How was pset 1?

What have we learnt last time?

- Log is good
- Luminance is different from chrominance
- Separate components:
  - Low and high frequencies
- Strong edges are important

Homomorph filtering

- Oppenhein, in the sixties
- Images are the product of illumination and albedo
  - Similarly, many sounds are the product of an envelope and a modulation
- Illumination is usually slow-varying
- Perform albedo-illumination using low-pass filtering of the log image

What's great about the bilateral filter

- Separate image into two components
- Preserve strong edges
- Non-iterative
  - More controllable, stable
- Can be accelerated

Bilateral filtering on meshes

- [Link](http://www.cs.tau.ac.il/~dcor/private_MESHESultybih03.pdf)
- [Link](http://people.csail.mit.edu/thouis/JDD03.pdf)
Questions?

Today: 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

Problems with direct cloning

Solution: clone gradient

Gradients and grayscale images

• Grayscale image: \( n \times n \) scalars
• Gradient: \( n \times n \) 2D vectors
• Overcomplete!
• What’s up with this?
• Not all vector fields are the gradient of an image!
• Only if they are curl-free (a.k.a. conservative)
  – But it does not matter for us

Today message I

• Manipulating the gradient is powerful
Today message II

- Optimization is powerful
  - In particular least square

- Good Least square optimization reduces to a big linear system
  - We are going to spend our time going back and force between minimization and setting derivatives to zero.
  - Your head will spin.
- Linear algebra is your friend
  - Big sparse linear systems can be solved efficiently

Today message III

- Toy examples are good to further understanding
- 1D can however be overly simplifying, n-D is much more complicated

Questions?

Seamless Poisson cloning

- Given vector field \( v \) (pasted gradient), find the value of \( f \) in unknown region that optimizes:

\[
\min_{f} \int_{\Omega} |\nabla f - v|^2 \quad \text{with} \quad f|_{\partial \Omega} = f' |_{\partial \Omega}
\]

Discrete 1D example: minimization

- Copy

\[
\begin{array}{cccccccc}
  0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
  0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\end{array}
\]

- 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 \((f_6-f_5)^2\)

1D example: minimization

- Copy

\[
\begin{array}{cccccccc}
  0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
  0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\end{array}
\]

- Min \(\min((f_2-6)^2)\)
- Min \(\min((f_3-f_2)^2)\)
- Min \(\min((f_4-f_3)^2)\)
- Min \(\min((f_5-6)^2)\)
1D example: big quadratic

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

1D example: derivatives

- Min \((f_2^2+49-14f_2 + f_3^2+f_2^2+1-2f_3f_2 +2f_3-2f_2 + f_4^2+f_3^2+4-2f_4 -4f_4+4f_3 + f_5^2+f_4^2+1-2f_5f_4 +2f_5-2f_4 + f_5^2+4-4f_5)\)

  Denote it \(Q\)

1D example: set derivatives to zero

- \(\frac{\partial Q}{\partial f_2} = 2f_2 + 2f_3 - 2f_2 - 16\)
- \(\frac{\partial Q}{\partial f_3} = 2f_2 - 2f_3 + 2 + 2f_2 - 2f_3 + 4\)
- \(\frac{\partial Q}{\partial f_4} = 3f_4 - 4 + 2f_2 - 2f_2 - 2\)
- \(\frac{\partial Q}{\partial f_5} = 2f_2 - 2f_3 + 2 + 2f_2 - 4\)

  \(\Rightarrow\) 
  \[
  \begin{pmatrix}
  4 & -2 & 0 & 0 \\
  -2 & 4 & -2 & 0 \\
  0 & -2 & 4 & -2 \\
  0 & 0 & -2 & 4
  \end{pmatrix}
  \begin{pmatrix}
  f_2 \\
  f_3 \\
  f_4 \\
  f_5
  \end{pmatrix}
  =
  \begin{pmatrix}
  16 \\
  -6 \\
  -6 \\
  6
  \end{pmatrix}
  \]

1D example: remarks

- 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
Questions?

Let’s try to further analyze

- What is a simple case?

