Image Blending & Compositing

Admin

- Please fill in feedback sheets
- Assignment 2 due today
  - Can have extension until Wed. if you need it
  - But MUST be in by then
  - I need to submit mid-term grades

Overview

- Image blending & compositing
  - Poisson blending
  - Cutting images (GraphCuts)
- Panoramas
  - RANSAC/Homographies
  - Brown and Lowe '03

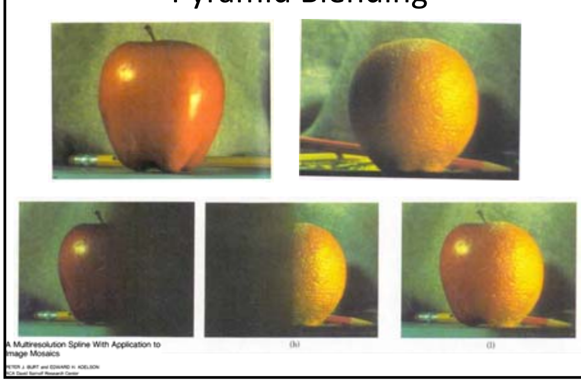
Overview

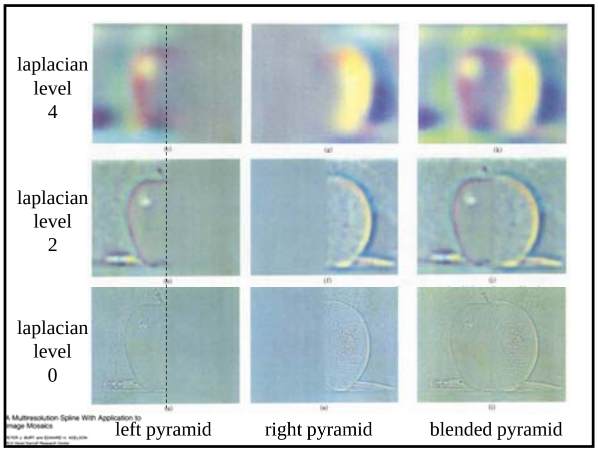
- Image blending & compositing
  - Poisson blending
  - Cutting images (GraphCuts)
- Panoramas
  - RANSAC/Homographies
  - Brown and Lowe '03

Image Blending -- Recap

- Pyramid blending
  - Multi-scale decomposition of image
  - Scale of feathering given by Gaussian pyramid of mask
  - In assignment 2

Pyramid Blending





Overview

- Image blending & compositing
  - Poisson blending
  - Cutting images (GraphCuts)
- Panoramas
  - RANSAC/Homographies
  - Brown and Lowe '03

Gradient manipulation

Idea:

- **Human visual system is very sensitive to gradient**
- **Gradient encode edges and local contrast quite well**

- **Do your editing in the gradient domain**
- **Reconstruct image from gradient**

∇

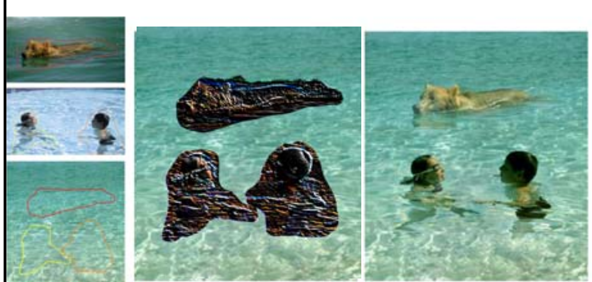
- **Various instances of this idea, I'll mostly follow Perez et al. Siggraph 2003 http://research.microsoft.com/vision/cambridge/papers/perez_siggraph03.pdf**

Slide credit: F. Durand

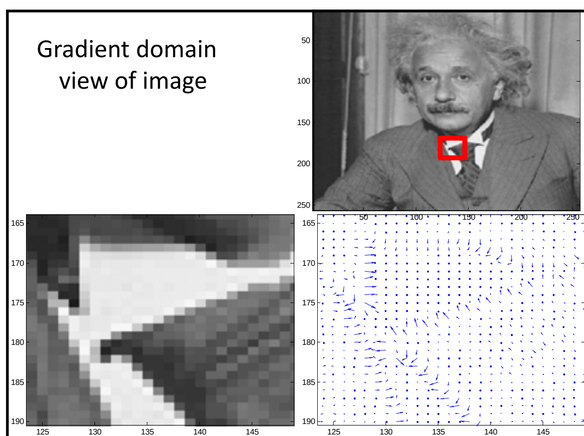
Cloning of intensities


sources/destinations   cloning

Gradient domain cloning


sources/destinations   cloning   seamless cloning

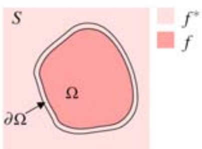
Gradient domain view of image

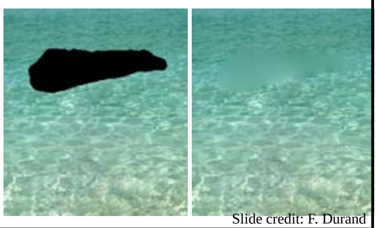


Membrane interpolation

- **Laplace equation (a.k.a. membrane equation)**

$$\min_f \iint_\Omega |\nabla f|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

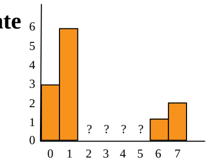




Slide credit: F. Durand

1D example: minimization

- **Minimize derivatives to interpolate**



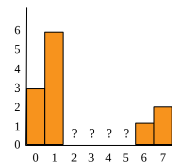
- **Min $(f_2-f_1)^2$**
- **Min $(f_3-f_2)^2$**
- **Min $(f_4-f_3)^2$**
- **Min $(f_5-f_4)^2$**
- **Min $(f_6-f_5)^2$**

