Are you getting the whole picture?

- Compact Camera FOV = 50 x 35°
Are you getting the whole picture?

- Compact Camera FOV = 50 x 35°
- Human FOV = 200 x 135°
Are you getting the whole picture?

- Compact Camera FOV = 50 x 35°
- Human FOV = 200 x 135°
- Panoramic Mosaic = 360 x 180°
Panorama
A wide-angle representation of the scene

Panorama of Along the River During Qingming Festival
18th century remake of a 12th century original by Chinese artist Zhang Zeduan

Image from Wikipedia
Panorama: Cinema for the early 19th century

Burford’s Panorama, Leicester Square, London, 1801

Painting by Robert Mitchell
Panoramas with wide-angle optics

http://www.0-360.com

AF DX Fisheye–NIKKOR 10.5mm f/2.8G ED
Rotation cameras

Idea
- rotate camera or lens so that a vertical slit is exposed

Swing lens
- rotate the lens and a vertical slit (or the sensor)
- typically can get 110-140 degree panoramas
- Widelux, Seitz, ...

Full rotation
- whole camera rotates
- can get 360 degree panoramas
- Panoscan, Roundshot, ...
Swing-lens panoramic images

San Francisco in ruins, 1906

101 Ranch, Oklahoma, circa 1920
Flatback panoramic camera

Lee Frost, Val D’Orcia, Tuscany, Italy

NVIDIA Research

Wednesday, February 15, 12
Disposable panoramic camera
wide-angle lens, limited vertical FOV
Stitching images to make a mosaic
Stitching images to make a mosaic

Given a set of images that should stitch together by rotating the camera around its center of perspective:

1. **Step 1**: Find corresponding features in a pair of images.
2. **Step 2**: Compute transformation from 2nd to 1st image.
3. **Step 3**: Warp 2nd image so it overlays 1st image.
4. **Step 4**: Blend images where they overlap one another.

repeat for 3rd image and mosaic of first two, etc.
Aligning images: Translation?

Translations are not enough to align the images

left on top

right on top
A pencil of rays contains all views

Can generate any synthetic camera view as long as it has **the same center of projection**!

... and scene geometry does not matter ...
Reprojecting an image onto a different picture plane

the sidewalk art of Julian Beever

the view on any picture plane can be projected onto any other plane in 3D without changing its appearance as seen from the center of projection
The mosaic has a natural interpretation in 3D
- the images are reprojected onto a common plane
- the mosaic is formed on this plane
- mosaic is a *synthetic wide-angle camera*
Which transform to use?

Translation 2 unknowns

Affine 6 unknowns

Perspective 8 unknowns

NVIDIA Research

Wednesday, February 15, 12
Homography

- Projective mapping between any two PPs with the same center of projection
  - rectangle should map to arbitrary quadrilateral
  - parallel lines aren’t
  - but must preserve straight lines

is called a Homography

\[
\begin{bmatrix}
w x' \\
w y' \\
w
\end{bmatrix} =
\begin{bmatrix}
h_{11} & h_{12} & h_{13} \\
h_{21} & h_{22} & h_{23} \\
h_{31} & h_{32} & h_{33}
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
1
\end{bmatrix}
\]

To apply a homography \( H \)

- compute \( p' =Hp \) \( \text{(regular matrix multiply)} \)
- convert \( p' \) from homogeneous to image coordinates \([x', y']\) \( \text{(divide by } w) \)
Homography from mapping quads

Figure 2.8: Quadrilateral to quadrilateral mapping as a composition of simpler mappings.
Homography from \( n \) point pairs \((x,y ; x',y')\)

- Multiply out
  \[
  \begin{align*}
  wx' &= h_{11} x + h_{12} y + h_{13} \\
  wy' &= h_{21} x + h_{22} y + h_{23} \\
  w &= h_{31} x + h_{32} y + h_{33}
  \end{align*}
  \]

- Get rid of \( w \)
  \[
  \begin{align*}
  (h_{31} x + h_{32} y + h_{33})x' - (h_{11} x + h_{12} y + h_{13}) &= 0 \\
  (h_{31} x + h_{32} y + h_{33})y' - (h_{21} x + h_{22} y + h_{23}) &= 0
  \end{align*}
  \]

