Proposal: Appendix 2

Mobile Scanning Systems:

A System Ready for Deployment
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Introduction</strong></td>
<td>- Posters</td>
</tr>
<tr>
<td>2</td>
<td>Understanding Mobile Scanning</td>
<td>- Pictorial presentation of system requirements and capabilities</td>
</tr>
<tr>
<td>3</td>
<td>UTK's Mobile Scanning Approach</td>
<td>- Leveraging of various sensors to develop the UTK Mobile Scanning System.</td>
</tr>
<tr>
<td>4</td>
<td>Research Plans:</td>
<td>- Integration and visualization of test sites, including the Knoxville Proving Grounds.</td>
</tr>
<tr>
<td>6</td>
<td>Research Plans:</td>
<td>Detail Enhancing Denoising of Digitized 3D Models from a Mobile Scanning System.</td>
</tr>
<tr>
<td>7</td>
<td>Research Plans:</td>
<td>Tire-Soil interaction using temporal stereo</td>
</tr>
<tr>
<td>8</td>
<td>UTK Publications</td>
<td></td>
</tr>
</tbody>
</table>
Developing detailed a priori 3D models of large environments to aid in robotic navigation tasks

Brad Grinstead, Andreas Koschan, and Mongi A. Abidi
Imaging, Robotics, and Intelligent Systems Laboratory, The University of Tennessee

Motivation
The research objectives for the Fast Digitization Project are:

- To develop a system to acquire realistic 3D models of large-scale real-world environments
- To provide accurate, to-scale 3D models at appropriate quality for aiding robotic navigation tasks

Accomplishments
- Initial data acquisition system developed and tested
- 3D models have been acquired and processed at a variety of resolutions, incorporating geometry and color textures
- Fusion (manual) of GPS, INS, video, and range data
- Differential GPS, RTK-IMU and color texture
- 3D geometry and texture
- 47,000 triangles
- 100% positional accuracy
- 350 m range,
- 5 cm accuracy
- 0.5° accuracy
- 80° vertical FoV
- 0 m
- 30 m
- Z210

Technical Approach

Data Acquisition

3D Range Image

Reconstructed Model

3D Geometry of Entire Building (shown as cloud of points)

Textured 3D Model

Future Work
- Using high resolution video sequences to augment our path and orientation estimation
- Incorporate multiple scanning views to enhance model verisimilitude
- Extend our pose estimation algorithm in a more robust framework
- Utilize multiple downward-looking laser scanners to investigate vehicle impact on assorted terrain types (e.g., tire tracks left in soft soil)
Building large scale 3D models for automotive testing under realistic conditions

Brad Grinstead, Andreas Koschan, Andrei Gribok, and Mongi A. Abidi
Imaging, Robotics, and Intelligent Systems Laboratory, The University of Tennessee

Motivation
The research objectives for the UTK Mobile Scanning Project are:

• To develop a system to acquire realistic 3D models for large-scale real-world environments.
• To utilize the acquired data to develop better models for vehicle simulation and testing under realistic conditions.

Fusion
High-res video sequences

Accomplishments
The research objectives for the UTK Mobile Scanning Project have been implemented (manual) of GPS, INS, video, and range data hardware have been used to develop a system to acquire realistic 3D models, incorporating multiple sensors, and process the data to provide an accurate and realistic 3D model.

Technical Approach

Once the data has been acquired, and the data fusion performed, the resulting models are then processed to: fill holes, complete textures, reduce noise effects, and develop multi-resolution models so that appropriate levels-of-detail can be retrieved as necessary for the application at hand.

Some areas of the model are sampled at too low resolution to make out the details. Often, these areas are important features for the target application. In order to not have to reacquire the data at a higher resolution, we are developing an algorithmic method - based on kriging - to refine the data by locally inserting new information. Below are some results on using this method to clean up a sign on a building.

Future Work

• Automate the detail-enhancement processing
• Develop a 3D surface analysis tool for decomposing a 3D surface into component local frequencies
• Integrate the surface analysis with an anisotropic denoising algorithm
• Increase the robustness of the pose estimation procedure to noise effects
Vehicle-borne Scanning for Detailed 3D Terrain Model Generation

Brad Grinstead
Andreas Koschan, David Page, Andrei Gribok, and Mongi A. Abidi

November 2, 2005
Applications
Large-scale Terrain Scanning

Can deliver accurate, photo-realistic, geo-referenced 3D models of road terrain.
Why not use…?

Profilometer Data

Accumulated over space

Image from DoD website

3D model for simulations

Real-world terrain cannot be easily modeled in a functional form!
Why Mobile Scanning?


Enhanced Simulations & Robotic Navigation

http://www.drive.cranfield.ac.uk/cfml/l3dorn.cfm

http://www.robotics.utexas.edu/
How?
Mobile Scanning System Concept
To develop a system capable of delivering accurate, photo-realistic, geo-referenced 3D models of large-scale terrain.

The data will be acquired in real-time at an appropriate resolution for the application at hand, using a modular system.
Pipeline

- Video Sequence
  - Feature Detection and Matching
  - Pose Estimation from Video
- GPS
- INS
  - 3D Position and Orientation
  - Combined Pose Estimation
- Range Profiles
  - Range Profile Alignment
  - Data Fusion

3D Model Processing
Approach

- **Texture**: JVC GR-HD1 HDTV camera
- **Position**: Leica GP5500 Differential RTK 2cm positional accuracy
- **Orientation**: Xsens MT9 0.5° accuracy IMU

Data Acquisition → 3D Range Image

Data Fusion → Yields

- Textured 3D Model
- Shopping Strip
- Data Processing
Data Types

Range Data
- **Riegl LMS-Z210** laser range scanner
  - 350 m range, 80° vertical field of view, **5 cm accuracy**
- **Sick LMS200** laser range scanner
  - 8m range, 180 ° field of view, **1 cm accuracy**

High-resolution Video
- **JVC GR-HD1** high definition camcorder

Position
- **Leica System 500** Global Positioning System (GPS)
  - Differential Real Time Kinematic (RTK) system
  - 10 measurements/second, **2 cm accuracy**

Orientation
- **XSens MT9** Inertial Measurement Unit (IMU)
  - 100 measurements/second, **0.5° accuracy**
Direct Self-localization

- Global Positioning System
- Inertial Measurement Unit
- Global Positioning System

- Inertial Measurement Unit
Indirect Self-localization

Video Sequence

Feature Matching
- Detection
- Robust tracking

Pose From Motion

Relate the images
Recover the motion

Image Sequence → Detect Features → Track Features → Estimate Motion → Introduce Scale → 6DOF Motion

motion feedback

Image, Robotics, and Intelligent Systems
IRIS
Fusion of multi-sensor data

GPS curve sampled at 10 Hz.

\[ P_t = [x_g^t, y_g^t, z_g^t]^T \]

IMU data at 100 Hz
\((\omega, \phi, \kappa)\)

Video recorded at 30 frames/sec

Time synchronization is the key.

\[ R_tD_t + P_t = W_t \]

\[ D_t = [x_r^t, y_r^t, z_r^t]^T \]

Range Profiles @
30 Hz if using the SICK
21 Hz if using the Riegl
Processing the Model

**Reducing the effects of localization error**

\[ p(y_i | x_i) \propto \exp \left( -\frac{1}{2} (x_i - x_i')^T A^{-1} (x_i - x_i') \right) \]

Criterion: Maximize \( p(y_i | x_i) p(\Delta x_i | x_i, x_{i-1}) p(x_i | x_{i-1}) \)

**Removing redundant geometry**

\[ d_{\text{cross}} = \frac{\Delta s}{\sin(\Delta \theta)} \]

\[ \text{filter}(r) = \begin{cases} r \leq d_{\text{cross}}, & \text{keep} \\ \text{otherwise}, & \text{discard} \end{cases} \]
Video mosaicking through Phase Correlation

\[
\frac{F(\xi, \eta)F^*(\xi, \eta)}{|F(\xi, \eta)F'(\xi, \eta)|} = e^{i2\pi(\xi x_0 + \eta y_0)}
\]

Why not use this all the time?

Method fails because the texture itself looks like impulse noise.
Video mosaicking through Feature Matching

We can calculate the camera transformation from 7 matching pairs, but there is no guarantee of robustness to noise.

We use Random Sampling and Consensus (RANSAC) to robustly calculate the translation and rotation of the camera between views.

- Extract features
- Compute potential matches
- While (%inliers<1)
  - Select a minimal sample of 7 matches
  - Compute $F$
  - Determine inliers
- Refine $F$ based on all inliers

\[
\begin{bmatrix}
    F_{11} & F_{12} & F_{13} \\
    F_{21} & F_{22} & F_{23} \\
    F_{31} & F_{32} & F_{33}
\end{bmatrix}
\begin{bmatrix}
    u \\
    v \\
    1
\end{bmatrix}
= 
\begin{bmatrix}
    u' \\
    v' \\
    1
\end{bmatrix}
\]
Processing the Model

Detail enhancement and local geometry refinement

1. Estimate the spatial correlation between samples
2. Construct an ideal model that best fits the calculated correlation
3. Estimate new surface values using Kriging

\[
\gamma(d) = \frac{1}{2N(d)} \sum_{i=1}^{N(d)} (z_i - z_{j+h})^2
\]

Spatial Correlation

\[
Z_e = \sum W_i \cdot Z_i
\]

Estimation
Processing the Model

### Noise Removal

- Original model: 363843 triangles, 185345 points
- Reduced to 25%: 90893 triangles, 48595 points
- Reduced to 2.5%: 9075 triangles, 6642 points

### Adaptive Simplification

- Initial Model
- Multiresolution Analysis and Denoising

Diagram showing the process of noise removal and adaptive simplification.
Early Experiments

Data Fusion

The red line is the GPS path that our mobile system traversed. We have shown the raw frame data from the camera in the left and the mosaic after integration using the GPS information.

Complete model: 750,000 triangles
Acquired 10 m worth of data in 18 seconds

Textured 3D Model
Early Experiments
Women’s Basketball Hall of Fame

Vehicle Path

3D Range Data

Point Cloud

Video Mosaic

Textured 3D Model

Complete model: 2.8 million triangles
Acquired 600 m worth of data in 52 seconds
More Experiments
Morrel Center Strip Mall

Hardware vs. Video Path Comparison

Gaps in the GPS path are **deliberate**, simulating loss of satellite lock, demonstrating the need to a hybrid self-localization system.

**Qualitative results** of the combination of the localization modules with the laser range data.
Larger-scale Experiments

UTK Facilities

Complete model: 2.5 million triangles
Acquired 419 m worth of data in 5 minutes

GPS Path
Imaged Area
Zoomed in View
Entire Path Scanned

2m wide and 8m long
Larger-scale Experiments
Downtown West Mall

Complete model: 5.3 million triangles
Acquired 1.4 km worth of data in 18 minutes
Applications

Model Building for Vehicle Simulation

Road Terrain Mapping

Road terrain mapping using a mobile scanning system to generate accurate 3D terrain models for vehicle-environment and soil-tire interaction simulators.

From reality testing to virtual simulations…

Image from Aberdeen Proving Grounds, Maryland
Soil mechanics must be accounted for in off-road vehicle simulation in order to accurately predict vehicle handling performance. **Accurate and validated** soil models are in high demand for real-time dynamic simulation environments.

Validation of simulated models ...

Soil Mechanics and Simulation

Tire Impression in Loose Soil

3D Terrain Model

Synthetic Model from Simulation

US Army Corps of Engineers

www.crrel.usace.army.mil/research/factsheets/tiremodel.doc
Applications
Close-support Inspection of Pavement

Pavement Inspection

Automatic detection of deterioration (cracks, potholes, etc.) and rating of pavement and off-road surfaces that can negatively affect the vehicles moving on them.

Manual Inspection?

Different types of distresses
www.faa.gov

Inspection and categorizing ...

Pavement patch and the scanned 3D model
Summary

• We have developed a modular Mobile Scanning System that can acquire dense, accurate 3D and color data in real time
• We have developed the tools to combine that data and provide accurate, photo-realistic, geo-referenced 3D models of large environments
• Experimentation has been done with a variety of system configurations on a large number of outdoor scenes of various sizes
Where Do We Go From Here?

- Robust interprofile registration
  - Further work on removing the effects of noise on the pose measurements and fusion

- 2D profile correction
  - Evaluating the “goodness” of measured points
  - Feature detection on a per-profile basis

- Detail enhancing denoising
  - A more intelligent denoising process
  - Identify 3D structures that are on the same order of magnitude as the system noise, and algorithmically enhance them while removing noise on the model

- Further experimentation
  - Off road experiments
  - More surface types
    - Gravel, grass, dirt, etc
Outdoor Scene Modeling Using a Mobile Multi-Sensor Platform

Brad Grinstead
May 19, 2004

UTK Advisors: Mongi Abidi, Andreas Koschan
Why?

