Robust Log-Polar Registration

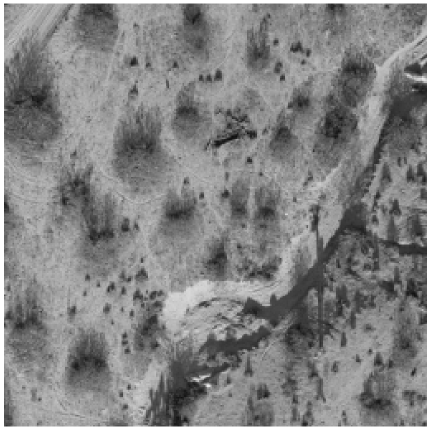
Prof. George Wolberg
Dept. of Computer Science
City College of New York

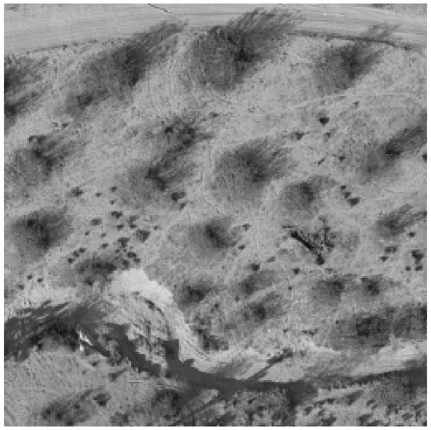
Objectives

- In this lecture we review image registration, a process to align multiple images together.
  - Affine registration
  - Perspective registration
  - Polar coordinates
  - Log-polar algorithm
  - Examples

Introduction

- Image registration refers to the problem of aligning a set of images.
- The images may be taken at different times, by different sensors, or from different viewpoints.





Applications

- Multisensor data fusion
  - integrating information taken from different sensors
- Image analysis / surveillance / change detection
  - for images taken at different times/conditions
- Image mosaics
  - generating large panoramas from overlapping images
- Super-resolution images / noise removal
  - integrating multiple registered images

Geometric Transformations

Geometric transformation models for registration:

- Rigid transformation
  - translation, rotation (3 parameters)
- Affine transformation
  - translation, rotation, scale, shear (6 parameters)
- Perspective transformation
  - affine, perspective foreshortening (8 parameters)
- Local transformation
  - terrain relief, nonlinear, nonrigid

Parameter Estimation (1)

Moderate perspective **$|\beta|, |\gamma| < 40^o$**	**Severe perspective** **$40^o < |\beta|, |\gamma| < 90^o$**
	

Parameter Estimation (2)

The objective function (similarity measure):

$$\chi^2(\mathbf{a}) = \sum_{i=1}^{N} (I_1(x,y) - I_2(Q_{\mathrm{a}}\{x', y'\}))^2$$

Q{} is the affine or perspective transform applied to image $I_{2.}$ We use the Levenberg-Marquardt method to minimize the above objective function for each pyramid level.

Affine Registration

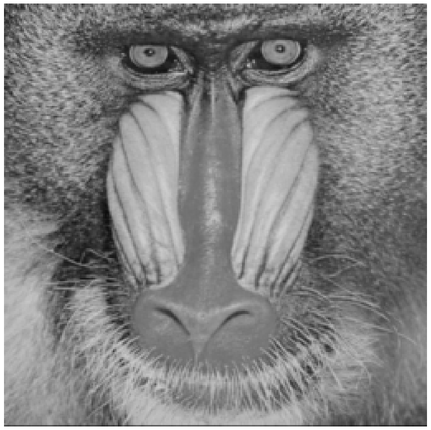


Image I_1

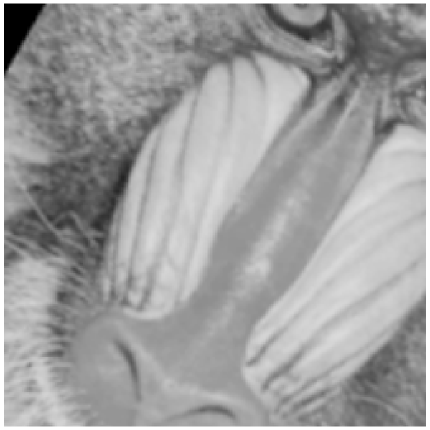
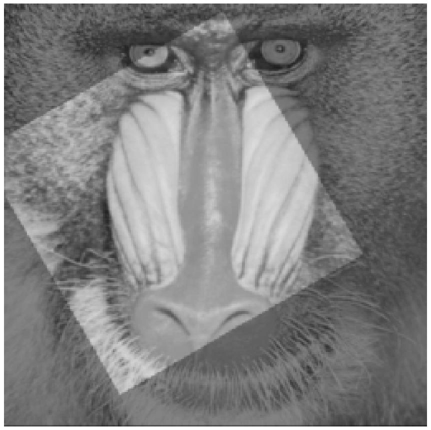