Membrane interpolation

- What if \( v \) is null?
- Laplace equation (a.k.a. membrane equation)
  \[
  \min \int_{\Omega} |\nabla f|^2 \, dx \text{ with } f|_{\partial \Omega} = f^*|_{\partial \Omega}
  \]

1D example: minimization

- Minimize derivatives to interpolate

1D example: derivatives

- Minimize derivatives to interpolate

1D example: set derivatives to zero.

Minimize derivatives to interpolate

\[
\begin{aligned}
\frac{\partial^2}{\partial x^2} f_2 &= 2f_2 - 2f_3 - 12 \\
\frac{\partial^2}{\partial x^2} f_3 &= 2f_3 - 2f_2 + 2f_4 - 2f_1 \\
\frac{\partial^2}{\partial x^2} f_4 &= 2f_4 - 2f_3 + 2f_5 - 2f_2 \\
\frac{\partial^2}{\partial x^2} f_5 &= 2f_5 - 2f_4 + 2f_6 - 2
\end{aligned}
\]

Denote it \( Q \)

\[
\begin{aligned}
\frac{\partial^2}{\partial x^2} f_2 &= 2f_2 + 2f_3 - 2f_3 - 12 \\
\frac{\partial^2}{\partial x^2} f_3 &= 2f_3 - 2f_2 + 2f_5 - 2f_1 \\
\frac{\partial^2}{\partial x^2} f_4 &= 2f_4 - 2f_3 + 2f_4 - 2f_2 \\
\frac{\partial^2}{\partial x^2} f_5 &= 2f_5 - 2f_4 + 2f_6 - 2 \\
\end{aligned}
\]

\[
\begin{pmatrix}
4 & -2 & 0 & 0 \\
-2 & 4 & -2 & 0 \\
0 & -2 & 4 & -2 \\
0 & 0 & -2 & 4
\end{pmatrix}
\begin{pmatrix}
f_2 \\
f_3 \\
f_4 \\
f_5
\end{pmatrix}
= \begin{pmatrix}
12 \\
0 \\
0 \\
2
\end{pmatrix}
\]
**1D example**

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

\((-1 \ 2 \ -1)\)

is a second derivative filter

![Graph showing 1D example](image)

\[
\begin{pmatrix}
  4 & -2 & 0 & 0 \\
  -2 & 4 & -2 & 0 \\
   0 & -2 & 4 & -2 \\
   0 &  0 & -2 & 4
\end{pmatrix}
\begin{pmatrix}
  f_1 \\
  f_2 \\
  f_3 \\
  f_4
\end{pmatrix}
= \begin{pmatrix}
  12 \\
   0 \\
   0 \\
  -2
\end{pmatrix}
\begin{pmatrix}
  f_1 \\
  f_2 \\
  f_3 \\
  f_4
\end{pmatrix}
= \begin{pmatrix}
  5 \\
   4 \\
   3 \\
   2
\end{pmatrix}
\]

**Intuition**

- In 1D: just linear interpolation!
  - The min of \( f' \) is the slope integrated over the interval
- Locally, if the second derivative was not zero, this would mean that the first derivative is varying, which is bad since we want \( f' \) to be minimized
- Note that, in 1D: by setting \( f'' \), we leave two degrees of freedom. This is exactly what we need to control the boundary condition at \( x_1 \) and \( x_2 \)

![Graph showing Intuition](image)

**Membrane interpolation**

- What if \( v \) is null?
- Laplace equation (a.k.a. membrane equation)
  \[
  \min_{\Omega} \int_{\Omega} |\nabla f|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}
  \]
- 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 \( \Delta \) is similar to \((-1 \ 2 \ -1)\) in 1D
- Kind of the idea that we want a minimum, so we kind of derive and get a simpler equation

![Graph showing Membrane interpolation](image)

**Questions?**

- What is \( v \) is not null?

![Questions](image)
What if \( v \) is not null?

• 1D case

Seamlessly paste \( v \) onto \( x_1, x_2 \)

Just add a linear function so that the boundary condition is respected.

