Efficient Structure-Aware Selection Techniques for 3D Point Cloud Visualizations with 2DOF Input

Lingyun Yu    Konstantinos Efstathiou
Petra Isenberg    Tobias Isenberg
The Problem: Selection of 3D Subspaces

- 3D spatial data—basis of many visualization research questions
- **problem**: why/how to efficiently select subspaces in 3D?
The Problem: Selection of 3D Subspaces

- 3D spatial data—basis of many visualization research questions
- **problem**: why/how to efficiently select subspaces in 3D?
  - filtering not always possible
The Problem: Selection of 3D Subspaces

- 3D spatial data—basis of many visualization research questions
- **problem**: why/how to efficiently select subspaces in 3D?
  - filtering not always possible
  - raycasting requires large objects

[Wingrave & Bowman, 2005]
The Problem: Selection of 3D Subspaces

- 3D spatial data—basis of many visualization research questions

**problem:** why/how to efficiently select subspaces in 3D?
- filtering not always possible
- raycasting requires large objects
- want spatial selection from 2D (Tablet Freehand Lasso)

[Lucas & Bowman, 2005]
The Problem: Selection of 3D Subspaces

- 3D spatial data—basis of many visualization research questions

- **problem:** why/how to efficiently select subspaces in 3D?
  - filtering not always possible
  - raycasting requires large objects
  - want spatial selection from 2D (Tablet Freehand Lasso)
The Problem: Selection of 3D Subspaces

- 3D spatial data—basis of many visualization research questions

**problem:** why/how to efficiently select subspaces in 3D?
- filtering not always possible
- raycasting requires large objects
- want spatial selection from 2D (Tablet Freehand Lasso)
- iterative selection too tedious
The Problem: Selection of 3D Subspaces

- 3D spatial data—basis of many visualization research questions

**problem:** why/how to efficiently select subspaces in 3D?
- filtering not always possible
- raycasting requires large objects
- want spatial selection from 2D (Tablet Freehand Lasso)
- iterative selection too tedious
The Problem: Selection of 3D Subspaces

- 3D spatial data—basis of many visualization research questions

- **problem**: why/how to efficiently select subspaces in 3D?
  - filtering not always possible
  - raycasting requires large objects
  - want spatial selection from 2D (Tablet Freehand Lasso)
  - iterative selection too tedious
The Problem: Selection of 3D Subspaces

- 3D spatial data—basis of many visualization research questions

**problem:** why/how to efficiently select subspaces in 3D?

- filtering not always possible
- raycasting requires large objects
- want spatial selection from 2D (Tablet Freehand Lasso)
- iterative selection too tedious
The Problem: Selection of 3D Subspaces

- 3D spatial data—basis of many visualization research questions

**problem:** why/how to efficiently select subspaces in 3D?
- filtering not always possible
- raycasting requires large objects
- want spatial selection from 2D (Tablet Freehand Lasso)
- iterative selection too tedious
The Problem: Selection of 3D Subspaces

- 3D spatial data—basis of many visualization research questions

**problem:** why/how to efficiently select subspaces in 3D?
- filtering not always possible
- raycasting requires large objects
- want spatial selection from 2D (Tablet Freehand Lasso)
- iterative selection too tedious
The Problem: Selection of 3D Subspaces

- 3D spatial data—basis of many visualization research questions

**Problem:** why/how to efficiently select subspaces in 3D?
  - filtering not always possible
  - raycasting requires large objects
  - want spatial selection from 2D (Tablet Freehand Lasso)
  - iterative selection too tedious
The Problem: Selection of 3D Subspaces

- 3D spatial data—basis of many visualization research questions

- **problem**: why/how to efficiently select subspaces in 3D?
  - filtering not always possible
  - raycasting requires large objects
  - want spatial selection from 2D (Tablet Freehand Lasso)
  - iterative selection too tedious
The Problem: Selection of 3D Subspaces

- 3D spatial data—basis of many visualization research questions

- **Problem:** why/how to efficiently select subspaces in 3D?
  - filtering not always possible
  - raycasting requires large objects
  - want spatial selection from 2D (Tablet Freehand Lasso)
  - iterative selection too tedious
A New Interactive Selection Technique