- Create a new system \( Ah = 0 \)
  Each point constraint gives two rows of \( A \)
  \[
  \begin{bmatrix}
  -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x'
  \\
  0 & 0 & 0 & -x & -y & -1 & xy' & yy' & y'
  \end{bmatrix}
  \]

- Solve with singular value decomposition of \( A = USV^T \)
  Solution is in the nullspace of \( A \)
  the last column of \( V \) (= last row of \( V^T \))

\( \begin{bmatrix} w x' \\ w y' \\ w \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \)
from numpy import *

# create 4 random homogen. points
X = ones([3,4])  # the points are on columns
X[2, :] = random.rand(2,4)  # first row x coord, second y coord, third w = 1
x, y = X[0], X[1]

# create projective matrix
H = random.rand(3,3)

# create the target points
Y = dot(H, X)

# homogeneous division
YY = (Y / Y[2])[:, 2, :]
u, v = YY[0], YY[1]

A = zeros([8, 9])
for i in range(4):
    A[2*i] = [-x[i], -y[i], -1, 0, 0, 0, x[i] * u[i], y[i] * u[i], u[i]]
    A[2*i+1] = [0, 0, 0, -x[i], -y[i], -1, x[i] * v[i], y[i] * v[i], v[i]]

[u, s, vt] = linalg.svd(A)

# reorder the last row of vt to 3x3 matrix
HH = vt[-1, :].reshape([3, 3])

# test that the matrices are the same (within a multiplicative factor)
print H - HH * (H[2,2] / HH[2,2])
Summary of perspective stitching

- Pick one image, typically the central view (red outline)
- Warp the others to its plane
- Blend
Example

perspective reprojection

Pics: Marc Levoy

common picture plane of mosaic image
Using 4 shots instead of 3
Back to 3 shots

NVIDIA Research

cylindrical reprojection

surface of

Wednesday, February 15, 12
Back to 3 shots

cylindrical reprojection

surface of
Back to 3 shots

perspective reprojection
Cylindrical panoramas

- What if you want a 360° panorama?
- Project each image onto a cylinder
- A cylindrical image is a rectangular array
Cylindrical panoramas

What if you want a 360° panorama?

Project each image onto a cylinder
A cylindrical image is a rectangular array
To view without distortion
reproject a portion of the cylinder onto a picture plane representing the display screen
2\textsuperscript{nd} reprojection to a plane for display

Imagine photographing the inside of a cylinder that is wallpapered with this panorama

- if your FOV is narrow, your photo won’t be too distorted
Demo

http://graphics.stanford.edu/courses/cs178/applets/projection.html
Changing camera center

Does it still work?

PP1
PP2
synthetic PP

Wednesday, February 15, 12
Where to rotate? Nodal point?

http://www.reallyrightstuff.com/pano/index.html

If you aim a ray at one of the nodal points, it will be refracted by the lens so it appears to have come from the other, and with the same angle with respect to the optical axis.
Rotate around center of lens perspective

Many instructions say rotate around the nodal point

- wrong! http://toothwalker.org/optics/misconceptions.html#m6

Correct: the entrance pupil

- the optical image of the physical aperture stop as 'seen' through the front of the lens
- due to the magnifying effect of the front lens, the entrance pupil's location is nearer than that of the physical aperture
Test for parallax

Figure 3. Configuration to reveal the presence or absence of parallax. The subject is first placed at the left side of the frame, and subsequently at the right side after rotation of the camera about a vertical axis with the help of a panoramic tripod head.

http://toothwalker.org/optics/cop.html#stitching
Correct center of rotation → no parallax

Figure 4. Rotation about an axis through the entrance pupil.