With $f_1=6$ $f_6=1$

Slide credit: F. Durand

1D example: derivatives

- **Minimize derivatives to interpolate**



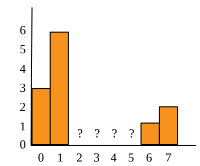
Min $(f_2^2+36-12f_2$
$+ f_3^2+f_2^2-2f_3f_2$
$+ f_4^2+f_3^2-2f_3f_4$
$+ f_5^2+f_4^2-2f_5f_4$
$+ f_5^2+1-2f_5)$
Denote it Q

$$\frac{dQ}{df_2} = 2f_2 + 2f_2 - 2f_3 - 12$$
$$\frac{dQ}{df_3} = 2f_3 - 2f_2 + 2f_3 - 2f_4$$
$$\frac{dQ}{df_4} = 2f_4 - 2f_3 + 2f_4 - 2f_5$$
$$\frac{dQ}{df_5} = 2f_5 - 2f_4 + 2f_5 - 2$$

Slide credit: F. Durand

1D example: set derivatives to zero

- **Minimize derivatives to interpolate**



$$\frac{dQ}{df_2} = 2f_2 + 2f_2 - 2f_3 - 12$$
$$\frac{dQ}{df_3} = 2f_3 - 2f_2 + 2f_3 - 2f_4$$
$$\frac{dQ}{df_4} = 2f_4 - 2f_3 + 2f_4 - 2f_5$$
$$\frac{dQ}{df_5} = 2f_5 - 2f_4 + 2f_5 - 2$$

==>

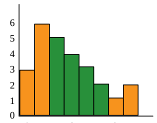
$$\begin{pmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 4 \end{pmatrix} \begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 12 \\ 0 \\ 0 \\ 2 \end{pmatrix}$$

Slide credit: F. Durand

1D example

- **Minimize derivatives to interpolate**
- **Pretty much says that second derivative should be zero**

  (-1 2 -1)
  is a second derivative filter



$$\begin{pmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 4 \end{pmatrix} \begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 12 \\ 0 \\ 0 \\ 2 \end{pmatrix} \qquad \begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 5 \\ 4 \\ 3 \\ 2 \end{pmatrix}$$

Slide credit: F. Durand

Membrane interpolation

- **Laplace equation (a.k.a. membrane equation)**

$$\min_f \iint_\Omega |\nabla f|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

- **Mathematicians will tell you there is an Associated Euler-Lagrange equation:**

$$\Delta f = 0 \text{ over } \Omega \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

  – Where the Laplacian Δ is similar to -1 2 -1in 1D
- **Kind of the idea that we want a minimum, so we kind of derive and get a simpler equation**

Slide credit: F. Durand

Seamless Poisson cloning

- Given vector field *v* (pasted gradient), find the value of *f* in unknown region that optimize:
- Previously, v was null

$$\min_f \iint_\Omega |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

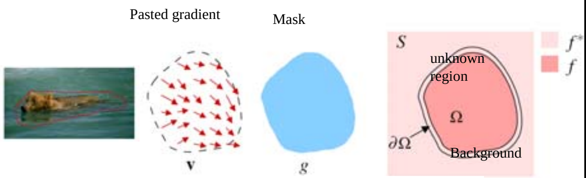


Figure 1: **Guided interpolation notations.** Unknown function f interpolates in domain Ω the destination function f^*, under guidance of vector field $\mathbf{v}$, which might be or not the gradient field of a source function g.

Slide credit: F. Durand

What if *v* is not null: 2D

- Variational minimization (integral of a functional) with boundary condition

$$\min_f \iint_\Omega |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega},$$

- Euler-Lagrange equation:

$$\Delta f = \operatorname{div}\mathbf{v} \text{ over } \Omega, \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

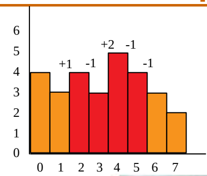
where $\operatorname{div}\mathbf{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$ is the divergence of $\mathbf{v} = (u, v)$

- (Compared to Laplace, we have replaced $\Delta = 0$ by $\Delta = \text{div}$)

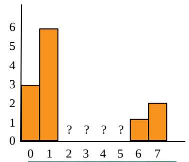
Slide credit: F. Durand

Discrete 1D example: minimization

- Copy



to



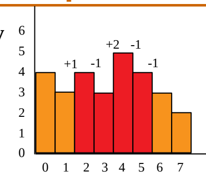
- Min $((f_2-f_1)-1)^2$
- Min $((f_3-f_2)-(-1))^2$
- Min $((f_4-f_3)-2)^2$
- Min $((f_5-f_4)-(-1))^2$
- Min $((f_6-f_5)-(-1))^2$

With
$f_1=6$
$f_6=1$

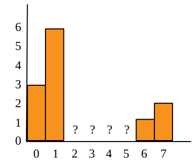
Slide credit: F. Durand

1D example: minimization

- Copy



to

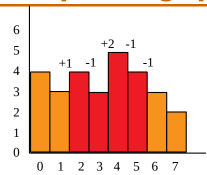


- Min $((f_2-6)-1)^2$ ==> $f_2^2+49-14f_2$
- Min $((f_3-f_2)-(-1))^2$ ==> $f_3^2+f_2^2+1-2f_3f_2 +2f_3-2f_2$
- Min $((f_4-f_3)-2)^2$ ==> $f_4^2+f_3^2+4-2f_3f_4 -4f_4+4f_3$
- Min $((f_5-f_4)-(-1))^2$ ==> $f_5^2+f_4^2+1-2f_5f_4 +2f_5-2f_4$
- Min $((1-f_5)-(-1))^2$ ==> $f_5^2+4-4f_5$

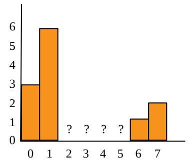
Slide credit: F. Durand

1D example: big quadratic

- Copy



to

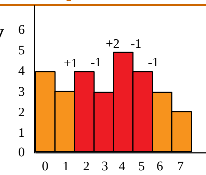


- Min ($f_2^2+49-14f_2$
  $+ f_3^2+f_2^2+1-2f_3f_2 +2f_3-2f_2$
  $+ f_4^2+f_3^2+4-2f_3f_4 -4f_4+4f_3$
  $+ f_5^2+f_4^2+1-2f_5f_4 +2f_5-2f_4$
  $+ f_5^2+4-4f_5$)
  Denote it Q

