Signal Processing
From Images to Surfaces

Misha Kazhdan
Johns Hopkins University
Goal

Extend image-processing techniques to surfaces:

1. Gradient Domain
Goal

Extend image-processing techniques to surfaces:

1. Gradient Domain
2. Shock Filters
Goal

Extend image-processing techniques to surfaces:

1. Gradient Domain
2. Shock Filters
3. Optical Flow

Moving Gradients: A Path Based Method for Plausible Image Interpolation. [Mahajan et al., 2009]
Outline

• Motivation

• **Processing Tools**
  – Screened Poisson Equation
  – Flow Fields/Lines

• Extensions to Signals on Surfaces

• Conclusion
1. **Screened Poisson Equation:**

Given a 2D domain $\Omega$, a function $g$, and a vector field $\mathbf{v}$, solve for the function $f$ minimizing:

$$E(f) = \int_{\Omega} \alpha \| f - g \|^2 + \| \nabla f - \mathbf{v} \|^2$$

- **value-fitting**
- **gradient-fitting**
1. **Screened Poisson Equation:**

Given a 2D domain $\Omega$, a function $g$, and a vector field $\vec{v}$, solve for the function $f$ minimizing:

$$E(f) = \int_{\Omega} \alpha \| f - g \|^2 + \| \nabla f - \vec{v} \|^2$$

$$\downarrow$$

$$(\alpha \cdot 1 - \Delta)f = \alpha \cdot 1 \cdot g - \text{div}(\vec{v})$$
2a. Flow Fields/Lines:

Given a 2D domain $\Omega$ and a vector field $\vec{v}$, a flow-line of $\vec{v}$ is a curve $\gamma_p$ such that:

$$\gamma_p(0) = p \quad \text{and} \quad \gamma_p'(t) = \vec{v}(\gamma_p(t)).$$
2b. Flow Fields/Lines:
Given a 2D domain $\Omega$ and a vector field $\vec{v}$, the *advection* of a function $f$ along $\vec{v}$ is the function:

$$[\text{Adv}_{\vec{v}}(f)](p) = f(\gamma_p(-1)).$$
2c. Flow Fields/Lines:
Given..., for small $t$ we have:

\[ \text{Adv}_{t \cdot \hat{\nu}}(f) - f \approx -t \cdot \langle \nabla f, \hat{\nu} \rangle \]

\[ \Downarrow \]

\[ \frac{\partial f}{\partial t} = -\langle \nabla f, \hat{\nu} \rangle \]
1. Screened Poisson Equation:

\[(\alpha \cdot 1 - \Delta)f = \alpha \cdot 1 \cdot g - \text{div}(\vec{v})\]

On a mesh:
Geometry-Processing Tools

1. Screened Poisson Equation:

\[(\alpha \cdot 1 - \Delta)f = \alpha \cdot 1 \cdot g - \text{div}(\vec{v})\]

On a mesh:

- \(f, g \rightarrow\) maps from vertices to real values
1. Screened Poisson Equation:

\[
(\alpha \cdot 1 - \Delta)f = \alpha \cdot 1 \cdot g - \text{div}(\vec{v})
\]

On a mesh:

- \(f, g\) → maps from vertices to real values
- \(\vec{v}\) → a map from triangles to tangent vectors
1. Screened Poisson Equation:

\[
(\alpha \cdot 1 - \Delta) f = \alpha \cdot 1 \cdot g - \text{div}(\hat{\mathbf{v}})
\]

On a mesh:

- $f, g \rightarrow$ maps from vertices to real values
- $\mathbf{v} \rightarrow$ a map from triangles to tangent vectors
- $\Delta \rightarrow$ the cotangent Laplacian
Geometry-Processing Tools

1. Screened Poisson Equation:
\[
(\alpha \cdot \text{1} - \Delta) f = \alpha \cdot \text{1} \cdot g - \text{div}(\mathbf{v})
\]

On a mesh:
- \(f, g\) → maps from vertices to real values
- \(\mathbf{v}\) → a map from triangles to tangent vectors
- \(\Delta\) → the cotangent Laplacian
- \(\text{1}\) → the mass-matrix
1. Screened Poisson Equation:

\[(\alpha \cdot 1 - \Delta)f = \alpha \cdot 1 \cdot g - \text{div}(\vec{v})\]

