Urban Routing from Aerial Surveillance

Students:
M. Chen¹, Z. Clawson², A. Issaei³, S. Ramirez⁴, D. Richmond⁵, J. Wang⁶, E. Wu⁷

Project presenter:
John Peach, MIT Lincoln Laboratory

Faculty Mentors:
Cammey Cole Manning§, Ilse Ipsen†, Pierre Gremaud†

[§] Meredith College, [†] NC State

July 23, 2013
Outline

1. **Introduction & motivation**

2. **Road detection**

3. **Image stitching**

4. **Path planning**
   - Define *cost of travel*

5. **Conclusions & future work**
1. Introduction & motivation
Introduction

• **Problem:** Given aerial images:

  1. Create *map* of *roads* (road detection)
  2. Plan *optimal paths* on *roads*
Motivation

• **Applications:**
  - Military & combat situations
  - Evacuation route planning
  - Humanitarian crises
Prior work

- **Road Detection**
  - Anti-parallel edges & energy snakes \([\text{Peysakhov}, 2005]\)
  - Histogram threshold based algorithms \([\text{Lin & Saripalli}, 2012]\)

- **Image stitching & map-making**
  - Consistent matching \([\text{Li & Ma}, 2006]\)
  - Local invariant features \([\text{Brown & Lowe}, 2005]\)

- **Path Planning**
  - The literature is incredibly vast
Our method

Road detection + map-making + path planning:

1. **Road detection**
   - Uses **HSV** color model
   - Filters & masks used to detect roads
   - Image becomes **binary**: *road & not road*

2. **Map making & image stitching**
   - **SIFT** algorithm
   - **RANSAC** algorithm

3. **Path planning**
   - Solution to Eikonal PDE yields optimal paths
   - Requires a **cost function**
2. Road detection
HSV color model

- HSV color model $\implies$ image becomes H, S, and V
- HSV = *Hue, Saturation, & Value*
- H, S, and V are between 0 and 1
HSV of image:  H layer
HSV of image: S layer
HSV of image: All H, S, and V
Example: Raleigh neighborhood
HSV of image: All H, S, and V layers
Our algorithm for choosing H, S, or V

1. **Calculate** $\text{mean}(V)$
   1. **If** $\text{mean} \geq 0.45$
      - Choose **H** to isolate the road
   2. **If** $0.39 < \text{mean} < 0.45$
      - Choose **S** to isolate the road
   3. **If** $\text{mean} \leq 0.39$
      - Choose **V** to isolate the road

2. **Color removal** (e.g. trees)

3. **Display result**
NCSU Example: \( H \) layer

\[
\text{mean} = 0.51 \geq 0.45
\]
Raleigh 1 Example: S layer

\[ \text{mean} = 0.41 \in (0.39, 0.45) \]
Raleigh 2 Example: \( V \) layer

\[
\text{mean} = 0.35 \leq 0.39
\]
3. Image stitching
Image stitching

- **Two images:**

- **Stitch images** to form bigger picture

- **How?** SIFT and RANSAC algorithms
SIFT algorithm

• Locates **corners** in images

• Points are matched **very well**!
Result of stitching
Road detection results

- **Left image:** Belltower hides road
Road detection with stitching
4. Path planning
Eikonal equation $\iff$ optimal trajectories

- **Cost map** $C$ gives **cost** of travel

- **Eikonal PDE**

\[
\begin{cases}
|\nabla u(x)| &= C(x) & \text{for every } x \text{ on image} \\
u(t) &= 0 & \text{for target position } t
\end{cases}
\]

- $u(x)$ = the **minimum-total-cost** to travel from $x$ to $t$

- $u$ is found $\implies$ recover **optimal trajectory**
Optimal trajectory \( y^*(\cdot) \) gives

\[
u(x) = \int_0^T C(y^*(t)) \, dt.
\]
Cost map example: The image

- We developed an algorithm to construct $C$ from image
Cost map example: The cost map

- We *developed* an *algorithm* to construct $C$ from image.
NCSU example: Original 3 images

Now stitch them together!
NCSU example: 3 stitched images
NCSU example: Detected roads from 3 originals

Now stitch them together!
NCSU example: Detected road from stitch
NCSU example: Optimal trajectory
Raleigh example: Original image
Raleigh example: Detected road
Raleigh example: Cost map
Raleigh example: Optimal trajectories
5. Conclusions & future work
Conclusions

• New method for road detection and path planning

• HSV color model for road detection

• SIFT + RANSAC for image stitching

• Eikonal PDE + cost function for path planning

• Tested on many examples & works well
Future work

- **Road detection**
  - Detect and remove any 3D object (Structure from motion)
  - Further refinement of parameters
  - Detect other traversable areas

- **Map making / Image stitching**
  - Remove ‘ghost’ images
  - Use images with little/no overlap

- **Path planning**
  - Use machine learning methods to improve cost map
  - Predict path with only a partial map of the area
Thanks!