Enhanced Simulations & Robotic Navigation

http://www.robotics.utexas.edu/

Concept

Profile Sequence → Textured 3D Model
Objectives

Develop a system for the **fast digitization** of real environments

- Utilize vehicle-borne scanning
  - Scanner system
  - Hardware for inter-profile alignment
- Increased verisimilitude over previous methods
- Results can be used for a variety of tasks
  - Realism in simulations
  - Driver performance evaluation
  - Aid in robotic navigation tasks

Overall Approach

Data Acquisition

- JVC GR-HD1 HDTV camera
- Leica GPS500 Differential RTK 2cm positional accuracy
- XSens MT9 0.5° accuracy IMU

Data Fusion

Yields

Textured 3D Model
Data Types

Range Data
  - **Riegl LMS-Z210 laser range scanner**
    - 350 m range, 80° vertical field of view, 5 cm accuracy

High-resolution Video
  - **JVC GR-HD1 high definition camcorder**

Position
  - **Leica System 500 Global Positioning System (GPS)**
    - Differential Real Time Kinematic (RTK) system
    - 10 measurements/second, 2 cm accuracy

Orientation
  - **Xsens MT9 Inertial Measurement Unit (IMU)**
    - 100 measurements/second, 0.5° accuracy
Individual profiles are converted to a global reference frame

\[ V_p = \ell + Rv_p \]

\( R \) is formed from the roll, pitch, and yaw parameters of the scanner.
Pose Estimation - Hardware

- Global Positioning System
- Inertial Measurement Unit
Pose Estimation from Video

Use standard feature matching and tracking from video sequences to provide additional pose information to enhance current system.
Hybrid Pose Estimation System

GPS

INS

3D Position & Orientation

Video Sequence

3D Points

Motion State Estimation

Estimation of Scene Structure

Camera Pose
(predicted feature locations)

Pose Estimation
Model Processing

Level of Detail (LOD) representation allows view-dependent triangulation.

We can also use multiresolution analysis to perform intelligent denoising of surface models.
Results – Strip Mall

Complete model: 823,000 triangles

Acquired 240 m worth of data in 25 seconds
Results – Women’s Basketball Hall of Fame

Image of building

3D geometry

Composite texture image

3D Model

Complete model: 2.8 million triangles

Acquired 600 m worth of data in 52 seconds
Results - Terrain Modeling

Parking Lot Test Run

~400 m
900k points
1.8 million triangles

Range image section

Section of 3D model

3D section of point cloud
Wrap Up

- Initial data acquisition system has been developed and tested for the fast digitization of environments
  - 3D models have been acquired and processed at a variety of resolutions, incorporating both geometry and color textures
  - Pose instrumentation has been integrated

The acquired models are true to life in terms of geometry, scale, and color

These models are appropriate for aiding in a variety of tasks including: realistic simulations, robotic navigation, etc.
Future Efforts

**Data Regularization**
Use **mathematical techniques** to remove the effects of inconsistent and noisy data. Modeling sources include GPS phase intervals, INS error models, known geometry/timing error models of range scanner.

**Vehicle Impact Evaluation**
Use **multiple downward-looking scanners** to measure the impact of the vehicle on various terrain types (e.g., tire treads left in soft soil).
Multi-perspective Mosaics and Layered Representation for Scene Visualization
Jin-Choon Ng
Wide-Angle Vision and Synthetic Environments

- Wide-angle representations of environments assist scene visualization.
- Increased range of perception.
- Enhanced realism.

Contemporary painting of the construction of a Panorama Rotunda.
c.1830
http://www.acmi.net.au/AIC/PANORAMA.html

Cross section of Robert Barker’s Panorama, Leicester Square, London, 1789
http://www.acmi.net.au/AIC/PANORAMA.html
Motivation

• Video sequences from road scenes:
  Wide-angle views may be used as texture for 3D reconstruction of road scenes.

• Video sequences for under-vehicle inspection:
  Wide angle views facilitate the inspection process.
Motivation
Motivation

• Why not use an omni-directional image capture system (e.g. catadioptric system, fish-eye lens)?

  [Image]

  http://www.inrialpes.fr/movi/people/Sturm/

• Requires Specialized Equipment
• Loss of Image Resolution
• Distortion of Images
Digital Image Mosaicing

- **Mosaic**: An image that is a composite of several smaller images.

- Techniques for the creation of mosaics using computers:
  Fast generation of seamless mosaics e.g. panoramas

http://www.cs.rochester.edu/u/kyros/Courses/CS290B/Lectures/lecture-19/sld027.htm
Digital Panoramas

- A stationary camera, rotated around its optical center, captures the input images.
- Creation of a cylindrical/spherical panorama:
  - input images are reprojected into cylindrical/spherical coordinates based on the focal length (or field of view) of the camera.
  - the correct alignments between the warped images are determined
  - images are merged to form the panorama
Digital Panoramas

- Limitation of panoramic mosaicing:
  - Camera movement is restricted to rotation about the optical center.
  - Cannot mosaic image sequences displaying large motion parallax.

- Solution: *multi-perspective mosaicing.*
Digital Panoramas

Stationary camera, Panning motion
Digital Panoramas

Translating camera
Multi-perspective Mosaicing

- Camera motion is no longer restricted to panning motion from a single perspective; the camera’s optical center moves.
- Each camera image may represent a unique perspective of the scene.
- Strips are sampled from each input image and combined to form the mosaic.
Previous Work


Zhu et al [21], “Stereo mosaics from a moving camera for environmental monitoring,” 1999

Previous Work


Multi-perspective Mosaicing

- Two cases:
  1. The camera translates past the scene, and motion parallax is relatively small.
  2. The camera translates past the scene, and the scene exhibits large motion parallax.
Layered-Mosaic Representation

- Case 2: Motion parallax is large.
- The scene is no longer assumed to exist on a single plane; the scene is represented as a series of planar layers.

- A unique inter-frame velocity is associated with each layer.
- Occlusions are present.
Layered-Mosaic Representation

- Why represent the scene as a series of layers?

- Sample each element in the scene according to its velocity.
- Deal with occlusions.

Layered-Mosaic Representation

Preprocessing
- Barrel Distortion Correction
- Angle Compensation

Registration
- Motion analysis
- Spatial Support Determination

Merging
- Strip Selection
- Layer composition

Output Mosaics

Input Images

Model Initialization
Layered-Mosaic Representation

- Model Initialization is performed manually
  - Number of layers ($n$)
  - Velocity ($x_n, y_n$) associated with each layer
Layered-Mosaic Representation

- Lucas-Kanade Motion Analysis

\[
\begin{bmatrix}
\sum w \nabla I_x^2 & \sum w \nabla I_x \nabla I_y \\
\sum w \nabla I_x \nabla I_y & \sum w \nabla I_y^2
\end{bmatrix}
\begin{bmatrix}
u \\
v
\end{bmatrix}
= -\begin{bmatrix}
\sum w \nabla I_x \nabla I_t \\
\sum w \nabla I_y \nabla I_t
\end{bmatrix}
\]

(6)

- A two-scale coarse-to-fine implementation is used.
Layered-Mosaic Representation

Input images, $D_1, D_2, D_3, D_4$, and $D_5$

$\sigma, \mu$

Rescale input images with rescale factor $f$. $v_{\text{lim}} = 2f_2$

Perform spatiotemporal smoothing on $D_1, D_2, D_3, D_4$, and $D_5$

$f_1, f_2$

Compute intensity gradients $I_x, I_y$, and $I_t$ of each pixel in $D_3$. $f = f_1$

Smooth intensity gradients.

Solve Equation (6) for $(u', v')$ for each pixel in $D_3$. $f = f_2$

no

Resize flow field to original resolution. Discard all $(u,v)$ vectors with $|f| < v_{\text{lim}}$.

yes

Resize second flow field to original resolution. Substitute previously discarded $(u,v)$ vectors in first flow field with corresponding vectors from second flow field.

$(u,v)$ for each pixel in $D_3$
Layered-Mosaic Representation

Yosemite sequence

Flow field partially computed using $f_1$ scale

Flow field completed using $f_2$ scale

Flow of sequence
Layered-Mosaic Representation

- A hierarchical morphological operator is used to reduce the noise in the results of segmentation of each video frame.
Layered-Mosaic Representation

Sample video sequence

Computed pixel velocities for one frame in the sequence
-5 pixels/frame
-4 pixels/frame
-3 pixels/frame
-2 pixels/frame
-1 pixels/frame

Segmentation of that frame
(Each segment is associated with a different layer)
Layered-Mosaic Representation

• Forming the Reference and Peripheral Mosaics

\[ \text{dist} \times \text{dist} \times \text{dist} \times \text{dist} \]

• Parameters:
  – The number of mosaics used to form the layer, \( k \).
  – The distance between the strips, \( \text{dist} \).
Layered-Mosaic Representation

- Layer Composition
Layered-Mosaic Representation

- Layer Composition: Dealing with occlusion.
Layered-Mosaic Representation

Warren, 504 Frames

BBHall, 914 Frames
Layered-Mosaic Representation

Sample frame from Warren sequence

Intended layer assignment

Sample frame from BBHall sequence

Intended layer assignment

Sample frame from BBHall sequence

Intended layer assignment
Layered-Mosaic Representation

- Sample frame from Warren sequence
- Flow field computed using Lucas-Kanade algorithm
- Rough segmentation based on model initialization
- Refinement of segmentation after hierarchical morphological operation
Layered-Mosaic Representation

Sample frame from BBHall sequence

Flow field computed using Lucas-Kanade algorithm

Rough segmentation based on model initialization

Refinement of segmentation after hierarchical morphological operation
Layered-Mosaic Representation

Intended layer assignment

Experimental result
Layered-Mosaic Representation

Sample of reference and peripheral mosaics created for layer composition of Warren sequence
Layered-Mosaic Representation

Sample of reference and peripheral mosaics created for layer composition of BBHall sequence
Layered-Mosaic Representation
Layered-Mosaic Representation
Conclusions

• Two methods: Single-mosaic representation and Layered-mosaic representation.

• Single-mosaic representation:
  – Phase correlation as a registration technique for multiperspective mosaicing

• Layered-mosaics representation:
  – Representation of scenes as mosaics of depth layers
Future Work

- Extend to cases of surfaces displaying more general motion.
- Automate model initialization.
- Improve spatiotemporal segmentation.
- Real-time implementation.
Outlier Rejection to Aid Pose from Video

Brad Grinstead
February 10, 2004
**Goals**

Develop a semi-automatic system for digitizing large-scale real world environments

- Utilize vehicle-borne scanning
  - Scanner system
  - Pose estimation sensors and data fusion
  - Video processing
Mobile Range Scanning Concept

Profile Sequence

Textured 3D Model
Road Scanning for Terrain Modeling

http://arc.engin.umich.edu/arc/conference/conf97/case3.pdf
Individual profiles are converted to a global reference frame

\[ V_P = \ell + R v_p \]

\( R \) is formed from the roll, pitch, and yaw parameters of the scanner.
Instrumented Pose Estimation

Position
  – Global Positioning System (GPS)

Orientation
  – Inertial Measurement Unit (IMU)
    *data can be processed to provide positioning also (INS)

Other
  – Odometer
  – Bearing meters
  – Velocimeters
Pose from Video

- Relate the images
- Recover the motion

Video Sequence

Feature Matching
- Detection
- Robust tracking

Pose From Motion
Hybrid Pose Estimation System

- GPS
- INS

3D Position & Orientation

Motion State Estimation

Video Sequence

Camera_pose (predicted feature locations)

3D Points

Estimation of Scene Structure

Pose Estimation
Pose From Video - Process

- Obtain calibration parameters $\mathbf{K}$ for camera
- Remove radial and tangential distortion effects
- Find corresponding points
- Calculate Fundamental Matrix $\mathbf{F}$ from corresponding points
- Use $\mathbf{K}$ and $\mathbf{F}$ to calculate pose
Feature Correspondence

- Standard Harris Corner Detector (Harris ’88)
- Shi-Tomasi Texturedness Detector (Shi ’94)
- Local Jet (Schmid ’97)
- Local Binary Patterns (Tian ’03)
- Others

- Normalized cross-correlation
- Dissimilarity metrics
- Texture Matching
- Others
Epipolar Geometry Estimation

- Use some feature detection to identify matching features
- We can calculate $F$ from 7 matching pairs, but there is no guarantee of robustness to noise
Pose Estimation

Essential Matrix - calculated from $F$ and $K$

$E = K^T F K$

Calculate translation direction - $\min\|ET_s\|$

$T_s$ is the unit eigenvector with the smallest eigenvalue of the matrix $EE^T$

Calculate orientation - $\min\|(R^T [-T_s]_x - E^T)\|$
Pose from Video Problems

- Grayscale images don’t always provide good features
- Current feature tracking methods still have problems with noisy features
- Moving foreground/background causes problems with current feature matching algorithms
- No matter the feature matching algorithm, the output will contain false matches
Outlier Rejection - Background

- Robust geometry estimation algorithms
  - RANSAC
    - Zhang '95, Pollefeys '97, Hartley '00, Salvi '01

- Filtering techniques
  - Image Residual (Shi '94)
  - Illumination Invariant Residual (Tommasini '98)
  - Triplet Consistency (Chua '00)
  - Mutual Best Match (Pollefeys '97, Hartley '00, Corke '04)
Outlier Rejection - RANSAC

- Extract features
- Compute potential matches
- While (%inliers<\(\Gamma\))
  - Select a minimal sample of 7 matches
  - Compute \(F\)
  - Determine inliers
- Refine \(F\) based on all inliers

Time complexity is based on the %outliers present in the data

\[N = \text{ceil}\left(\frac{\log(1 - \Gamma)}{\log(1 - (1 - \varepsilon)^n)}\right)\]

where \(\varepsilon\) is the percentage of outliers present in the data
Outlier Rejection – Mutual Best Match (MBM)
Observations
Oriented Tracks

- There is evidence that feature matches that correspond to the scene’s epipolar geometry exhibit similar behavior in their motion tracks.
- To filter out wrong matches, we simply have to identify the common behavior of the correct features.
ROR: Rejection of Outliers by Rotation

- ROR (Adam ’01) – determines common behavior by performing a series of virtual rotations to the images in an attempt to “shake away” the outliers

1. Start with a set of feature correspondences between images A and B
2. Perform \( K \) random virtual rotations of B. For each rotation…
   a) Find the mode of the distribution of segment directions, \( m_k \)
   b) Find the angular difference \( d_{ik} \) between each segment and the mode
   c) Find the minimal window about \( m_k \) that contains \( q \) percent of the segments
3. Order the \( K \) rotations according to increasing window widths
4. For each feature, compute the average angular difference \( D_i \) from the mode
5. Compute the mode \( T \) of the average distances
6. Reject those features for which \( D_i < T + \alpha \)
Rejection of Outliers by Oriented Tracks

- By using some assumptions supported by our mobile scanning system, we can greatly simplify the problem
  - Video camera is mounted so that its optical axis is orthogonal to the main direction of vehicle motion
  - Limited mobility of the platform means that motion about or along the optical axis is small compared to the orthogonal motion
  - High video acquisition rate means inter-frame motion is relatively small (< 50 pixels)
Rejection of Outliers by Oriented Tracks

1. Start with an initial set of feature correspondences
2. Estimate the probability distribution of the feature track directions
   a) Use kernel density estimation to calculate the mode
   b) Automatically calculate kernel width using Plug-In method
3. Determine the orientation vector that is “on average” parallel to the feature tracks from the estimated density function as the mode of the distribution $m$
4. For each feature calculate the angular difference $d_i$ from $m$
5. Reject features whose angular difference is greater than some threshold $\alpha$
Mode Calculation

- Histograms are notoriously sensitive to initialization
  - Kernel density estimation is a better way to go

\[ \hat{pdf}(x) = \frac{1}{nh} \sum_{i=1}^{n} K \left( \frac{x - X_i}{h} \right) \]
Determining Kernel Width

What effect does the kernel width have on our approximation?