(Review) Seamless Poisson cloning

• Given vector field \( v \) (pasted gradient), find the value of \( f \) in unknown region that optimize:

\[
\min_f \int_\Omega |\nabla f - v|^2 \text{ with } f|_{\partial\Omega} = f''|_{\partial\Omega}.
\]

Pasted gradient Mask

unknown region

Poisson equation with Dirichlet conditions

In 2D, if \( v \) is conservative

• If \( v \) is the gradient of an image \( g \)
• Correction function \( \hat{f} \) so that \( f = g + \hat{f} \)
• \( \hat{f} \) performs membrane interpolation over \( \Omega \):

\[
\Delta \hat{f} = 0 \text{ over } \Omega, \quad \hat{f}|_{\partial\Omega} = (f'' - g)|_{\partial\Omega}
\]

In 2D, if \( v \) is NOT conservative

• Also need to project the vector field \( v \) to a conservative field
• And do the membrane thing
• Of course, we do not need to worry about it, it’s all handled naturally by the least square approach.

What if \( v \) is not null: 2D

• Variational minimization (integral of a functional) with boundary condition:

\[
\min_f \int_\Omega |\nabla f - v|^2 \text{ with } f|_{\partial\Omega} = f''|_{\partial\Omega}.
\]

• Euler-Lagrange equation:

\[
\Delta f = \text{div} v \text{ over } \Omega, \quad f|_{\partial\Omega} = f''|_{\partial\Omega}
\]

where \( \text{div} v = \frac{\partial v_1}{\partial x_1} + \frac{\partial v_2}{\partial x_2} \) is the divergence of \( v = (v_1, v_2) \)

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

1D example

• Copy \( f \) to \( \hat{f} \)

Difference Solve Laplace Add Result

In 2D, if \( v \) is conservative

• If \( v \) is the gradient of an image \( g \)
• Correction function \( \hat{f} \) so that \( f = g + \hat{f} \)
• \( \hat{f} \) performs membrane interpolation over \( \Omega \):

\[
\Delta \hat{f} = 0 \text{ over } \Omega, \quad \hat{f}|_{\partial\Omega} = (f'' - g)|_{\partial\Omega}
\]

In 2D, if \( v \) is NOT conservative

• Also need to project the vector field \( v \) to a conservative field
• And do the membrane thing
• Of course, we do not need to worry about it, it’s all handled naturally by the least square approach.
Questions?

Recap

- Find image whose gradient best approximates the input gradient
  - least square Minimization
- Discrete case: turns into linear equation
  - Set derivatives to zero
  - Derivatives of quadratic ==> linear
- Continuous: turns into Euler-Lagrange form
  - $\Delta f = \text{div} v$
- When gradient is null, membrane interpolation
  - Linear interpolation in 1D

Fourier perspective

- Gradient in Fourier?
  - Multiply coeff by $i \omega$
- Parseval theorem?
  - Integral of square is the same in space & frequency
    $\int f(x)^2 \, dx = \int \hat{f}(\omega)^2 \, d\omega$
- Least square on gradient?
  - Least square in Fourier with weight $\omega$
  - Tries to respect high frequencies at the potential cost of low frequencies

Fourier interpretation

- Least square on gradient $\min \int |\nabla f - \nabla v|^2$ with $f|_{\partial Q} = g|_{\partial Q}$
- Parseval anybody?
  - Integral of squared stuff is the same in Fourier and primal
- What is the gradient/derivative in Fourier?
  - Multiply coefficients by frequency and $i$
- Seen in Fourier, Poisson editing does a weighted least square of the image where low frequencies have a small weight and high frequencies a big weight

Questions?

Warning:

- What follows is not strictly necessary to implement Poisson image editing
- But
  - It helps understand the properties of the equation
  - It helps to read the literature
  - It's cool math
Calculus

Simplified version:
- Want to minimize $g(x)$ over the space of real values $x$
- Derive and set $g'(x) = 0$

Now we have a more complex equation; we want to minimize a variational equation over the space of functions $f$

$$\min \int \int_{\Omega} |\nabla f|^2 \text{ with } f|_{\partial \Omega} = f^a|_{\partial \Omega}$$

- It's a complex business to derive wrt functions
  - In general, derivatives are well defined only for functions over 1D domains