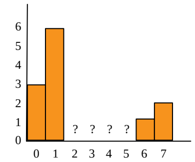
Slide credit: F. Durand

1D example: derivatives

- Copy



to



Min ($f_2^2+49-14f_2$
$+ f_3^2+f_2^2+1-2f_3f_2 +2f_3-2f_2$
$+ f_4^2+f_3^2+4-2f_3f_4 -4f_4+4f_3$
$+ f_5^2+f_4^2+1-2f_5f_4 +2f_5-2f_4$
$+ f_5^2+4-4f_5$)
Denote it Q

$$\frac{dQ}{df_2} = 2f_2 + 2f_2 - 2f_3 - 16$$

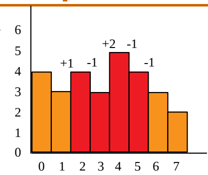
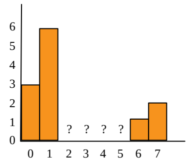
$$\frac{dQ}{df_3} = 2f_3 - 2f_2 + 2 + 2f_3 - 2f_4 + 4$$

$$\frac{dQ}{df_4} = 2f_4 - 2f_3 - 4 + 2f_4 - 2f_5 - 2$$

$$\frac{dQ}{df_5} = 2f_5 - 2f_4 + 2 + 2f_5 - 4$$

Slide credit: F. Durand

1D example: set derivatives to zero

- **Copy**  **to** 

$\frac{dQ}{df_2} = 2f_2 + 2f_2 - 2f_3 - 16$

$\frac{dQ}{df_3} = 2f_3 - 2f_2 + 2 + 2f_3 - 2f_4 + 4$

$\frac{dQ}{df_4} = 2f_4 - 2f_3 - 4 + 2f_4 - 2f_5 - 2$

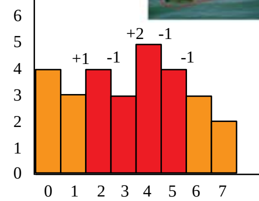
$\frac{dQ}{df_5} = 2f_5 - 2f_4 + 2 + 2f_5 - 4$

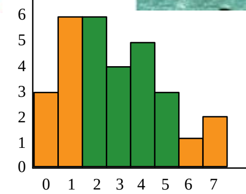
==> $$\begin{pmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 4 \end{pmatrix} \begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 16 \\ -6 \\ 6 \\ 2 \end{pmatrix}$$

Slide credit: F. Durand

1D example

- **Copy**  **to** 



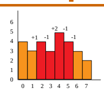
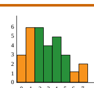


$$\begin{pmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 4 \end{pmatrix} \begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 16 \\ -6 \\ 6 \\ 2 \end{pmatrix}$$

$$\begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 6 \\ 4 \\ 5 \\ 3 \end{pmatrix}$$

Slide credit: F. Durand

1D example: remarks

- **Copy**  **to**  $\begin{pmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 4 \end{pmatrix} \begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 16 \\ -6 \\ 6 \\ 2 \end{pmatrix}$

- **Matrix is sparse**
- **Matrix is symmetric**
- **Everything is a multiple of 2**
  - because square and derivative of square
- **Matrix is a convolution (kernel -2 4 -2)**
- **Matrix is independent of gradient field. Only RHS is**
- **Matrix is a second derivative**

Slide credit: F. Durand

What if *v* is not null: 2D

- **Variational minimization (integral of a functional) with boundary condition**

$$\min_f \iint_\Omega |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega},$$

- **Euler-Lagrange equation:**

$$\Delta f = \text{div}\mathbf{v} \text{ over } \Omega, \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

where $\text{div}\mathbf{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$ is the divergence of $\mathbf{v} = (u, v)$

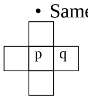
- **(Compared to Laplace, we have replaced Δ =0 by Δ = div)**

Slide credit: F. Durand

Discrete Poisson solver

- **Two approaches:**
  - Minimize variational problem $\min_f \iint_\Omega |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$.
  - Solve Euler-Lagrange equation $\Delta f = \text{div}\mathbf{v} \text{ over } \Omega, \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$

  In practice, variational is best
- **In both cases, need to discretize derivatives**
  - Finite differences over 4 pixel neighbors
  - We are going to work using pairs
    - Partial derivatives are easy on pairs
    - Same for the discretization of v



Slide credit: F. Durand

Discrete Poisson solver

- **Minimize variational problem** $\min_f \iint_\Omega |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$.

$$\min_{f|_\Omega} \sum_{\langle p,q\rangle \cap \Omega \neq \emptyset} (f_p - f_q - v_{pq})^2, \text{ with } f_p = f_p^*, \text{for all } p \in \partial\Omega$$

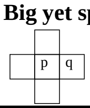
Discretized gradient; (all pairs that are in Ω); Discretized v: g(p)-g(q); Boundary condition

- **Rearrange and call N_p the neighbors of p**

$$\text{for all } p \in \Omega, \quad |N_p| f_p - \sum_{q \in N_p \cap \Omega} f_q = \sum_{q \in N_p \cap \partial\Omega} f_q^* + \sum_{q \in N_p} v_{pq}$$

Only for boundary pixels

- **Big yet sparse linear system**



Slide credit: F. Durand

Discrete Poisson solver

- **Minimize variational problem** $\min_f \iint_\Omega |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$.