On a mesh:

- \(f, g \rightarrow\) maps from vertices to real values
- \(\vec{v} \rightarrow\) a map from triangles to tangent vectors
- \(\Delta \rightarrow\) the cotangent Laplacian
- \(1 \rightarrow\) the mass-matrix
- \(\text{div} \rightarrow d^t \cdot \Lambda:\)
  - \(\Lambda:\) diagonal with triangle areas
  - \(d:\) the gradient operator
Geometry-Processing Tools

2. Flow Fields/Lines:
\[ \gamma_p(0) = p \quad \text{and} \quad \gamma_p'(t) = \vec{v}(\gamma_p(t)) \]

Iteratively:
- Sample the flow field at \( p \).
- Take a small step in a straight line along the flow direction.
2. Flow Fields/Lines:

\[ \gamma_p(0) = p \quad \text{and} \quad \gamma'_p(t) = \vec{v}(\gamma_p(t)) \]

Iteratively:

– Sample the flow field at \( p \).
– Take a small step in a straight line along the flow direction.
2. Flow Fields/Lines:

\[ \gamma_p(0) = p \quad \text{and} \quad \gamma'_p(t) = \vec{v}(\gamma_p(t)) \]

Iteratively:

- Sample the flow field at \( p \).
- Take a small step in a straight line along the flow direction.
Geometry-Processing Tools

2. Flow Fields/Lines:
\[ \gamma_p(0) = p \quad \text{and} \quad \gamma'_p(t) = \vec{v}(\gamma_p(t)) \]

Iteratively:

– Sample the flow field at \( p \).

– Take a small step in a straight line along the flow direction.
Outline

• Motivation
• Tools of the Trade
• Extensions to Signals on Surfaces
  – Gradient Domain [Poisson]
  – Shock Filters [Advection]
  – Optical Flow [Poisson + Advection]
• Conclusion
Gradient Domain (Stitching)

Different exposures $\Rightarrow$ Seams in the panorama

Image(s) courtesy of Uytterdaele
Gradient Domain (Stitching)

- Copy interior gradients into $\tilde{v}$
- Set seam-crossing gradients to zero

\[
f^{out} = \arg\min_{f: \Omega \rightarrow \mathbb{R}} \int \|\nabla f - \tilde{v}\|^2 dp
\]
Gradient Domain (Stitching)

• Copy interior gradients into $\tilde{v}$
• Set seam-crossing gradients to zero

$$f^{out} = \arg \min_{f: \Omega \to \mathbb{R}} \int \| \nabla f - \tilde{v} \|^2 dp$$
Gradient Domain (Sharpening)

• Fit input colors: \( g = f^{in} \)
• Amplify input gradients: \( \tilde{v} = \beta \cdot \nabla f^{in} \)

\[
\begin{align*}
    f^{out} &= \operatorname{argmin}_{f: \Omega \to \mathbb{R}} \int_{\Omega} \alpha \left\| f - f^{in} \right\|^2 + \left\| \nabla f - \beta \cdot \nabla f^{in} \right\|^2 \\
\end{align*}
\]

\( \beta > 1 \)

Fourier Analysis of the 2D Screened Poisson Equation for Gradient Domain Problems. [Bhat et al. 2008]
Gradient Domain (Sharpening)

• Fit input colors: \( g = f^{in} \)
• Amplify input gradients: \( \tilde{v} = \beta \cdot \nabla f^{in} \)

\[
f^{out} = \arg\min_{f: \Omega \to \mathbb{R}} \int_{\Omega} \alpha \|f - f^{in}\|^2 + \|\nabla f - \beta \cdot \nabla f^{in}\|^2
\]

value-fitting  gradient-fitting
Gradient Domain (Sharpening)

- Fit input colors: \( g = f^{in} \)
- Amplify input gradients: \( \hat{v} = \beta \cdot \nabla f^{in} \)

\[
f^{out} = \arg\min_{f: \Omega \to \mathbb{R}} \int_{\Omega} \alpha \| f - f^{in} \|^2 + \| \nabla f - \beta \cdot \nabla f^{in} \|^2
\]