Image I_2

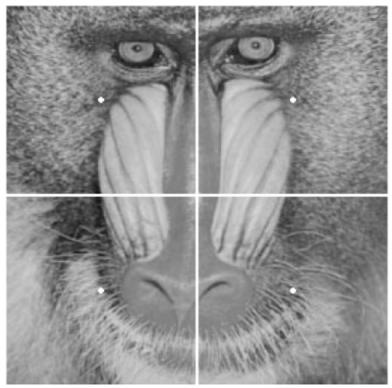


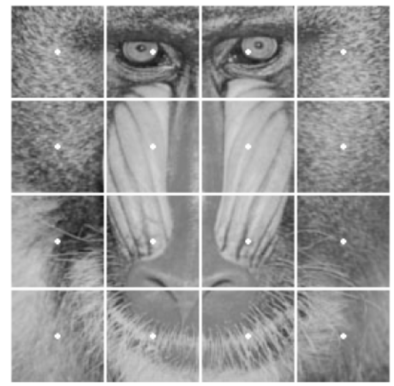
I_2 registered to I_1

Actual parameters: RST = $(30^\circ, 1.5, 30, 30)$

Estimated parameters: RST = $(29.99^\circ, 1.51, 29.51, 30.66)$

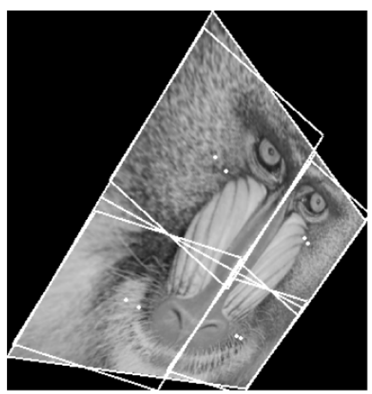
Local Affine Approximation

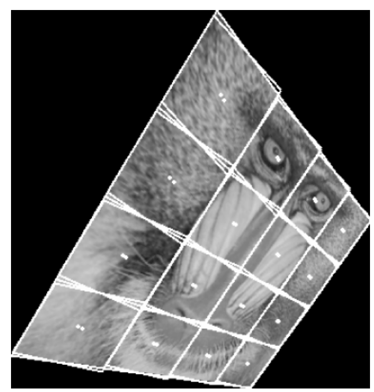




2 X 2 tiles

4 X 4 tiles

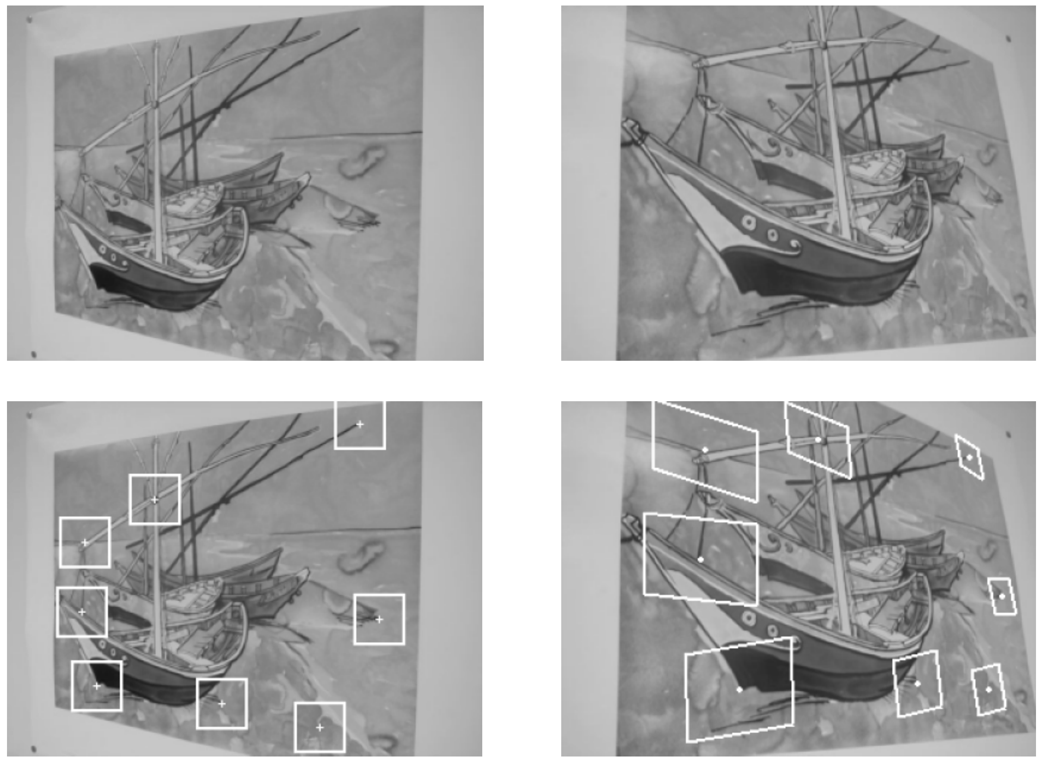




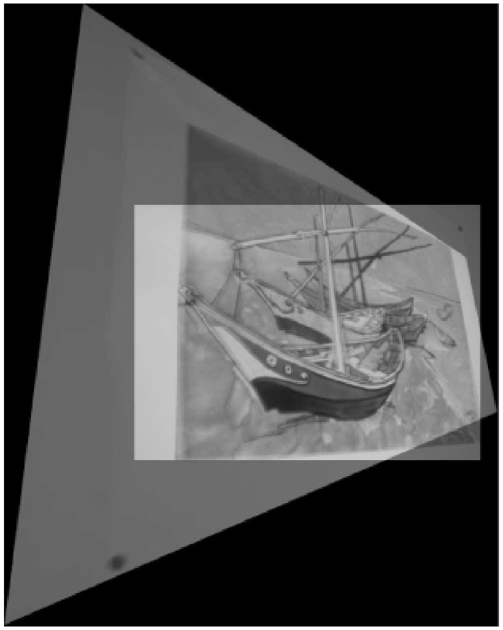
Piecewise Affine Approximation

- Consider I_2 subdivided into a regular grid of tiles
- For each tile T^i_2 search for similar tile T^i_1 in I_1 by performing affine registration using LMA
- Get collection of affine parameters and tile centers
- Plug into system of equations
- Solve for 8 persp. parameters using the pseudoinverse

Example (1)



Example (1)



Perspective Registration