Calculus of variation – 1D

- We want to minimize $\int_{x_1}^{x_2} f'(x)^2 \, dx$ with $f(x_1)=a, f(x_2)=b$
- Assume we have a solution $f$

Try to define some notion of 1D derivative wrt to a 1D parameter $\varepsilon$ in a given direction of functional space:

- For a perturbation function $\eta(x)$ that also respects the boundary condition (i.e. $\eta(x_1)=\eta(x_2)=0$) and scalar $\varepsilon$,
  the integral $\int (f'(x)+\varepsilon \eta'(x))^2 \, dx$ should be bigger than for $f'$ alone

Calculus of variation – 1D

$$\int_{x_1}^{x_2} \eta'(x) f'(x) \, dx$$

- How do we get rid of $\eta$? And still include the knowledge that $\eta(x_1)=\eta(x_2)=0$
- When we have an integral of a product and we are playing with derivatives, look into integration by parts
  - Now how do you remember integration by parts?
    - Integrate one, derive the other
  - It's about the derivative of a product in an integral
    $$[y/x]^2_1 = \int_{x_1}^{x_2} \frac{d}{dx} y \, dx$$
    $$= \int_{x_1}^{x_2} f'(x) y(x) + f(x) y'(x) \, dx$$

Calculus of variation – 1D

$$\int_{x_1}^{x_2} \eta'(x) f'(x) \, dx = 0$$

- Integrate by parts
  $$\int_{x_1}^{x_2} \eta'(x) f'(x) \, dx = \left[ \eta(x) f'(x) \right]_{x_1}^{x_2} - \int_{x_1}^{x_2} \eta(x) f''(x) \, dx$$
- We know that $\eta(x_1)=\eta(x_2)=0$
- We get
  $$\int_{x_1}^{x_2} \eta(x) f''(x) \, dx = 0$$
- Must be true for any $\eta$
- Therefore, $f''(x)$ must be zero everywhere
Summary

- Variational minimization (integral of a functional) with boundary condition
  \[ \min \int_{\Omega} |\nabla f|^2 \text{ with } f|_{\partial \Omega} = f'|_{\partial \Omega} \]
- Derive Euler-Lagrange equation:
  - Use perturbation function
  - Calculus of variation. Set to zero. Integrate by parts.
  \[ \Delta f = 0 \text{ over } \Omega \text{ with } f|_{\partial \Omega} = f'|_{\partial \Omega} \]
  Check out the hidden slides for detail

Questions?

Discrete solver: Recall 1D

<table>
<thead>
<tr>
<th>Copy</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ \nabla^2 f = 2f_2 + 2f_4 - 2f_6 - 16 \]
\[ \nabla^2 f = 2f_2 + 2f_4 + 2 + 2f_6 - 2f_4 + 4 \]
\[ \nabla^2 f = 2f_2 + 2f_4 + 4 + 2f_6 - 2 \]
\[ \nabla^2 f = 2f_2 - 2f_4 + 2f_6 - 4 \]

\[ \nabla^2 f \approx \frac{f_2 + f_4 - 2f_6 - 16}{4} \]
\[ \nabla^2 f \approx \frac{-2f_2 + 4 - 2f_4 + 2f_6 - 2}{4} \]
\[ \nabla^2 f \approx \frac{f_2 + f_4 - 2f_6 - 4}{4} \]

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

Discrete Poisson solver

- Two approaches:
  - Minimize variational problem \( \min \int_{\Omega} |\nabla f|^2 \text{ with } f|_{\partial \Omega} = f'|_{\partial \Omega} \)
  - Solve Euler-Lagrange equation \( \Delta f = \text{div } g \text{ over } \Omega \text{ with } f|_{\partial \Omega} = f'|_{\partial \Omega} \)
  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 \( v \)

Discrete Poisson solver

- Minimize variational problem \( \min \int_{\Omega} |\nabla f|^2 \text{ with } f|_{\partial \Omega} = f'|_{\partial \Omega} \)
  - Calculate gradient \( \nabla f \) at \( (p,q) \) for all \( p,q \) in \( \Omega \)
  - Boundary condition
    - Minimize \( \sum_{(p,q) \in \partial \Omega} (f_p - f_q - v_{pq})^2 \) with \( f_p = f'_p \) for all \( p \in \partial \Omega \)

