3D RECONSTRUCTION

Real-time 3D reconstruction in-door scene using moving Kinect

Jiakai Zhang, Hao Liu, YuXu
Reconstruction method

- Reconstruction from images
- Reconstruction from video
Using Kinect

- Infrared laser projector
- Monochrome CMOS sensor

Raw Depth Image
Demo
Kinect Raw data
Real-time Reconstruction
Input: 20 frames * 640 * 480 * 12 = 614.8 MB/s
Bilateral Filtering

\[ h(x) = k^{-1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi) c(\xi - x) s(f(\xi) - f(x)) d\xi \]
ICP 3D shape alignment

Figure 1: Point-to-plane error between two surfaces.

\[ M_{\text{opt}} = \arg\min_M \sum_i \left( (M \cdot s_i - d_i) \cdot n_i \right) \]

\[ M = T(t_x, t_y, t_z) \cdot R(\alpha, \beta, \gamma) \]

\[ \bar{M} = \begin{pmatrix} 1 & -\gamma & \beta & t_x \\ \gamma & 1 & -\alpha & t_y \\ -\beta & \alpha & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}. \]

\[ \min \sum_i \left( (\bar{M} \cdot s_i - d_i) \cdot n_i \right)^2 = \min \| Ax - b \|^2. \]

\( \text{SVD} \)
ICP 3D shape alignment

Demo
Pipeline

Raw Depth Image ($r_k$)

Noise Reduction Bilateral Filtering ($R_k$)

Measurement Compute Surface Vertex and Normal Map ($V_k, N_k$)

Pose Estimation ICP ($T_{gk}$)

Update Reconstruction TSDF ($S_k$)

Surface Prediction Ray cast ($V_k, N_k$)
The value in the cube corresponds to the signed distance to the closest zero crossing (surface).
Signed Distance Function

Truncated Signed Distance Function

\[ F_{R_k}(p) = \Psi\left( \lambda^{-1}\|t_{g,k} - p\|_2 - R_k(x) \right), \]

\[ \lambda = \|K^{-1}\dot{x}\|_2, \]

\[ x = \pi\left(KT_{g,k}^{-1}p\right), \]

\[ \Psi(\eta) = \begin{cases} \min\left(1, \frac{\eta}{\mu}\right) \text{sgn}(\eta) & \text{iff } \eta \geq -\mu \\ \text{null} & \text{otherwise} \end{cases} \]
TSDF

- Signed Distance Function
- **Truncated** Signed Distance Function
- Integrate the cubes from different position.

\[
F_k(p) = \frac{W_{k-1}(p)F_{k-1}(p) + W_{R_k}(p)F_{R_k}(p)}{W_{k-1}(p) + W_{R_k}(p)}
\]

\[
W_k(p) = W_{k-1}(p) + W_{R_k}(p)
\]
TSDF

2D Sample

Depth Map from Kinect
TSDF

2D Sample

Depth Map from Kinect
Depth Map from Kinect
TSDF

2D Sample

Depth Map from Kinect
TSDF

2D Sample

Depth Map from Kinect
Depth Map from Kinect

TSDF

2D Sample
### Integration

- We have depth maps from different camera positions, how can we integrate them together?

- Integration? or update?

- Weighted? or add up?

- What makes integration possible?

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>-0.2</td>
<td>0.05</td>
<td>0.5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>-1</td>
<td>-0.8</td>
<td>-0.1</td>
<td>0.3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-0.5</td>
<td>0.05</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-0.5</td>
<td>0.1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>-1</td>
<td>-0.8</td>
<td>-0.05</td>
<td>0.3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>-1</td>
<td>-0.5</td>
<td>-0.03</td>
<td>0.5</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
We have depth maps from different camera positions, how can we integrate them together?

Integration? or update?

Weighted? or add up?

What makes integration possible?
## Integration

We have depth maps from different camera positions, how can we integrate them together?

- **Integration?** or **update**?
- **Weighted?** or add up?
- What makes integration possible?

<table>
<thead>
<tr>
<th></th>
<th>-0.2</th>
<th>0.05</th>
<th>0.5</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>-0.8</td>
<td>-0.1</td>
<td>0.3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-0.5</td>
<td>0.05</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-0.5</td>
<td>0.1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>-1</td>
<td>-0.8</td>
<td>-0.05</td>
<td>0.3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>-1</td>
<td>-0.5</td>
<td>-0.03</td>
<td>0.5</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Only part of distance data is needed, so we can truncate the distance.

- To get the surface behind the surface. The camera is moving!
To get the surface behind the surface. The camera is moving!

Only part of distance data is needed to represent the object, so we can truncate the distance.
To get the surface behind the surface. The camera is moving!

Only part of distance data is needed to represent the object, so we can truncate the distance.
Only part of distance data is needed to represent the object, so we can truncate the distance.

To get the surface behind the surface. The camera is moving!
Pipeline

Raw Depth Image

Noise Reduction Bilateral Filtering

Measurement
Compute
Surface Vertex
and Normal Map

Pose Estimation ICP

Update Reconstruction TSDF

Surface Prediction Ray-cast

$r_k$

$R_k$

$V_k, N_k$

$T_{gk}$

$S_k$

$V_k, N_k$
Cast only, no chasing.

Transfer the TSDF cube into something the computer can understand, Vertex fusion.

Take a photo using X-ray.
**RAY CASTING**

-1 | -0.2 | 0.05 | 0.5 | 1 | 1
-1 | -0.8 | -0.1 | 0.3 | 1 | 1
-1 | -1 | -0.5 | 0.05 | 1 | 1
-1 | -1 | -0.5 | 0.1 | 1 | 1
-1 | -0.8 | -0.05 | 0.3 | 1 | 1
-1 | -0.5 | -0.03 | 0.5 | 1 | 1

- Detect the sign change.
- Two scales search
- Linear regression
<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>-0.2</td>
<td>0.05</td>
<td>0.5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>-1</td>
<td>-0.8</td>
<td>-0.1</td>
<td>0.3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-0.5</td>
<td>0.05</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-0.5</td>
<td>0.1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>-1</td>
<td>-0.8</td>
<td>-0.05</td>
<td>0.3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>-1</td>
<td>-0.5</td>
<td>-0.03</td>
<td>0.5</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

- Detect the sign change.
- Two scales search
- Linear regression
- Normal Vectors

RAY CASTING
Real-time Reconstruction

Demo


[7] Kok-Lim Low Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration
Thanks !!!