Glare Removal

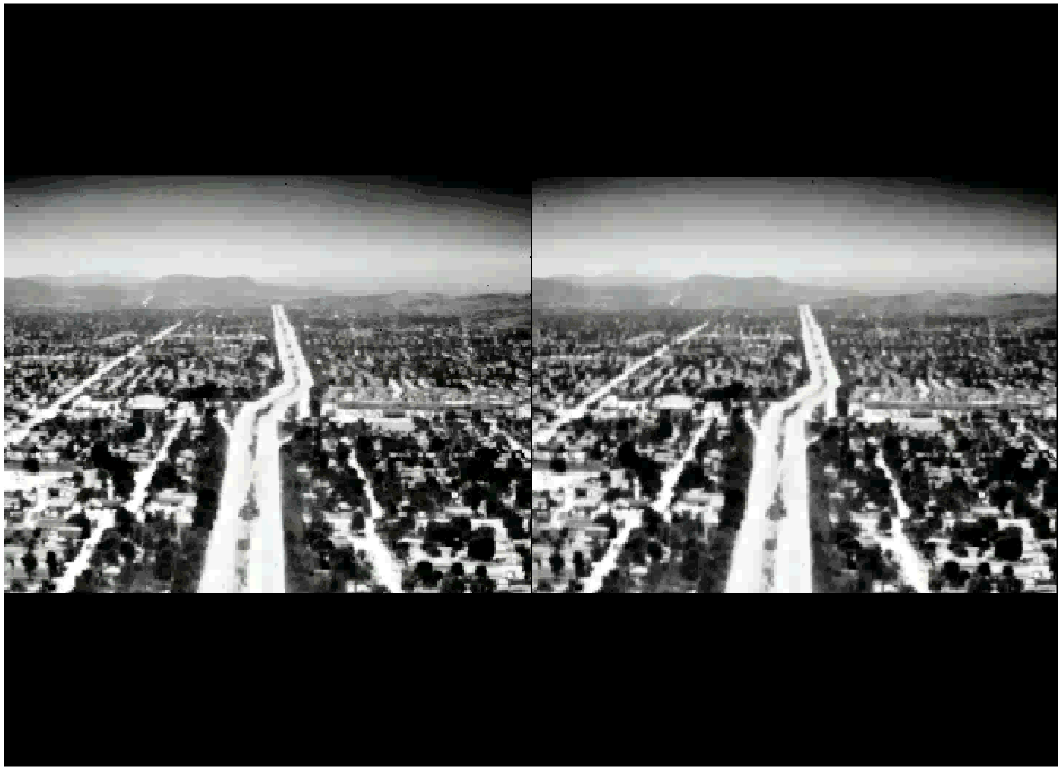
- The camera's flash introduces a highlight.
- Each highlight will appear at a different location on each image when captured from different viewpoints.
- If we register these images onto a common frame, we can render the final image by fusing the registered images.

Example (2)

- Video mosaic

Example (3)

- Video stabilization

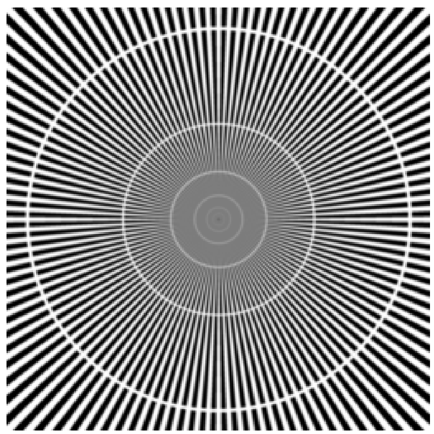


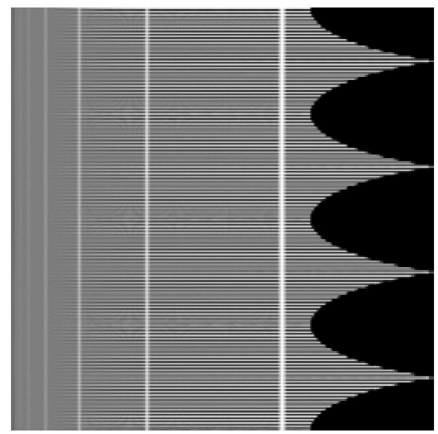
Discussion

- Method: Levenberg-Marquadt Alg. (nonlinear least squares)
- Benefit: estimation of real-valued parameters
- Drawback: the registered images must be fairly close in scale (<1.5), rotation (<45°), and translation
- Solution: log-polar registration

Polar Coordinates

Radial lines in *(x,y)* Cartesian space map to horizontal lines in *(r,θ)* polar coordinate space.