Setting \( f^{in} \) to the positions of the vertices in 3D
Outline

• Motivation
• Tools of the Trade
• Extensions to Signals on Surfaces
  – Gradient Domain [Poisson]
  – Shock Filters [Advection]
  – Optical Flow [Poisson + Advection]
• Conclusion
Shock Filters

[Osher and Rudin, 1990]:

Progressively sharpen a signal so that:

- Extrema preserved
- Edges pronounced
  - Lower-valued side $\rightarrow$ local minimum
  - Higher-valued side $\rightarrow$ local maximum
Shock Filters

[Osher and Rudin, 1990]:

Progressively sharpen a signal so that:

- Extrema preserved $\rightarrow \mathcal{F}$ vanishes with the gradient
- Edges pronounced $\rightarrow \mathcal{G}$ gives the sign w.r.t. the edge

\[
\frac{df}{dt} = \mathcal{F}(f) \cdot \mathcal{G}(f)
\]

\[
\mathcal{F}(f) = \|\nabla f\|^2
\]

\[
\mathcal{G}(f) = -\frac{\partial^2 f}{\partial (\nabla f / \|\nabla f\|)^2}
\] (Second derivative in the gradient direction)

Shock Filters

Method of Characteristics:
We can re-write the PDE:
\[
\frac{df}{dt} = F(f) \cdot G(f) = -\langle \nabla f, H_f \cdot \nabla f \rangle
\]
This describes the advection of \( f \) along the flow:
\[
\vec{v} = H_f \cdot \nabla f
\]

\[F(f) = \|\nabla f\|^2\]
\[G(f) = -\frac{\partial^2 f}{\partial (\nabla f / \|\nabla f\|)^2} = -\frac{1}{\|\nabla f\|^2} \langle \nabla f, H_f \cdot \nabla f \rangle\]
Shock Filters

Sharpen $f$ by advecting along the flow:

$$\mathbf{\nabla} = \frac{1}{2} \nabla ||\nabla f||^2$$

---

ShockAdvect($f$, $t$)

1. $P \leftarrow ||\nabla f||^2$ // potential
2. $\mathbf{\nabla} \leftarrow \frac{1}{2} \nabla P$ // flow field
3. return Advect($f$, $\mathbf{\nabla}$, $t$)
Shock Filters

Intuitively:

Values are transported along flow lines of the potential’s gradient, moving from the (local) minima to maxima:

– [Minima] Critical points of the input
– [Maxima] Edges of the output

⇒ “Piecewise constant” image with input extrema advected out to the edges.

\[ P = \|\nabla f\|^2 \]

\[ \vec{v} = \nabla P / 2 \]
Shock Filters

Sharpen $f$ by advecting along the flow:

$$\vec{v} = \frac{1}{2} \nabla \| \nabla f \|^2$$
Shock Filters

Sharpen $f$ by advecting along the flow:

$$\vec{v} = \frac{1}{2} \nabla \| \nabla f \|^2$$
Shock Filters

Sharpen $f$ by advecting along the flow:

$$\ddot{v} = \frac{1}{2} \nabla \| \nabla f \|^2$$

Setting $f$ to the normals of the vertices
Outline

• Motivation
• Tools of the Trade
• Extensions to Signals on Surfaces
  – Gradient Domain [Poisson]
  – Shock Filters [Advection]
  – Optical Flow [Poisson + Advection]
• Conclusion
Optical Flow

Video Textures:
Given a video, generate “a continuous, infinitely varying stream of video images”.