- **spatial selection** rather than object-based selection
- **two-dimensional input** (PC, touch displays)
- **2D lasso** interaction: **intended** selection
- **structure-aware selection** in 3D depth
A New Interactive Selection Technique

- **spatial selection** rather than object-based selection
- **two-dimensional input** (PC, touch displays)
- **2D lasso** interaction: **intended** selection
- **structure-aware selection** in 3D depth
A New Interactive Selection Technique

- **spatial selection** rather than object-based selection
- **two-dimensional input** (PC, touch displays)
- **2D lasso** interaction: **intended** selection
- **structure-aware selection** in 3D depth
A New Interactive Selection Technique

- *spatial selection* rather than object-based selection
- *two-dimensional input* (PC, touch displays)
- *2D lasso* interaction: *intended* selection
- *structure-aware selection* in 3D depth
A New Interactive Selection Technique

- **spatial selection** rather than object-based selection
- **two-dimensional input** (PC, touch displays)
- **2D lasso** interaction: **intended** selection
- **structure-aware selection** in 3D depth
A New Interactive Selection Technique

- **spatial selection** rather than object-based selection
- **two-dimensional input** (PC, touch displays)
- **2D lasso** interaction: *intended* selection
- **structure-aware selection** in 3D depth
A New Interactive Selection Technique

- **spatial selection** rather than object-based selection
- **two-dimensional input** (PC, touch displays)
- **2D lasso** interaction: intended selection
- **structure-aware selection** in 3D depth
A New Interactive Selection Technique

- **spatial selection** rather than object-based selection
- **two-dimensional input** (PC, touch displays)
- **2D lasso** interaction: **intended** selection
- **structure-aware selection** in 3D depth
A New Interactive Selection Technique

- **spatial selection** rather than object-based selection
- **two-dimensional input** (PC, touch displays)
- **2D lasso** interaction: **intended** selection
- **structure-aware selection** in 3D depth
A New Interactive Selection Technique

- **spatial selection** rather than object-based selection
- **two-dimensional input** (PC, touch displays)
- **2D lasso** interaction: **intended** selection
- **structure-aware selection** in 3D depth
A New Interactive Selection Technique

- *spatial selection* rather than object-based selection
- *two-dimensional input* (PC, touch displays)
- *2D lasso* interaction: *intended* selection
- *structure-aware selection* in 3D depth
A New Interactive Selection Technique

- **spatial selection** rather than object-based selection
- **two-dimensional input** (PC, touch displays)
- **2D lasso** interaction: **intended** selection
- **structure-aware selection** in 3D depth
A New Interactive Selection Technique

- observation: similar constraints as in sketch-based modeling:
  → definition of 3D space based on 2D input

[Igarashi et al., 1999]
TeddySelection

- sketch-based modeling: Teddy by Igarashi et al. [1999]
  - model in 3D based on sketched outline
  - not directly usable for selection
TeddySelection: Principle

1. draw lasso
TeddySelection: Principle

1. draw lasso
2. input polygon triangulation
TeddySelection: Principle

1. draw lasso
2. input polygon triangulation
3. mapping particles to triangles
TeddySelection: Principle

1. draw lasso
2. input polygon triangulation
3. mapping particles to triangles
4. selection mesh construction
   - 1\textsuperscript{st} binning to fit generalized cylinder to data
TeddySelection: Principle

1. draw lasso
2. input polygon triangulation
3. mapping particles to triangles
4. selection mesh construction
   - 1\textsuperscript{st} binning to fit generalized cylinder to data
   - 2\textsuperscript{nd} binning per triangle, find closest/furthest dense bin (density > 4 • expected density)
TeddySelection: Principle

1. draw lasso
2. input polygon triangulation
3. mapping particles to triangles
4. selection mesh construction
   - 1\textsuperscript{st} binning to fit generalized cylinder to data
   - 2\textsuperscript{nd} binning per triangle, find closest/furthest dense bin (density > 4 \cdot expected density)
TeddySelection: Principle

1. draw lasso
2. input polygon triangulation
3. mapping particles to triangles
4. selection mesh construction
   - 1\textsuperscript{st} binning to fit generalized cylinder to data
   - 2\textsuperscript{nd} binning per triangle, find closest/furthest dense bin (density > 4 • expected density)
   - place outline vertices at average distance
TeddySelection: Principle

1. draw lasso
2. input polygon triangulation
3. mapping particles to triangles
4. selection mesh construction
   - 1\textsuperscript{st} binning to fit generalized cylinder to data
   - 2\textsuperscript{nd} binning per triangle, find closest/furthest dense bin (density > 4 \cdot expected density)
   - place outline vertices at average distance
   - inflate 2D mesh based on binning data
Video: TeddySelection
TeddySelection: Pros & Cons

- **benefit**
  - structure-aware selection
  - compact selection volume
  - fast selection ($\approx 0.2$ sec.)