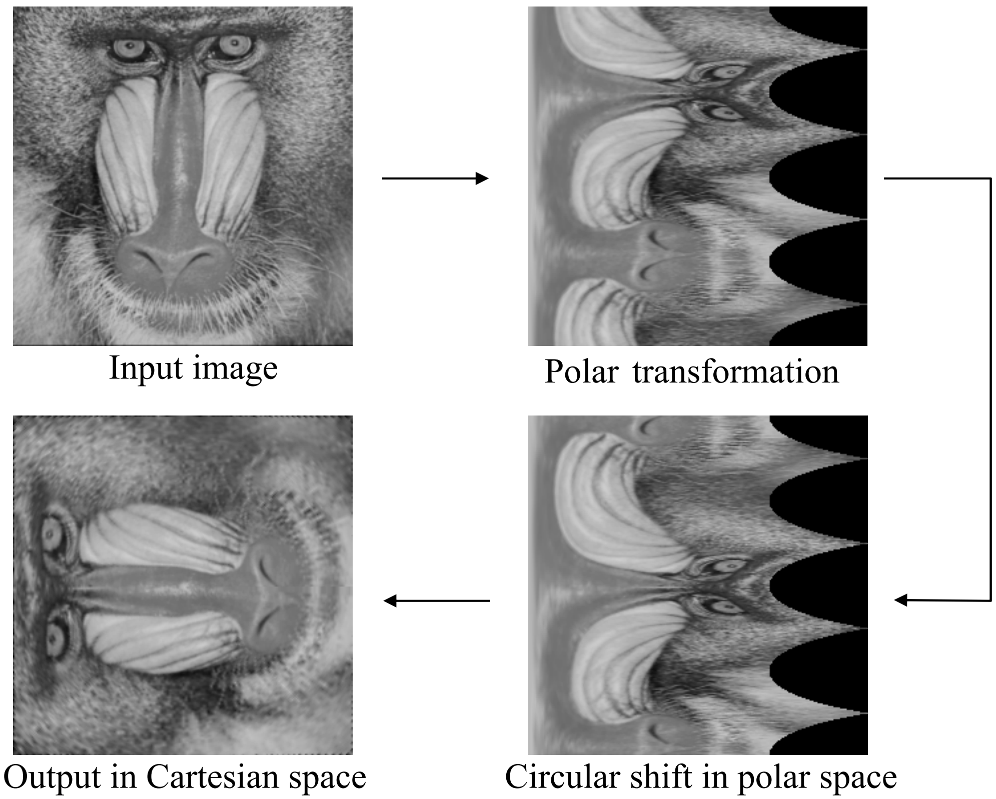




$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$$

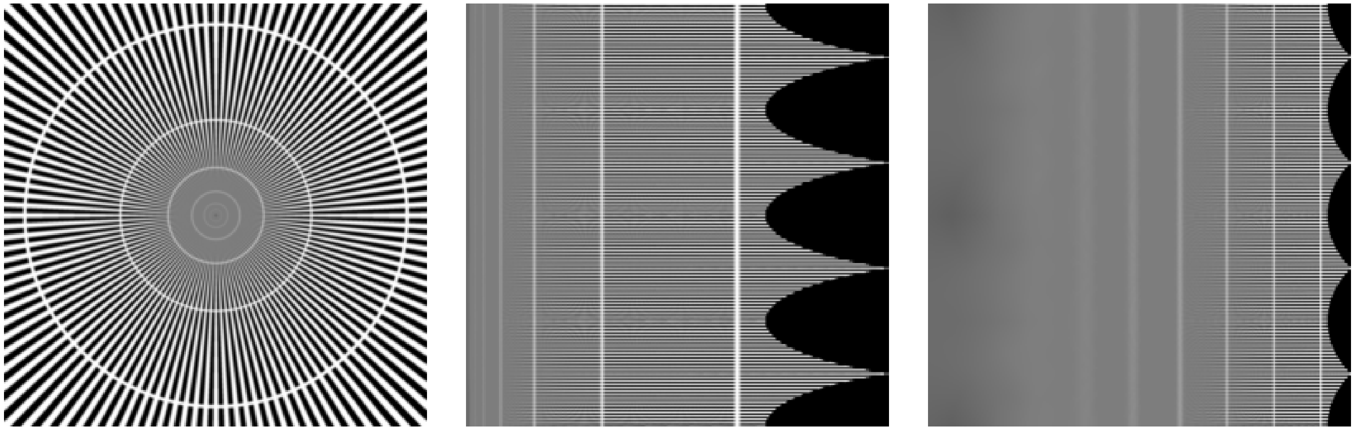
$$\theta = \tan^{-1} \frac{y - y_c}{x - x_c}$$