Discrete Poisson solver

- Minimize variational problem \( \min \int_{\Omega} |\nabla f|^2 \text{ with } f|_{\partial \Omega} = f'|_{\partial \Omega} \)
  - Calculate gradient \( \nabla f \) at \( (p,q) \) for all \( p,q \) in \( \Omega \)
  - Boundary condition
    - Minimize \( \sum_{(p,q) \in \partial \Omega} (f_p - f_q - v_{pq})^2 \) with \( f_p = f'_p \) for all \( p \in \partial \Omega \)

- Rearrange and call \( N_p \) the neighbors of \( p \)
  - Big yet sparse linear system only for boundary pixels

- Rearrange and call \( N_p \) the neighbors of \( p \)
  - Big yet sparse linear system only for boundary pixels
Result (eye candy)

Questions?

Recap

• Find image whose gradient best approximates the input gradient
  – least square Minimization
• Discrete case: turns into big sparse linear equation
  – Set derivatives to zero
  – Derivatives of quadratic ==> linear

Solving big matrix systems

• $Ax=b$
• You can use Matlab’s \( \backslash \) \  
  – (Gaussian elimination)
  – But not very scalable

Iterative solvers

Important ideas
• Do not inverse matrix
• Maintain a vector $x'$ that progresses towards the solution
• Updates mostly require to apply the matrix.
  – In many cases, it means you do no even need to store the matrix (e.g. for a convolution matrix you only need the kernel)
• Usually, you don’t even wait until convergence
• Big questions: in which direction do you walk?
  – Yes, very similar to gradient descent

Solving big matrix systems

• $Ax=b$, where $A$ is sparse (many zero entries)
• In Pset 3, we ask you to use conjugate gradient
Conjugate gradient

The Conjugate Gradient Method is the most prominent iterative method for solving sparse systems of linear equations. Unfortunately, many textbook treatments of the topic are written with neither illustrations nor intuition, and their victims can be found to this day babbling senselessly in the corners of dusty libraries. Nevertheless, the Conjugate Gradient Method is a composite of simple, elegant ideas that almost anyone can understand. Of course, a reader as intelligent as yourself will learn them almost effortlessly.

Ax=b

- A is square, symmetric and positive-definite
- When A is dense, you’re stuck, use backsubstitution
- When A is sparse, iterative techniques (such as Conjugate Gradient) are faster and more memory efficient

Simple example:

\[
\begin{bmatrix}
3 & 2 \\
2 & 6 \\
\end{bmatrix}
\begin{bmatrix}
x \\
-8 \\
\end{bmatrix}
\]

(Yeah yeah, it’s not sparse)

Turn Ax=b into a minimization problem

- Minimization is more logical to analyze iteration (gradient ascent/descent)
- Quadratic form
  \[ f(x) = \frac{1}{2} x^T A x + b^T x + c \]
  - c can be ignored because we want to minimize
- Intuition:
  - the solution of a linear system is always the intersection of n hyperplanes
  - A needs to be positive-definite so that we have a nice parabola

Gradient of the quadratic form

\[
\nabla f(x) = A x - b
\]

Not surprising: we turned Ax=b into the quadratic minimization (if A is not symmetric, conjugate gradient finds solution for \(\frac{1}{\lambda}(A^T + A)x = b\))

Steepest descent/ascent

- Pick gradient direction
- Find optimum in this direction

Residual

- At iteration i, we are at a point x(i)
- Residual r(i)=b-Ax(i)
- Cool property of quadratic form: residual = - gradient
Behavior of gradient descent

- Zigzag or goes straight depending if we’re lucky
  - Ends up doing multiple steps in the same direction

Conjugate gradient

- Smarter choice of direction
  - Ideally, step directions should be orthogonal to one another (no redundancy)
  - But tough to achieve
  - Next best thing: make them A-orthogonal (conjugate)
    - That is, orthogonal when transformed by $A$: $d_{(i)}^T Ad_{(i)} = 0$