- **$h$ too small**: Over repeated sampling, the spikes appear to shift.
- **$h$ too large**: Reduced variability at the cost of introducing bias.
- **$h$ just right**: Minimal bias, with reduced variability.
Optimizing Bandwidth

Minimize Asymptotic Mean Integrated Square Error

\[
\hat{AMISE}(f) = \frac{R(K)}{nh} + \frac{1}{4} h^4 \mu_2(K)^2 R(f^\gamma)
\]

Differentiating wrt. \( h \)

\[
h_{opt} = \left[ \frac{R(K)}{\mu_2(K)^2 R(f^\gamma) n} \right]^{1/5}
\]

Assuming pdf can be represented as a normal distribution

\[
h_{opt} = \left[ \frac{4 \sigma}{3n} \right]^{1/5}
\]


Choosing an Acceptance Window

- Empirically (current method)
- Data-driven
  - Gaussian Mixture Models
  - Distribution of the directions
  - Break point of acceptance
Gaussian Mixture Models

1. Use EM to model distribution as a combination of

2. Use the variance of the Gaussian model to estimate acceptance range

---

The trained model has parameters:

<table>
<thead>
<tr>
<th>Priors</th>
<th>Centers</th>
<th>Variances</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4329</td>
<td>-0.8038</td>
<td>1.9505</td>
</tr>
<tr>
<td>0.1348</td>
<td>2.3788</td>
<td>0.2968</td>
</tr>
<tr>
<td>0.4323</td>
<td>-0.2526</td>
<td>0.0041</td>
</tr>
</tbody>
</table>
Procedure

- Acquired data from both the **Sony Handicam** and the **JVC HD camcorder**
- For each image pair, the features were detected and matched
  - **Harris Corner Detector** combined with **Intensity Correlation**
- Both **MBM** and **OT** outlier rejection methods were applied
- RANSAC was used to estimate the **epipolar geometry**
  - **Accuracy** of each algorithm was determined by the number of features that lay within 2 pixels of their calculated epipolar lines (inliers)
  - **Efficiency** was determined by the amount of time required for outlier rejection and the number of RANSAC iterations
  - **1000 trials** of the RANSAC procedure were run for a statistical sampling of the results
Outlier Rejection Results – Warren, MI Video Sequence

<table>
<thead>
<tr>
<th>Algorithm</th>
<th># RANSAC Iterations</th>
<th>Outlier Rejection (s)</th>
<th>RANSAC (s)</th>
<th>Total (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oriented Tracks</td>
<td>4</td>
<td>0.12</td>
<td>0.09</td>
<td>0.21</td>
</tr>
<tr>
<td>Mutual Best Match</td>
<td>26</td>
<td>131.32</td>
<td>.61</td>
<td>131.93</td>
</tr>
</tbody>
</table>
Outlier Rejection Results – Downtown HD Video Sequence

<table>
<thead>
<tr>
<th>Algorithm</th>
<th># RANSAC Iterations</th>
<th>Outlier Rejection (s)</th>
<th>RANSAC (s)</th>
<th>Total (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oriented Tracks</td>
<td>18</td>
<td>0.20</td>
<td>0.18</td>
<td>0.38</td>
</tr>
<tr>
<td>Mutual Best Match</td>
<td>53</td>
<td>119.32</td>
<td>.96</td>
<td>120.28</td>
</tr>
</tbody>
</table>
Sony Handicam Images
Sony Handicam Images – Inliers & Timing Results

<table>
<thead>
<tr>
<th>Scene</th>
<th>Rejection Algorithm</th>
<th>RANSAC Iterations</th>
<th>Outlier Rejection (s)</th>
<th>RANSAC (s)</th>
<th>Total (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WBHoF</td>
<td>Oriented Tracks</td>
<td>5</td>
<td>0.22</td>
<td>0.18</td>
<td>0.40</td>
</tr>
<tr>
<td></td>
<td>Mutual Best Match</td>
<td>8</td>
<td>90.74</td>
<td>0.22</td>
<td>90.96</td>
</tr>
<tr>
<td>Baymont</td>
<td>Oriented Tracks</td>
<td>6</td>
<td>0.33</td>
<td>0.11</td>
<td>0.44</td>
</tr>
<tr>
<td></td>
<td>Mutual Best Match</td>
<td>18</td>
<td>95.11</td>
<td>0.26</td>
<td>95.37</td>
</tr>
<tr>
<td>Comerica</td>
<td>Oriented Tracks</td>
<td>5</td>
<td>0.23</td>
<td>0.13</td>
<td>0.36</td>
</tr>
<tr>
<td></td>
<td>Mutual Best Match</td>
<td>11</td>
<td>63.25</td>
<td>0.25</td>
<td>63.50</td>
</tr>
<tr>
<td>Restaurant</td>
<td>Oriented Tracks</td>
<td>3</td>
<td>0.27</td>
<td>0.08</td>
<td>0.35</td>
</tr>
<tr>
<td></td>
<td>Mutual Best Match</td>
<td>6</td>
<td>97.58</td>
<td>0.18</td>
<td>97.76</td>
</tr>
</tbody>
</table>
JVC HD Video Images
JVC HD Video Images – Inliers & Timing Results

<table>
<thead>
<tr>
<th>Scene</th>
<th>Rejection Algorithm</th>
<th>RANSAC Iterations×</th>
<th>Outlier Rejection (s)</th>
<th>RANSAC (s)</th>
<th>Total (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Garage</td>
<td>Oriented Tracks</td>
<td>8</td>
<td>0.33</td>
<td>0.19</td>
<td>0.52</td>
</tr>
<tr>
<td></td>
<td>Mutual Best Match</td>
<td>53</td>
<td>112.08</td>
<td>1.18</td>
<td>113.26</td>
</tr>
<tr>
<td>Gas Tank</td>
<td>Oriented Tracks</td>
<td>2</td>
<td>0.27</td>
<td>0.04</td>
<td>0.31</td>
</tr>
<tr>
<td></td>
<td>Mutual Best Match</td>
<td>9</td>
<td>90.61</td>
<td>0.18</td>
<td>90.79</td>
</tr>
<tr>
<td>The Hill</td>
<td>Oriented Tracks</td>
<td>4</td>
<td>0.38</td>
<td>0.09</td>
<td>0.47</td>
</tr>
<tr>
<td></td>
<td>Mutual Best Match</td>
<td>21</td>
<td>195.03</td>
<td>0.55</td>
<td>195.58</td>
</tr>
<tr>
<td>Storage Tank</td>
<td>Oriented Tracks</td>
<td>7</td>
<td>0.33</td>
<td>0.16</td>
<td>0.49</td>
</tr>
<tr>
<td></td>
<td>Mutual Best Match</td>
<td>36</td>
<td>128.88</td>
<td>0.82</td>
<td>129.70</td>
</tr>
</tbody>
</table>
What Happens If …
What Happens If …
Conclusions

• We have developed an outlier rejection method that can be used to enhance PfV on a robotic platform
  – Used as an additional pre-filtering stage between existing feature detection and epipolar geometry estimation stages

• Benefits
  – Reduces the number of outliers by up to 90%
  – Can be used on datasets containing where the number of false matches is as high as 80%
  – Effectively halves the amount of computation time over the standard MBM method
Improving Video-Based Robot Self Localization Through Outlier Removal

David Page

Brad Grinstead, Andreas Koschan, Andrei Gribok, and Mongi A. Abidi
February 14, 2006
Motivation

• In order to perform its duties effectively, a robot must know its position and orientation within its working environment
  – Known as self-localization or pose estimation

• Two main methods are used to do this
  – Direct (measured) Localization – typically used in outdoor environments
  – Indirect (inferred) Localization – typically used in indoor environments

• Optical navigation can be performed using inexpensive COTS video cameras
  – Motion estimated by determining the interframe motion between successive images
  – The quality of the motion estimate is dependant on the features matches between images
Example Scenario

Imagine a robot that is performing surveillance/inspection of a parking facility at an airport.

- Facility consists of both indoor and outdoor parking spots
- Robot must travel back and forth between parking areas
Feature-based Self-localization

- Relate the images
- Recover the motion

Video Sequence

Feature Matching

Pose From Motion

Image Sequence → Detect Features → Track Features → Estimate Motion → Introduce Scale → 6DOF Motion
Epipolar Geometry

• Epipolar geometry plays an important role in determining the **rigid transform** between 2 cameras.

• Correct estimation of the epipolar geometry is directly dependant on the **quality** of the feature matches provided.

• The feature matching task is **difficult** enough that even the most sophisticated algorithms will yield **false matches**, in addition to the true matches.
What Causes Feature Mismatches?

- Grayscale images don’t always provide good features
- Current feature tracking methods still have problems with noisy features
- Moving foreground/background causes problems with current feature matching algorithms

- No matter the feature matching algorithm, the output will contain false matches
Outlier Rejection - Background

• Robust geometry estimation algorithms
  – **RANSAC**
    • Zhang ’95, Pollefeys ’97, Hartley ’00, Salvi ’01

• Filtering techniques
  – Image Residual (Shi ’94)
  – Illumination Invariant Residual (Tommasini ’98)
  – Triplet Consistency (Chua ’00)
  – **Mutual Best Match** (Pollefeys ’97, Hartley ’00, Corke ’04)
Outlier Rejection - RANSAC

- Extract features
- Compute potential matches
- While ($\%\text{inliers} < \Gamma$)
  - Select a minimal sample of 7 matches
  - Compute $F$
  - Determine inliers
- Refine $F$ based on all inliers

Time complexity is based on the $\%$ outliers present in the data

$$N = \text{ceil}(\log(1 - \Gamma) / \log(1 - (1 - \varepsilon)^P))$$

where $\varepsilon$ is the percentage of outliers present in the data
Outlier Rejection – Mutual Best Match (MBM)
Observations
Rejection of Outliers by Oriented Tracks

1. Start with an initial set of feature **correspondences**
2. Estimate the **probability distribution** of the feature track directions
   a) Use **kernel density estimation** to calculate the mode
   b) Automatically calculate **kernel width** using Plug-In method
3. Determine the **orientation vector** that is “on average” parallel to the feature tracks from the estimated density function as the mode of the distribution \( m \)
4. For each feature calculate the **angular difference** \( d_i \) from \( m \)
5. **Reject features** whose angular difference is greater than some threshold \( \alpha \)
Mode Calculation

- Histograms are notoriously **sensitive** to initialization
  - Kernel density estimation is a better way to go

\[
\hat{pdf}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x-X_i}{h}\right)
\]
Determining Kernel Width

What effect does the kernel width have on our approximation?