Polar Coordinates: Example



Log-Polar Coordinates

Radial lines in *(x,y)* map to horizontal lines in *(log r, θ)*



Example

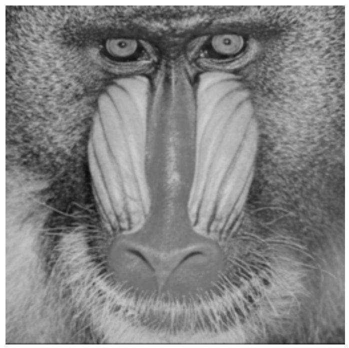
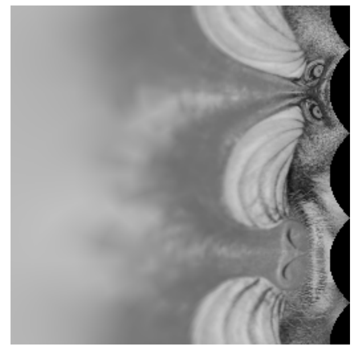


Image I_1



Log-polar(I_1)

Image I_2: 3x; 45°

Log-polar(I_2)

Log-Polar Registration

- <u>Benefit:</u> Correlation recovers rotation/scale (fast)
- <u>Drawback:</u> Image origins must be known
- <u>Solution:</u>

1. Crop central region I_1' from I_1
2. Compute I_{1p}', the log-polar transformation of I_1'
3. For all positions (x, y) in I_2:

    Crop region I_2'

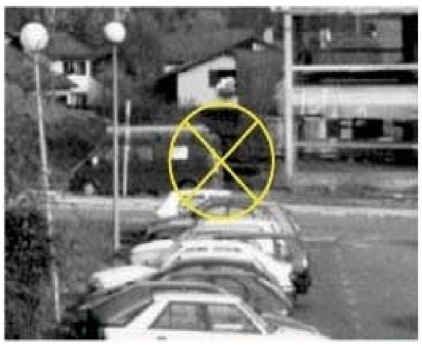
    Compute I_{2p}'

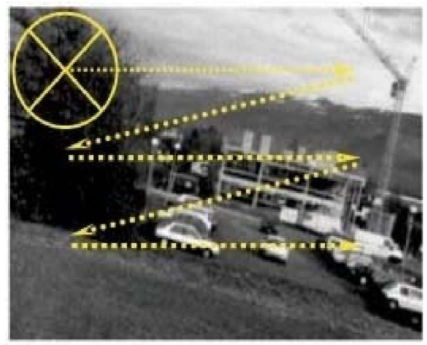
    Cross-correlate I_{1p}' and $I_{2p}' \rightarrow (dx, dy)$

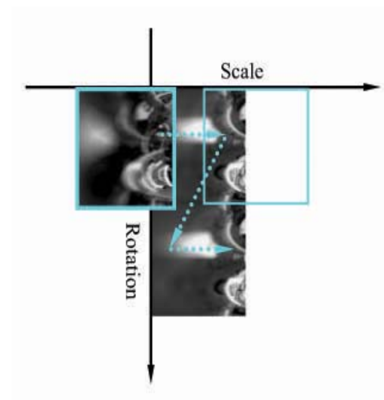
    if maximum correlation, save (x, y) and (dx, dy)

