Spectral Processing of Point-sampled Geometry

Outline

- Introduction
- Fourier transform
- Spectral processing pipeline
- Applications
  - Spectral filtering
  - Adaptive subsampling
- Summary
Introduction

Model Acquisition $\Rightarrow$ Point-based Geometry Processing $\Rightarrow$ Point Rendering

- Range scans
- Depth images
- ...

Spectral Methods

- QSplat
- Surfels
- ...

Spectral Transform

- Extend Fourier transform to 2-manifold surfaces

- Spectral representation of point-based objects
- Powerful methods for digital geometry processing
## Applications

- Spectral filtering:
  - Noise removal
  - Microstructure analysis
  - Enhancement

- Adaptive resampling:
  - Complexity reduction
  - Continuous LOD

## Fourier Transform

**input:** $x_k, k = 0, 1, \ldots, N - 1$

**output:** $X_n = \sum_{k=0}^{N-1} x_k e^{-i \frac{2\pi nk}{N}}, \quad n = 0, 1, \ldots, N - 1$

**Benefits:**
- Sound concept of frequency
- Extensive theory
- Fast algorithms

**Limitations:**
- Euclidean domain, global parameterization
- Regular sampling
- Lack of local control
## Fourier Transform

**Requirements:**
- Fourier transform defined on Euclidean domain
  - we need a global parameterization
- Basis functions are eigenfunctions of Laplacian operator
  - requires regular sampling pattern so that basis functions can be expressed in analytical form (fast evaluation)

**Limitations:**
- Basis functions are globally defined
  - Lack of local control

## Approach

Split model into patches that:
- are parameterized over the unit-square
  - mapping must be continuous and should minimize distortion
- are re-sampled onto a regular grid
  - adjust sampling rate to minimize information loss
- provide sufficient granularity for intended application (local analysis)
  - process each patch individually and blend processed patches
Overview

Processing stages are depicted as rectangles, rounded boxes represent input/output data of each stage. Gray background color indicates the preprocessing phase.

Patch Layout Creation

Clustering ⇒ Optimization

Samples ⇒ Clusters ⇒ Patches

The patch layout is created by first merging sample points into clusters and then merging clusters into patches.
Patch Layout Creation

- Parameterize patches by orthogonal projection onto base plane
- Bound normal cone to control distortion of mapping using smallest enclosing sphere
- Cone angle maximally $\pi$. Usually one takes: $\pi/2$

First put all points into a BSP tree. Now we successively merge clusters with a common parent in the BSP tree, until the leaves contain 25-100 sample points. These are input to the 2nd merging stage.
Patch Layout Creation

- Iterative, local optimization method
- Merge patches according to quality metric:

\[ \Phi = \Phi_S \cdot \Phi_{NC} \cdot \Phi_B \cdot \Phi_{Reg} \]

\( \Phi_S \): favour large patch size
\( \Phi_{NC} \): favour low curvature (flat patches)
\( \Phi_B \): favour roughly circular patches
\( \Phi_{Reg} \): spring energy regularization

Successively merge pairs of patches with the lowest value of \( F \) (highest quality gain) until normal cone condition is violated.

Patch Resampling

Patches are irregularly sampled without any connectivity information.

Resample patch onto regular grid using scattered data approximation (SDA).
Patch Resampling

Left: sample points in the parameter plane. Right: extended parameter domain with regular sampling. Red dots: boundary points from neighboring patches.

Patch Resampling

(1) Initial approximation by splatting (2) Lower resolution approximation by convolution (3) fill the holes by blending approximations at different resolutions
Spectral Filters

- 2D Discrete Fourier Transform (DFT)
  - Direct manipulation of spectral coefficient
- Filtering as convolution:
  \[ F(x \ast y) = F(x) \cdot F(y) \]
  - Convolution: \( O(N^2) \) \( \Rightarrow \) Multiplication: \( O(N) \)
- Inverse Fourier Transform
  - Filtered patch surface

---

Spectral Processing Pipeline

**Ideal Low-Pass**

**Gaussian Low-Pass**

**Band-Stop**

**Enhancement**

**Original**

**Zoom**
Spectral Resampling

Low-pass filtering
Band-limitation

Regular Resampling
Optimal sampling rate
(Sampling Theorem)
Error control (Parseval’s Theorem)

Power Spectrum Estimation

Left: normalized logarithmic plot of the power spectrum of a typical patch surface.
Right: idealized illustration for signal-to-noise ratio estimation.
Reconstruction

Filtering can lead to discontinuities at patch boundaries

Create patch overlap, blend adjacent patches

- Sampling rates
- Point positions
- Normals

Result: a gradual change of sampling density at patch boundaries
Spectral Processing Pipeline

Results

Surface Restoration

Original noise+blur  Gaussian Filter  Wiener Filter  Patch Layout
Interactive Filtering

Local smoothing (red circle) and enhancement (blue circle) with adaptive patch layout.

Adaptive Subsampling

4,128,614 pts. 287,163 pts. = 6.9% 98% of signal power
Results

Timings

<table>
<thead>
<tr>
<th>Time</th>
<th>Clustering</th>
<th>Patch Merging</th>
<th>SDA</th>
<th>Analysis</th>
<th>Reconstruction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>9%</td>
<td>38%</td>
<td>23%</td>
<td>4%</td>
<td>26%</td>
</tr>
</tbody>
</table>

Results

Timings

<table>
<thead>
<tr>
<th></th>
<th>Head</th>
<th>St. Matthew</th>
<th>David</th>
</tr>
</thead>
<tbody>
<tr>
<td>#points</td>
<td>460,800</td>
<td>3,382,866</td>
<td>4,128,614</td>
</tr>
<tr>
<td>#patches</td>
<td>256</td>
<td>595</td>
<td>2,966</td>
</tr>
<tr>
<td>Preprocess</td>
<td>10.9</td>
<td>117.2</td>
<td>128.3</td>
</tr>
<tr>
<td>Total</td>
<td>15.8</td>
<td>153.0</td>
<td>189.6</td>
</tr>
</tbody>
</table>

Timings (s) on a 1.1GHz AMD Athlon with 1.5 GByte main memory
Summary

- Versatile spectral decomposition of point-based models
- Effective filtering
- Adaptive resampling
- Efficient processing of large point-sampled models

Reference