- **$h$ too small**: Over repeated sampling, the spikes appear to shift.
- **$h$ too large**: Reduced variability at the cost of introducing bias.
- **$h$ just right**: Minimal bias, with reduced variability.
Optimizing Bandwidth

Minimize Asymptotic Mean Integrated Square Error

\[ \text{AMISE}(\hat{f}) = \frac{R(K)}{nh} + \frac{1}{4} h^4 \mu_2(K)^2 R(f^2) \]

Differentiating wrt. \( h \)

\[ h_{opt} = \left[ \frac{R(K)}{\mu_2(K)^2 R(f^2) n} \right]^{1/5} \]

Assuming pdf can be represented as a normal distribution

\[ h_{opt} = \left[ \frac{4 \sigma^{5}}{3n} \right]^{1/5} \]

Choosing an Acceptance Window

- Empirically (current method)
- Data-driven
  - Gaussian Mixture Models
  - Distribution of the directions
  - Break point of acceptance
Experimental Results

JVC HD Video Images

Storage Tanks

Intersection

Chemical Plant
## Experimental Results

### Inliers & Timing Results

<table>
<thead>
<tr>
<th></th>
<th>Tanks</th>
<th>Intersection</th>
<th>Chemical Plant</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total # of Features</strong></td>
<td>910</td>
<td>868</td>
<td>949</td>
</tr>
<tr>
<td><strong>True Matches</strong></td>
<td>463 (51%)</td>
<td>524 (60%)</td>
<td>696 (73%)</td>
</tr>
<tr>
<td><strong>Oriented Tracks</strong></td>
<td>416</td>
<td>458</td>
<td>503</td>
</tr>
<tr>
<td><strong>Mutual Best Match</strong></td>
<td>15 (4%)</td>
<td>132 (27%)</td>
<td>23 (5%)</td>
</tr>
<tr>
<td><strong>Filtered Matches</strong></td>
<td>356</td>
<td>482</td>
<td>503</td>
</tr>
<tr>
<td><strong>False Matches Accepted (Misses)</strong></td>
<td>16 (5%)</td>
<td>100.73</td>
<td>19 (1% )</td>
</tr>
<tr>
<td><strong>RANSAC Iterations</strong></td>
<td>0.16</td>
<td>0.77</td>
<td>0.45</td>
</tr>
<tr>
<td><strong>Total (s)</strong></td>
<td>0.33</td>
<td>0.46</td>
<td>0.26</td>
</tr>
</tbody>
</table>

1. Median number of RANSAC iterations taken from 1000 trials.
2. Median time for RANSAC iterations taken from 1000 trials.
Conclusions

• We have presented a method for reducing the feature mismatches used for estimating a robot’s motion from a video sequence
  – Based on the observation that true feature matches exhibit a common behavior in their frame-to-frame trajectories
  – Our algorithm shows improved performance over the standard method (MBM)
    • Higher percentage of inliers in the resultant data
    • Faster computation time
  – Can provide true/false match ratios of up to 95% from data whose initial ratios were as low as 40%
Future Efforts

• Currently, the **dominant motion** of the system is restricted to being **orthogonal** to the camera axis
  – Utilizing the **divergence** and **curl** of the feature tracks could provide less restrictive motion estimation

• During certain turning scenarios, there is **more than one** dominant motion present
  – We can utilize **Gaussian Mixture Models** to identify when multiple dominant motions are present, and infer the motion correctly
A comparison of pose estimation techniques:
Hardware vs. video

Brad Grinstead, David Page,
Andreas Koschan, Mongi A. Abidi
March 29, 2005
Motivation

• In order to perform its duties effectively, a robot must know its position and orientation within its working environment
  – Known as self-localization or pose estimation

• Two main methods are used to do this
  – Direct (measured) Localization – typically used in outdoor environments
  – Indirect (inferred) Localization – typically used in indoor environments

• Occasionally we need to use a combination of these two methods to meet our application needs
  – Wouldn’t it be nice to have a measure of relative accuracy when we switch back and forth between pose estimation modalities?
Example Scenario

Imagine a robot that is performing surveillance/inspection of a parking facility at an airport.

- Facility consists of both indoor and outdoor parking spots
- Robot must travel back and forth between parking areas
Evaluation Method

- Use our Mobile Scanning System to qualitatively and quantitatively analyze the different self-localization methods.
Direct Self-localization

Global Positioning System

Inertial Measurement Unit
Indirect Self-localization

Video Sequence

- Feature Matching
  - Detection
  - Robust tracking

Relate the images

Recover the motion

Pose From Motion

R, T

Image Sequence → Detect Features → Track Features → Estimate Motion → Introduce Scale → 6DOP Motion

motion feedback
Experimental Testbed
UTK’s Mobile Scanning System
Approach

Data Acquisition
- Shopping Strip
- JVC GR-HD1 HDTV camera
- Leica GPS500 Differential RTK 2cm positional accuracy
- XSens MT9 0.5° accuracy IMU

Data Fusion

Yields

3D Range Image

Textured 3D Model
Data Types

Range Data
  - Riegl LMS-Z210 laser range scanner
    • 350 m range, 80° vertical field of view, 5 cm accuracy

High-resolution Video
  - JVC GR-HD1 high definition camcorder

Position
  - Leica System 500 Global Positioning System (GPS)
    • Differential Real Time Kinematic (RTK) system
    • 10 measurements/second, 2 cm accuracy

Orientation
  - XSens MT9 Inertial Measurement Unit (IMU)
    • 100 measurements/second, 0.5° accuracy
Current Results – Strip Mall

Gaps in the GPS path are deliberate, simulating loss of satellite lock, demonstrating the need to a hybrid self-localization system.

Hardware vs. Video Path Comparison

Qualitative results of the combination of the localization modules with the laser range data.
Orientation Angles

Orientations shown are with respect to the vehicle’s chassis axes.

Rotations about the vehicle’s x- and y-axis are similar between the 2 methods, but the inherent inaccuracies in the video processing cause the yaw angles to vary significantly.
Remarks

• We have presented preliminary work on evaluating direct vs. indirect methods of robot self-localization
  – We have implemented both types of pose estimation
  – We have shown preliminary results for the qualitative evaluation of the 2 methods
    • Instrumented approaches give accurate information under specific circumstances
    • Pose from Video gives continuous information, with lower accuracy than the best instrumented data can provide
Future Efforts

• **Quantitative** analysis of the 2 methods
  – Gold-standard approach, where we have a **defined vehicle path**, and the estimated vehicle paths from the 2 methods are compared to the standard

• Testing of a **wider variety** of environments and conditions
Knoxville Proving Grounds
Data Acquisition, Integration, and Visualization

Brad Grinstead
Sreenivas Sukumar, Muharrem Mercimek, Anis Drira, and
Nikhil Naik
Rangan, Muharrem, Anis, Nihkil, and I spent January 20 using the Mobile Scanning Cart to acquire data from the Cellular Fields soccer complex walking paths.

The cart has full instrumentation, with 2 laser range scanners, GPS, INS, and video. 2 different computers are set up to acquire the data from the devices, and a heavy-duty battery and inverter are used to power the system.

We spent 4 hours collecting over 3 km worth of video, GPS, INS, and laser range data from 2 different scanners.

- Over 500 MB of raw data (not video) scanned
- Average Velocity: 1.08 m/s
- Average Riegl Interprofile Distance: 5 cm
- Average Sick Interprofile Distance: 3 cm
Data Collection Pictures

At the test area...

Setting up the GPS Base station

Scanning mechanism

Scanning Crew

Scanning terrain data with the setup
GPS Data Collection

Blue Line is the GPS Path for the loops that we collected.
Loop Animations
Still Pictures of Path
Segment of Processed Data
Mobile Scanning System – Selected Models
Data Acquisition, Integration, and Visualization
2003-2005

Brad Grinstead
Results – Strip Mall

Complete model: 823,000 triangles

Acquired 240 m worth of data in 25 seconds
Current Results – Strip Mall

Hardware vs. Video Path Comparison

Gaps in the GPS path are deliberate, simulating loss of satellite lock, demonstrating the need to a hybrid self-localization system.

Qualitative results of the combination of the localization modules with the laser range data.
Orientation Angles

Orientations shown are with respect to the vehicle’s chassis axes.

Rotations about the vehicle’s x- and y-axis are similar between the 2 methods, but the inherent inaccuracies in the video processing cause the yaw angles to vary significantly.
Results – Women’s Basketball Hall of Fame

Image of building

Subset of the textured 3D model

Complete model: 2.8 million triangles

Acquired 600 m worth of data in 52 seconds
BILO Supermarket – test application

Example Application
Low resolution 3D model of the BILO supermarket embedded into a simulation of parking lot inspection.
Data Fusion - Mall

Range Image

Position

Orientation

3D point cloud

Subset of 3D model
Results - Terrain Modeling

Parking Lot Test Run
~400 m
900k points
1.8 million triangles

Range image section

Section of 3D model

3D section of point cloud
Results – Terrain Modeling 2

Aerial View of target scene, with GPS path overlaid

Parking Lot Test Run 2
~1200 m
2 million points
4.3 million triangles

3D section of point cloud

3D surface
Model Building for Simulation and Testing Under Uncertain Conditions

Brad Grinstead
Andreas Koschan, David Page, and Mongi A. Abidi

*Imaging, Robotics, and Intelligent Systems Laboratory*
The University of Tennessee

April 19, 2006
Motivation

- Digitized models often have Regions of Interest (ROIs) that contain important small-scale details
  - Geometric features at or near the scanning resolution
- Software-based geometry refinement can preserve/enhance these details
  - Recovery of small-scale details
  - Ability to set up the acquisition to acquire at a lower resolution
Where Can It Be Applied?

• Large-scale mobile scanning applications
  – Virtual tours
  – Driving simulations

• Small-scale mobile scanning
  – Terrain/Tire impact
  – Individual scans for inspection
  – Model-verification through experimentation

• Any application where local frequency information and refinement of geometry would be useful!
Example Application
Terrain Modeling & Vehicle-Soil Impact

Aerial View

GPS Path

Full Data ~7 km

Zoom ~1 km

Zoom ~10 m

Tight Zoom ~0.5 m
Showing Tire Tread (synthetic)
Example Application
Pavement Inspection

Satellite image of Knoxville Airport

Pavement Distress

Need for an automatic system

Raw Point Cloud

3D Model of a Small Pavement Crack
System Block Diagram

3D range sensors
- RIEGL
- SICK
- IVP

Position and orientation sensors
- Leica - GPS
- Xsens IMU

Visual
- Sony

Thermal
- Indigo

Multi-sensor Alignment

Inter-profile Alignment

Multi-modality Data Integration

Stationary 3D Scan

3D Scan Under Motion
Where Does the Uncertainty Come From?

\[ P_t = \begin{bmatrix} x'_g, y'_g, z'_g \end{bmatrix}^T \]

GPS curve sampled at 10 Hz.

IMU data @ 100 Hz \((\omega, \phi, \kappa)\)

Video recorded at 30 frames/sec

\[ D_t = \begin{bmatrix} x'_r, y'_r, z'_r \end{bmatrix}^T \]

Range Profiles @
- 35 Hz 4m wide SICK
- 21 Hz 300 m range Riegl
- 2000 Hz and 50cms wide IVP

\[ R, D, + P = W \]
Geometry Refinement

• We need to “add” data in the ROIs
  – Locally increasing the resolution

• Kriging has been shown to outperform all other approximation methods, under certain circumstances
  – Ideal for non-uniform, sparsely sampled data
  – Built-in criteria for goodness of fit
  – No assumptions made about regular sampling, number of data points, etc.
  – No assumptions about the type of data being approximated

• Kriging Process
  – Estimate the underlying spatial correlations >>
  Construct an ideal model >> Estimate new surface values

\[
\begin{pmatrix}
\gamma(d_{11}) & \cdots & \gamma(d_{1n}) & 1 \\
\vdots & \ddots & \vdots & \vdots \\
\gamma(d_{m1}) & \cdots & \gamma(d_{mn}) & 1 \\
1 & 1 & 1 & 0
\end{pmatrix}
\begin{pmatrix}
W_1 \\
\vdots \\
W_n \\
\lambda
\end{pmatrix}
= \begin{pmatrix}
\gamma(d_{1p}) \\
\vdots \\
\gamma(d_{np}) \\
\end{pmatrix}
\]
Kriging

Punctual kriging uses points to estimate the values of other points.

\[ Z_e = \sum W_i \cdot Z_i \]  \hspace{1cm} (1)

Ideally, kriging attempts to choose the optimal weights that produce the minimum estimation error

\[ \varepsilon_p = Z_e(p) - Z_a(p) \]  \hspace{1cm} (2)

where the scatter of the estimates about the actual value has an estimation variance of

\[ \sigma^2 = \frac{\sum_{i=1}^{n} [Z_e(p_i) - Z_a(p_i)]^2}{n} \]  \hspace{1cm} (3)

Therefore, we minimize \( \sigma^2 - \lambda (\sum W_i - 1) \) wrt, \( W_1, W_2, ..., W_n \) and \( \lambda \).

Solving (4) for \( W \) yields the weights necessary to perform kriging for a single estimate.

\[
\begin{bmatrix}
\gamma(d_{11}) & \cdots & \gamma(d_{1n}) & 1 & W_1 \\
\vdots & \ddots & \vdots & \vdots & \vdots \\
\gamma(d_{m1}) & \cdots & \gamma(d_{mn}) & 1 & W_n \\
1 & 1 & 1 & 0 & \lambda
\end{bmatrix}
= \begin{bmatrix}
\gamma(d_{1p}) \\
\vdots \\
\gamma(d_{np})
\end{bmatrix}
\]  \hspace{1cm} (4)
Automating the Kriging Process

- Regularization to account for measurement errors and prevent oversmoothing
  - Tikhonov regularization, with $\lambda$ chosen via L-curve analysis
- Estimation of Variogram and its Model
  - Least squares fit of a number of models
  - Model with lowest residual is chosen
- Automatic selections of Regions of Interest
  - ROIs are different in appearance and functionality by definition
  - Region growing by curvature analysis
Segmentation via Curvature

- ROIs are **notably different** from surrounding geometry
- We can see this as a change in curvature and **segment** based on curvatures
  - **Seed** triangles >> **Region growing and merging** based on similar curvatures >> **Patch culling** to remove regions with a small # of triangles

For a quantitative comparison of geometry refinement techniques, we needed a ground truth model for comparison. The synthetic model was then used as the input to our Mobile Scanning System Simulator. The “range scans” output from our MSS Simulator were then used as the input data for geometry refinement. The refined models were then compared back to the reference model for performance evaluation.
Creating the Model

Synthetic Tire Tread model created by embossing a B&W tire tread pattern on a scaled DEM using RapidForm’s Engraving tool.