4. Scale $\leftarrow dx$
5. Rotation $\leftarrow dy$
6. Translation $\leftarrow (x, y)$

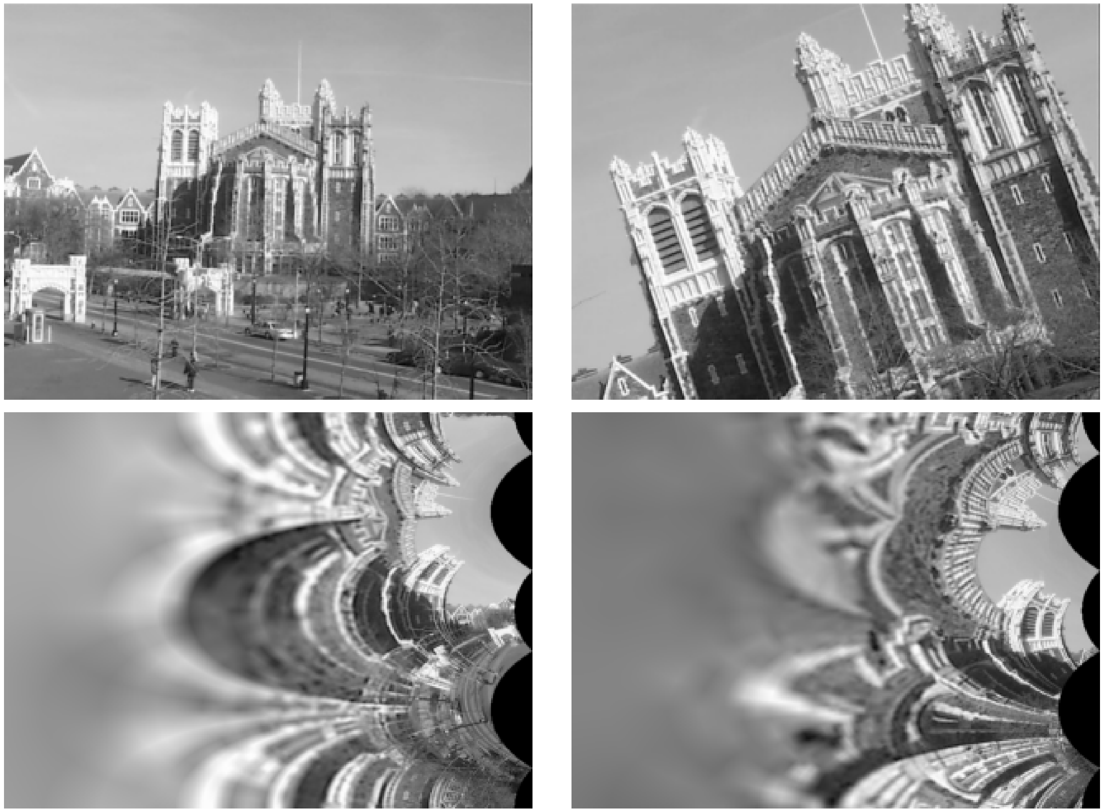
Scanning Window



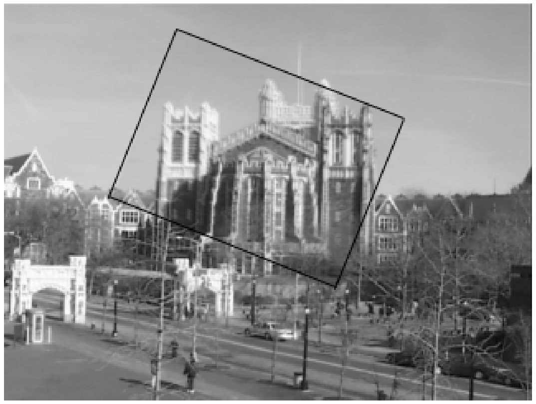




Example



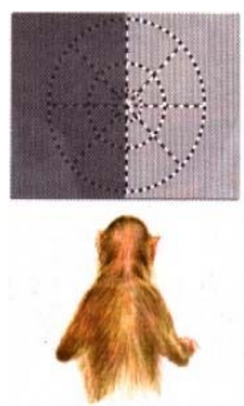
Example

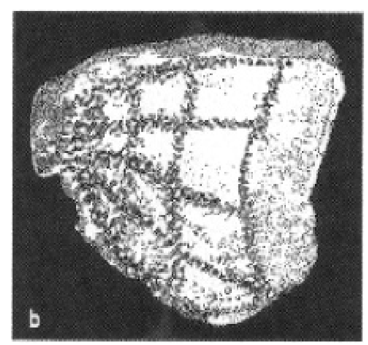


Rotation = -21°
Scale = 0.469
Translation = (-11,-51)

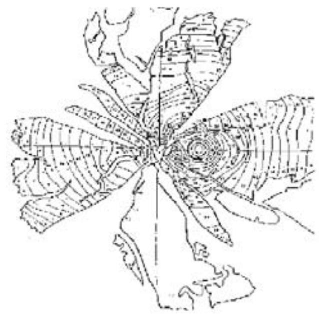
Biological Motivations (1)

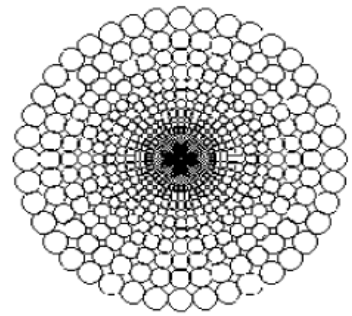
- Despite the difficulties of nonlinear processing, the log-polar transform has received considerable attention. [Marshal41] et. al. and [Hubel74] discovered a log-polar mapping in the primate visual system and this conformal map is an accepted model of the representation of the retina in the primary visual cortex in primates [Schwartz79] [Weinshall87].





Biological Motivations (2)





The left figure shows contours of cone cell density. In fact, the density of the ganglion cells which transmit information out of the retina is distributed logarithmically. From Osterberg, G. (1935)

Benefits:

1. Reduce the amount of information traversing the optical nerve while maintaining high resolution in the fovea and capturing a wide field of view in the periphery.

2. Invariant to scale and rotation.

Biological Motivations (3)

Fovea based vision

Related Work

- The Fourier-Mellin transform uses log-polar mapping to align images related by scale, rotation, and translation {***Casasent76, Decastro87, Chen94, Reddy96, Lucchese96, Chang97, Lucchese97a, Lucchese97b, Stone99, Stone03, Keller03***}.
- Detect lines in log-Hough space {***Weiman79, Giesler98, Weiman90, Young00***}.
- Recognizing 2D objects that were arbitrarily rotated or scaled {***Sandini92, Ferrari95***}.
- Tracking a moving object {***Capurro97***}.
- Estimation of time-to-impact from optical flow {***Sandini91***}
- Finding disparity map in stereo images {***Sandini01***}.
- Using log-polar transform to calculate time-to-crash for mobile vehicles {***Pardo02***}.
- A foveated binocular stereo system using log polar transforms {***Bernardino02***}.