Figure 5. Rotation about an axis through the front nodal point.
Cylindrical projection

- Map 3D point \((X, Y, Z)\) onto cylinder
  \[
  (\tilde{x}, \tilde{y}, \tilde{z}) = \frac{1}{\sqrt{X^2 + Z^2}} (X, Y, Z)
  \]

- Convert to coordinates on unit cylinder

- Convert to image coordinates on unwrapped cylinder
  \[
  (\tilde{x}, \tilde{y}) = (f\theta, fh) + (\tilde{x}_c, \tilde{y}_c)
  \]
Cylindrical projection
Focal length affects warping

Image 384x300
f = 180 (pixels)

f = 280
f = 380
Focal length is (very!) camera dependent

- Can get a rough estimate by measuring the FOV:
  - if the sensor size is known…

- Can use the EXIF tag
  - might not give the correct value

- Can use several images together
  - find $f$ that makes them match

- Etc.
Assembling the panorama

Stitch pairs together, blend, then crop
Problem: Drift

- **Vertical Error accumulation**
  - small (vertical) errors accumulate over time
  - apply correction so that sum = 0 (for 360° panorama)

- **Horizontal Error accumulation**
  - can reuse first/last image to find the right panorama radius
Spherical projection

- Map 3D point \((X,Y,Z)\) onto sphere

\[
(\hat{x}, \hat{y}, \hat{z}) = \frac{1}{\sqrt{X^2 + Y^2 + Z^2}} (X,Y,Z)
\]

- Convert to spherical coordinates

\[
(\sin\theta \cos\phi, \sin\phi, \cos\theta \cos\phi) = (\hat{x}, \hat{y}, \hat{z})
\]

- Convert to spherical image coordinates

\[
(\bar{x}, \bar{y}) = (f\theta, f\phi) + (\bar{x}_c, \bar{y}_c)
\]

unwrapped sphere
Spherical Projection
We need to match (align) images
Detect feature points in both images
Find corresponding pairs
Use these pairs to align images
Problem 1:

Detect the *same* point *independently* in both images

*no chance to match!*

**We need a repeatable detector**
Problem 2:
For each point correctly recognize the corresponding one

We need a reliable and distinctive descriptor
Harris Corners: The Basic Idea

- We should easily recognize the point by looking through a small window
- Shifting a window in any direction should give a large change in intensity
Harris Detector: Basic Idea

“flat” region: no change in all directions

“edge”: no change along the edge direction

“corner”: significant change in all directions
Harris Detector: Mathematics

Window-averaged change of intensity for the shift \([u,v]\\):

\[
E(u, v) = \sum_{x,y} w(x, y) \left[ I(x + u, y + v) - I(x, y) \right]^2
\]

Window function \(w(x, y) = \) 1 in window, 0 outside

Shifted intensity

Intensity

Window function \(w(x, y) = \)

1 in window, 0 outside

or

Gaussian
Harris Detector: Mathematics

Expanding $E(u,v)$ in a 2nd order Taylor series expansion, we have, for small shifts $[u,v]$, a bilinear approximation:

$$E(u, v) \approx [u, v] \begin{bmatrix} M \\ u \\ v \end{bmatrix}$$

where $M$ is a $2 \times 2$ matrix computed from image derivatives:

$$M = \sum_{x, y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$
Eigenvalues $\lambda_1, \lambda_2$ of $M$ at different locations

$\lambda_1$ and $\lambda_2$ are large
Eigenvalues $\lambda_1, \lambda_2$ of $M$ at different locations

large $\lambda_1$, small $\lambda_2$
Eigenvalues $\lambda_1$, $\lambda_2$ of $M$ at different locations

small $\lambda_1$, small $\lambda_2$
Classification of image points using eigenvalues of $M$:

- $\lambda_1$ and $\lambda_2$ are small; $E$ is almost constant in all directions
- $\lambda_1$ and $\lambda_2$ are large,
  $\lambda_1 \sim \lambda_2$; $E$ increases in all directions
- $\lambda_1 \gg \lambda_2$ (Edge)
- $\lambda_2 \gg \lambda_1$ (Corner)
Harris Detector: Mathematics

Measure of corner response:

\[ R = \det M - k \left( \text{trace } M \right)^2 \]

\[ M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \]

\[ \det M = \lambda_1 \lambda_2 \]

\[ \text{trace } M = \lambda_1 + \lambda_2 \]

\( k \) - empirical constant, \( k = 0.04 - 0.06 \)
Harris Detector: Mathematics

- $R$ depends only on eigenvalues of $M$
- $R$ is large for a corner
- $R$ is negative with large magnitude for an edge
- $|R|$ is small for a flat region
Harris Detector: Workflow