Tire Tread Model

- 500K triangles
- High resolution about the tire tread
- No planar surfaces
- Realistic impression surface for experimentation

2cm per profile scan
Geometry Refinement Comparison

<table>
<thead>
<tr>
<th>Method</th>
<th>Min (mm)</th>
<th>Max (mm)</th>
<th>Mean (mm)</th>
<th>Median (mm)</th>
<th>Variance</th>
<th>RMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>9.59 e-5</td>
<td>25.95</td>
<td>2.52</td>
<td>1.52</td>
<td>1.01 e-2</td>
<td>4.04</td>
</tr>
<tr>
<td>Cubic</td>
<td>9.61 e-5</td>
<td>25.80</td>
<td>2.58</td>
<td>1.63</td>
<td>8.41 e-3</td>
<td>3.84</td>
</tr>
<tr>
<td>IDW2</td>
<td>13.42 e-3</td>
<td>44.83</td>
<td>9.84</td>
<td>8.19</td>
<td>5.76 e-2</td>
<td>12.43</td>
</tr>
<tr>
<td>Modified Kriging</td>
<td>6.19 e-4</td>
<td>25.18</td>
<td>2.50</td>
<td>1.17</td>
<td>7.59 e-3</td>
<td>3.78</td>
</tr>
</tbody>
</table>
Effect of Training Data

Since the data that is used to create the variogram model is used to drive the estimation process, using data outside the ROIs for training is counterproductive!

Six training regions were selected, threes each from the ROI and non-ROI portions of the data.

These regions were used to develop the Kriging model and for geometry refinement.
Experimental Results

Scanning Resolution

Subsample a 3D model acquired from an accurate high-resolution laser range scanner.

Perform geometry refinement using our modified kriging and other methods.

Compare the refined models to each other and to the original.
Geometry Refinement of Low Resolution Scan

A 400x300 mm model of a pavement crack scanned at a resolution of 1 mm is subsampled at intervals of 3 mm. Geometry refinement is then performed on the subsampled model.
What Did We Learn?

- **Implementation** and **experiments** show that this can be a very useful tool
- Have **manually** tweaked the parameters to demonstrate that it is possible to get good results
- Demonstration has been done on both **synthetic** and **real** data
- Experiments with **regularization** show improvements in data estimation
- **Automation** of kriging setup has been accomplished
- Automatic **segmentation** of the ROIs and subsequent performance analysis of using the ROI data for training has been done
Where Do We Go From Here?

- Need to define a **metric** that shows “good” details
- Where do you stop the refinement? And what **level of details** can you preserve from data of a given **quality**?
- What effects do various **noise** sources have on the reconstruction?
- Can we still recover the small-scale details in the presence of **irregular noise**?
- Which **applications** can we target for best use of this method? Tire/soil modelling, pavement inspection, cultural heritage modelling, others?
- Surface **comparison** between refined model and stationary scan. Can we get close to the “best model”?
Detail Enhancing Denoising of Digitized 3D Models from a Mobile Scanning System

Kriging Progress

Brad Grinstead
January 27, 2006
Motivation

Any method for digitizing 3D models of real-world objects produces a **noisy model**. Standard smoothing methods fail to preserve the interesting details of the object. I propose to work on an **information-preserving** denoising method that is capable of smoothing the areas of the model with little information, while preserving/enhancing the regions where details are present.
Project Pipeline

- Video Sequence
  - Feature Detection and Matching
    - Pose Estimation from Video
  - GPS
    - 3D Position and Orientation
- INS
- Range Profiles
- Combined Pose Estimation
- Range Profile Alignment
  - Data Fusion
    - Kriging for Detail Enhancement
    - Surface Decomposition
    - Detail Preserving Denoising

Contributions
Where Can It Be Applied?

• Large-scale mobile scanning applications
  – Virtual tours
  – Driving simulations

• Small-scale mobile scanning
  – Terrain/Tire impact
  – Individual scans for inspection
  – Model-verification through experimentation

• Any application where local frequency information and refinement of geometry would be useful!
Example Application
Vehicle-Soil Impact

Synthetic Tire Tread model created by **embossing** a B&W tire tread pattern on a scaled DEM using RapidForm’s **Engraving** tool.

**Tire Tread Model**
- 500K triangles
- **High resolution** about the tire tread
- No planar surfaces
- **Realistic impression** surface for experimentation

2cm per profile scan
Example Application
Vehicle-Soil Impact

Tire/Soil Impressions

- Data acquired from Genex scanner, with Sophie's help
- 3 Models: 1 small tire, and 2 large tire
- Soil was loose and moist for good impressions
- VRML models for all experiments available
How Do We Do It?

• We need to “add” data in the regions of detail
  – Locally increasing the resolution
• Kriging has been shown to outperform all other approximation methods, under certain circumstances
  – Ideal for non-uniform, sparsely sampled data
  – Built-in criteria for goodness of fit
  – No assumptions made about regular sampling, number of data points, etc.
  – No assumptions about the type of data being approximated
Kriging for Surface Approximation

1. Estimate the **spatial correlation** between samples
2. Construct an **ideal model** that best fits the calculated correlation
3. **Estimate** new surface values using Kriging

\[
\begin{align*}
\gamma(d_{11}) & \quad \cdots \quad \gamma(d_{1n}) & \quad 1 & \quad W_1 \\
\vdots & \quad \ddots & \quad \vdots & \quad \vdots \\
\gamma(d_{m1}) & \quad \cdots \quad \gamma(d_{mn}) & \quad 1 & \quad W_n \\
1 & \quad 1 & \quad \cdots & \quad 0
\end{align*}
\]

\[
\begin{pmatrix}
\gamma(d_{ij}) \\
\vdots \\
\gamma(d_{ij}) \\
1
\end{pmatrix}
\]
Calculating the Spatial Correlation

- **Semi-variance (SV)** is a measure of the degree of spatial dependence between samples
- A **variogram** is a measure of how quickly things change on average
  - Directionally dependant
  - Measure of variance vs. distance
    - At $d=0$ the SV should be 0, and at larger distances the SV should increase

![Diagram of variograms](image)
Constructing the Variogram

\[
\gamma(d) = \frac{1}{2N(d)} \sum_{i=1}^{N(d)} (z_i - z_{i+d})^2
\]

(1)

\(d\) is a set distance window

\(N(d)\) is the number of points that distance apart

\(z_i\) is the data value of a local variable, taken at location \(i\)

Once the variogram data points are calculated, a parametric model is fit to the data. This model is then used to perform the kriging.
Kriging

Punctual kriging uses points to estimate the values of other points.

\[ Z_e = \sum W_i \cdot Z_i \quad (2) \]

Ideally, kriging attempts to choose the optimal weights that produce the minimum estimation error

\[ \varepsilon_p = Z_e(p) - Z_a(p) \quad (3) \]

where the scatter of the estimates about the actual value has an estimation variance of

\[ \sigma^2 = \frac{\sum_{i=1}^{n} [Z_e(p_i) - Z_a(p_i)]^2}{n} \quad (4) \]

Therefore, we minimize \( \sigma^2 - \lambda (\sum W_i - 1) \) wrt, \( W_1, W_2, \ldots, W_n \) and \( \lambda \)

\[
\begin{bmatrix}
\gamma(d_{11}) & \cdots & \gamma(d_{1n}) \\
\vdots & \ddots & \vdots \\
\gamma(d_{n1}) & \cdots & \gamma(d_{nn})
\end{bmatrix}
\begin{bmatrix}
W_1 \\
\vdots \\
W_n
\end{bmatrix}
= 
\begin{bmatrix}
\gamma(d_{1p}) \\
\vdots \\
\gamma(d_{np})
\end{bmatrix} \quad (5)
\]

Solving (5) for \( W \) yields the weights necessary to perform kriging for a single estimate.
Initial Results

Original Synthetic Model
1m x 1m

MS³ range scan

2cm per profile scan
3517 points

Linear Interpolation
160,801 points

Cubic Interpolation
160,801 points

Kriging w/ Langragian multiplier
160,801 points
Closeups

Original Synthetic Model

Linear Interpolation  Cubic Interpolation  Kriging w/ Lagrangian multiplier
Kriging Results
Interpolation Comparison

Noise-free case
Kriging Hurdles

- Some **hurdles** to overcome for “automated” geometry refinement are:
  - How to **automatically** setup the kriging process
  - How to **select** the regularization parameters?
  - What affect does the **training** region have on the estimation?
Estimating the Semivariogram

**Traditional Method**: Choose from a library of functions and use manual or least-squares fitting to fit the model to the calculated semivariances

**New Method**: Use a sum of kernels (similar to kernel density estimation) to build the function up from the calculated semivariances
Tikhonov Regularization

Traditional kriging uses a “nugget” effect to account for the micro-scale variation in measurements.

This is *inappropriate* when we know that the measurement error is not likely to be the same for different regions.

We can use *regularization* to account for local measurement errors and achieve better estimation results.
Regularization Results

- $\lambda$ found using L-curve analysis
- Some improvements in data fit.
- Perhaps another manner of choosing $\lambda$?

Original Synthetic Model
1m x 1m

2cm per profile scan
Noise-free case

Original Kriging
w/ Langragian multiplier

Automated Kriging
w/ Tikhonov solution
Affect of Training Regions

6 training subsets were chosen, 3 each from smooth and detailed patches.

Note how the Semivariograms within each group appear to be similar.
Training Region Comparison

Reference Model
40,558 points

Whole Trained
Trained on 40,558 points

Smooth1 Trained
Trained on 1,205 points

Smooth3 Trained
Trained on 3,364 points

Detail1 Trained
Trained on 2,153 points

Detail3 Trained
Trained on 4,217 points
Numerical Comparisons
Wrap-up

• **Kriging – What has been done**
  – Implementation and experiments show that this can be a very useful tool
  – Have manually tweaked the parameters to demonstrate that it is possible to get good results
  – Demonstration has been done on both synthetic and real data
  – Experiments with regularization show improvements in data estimation
  – Automation of kriging setup has been accomplished

• **Where do we go from here?**
  – Need to define a metric that shows “good” details
  – Where do you stop the refinement? And what level of details can you preserve from data of a given quality?
  – How to go about automatic selection of regions of interest?
  – Conference publications have been selected. Papers need to be written, and venue for Journal publication needs to be determined
Thank You

\[
\begin{bmatrix}
\gamma(d_1) & \ldots & \gamma(d_m) & 1 \\
\vdots & \ddots & \vdots & \vdots \\
\gamma(d_{m}) & \ldots & \gamma(d_{m}) & 1 \\
1 & 1 & 0 & \hat{\lambda}
\end{bmatrix}
\begin{bmatrix}
W_1 \\
\vdots \\
W_m \\
\gamma(d_{mp})
\end{bmatrix}
= \begin{bmatrix}
\gamma(d_{1p}) \\
\vdots \\
\gamma(d_{mp}) \\
1
\end{bmatrix}
\]
Detail Enhancing Denoising of Digitized 3D Models from a Mobile Scanning System

Brad Grinstead
October 17, 2005
Outline

• **Introduction**
• Mobile Scanning System
• False Match Rejection
• Detail Enhancement
• 3D Model Decomposition & Analysis
• Summary & Proposed Tasks
Why Detail Preserving Denoising?

Any method for digitizing 3D models of real-world objects produces a noisy model. Standard smoothing methods fail to preserve the interesting details of the object. I propose to work on an information-preserving denoising method that is capable of smoothing the areas of the model with little information, while preserving/enhancing the regions where details are present.
Project Pipeline

Video Sequence → Feature Detection and Matching → Pose Estimation from Video

GPS → 3D Position and Orientation → Combined Pose Estimation

INS → Range Profiles → Range Profile Alignment → Data Fusion

Data Fusion → Kriging for Detail Enhancement → Surface Decomposition → Detail Preserving Denoising

Contributions
Contributions

- **False Match Rejection via Oriented Tracks**
  - New *false match rejection* algorithm based on behavior observation of true matches throughout a sequence
  - Algorithm identifies the *common behavior* of the true feature matches and *filters out* non-conformists

- **Application of Kriging for Detail Enhancement**
  - Algorithmic *geometry refinement* process that provides an estimate of the goodness of fit
  - *Automatic* selection of parameters to *preserve* the natural edges present in the data

- **An Extension of Empirical Mode Decomposition for 3D Surface Analysis**
  - *State-of-the-art* extension to 3D signal processing

- **Detail-Enhancing Denoising of 3D Surfaces**
  - *Applications* for 3D signal processing using our developed techniques
Outline

• Introduction
• **Mobile Scanning System**
• False Match Rejection
• Detail Enhancement
• 3D Model Decomposition & Analysis
• Summary & Proposed Tasks
Concept

Profile Sequence → Textured 3D Model

May 15, 2006 Slide 7
Overall Approach

Data Acquisition

Shopping Strip

3D Range Image

Data Fusion

Yields

Textured 3D Model

Texture
JVC GR-HD1 HDTV camera

Position
Leica GPS500 Differential RTK 2cm positional accuracy

Orientation
Xsens MT9 0.5° accuracy IMU

Positioning
Leica GPS500 Differential RTK 2cm positional accuracy
Data Types

Range Data
- **Riegl LMS-Z210** laser range scanner
  - 350 m range, 80° vertical field of view, 5 cm accuracy

High-resolution Video
- **JVC GR-HD1** high definition camcorder

Position
- **Leica System 500** Global Positioning System (GPS)
  - Differential Real Time Kinematic (RTK) system
  - 10 measurements/second, 2 cm accuracy

Orientation
- **Xsens MT9** Inertial Measurement Unit (IMU)
  - 100 measurements/second, 0.5° accuracy
Morrel Center