Previous work: Fourier-Mellin

$$I_1(x,y) = I_2(s(x\cos(\theta_0) + y\sin(\theta_0) - x_0), s(-x\sin(\theta_0) + y\cos(\theta_0)) - y_0)$$

$$F_1(\omega_x,\omega_y) = \frac{1}{s} F_2\left(\frac{\omega_x \cos(\theta_0) + \omega_y \sin(\theta_0)}{s}, \frac{-\omega_x \sin(\theta_0) + \omega_y \cos(\theta_0)}{s}\right) e^{J(x_0\omega_x + y_0\omega_y)}$$

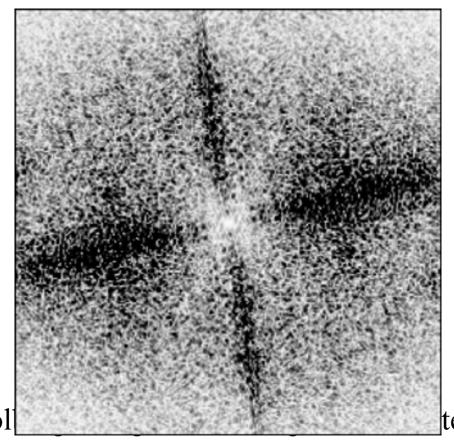
- The magnitude of spectra $|F_1|$ is a rotated and scaled replica of $|F_2|$. We can recover this rotation and scale by representing the spectra $|F_1|$ and $|F_2|$ in log-polar coordinates:

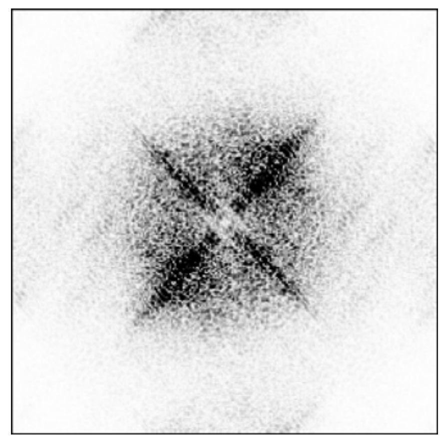
$$\left|F_1(\log r, \theta)\right| = \frac{1}{s}\left|F_2(\log r + \log s, \theta - \theta_0)\right|$$

- Phase-correlation can be used to recover the amount shift

Previous work: Fourier-Mellin

- Example: $s = 1.5$, $\theta = 36^\circ$





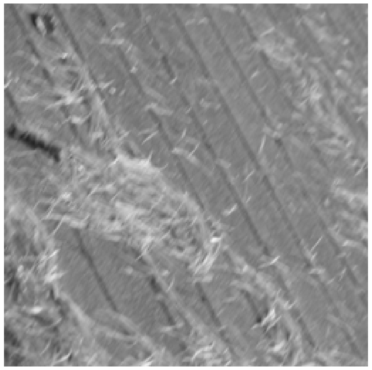
Previous work: Fourier-Mellin

- Benefit: We can search for the scale and rotation independent of the translations.

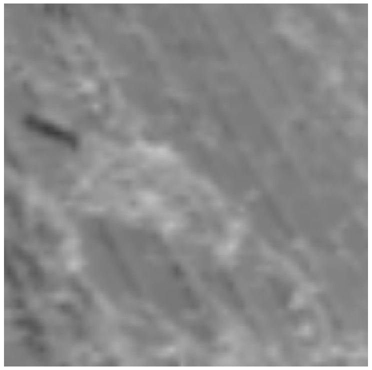
- Drawbacks: Large translation, scale, or mild perspective will alter the coefficients of the finite and discrete Fourier transform.

Previous work: Fourier-Mellin

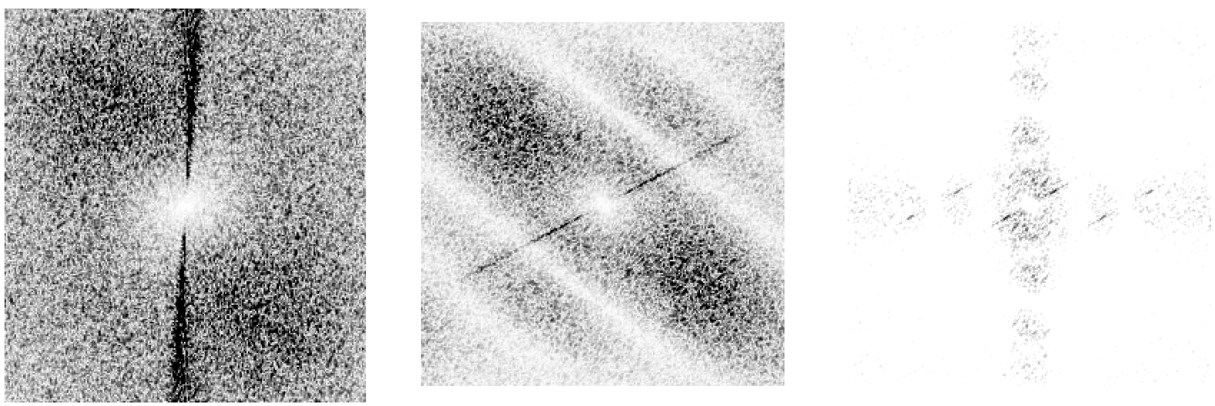
(a) Reference image



(b) Target image (real)