Discretized gradient

$$\min_{f|_\Omega} \sum_{\langle p,q \rangle \cap \Omega \neq \emptyset} (f_p - f_q - v_{pq})^2, \text{ with } f_p = f_p^*, \text{for all } p \in \partial\Omega$$

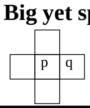
(all pairs that are in Ω) Discretized v: g(p)-g(q) Boundary condition

- **Rearrange and call N_p the neighbors of p**

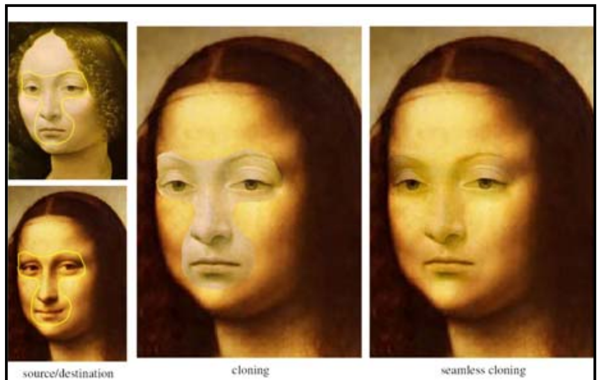
$$\text{for all } p \in \Omega, \quad |N_p| f_p - \sum_{q \in N_p \cap \Omega} f_q = \sum_{q \in N_p \cap \partial\Omega} f_q^* + \sum_{q \in N_p} v_{pq}$$

Only for boundary pixels

- **Big yet sparse linear system**



Slide credit: F. Durand



Perez et al. SIGGRAPH 03

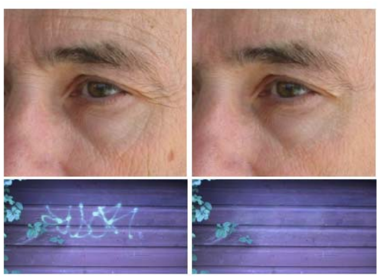


Figure 2: **Concealment**. By importing seamlessly a piece of the background, complete objects, parts of objects, and undesirable artifacts can easily be hidden. In both examples, multiple strokes (not shown) were used.

Perez et al. SIGGRAPH 03

Figure 5: **Monochrome transfer**. In some cases, such as texture transfer, the part of the source color remaining after seamless cloning might be undesirable. This is fixed by turning the source image monochrome beforehand.

Perez et al. SIGGRAPH 03

Seamless Image Stitching in the Gradient Domain*

Anat Levin, Assaf Zomet **, Shmuel Peleg, and Yair Weiss

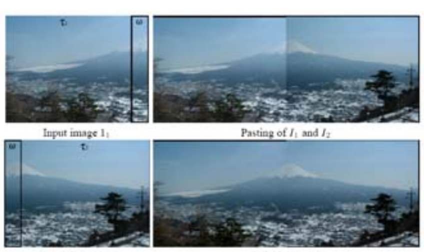


Fig. 1. Image stitching. On the left are the input images. ω is the overlap region. On top right is a simple pasting of the input images. On the bottom right is the result of the GIST1 algorithm.

$$E_p\left(\hat{I}; I_1, I_2, W\right) = d_p(\nabla\hat{I}, \nabla I_1, \tau_1 \cup \omega, W) + d_p(\nabla\hat{I}, \nabla I_2, \tau_2 \cup \omega, U - W) \quad (1)$$

where U is a uniform image, and $d_p(J_1, J_2, \phi, W)$ is the distance between J_1, J_2 on ϕ:

$$d_p(J_1, J_2, \phi, W) = \sum_{q \in \phi} W(q) \| J_1(q) - J_2(q) \|_p^p \quad (2)$$

with $\|\cdot\|_p$ denoting the ℓ_p-norm.

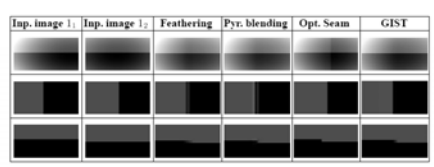
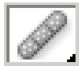


Fig. 2. Comparing stitching methods with various sources for inconsistencies between the input images. The left side of I_1 is stitched to right side of I_2. Optimal seam methods produce a seam artifact in case of photometric inconsistencies between the images (first row). Feathering and pyramid blending produce double edges in case of horizontal misalignments (second row). In case there is a vertical misalignments (third row), the stitching is less visible with Feathering and GIST.

What about Photoshop?

- Healing brush tool 
- Uses Poisson blending

Todor Georgiev
Sr. Research Scientist, Photoshop Group, Adobe Systems

Photoshop healing brush

http://www.photoshopsupport.com/tutorials/jf/retouching-tutorial/remove-wrinkles-healing-brush.html

Covariant Derivative = Perceptual Derivative

$$\frac{\partial}{\partial x} \rightarrow \frac{\partial}{\partial x} + A_x(x,y)$$

$$\frac{\partial}{\partial y} \rightarrow \frac{\partial}{\partial y} + A_y(x,y)$$

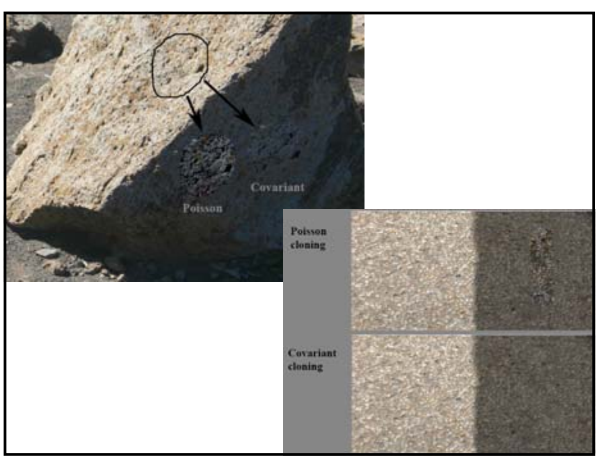
Our covariant Laplace equation:

$$\frac{\triangle f}{f} - 2\frac{grad f}{f}\cdot\frac{grad g}{g} - \frac{\triangle g}{g} + 2\frac{(grad g)\cdot(grad g)}{g^2} = 0$$

Compare this to **Poisson equation**:

$$\triangle f(x,y) = \triangle g(x,y)$$

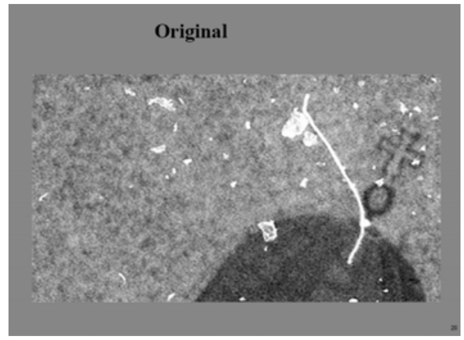
Both define gradient domain cloning. Which one is better?



