3D Reconstruction of Canal Profiles using Data Acquired from Teleoperated Boat

Supannee Tanathong, PhD
Content

1. Introduction
2. Hardware Description
3. Application Design and Implementation
4. Experimental Results
Content

1 Introduction
Over the past three years, Thailand has experienced widespread flooding in several regions across the country.

In order to reduce the hazard severity and minimize the area affected by flood, flood management personnel need to know the profiles of canals and waterways.

The profile data should, at least, include:

- Physical descriptions of canal banks
- Width between two sides of canal
- Depth of canal bed
- Water level
- Existing structures along the stretches of waterways

This will allow them to effectively direct the water flow in ways that can reduce the amount of flood water.
Solutions to measure/acquire canal profiles

- Field survey through direct measurement
- Terrestrial Laser Scanner (TLS)
- Airborne LiDAR Bathymetry (ALB)

To obtain the description and imaging view of canal banks and structures along the canals, we need

- A sort of TLS but with an added ability to produce continuous survey data.
- An inexpensive mechanism that can penetrate unclear water to the canal bed
In this research, we employ a mobile laser scanning technique for topographic survey of canal profiles.

- A teleoperated boat as a mobile platform
- A 2D laser scanner
- A Single-beam depth sounder
- A Global Positioning System (GPS)
- An Inertial Measurement Unit (IMU)
Content

3
Application Design and Implementation
• Development tools:
  
  o C/C++ language
  o OpenGL library (www.opengl.org – free from licensing requirements)
  o QT library (qt-project.org – open source license)
• Compute 3D point clouds

\[
LP^{GCS} = GP^{GPS} + R_{LCS}^{GCS} \times \begin{bmatrix} r \cdot \cos \theta & r \cdot \sin \theta & 0 \end{bmatrix}^T
\]

\[
DP^{GCS} = GP^{GPS} + R_{BCS}^{GCS} \times \begin{bmatrix} 0 & 0 & d \end{bmatrix}^T
\]

---

<table>
<thead>
<tr>
<th>Matrix</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_{LCS}^{GCS}$</td>
<td>$R_{GCS}^{GCS} \times R_{LCS}^{NED} \times R_{BCS}^{BCS}$</td>
</tr>
<tr>
<td>$R_{BCS}^{GCS}$</td>
<td>$R_{GCS}^{GCS} \times R_{BCS}^{NED}$</td>
</tr>
<tr>
<td>$R_{LCS}^{BCS}$</td>
<td>$\begin{bmatrix} 0 &amp; 0 &amp; -1 \ 1 &amp; 0 &amp; 0 \ 0 &amp; -1 &amp; 0 \end{bmatrix}$</td>
</tr>
<tr>
<td>$R_{BCS}^{NED}$</td>
<td>$\begin{bmatrix} 1 &amp; 0 &amp; 0 \ 0 &amp; \cos(\omega) &amp; -\sin(\omega) \ 0 &amp; \sin(\omega) &amp; \cos(\omega) \end{bmatrix} \times \begin{bmatrix} \cos(\phi) &amp; 0 &amp; \sin(\phi) \ 0 &amp; 1 &amp; 0 \ -\sin(\phi) &amp; 0 &amp; \cos(\phi) \end{bmatrix} \times \begin{bmatrix} \cos(\kappa) &amp; -\sin(\kappa) &amp; 0 \ \sin(\kappa) &amp; \cos(\kappa) &amp; 0 \ 0 &amp; 0 &amp; 1 \end{bmatrix}$</td>
</tr>
<tr>
<td>$R_{NED}^{GCS}$</td>
<td>$\begin{bmatrix} 0 &amp; 1 &amp; 0 \ 1 &amp; 0 &amp; 0 \ 0 &amp; 0 &amp; -1 \end{bmatrix}$</td>
</tr>
</tbody>
</table>

$^1$ In this work, the matrix is rotated around the Z axis first, then the Y axis, and the X axis.
```cpp
struct 3DPOINT
{   float X;   float Y;   float Z;   int C;   }

typedef std::vector<3DPOINT *> POINTCLOUD;

#define TOTAL_COLOR 188

struct R_G_B
{   int r;   int g;   int b;   }

R_G_B LOOKUP_TABLE [TOTAL_COLOR]
```
• Point cloud coloring
  
  o Ratio-based technique
    \[ C = (Z - \text{Min}_Z) \times \text{TOTAL\_COLOR}/(\text{Max}_Z - \text{Min}_Z) \]
  
  o Fixed-based technique
    \[ C = C\_OFFSET + (Z - \text{Base}_Z)/W \]
• Indoor datasets
Demo

- Outdoor datasets
Thank you for your attention

www.kgeo.org
KMUTT Geospatial Engineering and InnOvation Center (KGEO)
Institute of Field Robotics (FIBO)
King Mongkut’s University of Technology Thonburi