Complete model: 823,000 triangles

Acquired 240 m worth of data in 25 seconds
Downtown West Mall

Position

Orientation

Color Image

Range Image

Point Cloud of Entire Model

Complete model: 5.3 million triangles

Acquired 1.4 km worth of data in **18 minutes**

Subset of Textured 3D Model
Sources of Error (Noise)

**Radial:** Scanner Noise

**Mobile:** Scanner Noise + Localization errors
Outline

• Introduction
• Mobile Scanning System
• False Match Rejection
• Detail Enhancement
• 3D Model Decomposition & Analysis
• Summary & Proposed Tasks
Pose from Video Problems

- Grayscale images don’t always provide good features
- Current feature tracking methods still have problems with noisy features
- Moving foreground/background causes problems with current feature matching algorithms
- No matter the feature matching algorithm, the output will contain false matches
Outlier Rejection - Background

- Robust geometry estimation algorithms
  - RANSAC
    - Zhang ‘95, Pollefeys ‘97, Hartley ‘00, Salvi ‘01

- Filtering techniques
  - Image Residual (Shi ‘94)
  - Illumination Invariant Residual (Tommasini ‘98)
  - Triplet Consistency (Chua ‘00)
  - Mutual Best Match (Pollefeys ‘97, Hartley ‘00, Corke ‘04)
Outlier Rejection - RANSAC

- Extract features
- Compute potential matches
- While (%inliers<\(\Gamma\))
  - Select a minimal sample of 7 matches
  - Compute \(F\)
  - Determine inliers
- Refine \(F\) based on all inliers

Time complexity is based on the %outliers present in the data

\[ N = \text{ceil}(\log(1 - \Gamma) / \log(1 - (1 - \varepsilon)^P)) \]

where \(\varepsilon\) is the percentage of outliers present in the data
Outlier Rejection – Mutual Best Match (MBM)
Observations
Rejection of Outliers by Oriented Tracks

1. Start with an initial set of feature correspondences
2. Estimate the probability distribution of the feature track directions
   a) Use kernel density estimation to calculate the mode
   b) Automatically calculate kernel width using Plug-In method
3. Determine the orientation vector that is “on average” parallel to the feature tracks from the estimated density function as the mode of the distribution $m$
4. For each feature calculate the angular difference $d_i$ from $m$
5. Reject features whose angular difference is greater than some threshold $\alpha$
Mode Calculation

- Histograms are notoriously sensitive to initialization
  - Kernel density estimation is a better way to go

\[
\hat{pdf}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)
\]
Determining Kernel Width

What effect does the kernel width have on our approximation?

- **Too small (h):**
  - Over repeated sampling, the spikes appear to shift.

- **Too large (h):**
  - Reduced variability at the cost of introducing bias.

- **Just right (h):**
  - Minimal bias, with reduced variability.
Optimizing Bandwidth

Minimize Asymptotic Mean Integrated Square Error

$$AMISE(f) = \frac{R(K)}{nh} + \frac{1}{4} h^4 \mu_2(K)^2 R(f^\prime)$$

Differentiating wrt. $h$

$$h_{opt} = \left[ \frac{R(K)}{\mu_2(K)^2 R(f^\prime) n} \right]^{1/5}$$

Assuming $pdf$ can be represented as a normal distribution

$$h_{opt} = \left[ \frac{4 \sigma}{3n} \right]^{1/5}$$


Choosing an Acceptance Window

- Empirically (current method)
- Data-driven
  - Gaussian Mixture Models
  - Distribution of the directions
  - Break point of acceptance
Experimental Results
JVC HD Video Images
Experimental Results

Inliers & Timing Results

<table>
<thead>
<tr>
<th></th>
<th>Tanks</th>
<th>Intersection</th>
<th>Courthouse</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total # of Features</strong></td>
<td>910</td>
<td>868</td>
<td>949</td>
</tr>
<tr>
<td><strong>True Matches</strong></td>
<td>463 (51%)</td>
<td>524 (60%)</td>
<td>432 (46%)</td>
</tr>
<tr>
<td><strong>Oriented Tracks</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Mutual Best Match</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Oriented Tracks</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Mutual Best Match</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Filtered Matches</strong></td>
<td>416</td>
<td>458</td>
<td>503</td>
</tr>
<tr>
<td><strong>False Matches</strong></td>
<td>15 (4%)</td>
<td>132 (27%)</td>
<td>23 (5%)</td>
</tr>
<tr>
<td><strong>Accept Mutes</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>RANSAC Iterations</strong></td>
<td>6</td>
<td>30</td>
<td>6</td>
</tr>
<tr>
<td><strong>O3</strong></td>
<td>0.27</td>
<td>133.69</td>
<td>0.22</td>
</tr>
<tr>
<td><strong>RANSAC</strong></td>
<td>0.16</td>
<td>0.77</td>
<td>0.19</td>
</tr>
<tr>
<td><strong>Total (s)</strong></td>
<td>0.33</td>
<td>134.46</td>
<td>0.41</td>
</tr>
</tbody>
</table>
Outline

- Introduction
- Mobile Scanning System
- False Match Rejection

**Detail Enhancement**

- 3D Model Decomposition & Analysis

- Summary & Proposed Tasks
Why Kriging?

- We need to “add” data in the regions of detail
  - Locally increasing the resolution
- Kriging has been shown to outperform all other approximation methods, under certain circumstances
  - Ideal for non-uniform, sparsely sampled data
  - Built-in criteria for goodness of fit
  - No assumptions made about regular sampling, number of data points, etc.
  - No assumptions about the type of data being approximated
Kriging for Surface Approximation

1. Estimate the **spatial correlation** between samples
2. Construct an **ideal model** that best fits the calculated correlation
3. **Estimate** new surface values using Kriging
Calculating the Semi-variance

- **Semi-variance (SV)** is a measure of the degree of spatial dependence between samples.

- A **variogram** is a measure of how quickly things change on average.
  - Directionally dependant
  - Measure of variance vs. distance

- At d=0 the SV should be 0, and at larger distances the SV should increase.

![Variogram Diagram](image)
Constructing the Variogram

\[ \gamma(d) = \frac{1}{2N(d)} \sum_{i=1}^{N(d)} (z_i - z_{i+h})^2 \]  

(1)

- \( d \) is a set distance window
- \( N(d) \) is the number of points that distance apart
- \( z_i \) is the data value of a local variable, taken at location \( i \)

Once the variogram data points are calculated, a parametric model is fit to the data. This model is then used to perform the kriging.
Kriging

Punctual kriging uses points to estimate the values of other points.

\[ Z_e = \sum W_i \cdot Z_i \]  \hspace{1cm} (2)

Ideally, kriging attempts to choose the optimal weights that produce the minimum estimation error

\[ \varepsilon_p = Z_e(p) - Z_a(p) \]  \hspace{1cm} (3)

where the scatter of the estimates about the actual value has an estimation variance of

\[ \sigma^2 = \frac{\sum_{i=1}^{n} [Z_e(p_i) - Z_a(p_i)]^2}{n} \]  \hspace{1cm} (4)

Therefore, we minimize \( \sigma^2 - \lambda (\sum W_i - 1) \) wrt, \( W_1, W_2, \ldots, W_n \) and \( \lambda \)

\[
\begin{bmatrix}
\gamma(d_{11}) & \cdots & \gamma(d_{1n}) & 1 & W_1 \\
\vdots & \ddots & \vdots & \vdots & \vdots \\
\gamma(d_{n1}) & \cdots & \gamma(d_{nn}) & 1 & W_n \\
1 & 1 & 1 & 0 & \lambda
\end{bmatrix}
= \begin{bmatrix}
\gamma(d_{1p}) \\
\vdots \\
\gamma(d_{np})
\end{bmatrix}
\]  \hspace{1cm} (5)

Solving (5) for \( W \) yields the weights necessary to perform kriging for a single estimate.
Kriging Evaluation

Original Data (surface)

Original Data (points)

General exponential-Bessel model used
Kriging Results

Real Data – Original Implementation

Original

Kriging

Enhanced specularity to demonstrate details
In order to improve the performance of the kriging refinement, we manually tweaked the variogram model and regularization parameters:

- Forced the variogram model to fit the estimated semi-variances closer to \( d=0 \) than \( d=1 \) (normalized)
- Manually set the regularization parameters (experimentally) to preserve the edges
Kriging Results

Synthetic Data - Interpolation Comparison

Simulation results generated using our Mobile Scanning System Simulator (MS^3), which allows us to synthetically recreate conditions observed by our Mobile Scanning System.
Kriging Results

Synthetic Data - Interpolation Comparison

Noise-free case
Kriging Results
Synthetic Data - Interpolation Comparison

Used the MS³ to recreate the conditions for several scans of the Jared Sign model.

**Parameters:**
- Linear path
- Uniform velocity of 5 cm per profile
- Intraprofile sample spacing of 1.5 cm

**Modes:**
- Noise-free
- 5 cm noise
- 5 cm noise, smoothed

![Graph showing synthetic model comparisons](graph.png)
Outline

- Introduction
- Mobile Scanning System
- False Match Rejection
- Detail Enhancement
- **3D Model Decomposition & Analysis**
- Summary & Proposed Tasks
Empirical Mode Decomposition (EMD)

• Introduced in 1998 by Huang, et al. to analyze non-linear, non-stationary signals

• Data analysis technique that represents a 1D signal as a sum of zero-mean AM-FM signals

• Has been shown to out-perform all competing decomposition algorithms for certain signal types
EMD Assumptions

1. Non-linear, non-stationary signals (e.g., data accumulated from real experiments) contain 1 or more instantaneous frequency components

2. The characteristic time scales of the data are defined by the oscillating modes

3. These oscillating modes can be separated for analysis, and the final results can be obtained through integration of the components
A Simple Visual Example

Components

Signal
The EMD Algorithm

- While (#maxima + #minima > 3) // not converged
  1. Find extrema envelopes
  2. Calculate the signal mean
     - $m_i = (\text{Max}_i - \text{Min}_i)/2$
  3. Subtract the signal mean from the data
     - $h_i = X - m_i$
       - If ($h_i == \text{IMF}$)
         - $c_k = h_i$ // store IMF
         - $X = X - c_k$ // subtract IMF from signal
         - Repeat steps 1-3 on new approximation $X$
       - Else
         - Repeat steps 1-3 on the residual $h_i$
Extending EMD to 3D Surfaces

- Extension to 3D is non-trivial!
- Vector-valued details
- Definition of extrema on a 3D surface
- Generation of extrema envelopes
- Difference between 2 surfaces
Defining Maxima & Minima on a Surface

What do **extrema** on a surface look like?

In 1D:

In 3D:

It appears that we can identify extrema by the **surface curvature** at the vertex. We can use standard methods of computing curvature – such as Taubin, 1995 – to get the principle curvatures for the 3D surface at vertex $N_i$. 

![Minimum Curvatures](image1)

![Extrema](image2)
Volumetric Regularization

Unknown surface function $f(x)$ is found by minimizing the energy function

$$H[f] = \beta[f] + \frac{1}{\lambda} \sum_{i=1}^{n} (y_i - f(x_i))^2$$  \hspace{1cm} (6)

This cost functional is minimized by a sum of weighted radial basis functions

$$f(x) = P(x) + \sum_{i=1}^{n} \omega_i \phi(|x - c_i|)$$  \hspace{1cm} (7)

The weights and coefficients of the polynomial $P$ are solved by constructing a linear system from the specified constraints

$$\begin{bmatrix}
\phi(r_{i1}) + \lambda_1 & \ldots & \phi(r_{in}) & 1 & c_1 & \omega_1 & f(c_1) \\
\vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots \\
\phi(r_{m1}) & \ldots & \phi(r_{mn}) + \lambda_m & 1 & c_n & \omega_m & f(c_m) \\
1 & 1 & 1 & 0 & 0 & P_0 & 0 \\
c_1 & \ldots & cn & 0 & 0 & P & 0 \\
\end{bmatrix}$$

$$r_y = |c_i - c_j|$$  \hspace{1cm} (8)
Radial Basis Functions

The radial basic functions (RBFs) are radially symmetric functions centered at the constraint points. Dinh used RBFs with multiple orders of smoothness. The associated partial differential equation is

\[-\delta \Delta f + \Delta^2 f - \tau \Delta^3 f = 0 \]  

(9)

\(\delta\) controls the amount of 1st order smoothness, while \(\tau\) controls the amount of 3rd order smoothness.

The RBF that minimizes (9) exhibits only local influence (quickly falling towards 0 as \(r\) increases), leading to a sparse system matrix that can be solved quickly using sparse matrix solvers such as the conjugate gradients squared method.

\[
\phi(r) = \frac{1}{4\pi \delta^2 r} \left[ 1 + \frac{v e^{-\sqrt{\delta} r}}{v - w} - \frac{w e^{-\sqrt{\delta} r}}{v - w} \right] \\
v = 1 + \sqrt{1 - 4\tau^2 \delta^2} \frac{2\tau^2}{2\tau^2} \\
w = 1 - \sqrt{1 - 4\tau^2 \delta^2} \frac{2\tau^2}{2\tau^2} 
\]

(10)
Envelope Generation Example

Waterneck Model

- 1324 Surface Constraints
- 132 Exterior Constraints
- $\delta = 10, \tau = 0.01$

Original Model

Minima Envelope

Maxima Envelope

Marching Cubes voxel size = 0.03 mm
(0.006% of object length)

Minima/Maxima Envelope Deviation
**3D EMD Algorithm**

- X = 3D surface
- R = X
- done = false
- while (!done) //not converged
  - Find maxima and minima of R, based on curvatures
  - Use volumetric regularization to generate extrema envelopes
  - For each vertex Vi
    - Calculate the distance along the vertex normal Ni to maxima and minima envelopes (dMaxi, dMini)
    - Find the mean distance mi=(dMaxi-dMini)/2
    - “Subtract” mi from vertex Vi
  - End for
  - If (∆R < threshold) //not much change in surface
    - Store IMF
    - If (minimal surface)
      - done = true
    - Else
      - R = X-R
      - Repeat while
    - Else
      - Repeat while on modified R
  - End if
- End while
Visual Example

Waterneck

A1 (approximation)

A2

A3

C1 (details)

C2

C3
Visual Example
Waterneck - continued
Outline

- Introduction
- Mobile Scanning System
- False Match Rejection
- Detail Enhancement
- 3D Model Decomposition & Analysis
- **Summary & Proposed Tasks**
What Has Been Done

• **False Match Rejection via Oriented Tracks**
  - Used to **augment** our self-localization from video system
  - **Experiments** show that we can cut the feature matching and geometry estimation time in half.
  - Work has been published

• **Kriging**
  - **Implementation** and **experiments** show that this can be a very useful tool
  - Have **manually** tweaked the parameters to demonstrate that it is possible to get good results
  - Demonstration was done on both **synthetic** and **real** data

• **3D Surface Analysis**
  - Basic theory and implementation of **3D EMD extension** done
  - Initial experiments are **promising**
Where Do We Go From Here?