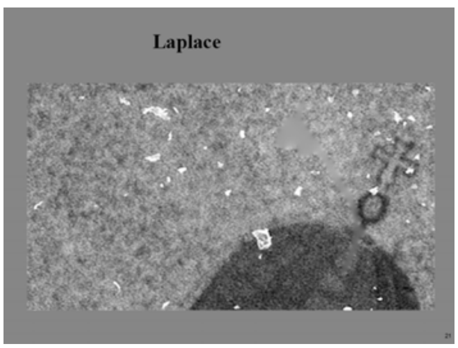
Differences to Laplacian pyramid blending

- No long-range mixing
  - Mixing of pixels at large scale in pyramid
- Gives exact solution to Poisson equation
  - First layer of Laplacian pyramid only gives approximate solution

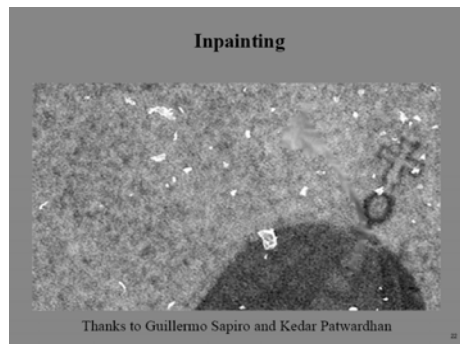
Todor Georgiev
Sr. Research Scientist, Photoshop Group, Adobe Systems



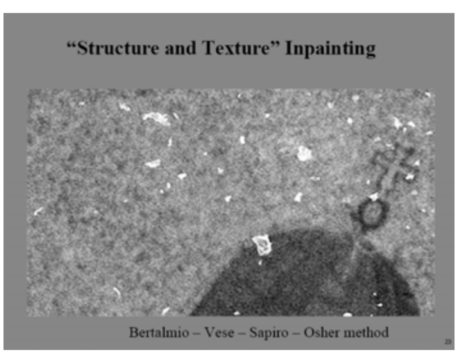
Todor Georgiev
Sr. Research Scientist, Photoshop Group, Adobe Systems



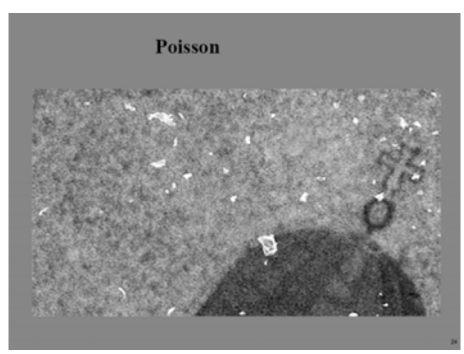
Todor Georgiev
Sr. Research Scientist, Photoshop Group, Adobe Systems



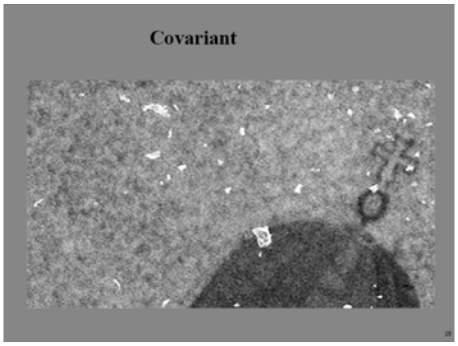
Todor Georgiev
Sr. Research Scientist, Photoshop Group, Adobe Systems



Todor Georgiev
Sr. Research Scientist, Photoshop Group, Adobe Systems



Todor Georgiev
Sr. Research Scientist, Photoshop Group, Adobe Systems



Todor Georgiev
Sr. Research Scientist, Photoshop Group, Adobe Systems

Overview

- Image blending & compositing
  - Poisson blending
  - Cutting images (GraphCuts)
- Panoramas
  - RANSAC/Homographies
  - Brown and Lowe '03

Don't blend, CUT!

Moving objects become ghosts

So far we only tried to blend between two images. What about finding an optimal seam?

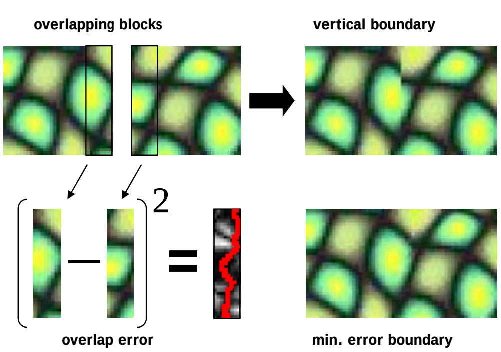
Slide credit: A. Efros

Davis, 1998

Segment the mosaic
- Single source image per segment
- Avoid artifacts along boundries
  - Dijkstra's algorithm

Slide credit: A. Efros

Minimal error boundary

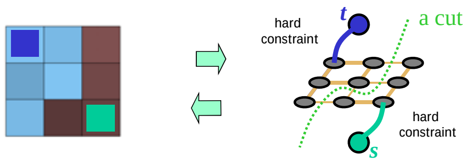


Slide credit: A. Efros

Graphcuts

What if we want similar "cut-where-things-agree" idea, but for closed regions?
- Dynamic programming can't handle loops

Graph cuts
(simple example à la Boykov&Jolly, ICCV'01)



Minimum cost cut can be computed in polynomial time

(max-flow/min-cut algorithms)

Slide credit: A. Efros

Kwatra et al, 2003

Actually, for this example, DP will work just as well…

Slide credit: A. Efros

Lazy Snapping

Interactive segmentation using graphcuts

Slide credit: A. Efros

Seamless Image Stitching in the Gradient Domain*

Anat Levin, Assaf Zomet **, Shmuel Peleg, and Yair Weiss

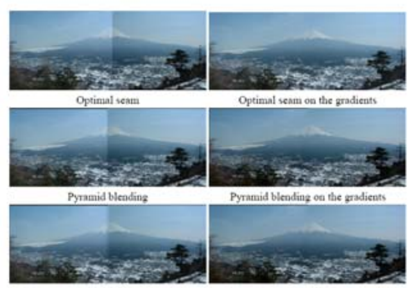


Fig. 3. Stitching in the gradient domain. The input images appear in Figure 1, with the overlap region marked by a black rectangle. With the image domain methods (top panels) the stitching is observable. Gradient-domain methods (bottom panels) overcome global inconsistencies.