Conjugate gradient

- For each step:
  - Take the residual (gradient)
  - Make it A-orthogonal to the previous ones
  - Find minimum along this direction
- Plus life is good:
  - In practice, you only need the previous one
  - You can show that the new residual $r(i+1)$ is already A-orthogonal to all previous directions $p$ but $p(i)$

Recap

- Poisson image cloning: paste gradient, enforce boundary condition
- Variational formulation $\min \int |I - f| \quad \text{with } f|_{\partial\Omega} = I'|_{\partial\Omega}$
- Also Euler-Lagrange formulation $\mathcal{L} = \text{div} \nabla f$, with $f|_{\partial\Omega} = I'|_{\partial\Omega}$
- Discretize variational version, leads to big but sparse linear system
- Conjugate gradient is a smart iterative technique to solve it

Questions?

Figure 2: Concealment. By inpainting 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.
Manipulate the gradient

- Mix gradients of g & f: take the max

Figure 8: Inserting one object close to another. With seamless cloning, an object in the destination image touching the selected region $\Omega$ bleeds into it. Bleeding is inhibited by using mixed gradients as the guidance field.

Reduce big gradients

- Dynamic range compression
- See Fattal et al. 2002

Figure 9: Local distortion changes. Applying an appropriate anti-aliasing transformation to the guidance field under the selection and then integrating back with a Poisson solver, removes locally the apparent distortion of an image. This is useful to highlight underexposed foreground objects or to induce specular reflections.
Questions?

Issues with Poisson cloning
- Colors
- Contrast
- The backgrounds in f & g should be similar

Improvement: local contrast
- Use the log
- Or use covariant derivatives (next slides)

Covariant derivatives & Photoshop
- Photoshop Healing brush
- Developed independently from Poisson editing by Todor Georgiev (Adobe)

Seamless Image Stitching in the Gradient Domain
- Anat Levin, Assaf Zomet, Shmuel Peleg, and Yair Weiss
  http://eprints.pascal-network.org/archive/00001062/01/tips05-blending.pdf
- Various strategies (optimal cut, feathering)

Photomontage
**Elder's edge representation**

- [http://elderlab.yorku.ca/~elder/publications/journals/ElderPAMI01.pdf](http://elderlab.yorku.ca/~elder/publications/journals/ElderPAMI01.pdf)

**Gradient tone mapping**

- Fattal et al. Siggraph 2002

---

**Gradient attenuation**

- From Fattal et al.

---

**Gradient tone mapping**


---

**Gradient tone mapping**


![Image](image.png)

Figure 1: (a) Grayscale version of a band image computed through PCA. (b) Grayscale version of the same image computed through our algorithm.

Questions?

**Retinex**

• Land, Land and McCann (inventor/founder of Polaroid)
• Theory of lightness perception (albedo vs. illumination)
• Strong gradients come from albedo, illumination is smooth

**Color2gray**

• Use Lab gradient to create grayscale images

![Image](image.png)

Figure 1: A color image (left) after applying important visual salient edges from a grayscale color image (middle). The Color2Gray algorithm (right) provides visible relief changes to the grayscale image. Image: Improvement Source by Glass Island, collage of AdobePhotoshop.

**Poisson Matting**

• Sun et al. Siggraph 2004
• Assume gradient of F & B is negligible
• Plus various image-editing tools to refine matte

\[ I = \alpha F + (1 - \alpha)B \]
\[ \nabla I = (F - B) \nabla \alpha + \alpha \nabla F + (1 - \alpha) \nabla B \]
\[ \nabla \alpha \approx \frac{1}{F - B} \nabla I \]

![Image](image.png)

Figure 1: Gradient of one from a complete scene. From left to right: a complete natural image for editing testing, an edge image where the color background is removed, a result from spectral to gradient image and the result from our algorithm.