- **Kriging**
  - Regularization to account for measurement errors
    - Would like to incorporate this into the Kriging formulation
    - How to **choose** parameters?
      - Autocorrelation, Total Variation, Tikhonov, Other?
  - Estimation of Variogram and its Model
    - Currently done arbitrarily
    - Would like to automate using something similar to **Kernel Density Estimation**
  - Automatic selections of Regions of Interest
    - Can it be done using our **3D Analysis** algorithm?
    - Could something simpler like curvature analysis work?
Where Do We Go From Here?

- **3D Empirical Mode Decomposition**
  - Is the current method the appropriate direction to take?
    - Many ways to get to a given destination
    - Why treat it *topologically* instead on *functionally*?
    - Perhaps a *space transform* would give better results
  - Is the current method of selecting extrema appropriate
    - Ohtake, et al, 2004 presented a *ridge/valley* extraction algorithm
  - Experimentation, experimentation, experimentation!
    - Current results only shown for 1 dataset, and that one not even from our Mobile Scanning System!
Where Can It Be Applied?

- Large-scale mobile scanning applications
  - Virtual tours
  - Driving simulations

- Small-scale mobile scanning
  - Terrain/Tire impact
  - Individual scans for inspection
  - Model-verification through experimentation

- **Any application where local frequency information would be useful!**
Publication Plan

• **Winter 2005**
  - Publish work that is currently being done on Automated Kriging for Preserving Natural Edges

• **Spring 2005**
  - Publish Oriented Tracks work in a conference, now that it has been accepted for Journal publication
  - Target Journal/Conference for the publication of the 3D EMD extension

• **Summer 2006**
  - Develop applications for our 3D analysis and refinement algorithms (Detail Enhancing Denoising) and submit to appropriate Journal/Conference
## Schedule

<table>
<thead>
<tr>
<th>Research Objective</th>
<th>Start Date</th>
<th>Completion Date</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile scanning survey</td>
<td>1/02</td>
<td>4/02</td>
<td>Done</td>
</tr>
<tr>
<td>Pose estimation survey</td>
<td>9/02</td>
<td>2/04</td>
<td>Done</td>
</tr>
<tr>
<td>Smoothing survey</td>
<td>12/04</td>
<td>2/05</td>
<td>Done</td>
</tr>
<tr>
<td><strong>Outlier Rejection</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Design</td>
<td>10/03</td>
<td>11/03</td>
<td>Done</td>
</tr>
<tr>
<td>Implementation</td>
<td>11/03</td>
<td>11/03</td>
<td>Done</td>
</tr>
<tr>
<td>Evaluation</td>
<td>11/03</td>
<td>1/04</td>
<td>Done</td>
</tr>
<tr>
<td>Publication</td>
<td>1/04</td>
<td>4/04</td>
<td>Done</td>
</tr>
<tr>
<td><strong>EMD extension</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Background Survey</td>
<td>6/04</td>
<td>6/04</td>
<td>Done</td>
</tr>
<tr>
<td>Design</td>
<td>6/04</td>
<td>11/05</td>
<td>In progress</td>
</tr>
<tr>
<td>Implementation</td>
<td>6/04</td>
<td>12/05</td>
<td>In progress</td>
</tr>
<tr>
<td>Evaluation</td>
<td>12/05</td>
<td>3/06</td>
<td>Future</td>
</tr>
<tr>
<td>Publication</td>
<td>3/06</td>
<td>4/06</td>
<td></td>
</tr>
<tr>
<td><strong>Detail enhancement and geometry refinement</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Background Survey</td>
<td>12/04</td>
<td>2/05</td>
<td>Done</td>
</tr>
<tr>
<td>Design</td>
<td>2/05</td>
<td>6/05</td>
<td>Done</td>
</tr>
<tr>
<td>Implementation</td>
<td>3/05</td>
<td>8/05</td>
<td>Done</td>
</tr>
<tr>
<td>Evaluation</td>
<td>8/05</td>
<td>11/05</td>
<td>In Progress</td>
</tr>
<tr>
<td>Publication</td>
<td>11/05</td>
<td>1/06</td>
<td>Future</td>
</tr>
</tbody>
</table>
List of Publications

Accepted for Publication


Submitted and Pending Review

Thank You
Abstract

3D imaging is a popular method for acquiring accurate models for a variety of applications. However, the size of the geometric features that can be modeled in this manner is dependent on the scanning system’s resolution. This paper presents a method that attempts to accurately reconstruct regions whose features are at or below the system’s scanning resolution, combining automatic region selection with a form of kriging. A curvature-based segmentation is followed by an automated geometry refinement procedure in which the model of spatial correlation between the irregularly sampled 3D data is automatically determined. Geometry refinement is done by a regularized kriging approach that is designed to preserve the sharp features typical to many 3D laser range applications. This method is validated on synthetic data, showing that the accuracy of our method is higher than that of its standard competitors. Then, the performance on real data is demonstrated through several examples.

1. Introduction

3D imaging is becoming a widely used method for acquiring 3D surface data for a variety of applications. Laser scanning has been used to generate Digital Elevation Maps (DEMs) [3, 10, 13] from airborne systems, urban 3D models [4, 6, 9] from ground-based systems, 3D models of small objects and indoor mapping [7, 14, 15], and more. Often, however, the modeled objects contain regions of interest (ROIs) that have small-scale details – defined as areas whose geometric feature dimensions are much smaller than the overall size of the object – that are not digitized at a high enough resolution to be accurately represented for further analysis.

In order to enhance the details of a region in an acquired dataset, additional sampling needs to be applied to the region of interest. Re-acquiring the data at a higher resolution may not be an option for many applications, due to the time and costs involved in performing the acquisition again. Therefore software-based geometry refinement is the only viable option of enhancing the existing 3D data. Software-based geometry refinement has a number of benefits, including: the ability to recover details that are near, or below, the low end of the scanning system’s resolution from an existing dataset; the ability to set up the acquisition to acquire at a lower resolution, based on the assumption that the refinement method will recover the details of interest from a lower resolution scan, yielding faster acquisition and smaller raw data files.

As an example of this type of data, Figure 1 shows a laser range dataset that was acquired at 5cm/profile resolution. This resolution is not fine enough to accurately represent the ideal (synthetic) 3D geometry seen in the bottom image of Figure 1.
been overlooked in the scanning process. Algorithmic geometry refinement could remove the need to reacquire that data. Geometry enhancement can also ease the task of model verification through laser scanning, crack detection from a mobile platform, or even modeling and verifying tire/soil interaction for automotive simulation and design. In fact, many 3D modeling applications that have regions of important details that are near, or below, the low end of the scanning system’s resolution and are much smaller than the overall dataset’s dimensions can benefit from this method.

In the majority of these applications, critical information is contained at the edges of the 3D structure. Preservation of these edges, or sharp features, is therefore the priority when implementing any geometry refinement method. Classical approaches for resampling range data are linear and cubic interpolation [1], and Inverse Distance Weighting (IDW). Statistical methods such as kriging have also been applied to 3D datasets [12, 18]. Despite the fact that 3D data tends to be sampled at very small separation distances, the uncertainty of resampling using these methods tends towards oversmoothing or deformation of the sharp features present in the data.

We propose a modified version of kriging that uses the spatial relationships present within the data to provide estimates of the underlying surface during the geometry refinement process, with a regularization component tailored towards preserving the edge information also present in the data. The geometry refinement process is fully automatic, from the selection of the ROIs to the final estimation of the underlying surface. We demonstrate the efficacy of this method on a synthetic dataset, as well as on data acquired from an existing mobile laser scanning system, such as the one used in [6]. The results show that our modified kriging method outperforms its common competitors in terms of accuracy and preservation of features.

The rest of the paper is presented as follows: Section 2 describes the fundamentals of geometry refinement, and the theory behind kriging in particular; Section 3 describes the modifications that we have made to automate the kriging process and tailor the output towards preserving edge information; Section 4 describes the experimental process used to validate this method, its comparison to existing interpolation methods, and its application to actual 3D laser scanning data. Section 5 concludes the paper with a summary of the work presented, a discussion of the results of our efforts, and a brief look at what the future holds.

2. Geometry refinement methods

Spatial interpolation has been defined as the procedure of estimating the value of a field variable at unsampled locations within the area covered by sample locations [17]. In the case of resampling 3D data from a laser scanning system, the process is one of geometry refinement based on the non-uniformly sampled surface points. The underlying assumption of any interpolation algorithm is that samples close to each other will likely be more similar than points that are further apart.

Linear and cubic interpolation on a non-uniformly sampled field variable is typically done by performing a Delaunay triangulation to optimally connect the sampled point into a network of triangles. Interpolation is then done by searching the network for the closest triangle and computing the linear or cubic interpolant from that triangle. This method of interpolation is typically quite fast, but is very dependant on the uniformity of the sampling of the data.

Inverse distance weighting is a weighted average interpolator, where the influence of a measured point on the new estimate decreases as the distance from to the location of the estimated point increases. The weighting power \( p \) controls how the influence tapers off with respect to distance \( d \):

\[
Z_e = \frac{\sum w_i Z_i}{\sum w_i},
\]

\[
w_i = d_i^{-p},
\]

where \( w_i \) is the weight associated with the measured value \( Z_i \), and \( Z_e \) is the estimation of the new value. Frequently the power used for performing the interpolation is attached to the IDW abbreviation, thus inverse distance weighting using a power of 2 is designated as IDW2.

In its native state, IDW uses all measured data points in the weighting process, but more frequently only the points within a user-specified radius are used. One of the characteristic effects of IDW is that it tends to generate a “bulls-eye” effect around measured points [16]. Smoothing parameters are sometimes introduced to alleviate this effect.

Kriging is a spatial interpolation method that gives estimates of new surface values based on a statistical model of the spatial correlation between measured values [2]. Kriging estimates the values at specified locations, using observed data to drive the process, optimized with respect to specific error criteria. This error criteria is the squared prediction
error at the unobserved locations. The measured data provide the ‘support’ for the estimated values, and the quality, size, shape, and orientation of the observed values influence the ability to accurately predict the unknown sample values. Because kriging is capable of modeling the data both as a trend and as a set of residuals, it is often considered to provide the most accurate surface value predictions of all other interpolation types under the right circumstances.

The estimation of the spatial correlation of surface data in kriging theory is typically done with the use of variograms. The variogram is a function which characterizes the dependence of data points measured at different locations. The variogram is calculated as:

\[ \gamma(d) = \frac{1}{2N(d)} \sum_{i=1}^{N(d)} (z_i - z_{i+h})^2, \]

where \( \gamma(d) \) is the semivariance at lag \( d \), \( z_i \) is the measurement of a regionalized variable taken at location \( i \), \( z_{i+h} \) is another measurement taken \( h \) intervals away, \( N(d) \) is the number of separating distances: number of points - number of lag intervals.

Once the experimental variogram has been calculated, an ideal parametric model is fit to the data. The ideal parametric model is used to simplify the estimation process and to increase the robustness to measurement errors.

Kriging is the actual process of using the parametric variogram model to estimate the surface value at the specified location. The most common form of kriging used in engineering applications is ordinary kriging (OK).

In ordinary kriging, the estimate of an unknown surface value uses a weighted estimate of other nearby points:

\[ Z_e(p) = \sum w_i Z(p_i). \]

Ideally, kriging attempts to minimize the error between the estimated point and the actual value. The variance of this error is the amount of scattering of the estimates \( Z_e \) about their true values \( Z_o \):

\[ \sigma^2 = \frac{1}{n} \sum [Z_e(p_i) - Z_o(p_i)]. \]

The estimation and its error are dependent on the weights chosen in (3). Optimal weights, therefore, would be those that produce the minimum estimation variance. These are found by solving a system of equations consisting of the weighted semivariances between measured points, and the estimated semivariances between the unknown point and the known values

\[
\begin{bmatrix}
\gamma(d_{11}) & \cdots & \gamma(d_{1n}) & \vdots \\
\vdots & \ddots & \vdots & \vdots \\
\gamma(d_{n1}) & \cdots & \gamma(d_{nn}) & \gamma(d_{n0})
\end{bmatrix}
\begin{bmatrix}
w_1 \\
\vdots \\
w_n \\
w_0
\end{bmatrix}
= 
\begin{bmatrix}
\gamma(d_{10}) \\
\vdots \\
\gamma(d_{n0})
\end{bmatrix},
\]

(5)

where \( \gamma(d_{ij}) \) is the semivariance between the \( i^{th} \) and \( j^{th} \) measured values and \( \gamma(d_{ii}) \) is the estimated semivariance between the \( i^{th} \) observation and the location of the evaluation point.