Compute corner response R
Harris Detector: Workflow

Find points with large corner response: \( R > \text{threshold} \)
Harris Detector: Workflow

Take only the points of local maxima of $R$
Harris Detector: Workflow
Harris Detector: Summary

- Average intensity change in direction $[u, v]$ can be expressed as a bilinear form:

$$E(u, v) \equiv [u, v] M \begin{bmatrix} u \\ v \end{bmatrix}$$

- Describe a point in terms of eigenvalues of $M$: measure of corner response

$$R = \lambda_1 \lambda_2 - k \left( \lambda_1 + \lambda_2 \right)^2$$

- A good (corner) point should have a large intensity change in all directions, i.e., $R$ should be large positive
Harris Detector: Invariant to rotation

Ellipse rotates but its shape (i.e., eigenvalues) remains the same

Corner response $R$ is invariant to image rotation
Almost invariant to intensity change

Partial invariance

✓ Only derivatives are used

=>

invariance to intensity shift \( I \rightarrow I + b \)

✓ Intensity scale: \( I \rightarrow a \cdot I \)
Not invariant to image scale!

All points will be classified as edges

Corner!
Point Descriptors

- We know how to detect points
- Next question:
  How to match them?

Point descriptor should be:
1. Invariant
2. Distinctive
Figure 12: The training images for two objects are shown on the left. These can be recognized in a cluttered image with extensive occlusion, shown in the middle. The results of recognition are shown on the right. A parallelogram is drawn around each recognized object showing the boundaries of the original training image under the affine transformation solved for during recognition. Smaller squares indicate the keypoints that were used for recognition.
Descriptor overview:

- Determine **scale** (by maximizing DoG in scale and in space), **local orientation** as the dominant gradient direction
- Use this scale and orientation to make all further computations invariant to scale and rotation
Descriptor overview:

- Determine **scale** (by maximizing DoG in scale and in space), **local orientation** as the dominant gradient direction
- Use this scale and orientation to make all further computations invariant to scale and rotation
- Compute **gradient orientation histograms** of several small windows (128 values for each point)
- Normalize the descriptor to make it invariant to intensity change
Registration in practice: tracking

Camera Module

Video Frames

Real-Time Tracking

Current location
Viewfinder alignment for tracking

Andrew Adams, Natasha Gelfand, Kari Pulli

Viewfinder Alignment
Eurographics 2008

http://graphics.stanford.edu/papers/viewfinderalignment/
Project gradients along columns and rows
... diagonal gradients along diagonals ...
... and find corners
Overlap and match the gradient projections
Apply the best translation to corners
Match corners, refine translation & rotation
System Overview

Camera Module

Video Frames

Real-Time Tracking

High Resolution Images

Current location

Panorama expansion

time
Hybrid multi-resolution registration

Initial guess

I.B. Image Based
F.B. Feature Based

Registration parameters

Progression of multi-resolution registration

Actual size

Applied to hi-res
Feature-based registration

Feature Detection (Harris corners) → Feature Matching (spherical coordinates) → RANSAC → Validity check

- Previous estimate
- Update search range
- Convert to spherical coordinates
- Apply the previous registration estimate
- Convert from spherical coordinates
- Best block cross-correlation match
- New estimate

Wednesday, February 15, 12
System overview

Camera Module

Real-Time Tracking

Video Frames

High Resolution Images

Image Warping

Image Registration
Image blending

- Directly averaging the overlapped pixels results in ghosting artifacts
  - Moving objects, errors in registration, parallax, etc.