Seamless Image Stitching in the Gradient Domain*

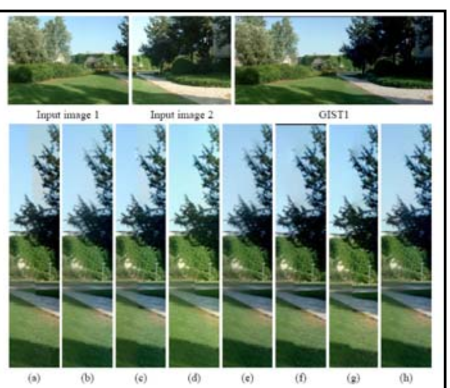


Fig. 4. Comparing various stitching methods. On top are the input image and the result of GIST1 under ℓ_1. The images on bottom are cropped results of various methods. (a)-Optimal seam, (b)-Feathering, (c)-Pyramid blending, (d)-Optimal seam on the gradients, (e)-Feathering on the gradients, (f)-Pyramid blending on the gradients, (g)-Poisson editing [10] and (h) GIST1 - ℓ_1. The

Photomontage video

Interactive Digital Photomontage

Aseem Agarwala[1] Mira Dontcheva[1] Maneesh Agrawala[2] Steven Drucker[2] Alex Colburn[2]
Brian Curless[1] David Salesin[1,2] Michael Cohen[2]
[1]University of Washington [2]Microsoft Research

Figure 1 From a set of five source images (of which four are shown on the left), we quickly create a composite family portrait in which everyone is smiling and looking at the camera (right). We simply flip through the stack and coursely draw strokes using the *designated source* image objective over the people we wish to add to the composite. The user-applied strokes and computed regions are color-coded by the borders of the source images on the left (middle).

Interactive Digital Photomontage

Aseem Agarwala[1] Mira Dontcheva[1] Maneesh Agrawala[2] Steven Drucker[2] Alex Colburn[2]
Brian Curless[1] David Salesin[1,2] Michael Cohen[2]
[1]University of Washington [2]Microsoft Research

Figure 6 We use a set of portraits (first row) to mix and match facial features, to either improve a portrait, or create entirely new people. The faces are first hand-aligned, for example, to place all the noses in the same location. In the first two images in the second row, we replace the closed eyes of a portrait with the open eyes of another. The user paints strokes with the *designated source* objective to specify desired features. Next, we create a fictional person by combining three source portraits. Gradient-domain fusion is used to smooth out skin tone differences. Finally, we show two additional mixed portraits.

Overview

- Image blending & compositing
  - Poisson blending
  - Cutting images (GraphCuts)
- Panoramas
  - RANSAC/Homographies
  - Brown and Lowe '03

Recognising Panoramas

M. Brown and D. Lowe,
University of British Columbia

Introduction

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

Introduction

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

Introduction

- Are you getting the whole picture?
  - Compact Camera FOV = 50 x 35°
  - Human FOV = 200 x 135°
  - Panoramic Mosaic = 360 x 180°

Why "Recognising Panoramas"?

- 1D Rotations (θ)
  - Ordering ⇒ matching images

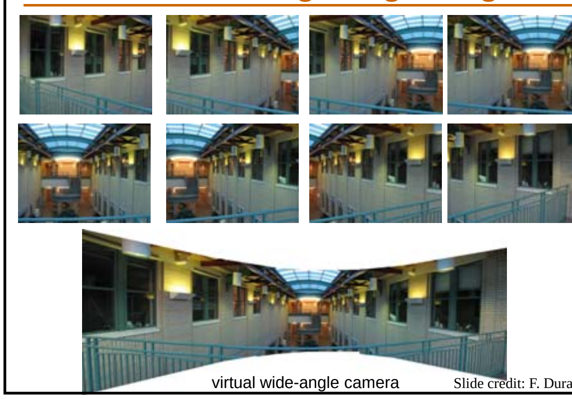
- 2D Rotations (θ, ϕ)
  - Ordering ⇏ matching images

Why "Recognising Panoramas"?

- 1D Rotations (θ)
  - Ordering ⇒ matching images

- 2D Rotations (θ, ϕ)
  - Ordering ⇏ matching images

Mosaics: stitching images together



virtual wide-angle camera

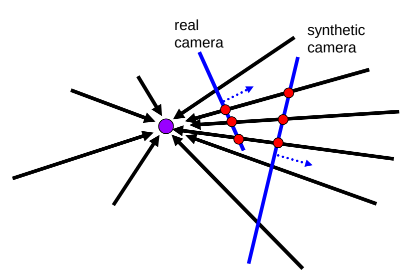
Slide credit: F. Durand

How to do it?

- **Basic Procedure**
  - Take a sequence of images from the same position
    - Rotate the camera about its optical center
  - Compute transformation between second image and first
  - Transform the second image to overlap with the first
  - Blend the two together to create a mosaic
  - If there are more images, repeat
- **…but wait, why should this work at all?**
  - What about the 3D geometry of the scene?
  - Why aren't we using it?

Slide credit: F. Durand

A pencil of rays contains all views



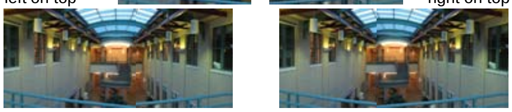
Can generate any synthetic camera view as long as it has **the same center of projection**!