- **criticism**
  - problems in sparse regions
  - volume always connected, does not work well for many small clusters
CloudLasso

- **goals**
  - same selection procedure as before
  - overcome limitations of TeddySelection
    → be able to treat clusters

- **concept**
  - base the selection volume on global particle density estimation
  - i.e., selection mesh based on density field
    → marching cubes algorithm
CloudLasso: Principle

1. draw lasso
CloudLasso: Principle

1. draw lasso
2. selection mesh construction
   - 1\textsuperscript{st} binning to fit generalized cylinder to data
CloudLasso: Principle

1. draw lasso

2. selection mesh construction
   - 1\textsuperscript{st} binning to fit generalized cylinder to data
   - fit regular grid (64 × 64 × 64) to enclose the lasso frustum
CloudLasso: Principle

1. draw lasso
2. selection mesh construction
   - 1\(^{st}\) binning to fit generalized cylinder to data
   - fit regular grid (64 \(\times\) 64 \(\times\) 64) to enclose the lasso frustum
   - use kernel density estimation on grid
CloudLasso: Principle

1. draw lasso
2. selection mesh construction
   - $1^{st}$ binning to fit generalized cylinder to data
   - fit regular grid ($64 \times 64 \times 64$) to enclose the lasso frustum
   - use kernel density estimation on grid
   - run marching cubes algorithm, but ensure to ignore parts outside lasso
• threshold adjustment possible interactively
CloudLasso: Results

• structure-aware selection
• separate clusters
• interactive adjustment of selection threshold
• performance:
  Marching Cubes: ≈ 0.4 sec.
  density estimation:
  4–6 sec. for ≈ 2 \cdot 10^5 particles
Video: CloudLasso Selection & Interaction
Evaluation & Validation: User Study

- informal feedback positive
- quantitative study to confirm
- restriction to 2 selection methods: CylinderSelection
  - base line (Tablet Freehand Lasso)
  - CloudLasso
    - subjectively best results
  both could be fine-tuned
- Boolean operations possible
Study Design

- 12 participants (4 female)
- 4 selection tasks (datasets)
- measurement of **time**, **error**, and **selection volume**
- questionnaire for subjective opinion
Study Results

- CloudLasso (CL) always faster than CylinderSelection (CS)
  - significant except galaxies
- two error metrics $F_1$ & MCC
  - CloudLasso always less error than CS
  - $F_1$ significant except galaxies
  - MCC significant for clusters & shell/core
- CloudLasso volume always smaller, significant for strings dataset
- CloudLasso the preferred technique for all participants
Discussion: CloudLasso vs. TeddySelection

- both **spatial** & **structure-aware** selection
- both based on lasso principle
- TeddySelection: connected selection
- CloudLasso: individual clusters
- CloudLasso can handle difficult cases
- both can be coupled with Cylinder-Selection using Boolean operations
Limitations

• performance
  – CloudLasso requires grid-based density estimation
  – slower than interactive speeds (≈ 4–6 seconds for ≈ 2 \cdot 10^5 particles)
  – but parallelizable / GPU; only needs to be computed once per scale level

• several parameter choices (e.g., # of bins)
  – parameters seem stable, not changed in our experiments
  – initial density threshold of CloudLasso suggested by algorithm

• set difference (subtraction) just with structure-aware selection not good
  – include operations with CylinderSelection results (e.g., for subtraction)
Application Domains / Future Work

- any particle-based dataset
- also abstract data such as 3D scatter plots → linked views
- huge datasets possible
- selection metrics other than density possible
- applicable to volume data with minor changes
Conclusion

- TeddySelection & CloudLasso: new **spatial, structure-aware** 3D selection techniques
- input: lasso drawn in 2D; output: 3D subspaces
- support complex spatial selections
- applications in many fields of visualization
- study showed that CloudLasso is superior to the traditional cylinder-based selection both in performance and overall preference
- smart selection techniques essential for **interactive** visualization
Thanks for your attention!