Photo by Chia-Kai Liang
Solution: Image labeling

Assign one input image each output pixel

Optimal assignment can be found by graph cut [Agarwala et al. 2004]
New artifacts

- Inconsistency between pixels from different input images
  - Different exposure/white balance settings
  - Photometric distortions (e.g., vignetting)
Solution: Poisson blending

- Copy the gradient field from the input image
- Reconstruct the final image by solving a Poisson equation
Seam finding is difficult when colors differ

No color correction

With color correction

Alpha blending

After labeling

Poisson blending
System Overview

Camera Module

Video Frames

Real-Time Tracking

High Resolution Images

Image Warping

Image Registration

Image Blending

Preview on Phone

Final Panorama

Photo by Marius Tico

NVIDIA Research

Wednesday, February 15, 12
Panorama Visualization

Trivial method:
- Show the whole panorama on the screen
- Zooming and panning
No Projection Method is Optimal

Zoom

Spherical Projection

Perspective Projection
Interpolate the Projection Coordinates

Weights are determined by a sigmoid function of zoom factor

Wednesday, February 15, 12
Capturing and Viewing Gigapixel Images

Johannes Kopf \(^1,2\)
Matt Uyttendaele \(^1\)
Oliver Deussen \(^2\)
Michael Cohen \(^1\)

\(^1\) Microsoft Research
\(^2\) Universität Konstanz
3,600,000,000 Pixels

Created from about 800 8 MegaPixel Images
BIG
Wide

“Normal” perspective projections cause distortions.
Deep

100X variation in Radiance

High Dynamic Range
Capture
Capturing Gigapixel Images
DeVignette
White Balance
Feature Points
Feature Matches
Radiometric Alignment

1 / 1000th of a second
1 / 10th of a second

High Dynamic Range
Radiometric Alignment

Laplacian Blend
Radiometric Alignment

Poisson Blend
Radiometric Alignment

Pure Radiometric
Radiometric Alignment

High Dynamic Range
Tile Pyramid
Photographer Alfred Zhao captured this 272 gigapixel image of the Shanghai skyline using the GigaPan EPIC Pro and a Canon 7D with a 400mm f/5.6 lens and 2x teleconverter attached. He was setup and started shooting at around 8:30am and after 12,000 images were in the bag, it was just before dusk. It took months to complete image and get the final 1.09TB file uploaded.

Just how big is a 272 gigapixel image? 1 gigapixel = 1000 megapixels = 1 billion pixels. That’s 272 BILLION pixels. Printed at standard resolution, this image would cover over 7000 billboards.

But now it’s done and Zhao holds a world record for the largest digital photo. There’s no time to rest though, as Zhao says, “This is not the end of my panorama journey, it is a new start, challenging the limit is an infinite process. New records will appear in the future, it is only a matter of time.”
Optimizing Content-Preserving Projections for Wide-Angle Images

Robert Carroll  
University of California, Berkeley

Maneesh Agrawala  
University of California, Berkeley

Aseem Agarwala  
Adobe Systems, Inc.

Figure 1: Wide-angle photographs can appear badly distorted under existing projections, such as the perspective, Mercator and stereographic projections. Perspective projection preserves linear structures in the scene, but distorts shapes of objects. Mercator and stereographic projections preserve shapes locally, but bend linear structures. Our projection is designed to both preserve local shape and maintain straight scene lines that are marked by the user with our interactive tool.
Push broom / slit scan panoramas
Photographing long scenes with multi-viewpoint panoramas

Aseem Agarwala\textsuperscript{1}    Maneesh Agrawala\textsuperscript{4}    Michael Cohen\textsuperscript{3}    David Salesin\textsuperscript{1,2}    Rick Szeliski\textsuperscript{3}

\textsuperscript{1}\textit{University of Washington}    \textsuperscript{2}\textit{Adobe Systems}    \textsuperscript{3}\textit{Microsoft Research}    \textsuperscript{4}\textit{UC Berkeley}

Abstract

We present a system for producing multi-viewpoint panoramas of long, roughly planar scenes, such as the facades of buildings along a city street, from a relatively sparse set of photographs captured with a handheld still camera that is moved along the scene. Our work is a significant departure from previous methods for creating multi-viewpoint panoramas, which composite thin vertical strips from a video sequence captured by a translating video camera, in that the resulting panoramas are composed of relatively large regions of ordinary perspective. In our system, the only user input required beyond capturing the photographs themselves is to identify the dominant plane of the photographed scene; our system then computes a panorama automatically using Markov Random Field optimization. Users may exert additional control over the appearance of the result by drawing rough strokes that indicate various high-level goals. We demonstrate the results of our system on several scenes, including urban streets, a river bank, and a grocery store aisle.