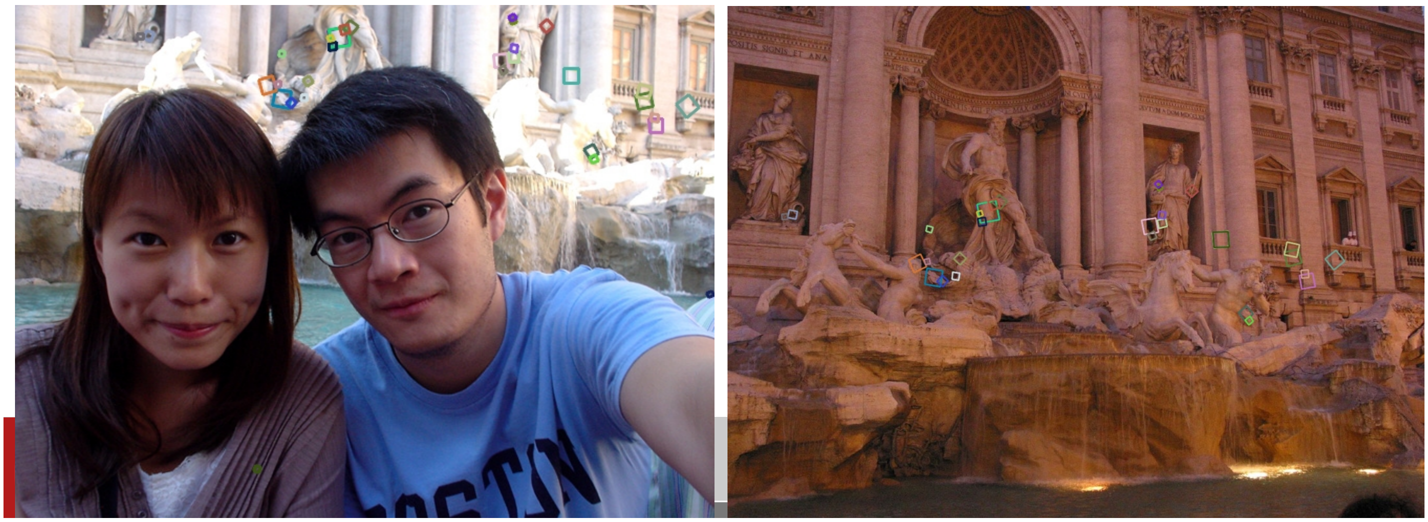


Adapted from a slide by David Lowe

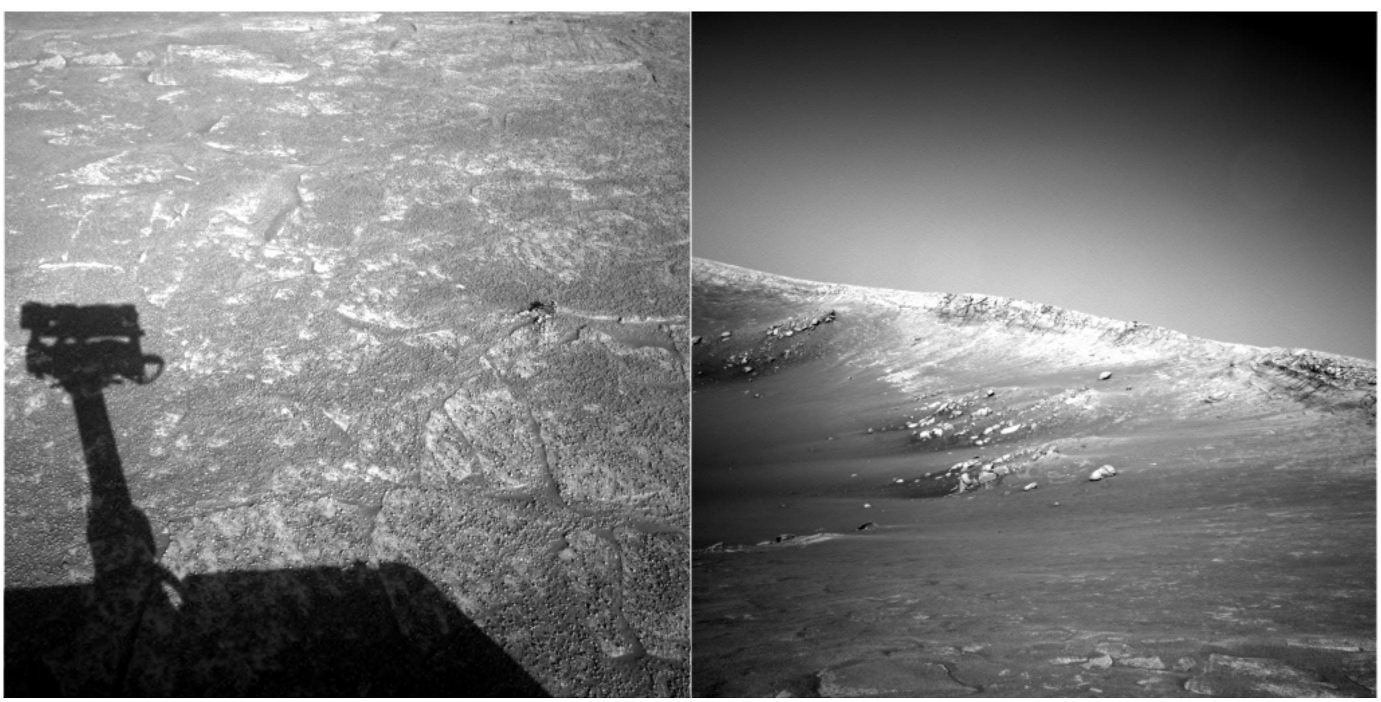
Cornell University

Properties of SIFT

- Extraordinarily robust matching technique
  - Can handle significant changes in illumination
    - Sometimes even day vs. night (below)
  - Fast and efficient—can run in real time
  - Lots of code available
    - http://people.csail.mit.edu/albert/ladypack/wiki/index.php/Known_implementations_of_SIFT

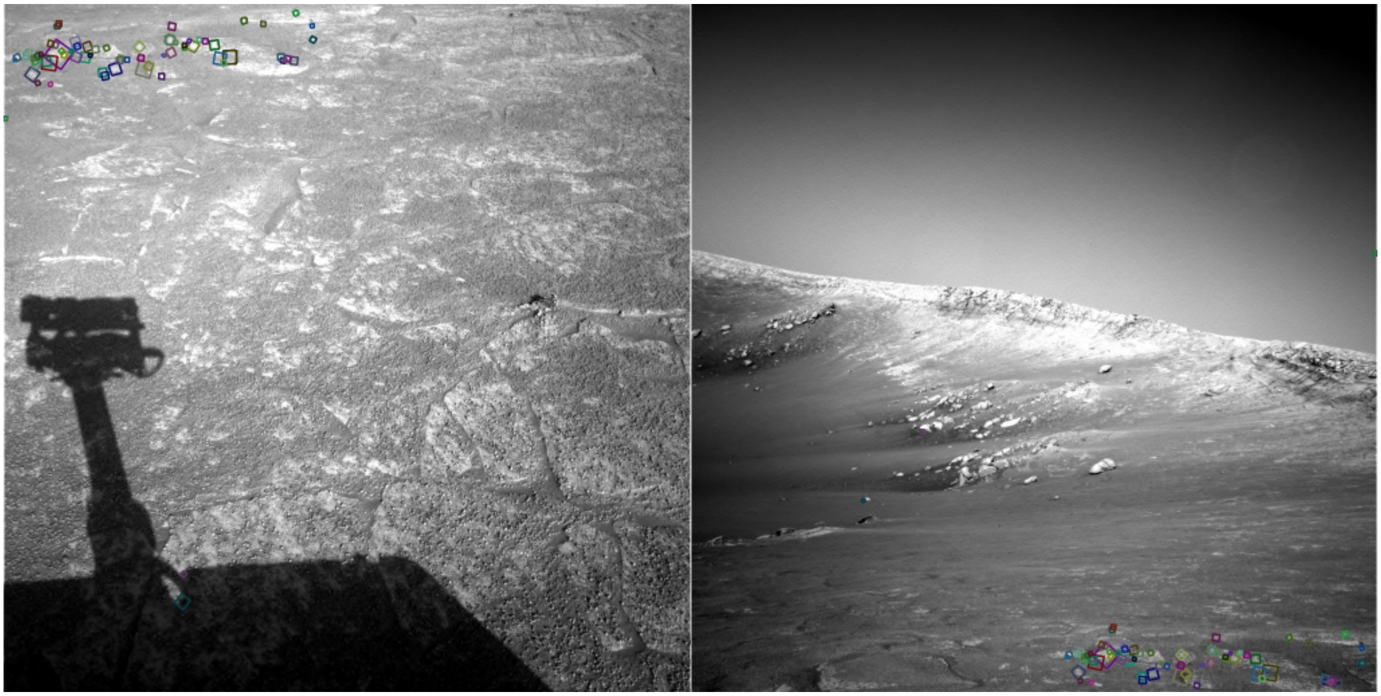


Do these two images overlap?



NASA Mars Rover images

Answer below



NASA Mars Rover images

Image transformations

- What about a general homogeneous transformation?

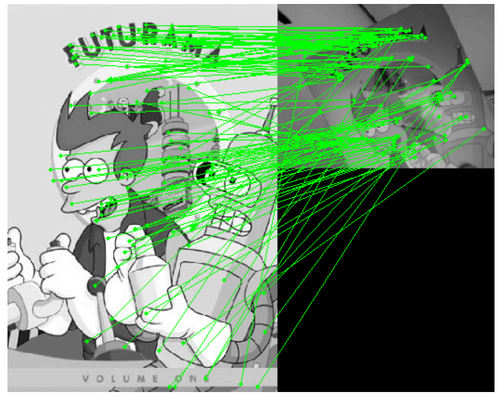
$$T = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} ax + by + c \\ dx + ey + f \\ 1 \end{bmatrix}$$

- Called a 2D *affine* transformation

Solving for image transformations

- Given a set of matching points between image 1 and image 2…



… can we solve for an affine transformation *T* mapping 1 to 2?