Video Textures. [Schödl, Szeliski, Salesin, and Essa, 2000]
Optical Flow

Extend video textures [Schödl et al., 2000] to 3D:
Optical Flow

Extend video textures [Schödl et al., 2000] to 3D:

– Identify similar windows in the video
– Fit a template to the windows
– Interpolate geometries
– Interpolate textures
Optical Flow

Image Interpolation:

Target
Optical Flow

**Image Interpolation (Linear):**
- Linear interpolation causes ghosting.
Optical Flow

Image Interpolation (Advected):
– Estimate optical flow field.
Optical Flow

Image Interpolation (Advected):
- Estimate optical flow field.
- Advect forward/backward and blend.
Optical Flow

Brightness Constancy [Lucas and Kanade, 1981]:
Solve for $\tilde{v}$ that advects the source/target towards each other:

$$E_{\tilde{v}} = \text{Adv} - \tilde{v} \cdot \nabla f_t - \text{Adv} \cdot \nabla f_s \approx \nabla f_t - \nabla f_s + \langle \nabla f_t + f_s, \tilde{v} \rangle$$

Smooth signals/vector-fields are implicitly mandated by working in a space that does not have high-frequencies.

Estimate $\tilde{v}$ hierarchically (coarse-to-fine):

– Advance the source/target along $\tilde{v}$
– Solve for the correcting flow
– Add the correcting flow to $\tilde{v}$
– Advance to the next level of the hierarchy
Optical Flow

Scale Space Formulation:

Smooth solutions are explicitly encouraged by:

– Smoothing the source and target at each level:

\[
E(\tilde{f}^{s/t}) = \|\tilde{f}^{s/t} - f^{s/t}\|^2 + \frac{\alpha}{4^l} \|\nabla \tilde{f}^{s/t}\|^2
\]

– Incorporating a smoothness term in the energy:

\[
E(\tilde{v}) = \|\text{Adv}_{\tilde{v}}(\tilde{f}^s) - \text{Adv}_{-\tilde{v}}(\tilde{f}^t)\|^2 + \frac{\alpha}{4^l} \|\nabla \tilde{v}\|^2
\]

Solve two Poisson equations per level.*

*In the second, the Laplacian is the vector-field (Hodge) Laplacian.
Optical Flow

Texture Interpolation:

Source                 Target                 Synthesized
Optical Flow

Texture Interpolation:

Linear Blend

Optical Flow Blend
Outline

• Motivation
• Tools of the Trade
• Extensions to Signals on Surfaces
• Conclusion
Conclusion

Extend fundamental image-processing operators to the context of surfaces
[Differential Operators / Flows]

Much of the heavy lifting has already been done for us

Well-established image-processing techniques carry over
[Gradient Domain / Shock Filters / Optical Flow]
Conclusion

Working with images makes simple things easier.

Fixed stencil Laplacian / Gradient / Divergence
Parallelization / Out-of-Core Streaming
Fast Fourier Transform
Conclusion

Working with images makes simple things easier.

Setting $\alpha$ be a \textit{spatially varying} weighting function
Estimating the Laplace-Beltrami Operator by Restricting 3D Functions. [Chuang et al., 2009]
Fast Mean-Curvature Flow via Finite Elements Tracking. [Chuang et al., 2011]
Interactive and Anisotropic Geometry Processing Using the Screened Poisson Equation. [Chuang et al., 2011]
Unconditionally Stable Shock Filters for Image and Geometry Processing. [Prada et al., 2015]
Motion Graphs for Unstructured Textured Meshes. [Prada et al., 2016]

Thank You!

Ming Chuang, Fabian Prada, Alvaro Collet, Linjie Luo, Benedict Brown
Szymon Rusinkiewicz, Hugues Hoppe

http://www.cs.jhu.edu/~misha/Code/PoissonMesh/GeometryEditor/
http://www.cs.jhu.edu/~misha/Code/AdvectionSharpening/
Properties:

- **Lagrangian Implementation**
  Sampling → antialiased output edges

- **One-Step Integration**
  Use a single (long) stream-line per pixel
Gradient Domain (Sharpening)

- Fit input colors: \( g = f^{in} \)
- Amplify input gradients: \( \tilde{v} = \beta \cdot \nabla f^{in} \)

\[
f^{out} = \arg\min_{f: \Omega \to \mathbb{R}} \int_{\Omega} \alpha \| f - f^{in} \|^2 + \| \nabla f - \beta \cdot \nabla f^{in} \|^2
\]

Setting \( \alpha \) to be an anisotropic scaling of the metric.