Slide credit: F. Durand

Aligning images: translation

left on top

right on top

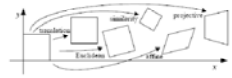


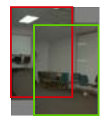
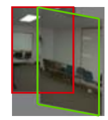
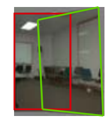
Translations are not enough to align the images

Slide credit: F. Durand

Back to Image Warping

Which t-form is the right one for warping PP1 into PP2?
e.g. translation, Euclidean, affine, projective



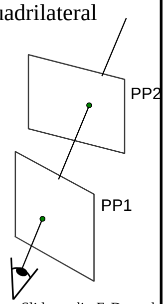
Translation	Affine	Perspective
		
2 unknowns	6 unknowns	8 unknowns

Slide credit: F. Durand

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
  - same as: project, rotate, reproject
- **called Homography**

$$\underset{\mathbf{p'}}{\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix}} = \underset{\mathbf{H}}{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}} \underset{\mathbf{p}}{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}$$



To apply a homography **H**
- Compute **p' = Hp** (regular matrix multiply)
- Convert **p'** from homogeneous to image coordinates

Slide credit: F. Durand

Homography for Rotation

Parameterize each camera by rotation and focal length

$$\mathbf{K}_i = \begin{bmatrix} f_i & 0 & 0 \\ 0 & f_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{R}_i = e^{[\boldsymbol{\theta}_i]_\times}, \; [\boldsymbol{\theta}_i]_\times = \begin{bmatrix} 0 & -\theta_{i3} & \theta_{i2} \\ \theta_{i3} & 0 & -\theta_{i1} \\ -\theta_{i2} & \theta_{i1} & 0 \end{bmatrix}$$

$$\tilde{\mathbf{u}}_i = \mathbf{H}_{ij}\tilde{\mathbf{u}}_j, \; \mathbf{H}_{ij} = \mathbf{K}_i\mathbf{R}_i\mathbf{R}_j^T\mathbf{K}_j^{-1}$$

For small changes in image position: $\mathbf{u}_i = \mathbf{u}_{i0} + \left.\frac{\partial \mathbf{u}_i}{\partial \mathbf{u}_j}\right|_{\mathbf{u}_{i0}} \Delta\mathbf{u}_j$

$$\tilde{\mathbf{u}}_i = \mathbf{A}_{ij}\tilde{\mathbf{u}}_j \qquad \mathbf{A}_{ij} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

Slide credit: F. Durand

Overview

- Feature Matching
  - SIFT Features
  - Nearest Neighbour Matching
- Image Matching
- Bundle Adjustment
- Multi-band Blending
- Results
- Conclusions

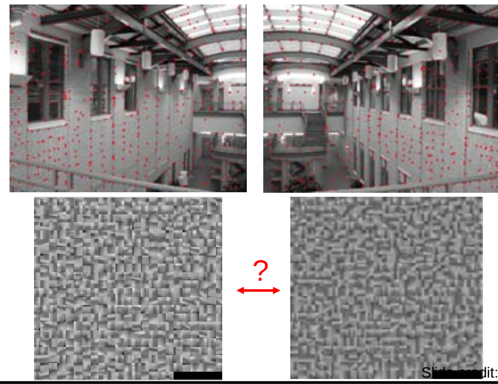
Invariant Features

- Schmid & Mohr 1997, Lowe 1999, Baumberg 2000, Tuytelaars & Van Gool 2000, Mikolajczyk & Schmid 2001, Brown & Lowe 2002, Matas et. al. 2002, Schaffalitzky & Zisserman 2002

SIFT Features

- Invariant Features
  - Establish invariant frame
    - Maxima/minima of scale-space DOG ⇒ x, y, s
    - Maximum of distribution of local gradients ⇒ θ
  - Form descriptor vector
    - Histogram of smoothed local gradients
    - 128 dimensions
- SIFT features are…
  - Geometrically invariant to similarity transforms,
    - some robustness to affine change
  - Photometrically invariant to affine changes in intensity

Feature matching



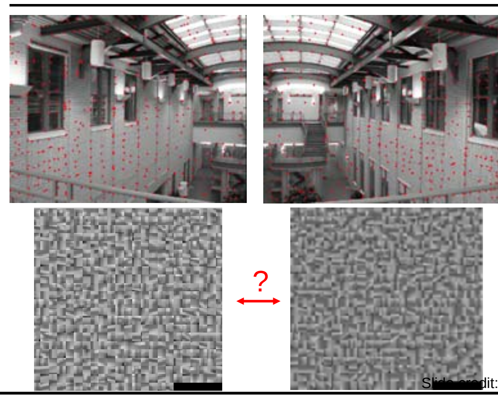
Sl[redacted]dit: A Efros

Feature matching

- Exhaustive search
  - for each feature in one image, look at *all* the other features in the other image(s)
- Hashing
  - compute a short descriptor from each feature vector, or hash longer descriptors (randomly)
- Nearest neighbor techniques
  - *k*-trees and their variants

Slide credit: A Efros

What about outliers?



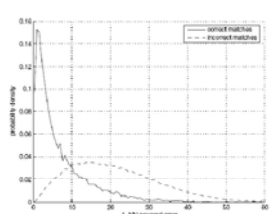
Sl[redacted]dit: A Efros

Feature-space outlier rejection

Let's not match all features, but only these that have "similar enough" matches?

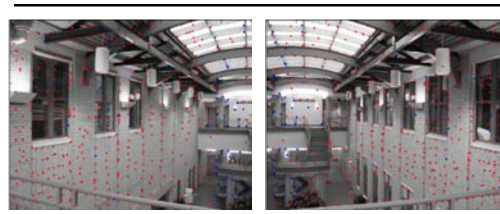
How can we do it?

- SSD(patch1,patch2) < threshold
- How to set threshold?



Slide credit: A Efros

Feature-space outliner rejection



Can we now compute H from the blue points?

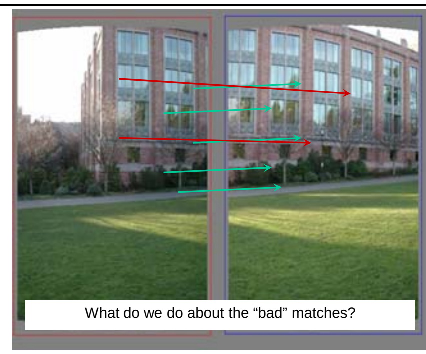
- No! Still too many outliers…
- What can we do?