Trivia: LingYun Yu’s nick name is Yun which in Mandarin (云) means “cloud”. So it’s really Yun’s Lasso and she is the LassoGirl … ;-)
What about HUGE datasets?

- interactive selection based on a well-chosen sample
- use of both LassoSelection and CylinderSelection
- generation of selection shapes, sequence of Boolean operations
- off-line application to the whole huge dataset (batch process)
What about properties other than density?

- density makes most sense for particle data
- other properties may make sense, e.g., for volume data
- property needs to be defined continuously
- need means to compute property on a grid for CloudLasso
- output always a mesh that encloses a volume in 3D space
- particles/voxels inside that volume are selected
What about precision issues?

- precise input possible (mouse, pen, algorithmic)
- adjustment possible after selection operation
- iterative selections possible
- selection is structure-aware, thus needs less precision
## Study Results – Errors: $F_1$

<table>
<thead>
<tr>
<th></th>
<th>Clusters</th>
<th>Galaxies</th>
<th>Shell/Core</th>
<th>Strings</th>
</tr>
</thead>
<tbody>
<tr>
<td>CloudLasso</td>
<td>.9789</td>
<td>.9866</td>
<td>.9980</td>
<td>.7494</td>
</tr>
<tr>
<td>CylinderSelection</td>
<td>.9759</td>
<td>.9855</td>
<td>.9960</td>
<td>.7303</td>
</tr>
<tr>
<td>$Z$</td>
<td>2.67</td>
<td>0.71</td>
<td>3.06</td>
<td>2.04</td>
</tr>
<tr>
<td>$\rho$</td>
<td>&lt;.01</td>
<td>.48</td>
<td>&lt;.01</td>
<td>.041</td>
</tr>
<tr>
<td></td>
<td>Clusters</td>
<td>Galaxies</td>
<td>Shell/Core</td>
<td>Strings</td>
</tr>
<tr>
<td>------------------</td>
<td>----------</td>
<td>----------</td>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>CloudLasso</td>
<td>.9765</td>
<td>.9733</td>
<td>.9974</td>
<td>.6519</td>
</tr>
<tr>
<td>CylinderSelection</td>
<td>.9731</td>
<td>.9712</td>
<td>.9948</td>
<td>.6305</td>
</tr>
<tr>
<td>$Z$</td>
<td>2.75</td>
<td>0.78</td>
<td>3.06</td>
<td>1.89</td>
</tr>
<tr>
<td>$p$</td>
<td>&lt;.01</td>
<td>.43</td>
<td>&lt;.01</td>
<td>.06</td>
</tr>
</tbody>
</table>
### Study Results – Errors: $V_{\text{selected}} / V_{\text{real}}$

<table>
<thead>
<tr>
<th></th>
<th>Clusters</th>
<th>Galaxies</th>
<th>Shell/Core</th>
<th>Strings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CloudLasso</strong></td>
<td>1.244</td>
<td>4.055</td>
<td>1.327</td>
<td>1.852</td>
</tr>
<tr>
<td><strong>CylinderSelection</strong></td>
<td>1.303</td>
<td>5.855</td>
<td>1.360</td>
<td>2.691</td>
</tr>
<tr>
<td><strong>Z</strong></td>
<td>0.94</td>
<td>1.49</td>
<td>1.57</td>
<td>2.98</td>
</tr>
<tr>
<td><strong>$p$</strong></td>
<td>0.347</td>
<td>0.14</td>
<td>0.875</td>
<td>&lt;0.01</td>
</tr>
</tbody>
</table>
Problem with Structure-Aware Subtraction
Problem with Structure-Aware Subtraction
Problem with Structure-Aware Subtraction
Problem with Structure-Aware Subtraction
Problem with Structure-Aware Subtraction
Problem with Structure-Aware Subtraction
Problem with Structure-Aware Subtraction
Problem with Structure-Aware Subtraction
Problem with Structure-Aware Subtraction