**Gradient camera?**

• Tumblin et al. CVPR 2005

http://www.cfar.umd.edu/~aagrawal/gradcam/gradcam.html

![Image](image.png)

Figure 2: Log gradient camera overview: intensity sensors organized into 4-pixel arrays share the same self-calibrating gain settings \( A \), and send log(\( f_1 \)) signals to A/D converters. Subtraction removes common-mode noise, and a linear “curl filter” solves corrects orthogonal gradient values or “dark” pixels, and a Poisson solver finds output values from gradients.
Poisson-ish mesh editing

- [http://portal.acm.org/citation.cfm?id=1057432.1057456](http://portal.acm.org/citation.cfm?id=1057432.1057456)
- [http://www.cad.zju.edu.cn/home/xudong/Projects/mesh_editing/main.html](http://www.cad.zju.edu.cn/home/xudong/Projects/mesh_editing/main.html)

Questions?

Alternative to membrane

- Thin plate: minimize second derivative
  \[
  \min_f \int \int (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2) \, dx \, dy
  \]

Inpainting

- More elaborate energy functional/PDEs
  [http://www-mount.ee.umn.edu/~guille/inpainting.htm](http://www-mount.ee.umn.edu/~guille/inpainting.htm)

Key references

- Elder, Image editing in the contour domain, 2001 [http://elderlab.yorku.ca/~elder/publications/journals/ElderPA/MI01.pdf](http://elderlab.yorku.ca/~elder/publications/journals/ElderPA/MI01.pdf)
- Poisson Image Editing Perez et al. [http://research.microsoft.com/vision/cambridge/papers/perez_siggraph03.pdf](http://research.microsoft.com/vision/cambridge/papers/perez_siggraph03.pdf)

Poisson, Laplace, Lagrange, Fourier, Monge, Parseval

- Fourier studied under Lagrange, Laplace & Monge, and Legendre & Poisson were around
- They all raised serious objections about Fourier’s work on Trigonometric series
  [http://www.ece.umd.edu/~taylor/Frame2.htm](http://www.ece.umd.edu/~taylor/Frame2.htm)
- [http://www.mathphysics.com/pde/history.html](http://www.mathphysics.com/pde/history.html)
- [http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Fourier.html](http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Fourier.html)
- [http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Parseval.html](http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Parseval.html)
Refs Laplace and Poisson

- http://farside.ph.utexas.edu/teaching/329/lectures/notes74.html
- http://en.wikipedia.org/wiki/Poisson's_equation
- http://www.colorado.edu/engineering/CAS/courses.d/AFEM.d/AFEM.Ch03.d/AFEM.Ch03.pdf

Gradient image editing refs

- http://research.microsoft.com/vision/cambridge/papers/perez_siggraph03.pdf
- Covariant Derivatives and Vision, Todor Georgiev (Adobe Systems)
- ECCV 2006
- http://www.merl.com/people/raskar/Flash05/
- http://research.microsoft.com/~carrot/new_page_1.htm

Links

- How to Get Your SIGGRAPH Paper Rejected, Jim Kajiya, SIGGRAPH 1993 Papers Chair, (link)
- Ted Adelson’s Informal guidelines for writing a paper, 1991. (link)
- Notes on technical writing, Don Knuth, 1989. (pdf)
- What's wrong with these equations, David Mermin, Physics Today, Oct., 1989. (pdf)
- Ten Simple Rules for Mathematical Writing, Dimitri P. Bertsekas (link)
- Advice on Research and Writing (at CMU)
- How (and How Not) to Write a Good Systems Paper by Roy Levin and David D. Redell
- Things I Hope Not to See or Hear at SIGGRAPH by Jim Blinn
- How to have your abstract rejected

Poisson image editing

- Two aspects
  - When the new gradient is conservative:
    - Just membrane interpolation to ensure boundary condition
  - Otherwise: allows you to work with non-conservative vector fields and
- Why is it good?
  - More weight on high frequencies
    - Membrane tries to use low frequencies to match boundaries conditions
    - Manipulation of the gradient can be cool (e.g. mix of the two gradients)
    - Manipulate local features (edge/gradient) and worry about global consistency later
- Smart thing to do: work in log domain
- Limitations
  - Color shift, contrast shift (depends strongly on the difference between the two respective backgrounds)

Other functionals

- I lied, some people have used smarted energy functions:
  - Todor Georgiev’s initial implementation of the Photoshop healing brush.