(c) Target image (synthetic)



Power spectra of (a), (b), and (c)

Example from INRIA

Example (1)

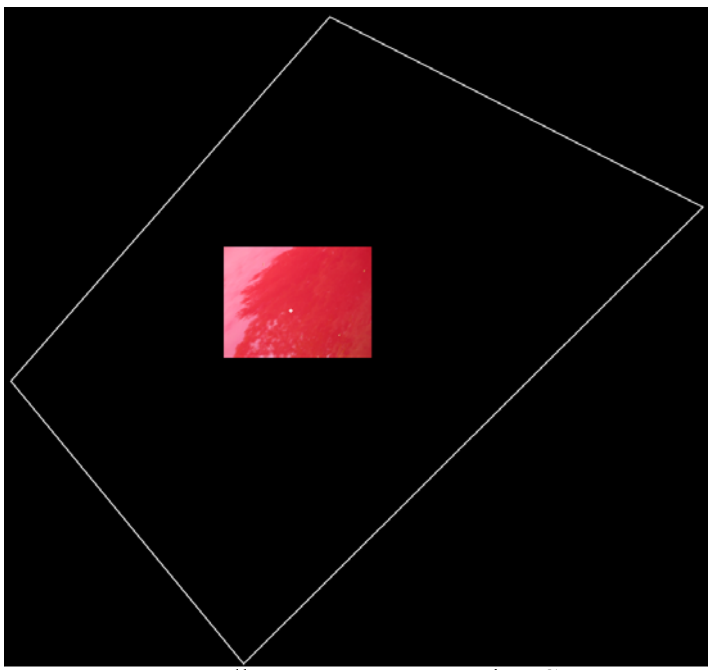
Example (2)

Example (3)

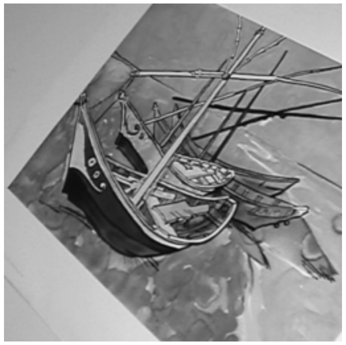
Reference image

Target image

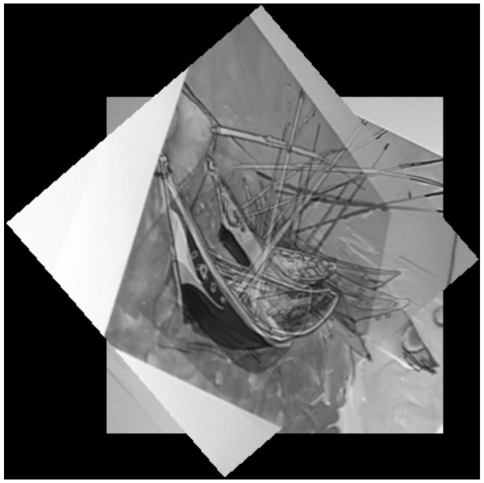
Recovered parameters:
$\theta = 228.17^\circ$, S = 3.528217
Tx=-32 Ty=35

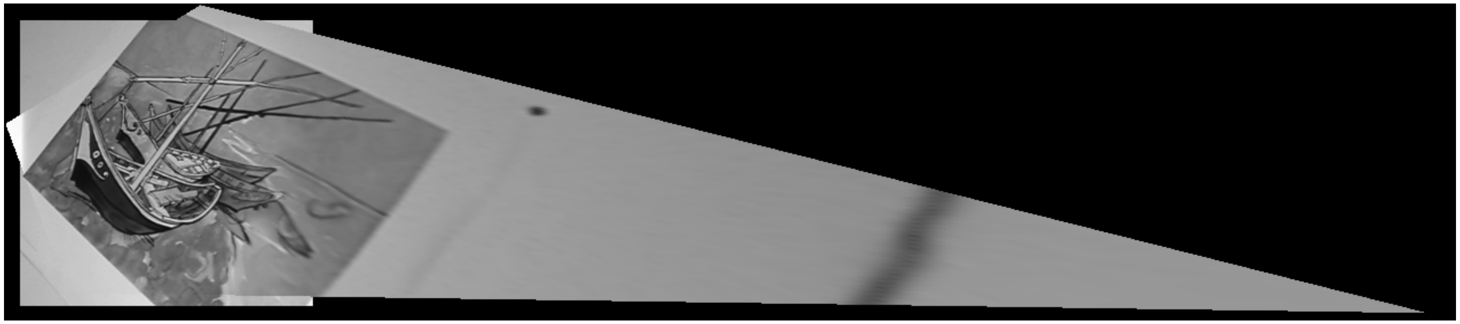


Example: Large Perspective (1)



Example: Large Perspective (2)





Summary

- Log-polar registration:
  - recovers large-scale rotation/scale/translation
  - applies fast correlation in log-polar space
  - exploits fast multiresolution processing
  - initial estimate for affine/perspective registration
- Perspective registration:
  - handles small distortions with subpixel accuracy
  - applies Levenberg-Marquardt algorithm
  - nonlinear least squares optimization
  - exploits fast multiresolution processing