Slide credit: A Efros

Overview

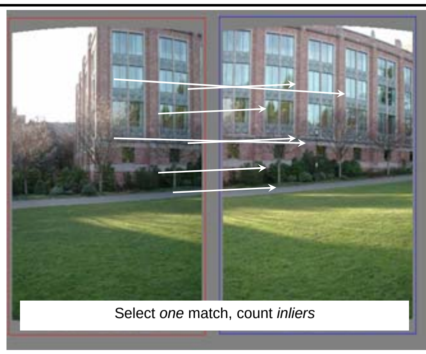
- Feature Matching
- Image Matching
  - RANSAC for Homography
  - Probabilistic model for verification
- Bundle Adjustment
- Multi-band Blending
- Results
- Conclusions

Matching features



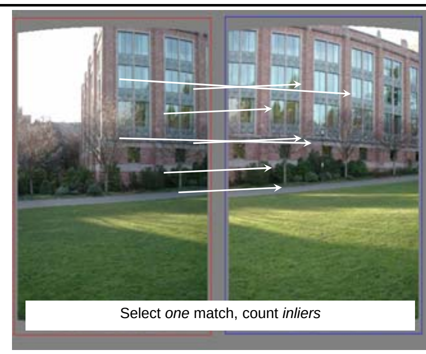
Slide credit: A Efros

RAndom SAmple Consensus



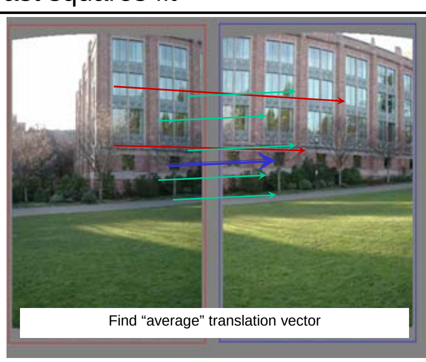
Slide credit: A Efros

RAndom SAmple Consensus



Slide credit: A Efros

Least squares fit



Slide credit: A Efros

RANSAC for estimating homography

$$\underbrace{\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix}}_{\mathbf{p'}} = \underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\mathbf{p}}$$

RANSAC loop:

1. Select four feature pairs (at random)
2. Compute homography H (exact)
3. Compute *inliers* where $SSD(p_i', \mathbf{H}\, p_i) < \varepsilon$
4. Keep largest set of inliers
5. Re-compute least-squares H estimate on all of the inliers

Slide credit: A Efros

RANSAC

Slide credit: A Efros

Overview

- Feature Matching
- Image Matching
- Bundle Adjustment
  - Error function
- Multi-band Blending
- Results
- Conclusions

Error function

- Sum of squared projection errors

$$e = \sum_{i=1}^{n} \sum_{j \epsilon \mathcal{I}(i)} \sum_{k \epsilon \mathcal{F}(i,j)} f(\mathbf{r}_{ij}^k)^2$$

  - n = #images
  - I(i) = set of image matches to image i
  - F(i, j) = set of feature matches between images i,j
  - r_{ij}^k = residual of k^{th} feature match between images i,j

- Robust error function

$$f(\mathbf{x}) = \begin{cases} |\mathbf{x}|, & \text{if } |\mathbf{x}| < x_{max} \\ x_{max}, & \text{if } |\mathbf{x}| \geq x_{max} \end{cases}$$

Bundle Adjustment

- New images initialised with rotation, focal length of best matching image

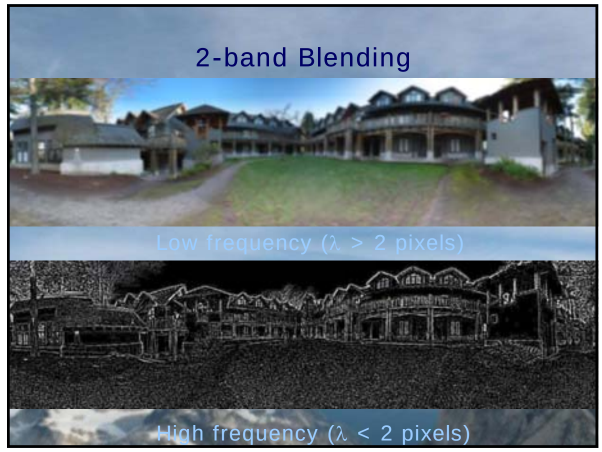
Bundle Adjustment

- New images initialised with rotation, focal length of best matching image

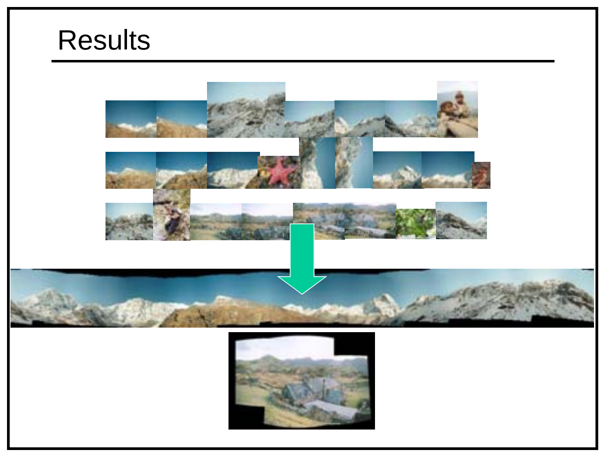
Overview

- Feature Matching
- Image Matching
- Bundle Adjustment
  - Error function
- Multi-band Blending
- Results
- Conclusions


Multi-band Blending
• Burt & Adelson 1983
– Blend frequency bands over range ∝ λ


2-band Blending
Low frequency (λ > 2 pixels)
High frequency (λ < 2 pixels)


Overview
• Feature Matching
• Image Matching
• Bundle Adjustment
– Error function
• Multi-band Blending
• Results
• Conclusions


Results