3. Our improvements

There are a number of issues with standard kriging that cause it to be less than ideal for geometry refinement in the presence of small-scale details. Firstly, the global optimization of the kriging process tends towards smooth surfaces that obscure, rather than enhance, the small-scale details present in the data. Secondly, kriging is often done with a number of steps involving manual parameter selection and initialization. The size of the training and estimation regions also poses a problem. Using the entire dataset for training is not beneficial when the refinement is going to be performed only on small ROIs. Finally, the manner in which kriging itself is performed lends to a robust solution, but an oversmoothed one. The rest of this paper discusses our solutions to these issues, followed by experiments that verify our solutions.

The regions selected for enhancement in this scenario are often much smaller than the dimensions of the entire collected dataset. They also typically stand out from the rest of the data by being different—in terms of shape, curvature, spatial frequency—than the surrounding areas. These regions can typically be picked out by a human observer rather easily. In order to automate this process, we use curvature information, calculated from the acquired data, to identify areas of uncommon curvature that are then considered our ROIs. The method that we have chosen to use is similar to that shown in [11].

The ROIs are generated by randomly selecting seed triangles from the surface, performing region growing around those seeds, and assimilating neighboring patches with similar curvatures. Finally, connected regions with fewer than a pre-specified number of triangles are rejected. What remains is a set of surface patches with a relatively constant curvature. The selection is then inverted to provide those regions that do not have a continuous curvature, and are thus our regions of interest. Figure 2 shows a visual demonstration of the procedure. The upper left image shows the seed triangles highlighted in blue. Shown
in the upper right is the segmentation result after several iterations of region growing and merging. The lower image shows the final segmentation results after culling, with the “smooth” regions highlighted in blue.

The ROIs are not only used to define the areas that need refinement, they are also used as the training data for the kriging process. By restricting the training data to lie only in the ROIs, we get an improvement in surface fit due to the removal of the influence of the data that is “not of interest”.

Typically, variograms are generated by manually selecting a model from a library of functions, and then using a least-squares method or manual parameter selection to fit the model to the estimated semivariogram. In order to automate this process, we perform a least-squares fitting of a number of ideal models, and the one with the best match to the data is used as the ideal variogram model for the kriging process. The least squares fitting is constrained to closely fit the model to those lags close to the source, since we are interested in the best local fit of the surface data. This process allows us to automatically select the appropriate model for the task, as well as providing a closer estimation of the underlying model near the estimation location.

In experimental systems, it is not possible to sample data at one location over and over without incurring some variation in the samples. This variation is due to system noise, environmental shifts, and other factors. Traditional kriging treats this micro-scale variation in a global fashion, adding a “nugget” effect to the variogram at \( d=0 \). However, this is inappropriate when it is known that the measurement variation in every region is likely to be different. Instead, we employ a regularization procedure that attempts to model the micro-scale variations independently. This regularized approach accounts for local measurement errors and helps to reduce the smoothing effect that the nugget parameter introduced.

Given that Equation (5) can be written as \( Ax=b \), we can solve the system using Ordinary Least Squares

\[
x = (A^TA)^{-1}A^Tb.
\]

This is equivalent to zero-order Tikhonov regularization. Since \((X'X)^{-1}\) is ill-conditioned, we introduce a regularization parameter \( \lambda \) to increase stability of the system

\[
x = (A^TA + \lambda I)^{-1}A^Tb.
\]

Here \( \lambda \) can be chosen to be either a scalar or a vector. It can be automatically determined through validation methods such as autocorrelation, or empirically chosen through experimentation. We suggest choosing a \( \lambda \) that reflects the amount of uncertainty present in the data acquisition system. Our \( \lambda \) was chosen using the L-curve analysis technique of [8].

4. Experimental results

In order to show that the augmentations made to the original kriging algorithm actually yield a better geometry refinement, a number of experiments were performed on synthetic datasets. Figure 3 shows a 2m x 2m synthetic model developed for tire/soil interaction. The model is an impression of a repeated tire tread pattern embedded into a varying terrain surface. This model contains regions of high detail, as well as smooth regions, and is useful in demonstrating the effects of a laser scanning system on such data, as well as providing a ground truth model for comparison of geometry refinement results.

A mobile laser scanning simulator was developed to synthetically recreate the process of using a laser range scanner to digitize an arbitrary surface. This simulator allows the user to specify parameters that control a virtual laser scanning system. These parameters include the scanning resolution along a laser profile, the resolution between profiles, the orientation of the “scanner” at each sampling stage, and the noise parameters of the system. The simulator then reads in a triangulated 3D model, performs the “scanning” using the specified parameters, and the output of this system was then used as the input data for 4 different interpolators: linear, cubic, IDW, and our modified kriging.
The input to the test algorithms was a synthetic range scan (Figure 3) with a vertical resolution of 4 mm and a horizontal resolution of 2 cm. This non-uniform sampling is consistent with many laser scanning systems, and is the cause of the majority of the undersampling problems of interest in this paper. Each of the 4 geometry refinement algorithms was then used to reconstruct the surface at a resolution of 5 mm in both the horizontal and vertical directions.

Figure 4 shows the results of the interpolation methods applied to the synthetic range scan generated from our simulator. For each of the interpolated models, we calculated the distance of each point to the reference model and color coded the surface according to its distance to the reference model. Surface points close to the ground truth are shown in blue moving through the spectrum to green and finally to red as the deviation grows large. It can be seen that our modified kriging algorithm has fewer areas of red, and more of blue and green, than the other methods, indicating that the interpolated surface follows closer to our ground truth model.

Figure 5 shows the distributions of the signed distance from the interpolated surfaces to the reference model. The tighter the distribution is centered at 0, the better it represents the underlying surface. From this graph it can be seen that our modified kriging holds the closest to the original surface, with the linear and cubic interpolation right behind. The poor results from IDW2 come from the aforementioned “bulls-eye” effects due to the irregularly sampled nature of our data. More global surface deviation results are given in Table 1. The fields in Table 1 list the minimum, maximum, mean, and median deviation (error) of the refined surface from the ground truth model, the variance of the errors, and the Root Mean Square error for each of the refinement methods tested. Note that our kriging algorithm outperforms the other methods across the board.

Figure 3. Geometry refinement validation using a synthetic ground truth model.

Figure 4. Results of geometry refinement methods applied to test dataset. Variations from the reference model are color coded. The lower images for each column are zoomed in photos of the upper right hand corners of the reconstructions.
Having shown that our modified method has the ability to outperform its competitors, we turn to the discussion of training regions and their effect on the refinement of our areas of interest. Figure 6 shows six regions selected from the ideal model to be used in this test, three each from the detailed and non-detailed regions. Semivariograms were trained on each of these regions independently, and the model was reconstructed at a resolution of 5 mm.

Figure 7 shows a graph relating the median distance to the reference model by surface patch. Each of the reconstructions is compared to the reference model as a whole, as well as to each of the individual training regions, in order to evaluate global vs. local performance. It can be seen from the graph that the reconstructions from the training regions in detailed areas have the best performance within the detailed regions, while not giving up any performance overall. This supports our claim that restricting the training set to only those regions that contain the small-scale details we are interested in enhancing is beneficial.

Figure 8 gives a visual representation of the location and magnitude of the errors in the surface reconstructed from region Detail3. The larger the dot, the larger the error in the reconstructed surface. Surface elevation contours have been overlain on the image for context. From this figure, it can be seen that the highest magnitude of errors occurs, unsurprisingly, in those areas that have the largest changes in curvature. Also, the largest errors occur in a vertical striping pattern that lies along the scan lines from the input data.

Having shown the validity of our algorithm on a synthetic dataset, the method was used to enhance data taken from real laser scanning systems. These datasets were acquired from mobile scanning platforms undergoing general motion. The sampling resolution is determined by the equipment specifications as well as the motion the platform is undergoing.

Table 1. Quantitative comparison of interpolation results

<table>
<thead>
<tr>
<th>Method</th>
<th>Min (mm)</th>
<th>Max (mm)</th>
<th>Mean (mm)</th>
<th>Median (mm)</th>
<th>Variance</th>
<th>RMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>9.59 e-5</td>
<td>25.95</td>
<td>2.52</td>
<td>1.32</td>
<td>1.01 e-2</td>
<td>4.04</td>
</tr>
<tr>
<td>Cubic</td>
<td>9.61 e-5</td>
<td>25.80</td>
<td>2.58</td>
<td>1.63</td>
<td>8.41 e-3</td>
<td>3.84</td>
</tr>
<tr>
<td>IDW2</td>
<td>13.42 e-3</td>
<td>44.83</td>
<td>9.84</td>
<td>8.19</td>
<td>5.76 e-2</td>
<td>12.43</td>
</tr>
<tr>
<td>Modified Kriging</td>
<td>6.19 e-4</td>
<td>25.18</td>
<td>2.50</td>
<td>1.17</td>
<td>7.59 e-3</td>
<td>3.78</td>
</tr>
</tbody>
</table>

Having shown that our modified method has the ability to outperform its competitors, we turn to the discussion of training regions and their effect on the refinement of our areas of interest. Figure 6 shows six regions selected from the ideal model to be used in this test, three each from the detailed and non-detailed regions. Semivariograms were trained on each of these regions independently, and the model was reconstructed at a resolution of 5 mm.

Figure 7 shows a graph relating the median distance to the reference model by surface patch. Each of the reconstructions is compared to the reference model as a whole, as well as to each of the individual training regions, in order to evaluate global vs. local performance. It can be seen from the graph that the reconstructions from the training regions in detailed areas have the best performance within the detailed regions, while not giving up any performance overall. This supports our claim that restricting the training set to only those regions that contain the small-scale details we are interested in enhancing is beneficial.

Figure 8 gives a visual representation of the location and magnitude of the errors in the surface reconstructed from region Detail3. The larger the dot, the larger the error in the reconstructed surface. Surface elevation contours have been overlain on the image for context. From this figure, it can be seen that the highest magnitude of errors occurs, unsurprisingly, in those areas that have the largest changes in curvature. Also, the largest errors occur in a vertical striping pattern that lies along the scan lines from the input data.

Having shown the validity of our algorithm on a synthetic dataset, the method was used to enhance data taken from real laser scanning systems. These datasets were acquired from mobile scanning platforms undergoing general motion. The sampling resolution is determined by the equipment specifications as well as the motion the platform is undergoing.

Figure 9 shows an image of a commercial building, as well as the raw data taken from a scanning platform that moved past the building in a semi-parallel fashion. Notice that the letters on the sign are undersampled. This is due to the scanning vehicle’s movement - approximately 5 cm per scanned profile. The bottom image in Figure 9 shows the refined surface output from our method. The structure on the
lettering is much clearer than in the original data, showing the enhancement.

The dataset in Figure 10 is from the same mobile scanning platform. Notice in the visual image how there is a regular grid structure present on the building that was undersampled by the scanning system. The vehicle’s motion corresponded to a sampling rate of approximately 10 cm per scanned profile. However, the few samples that are present are enough to partially reconstruct the grid, as seen in the lower image of Figure 10. This example shows that some minimum of sampling is required to be able to recover small-scale geometry from actual scanned data.

5. Discussion

Work has been presented on an automatic method for identifying and refining the surface geometry of small-scale details present in 3D laser scanning data. The method was validated on synthetic datasets, showing that the quality of the refined geometry was greater than that of both the original data as well as that from other interpolation methods. The method was then applied to real-world data, showing the type of enhancements that can be performed using this algorithm.

It should be noted that kriging, in general, is a computationally intensive method. Thus, the execution times for our geometry enhancement algorithm are larger than those from the competing methods we tested against. Thus, while we showed an improvement in the approximation of the underlying surface, it came at the cost of long execution times. The decision on whether the extra computation is worthwhile for the added benefit of improved surface accuracy will have to be weighed carefully, and will be application specific.

Future efforts for this research will likely be devoted towards continued automation and improvements in the accuracy. Among other topics, investigations involving nonparametric variogram estimation such as that shown in [5] are currently underway. Another topic in need of investigation is the minimal sampling required to reconstruct details of a given size. As we showed in Figure 10, there is indeed a certain sampling resolution required for each scale of resolution. An in-depth look at what these requirements are would be beneficial for future tasks in geometry refinement. In addition to the algorithmic investigations, data from applications such as cultural...
heritage preservation and tire/soil interaction are being examined for benefit from this method.

Acknowledgements

<acknowledgements removed for anonymity>

References

Mobile Scanning System for the Fast Digitization of Existing Roadways and Structures

Brad Grinstead¹, Sreenivas Sukumar¹, David Page¹, Andreas Koschan¹, David Gorsich², and Mongi A. Abidi¹

¹ Imaging, Robotics, and Intelligent Systems Laboratory
Department of Electrical and Computer Engineering
The University of Tennessee
Knoxville, TN, USA 37996
< bgrinste, ssrangan, dpage, akoschan, abidi@utk.edu >

² U. S. Army RDECOM Tank-Automotive Research
Development and Engineering Center
Warren, MI, USA 48397
< GorsichD@tacom.army.mil >
Mobile Scanning System for the Fast Digitization of Existing Roadways and Structures

Abstract

Purpose – To present a Mobile Scanning System for digitizing three-dimensional (3D) models of real-world terrain.

Design/methodology/approach – A combination of sensors (video, laser range, positioning, orientation) is placed on a mobile platform, which moves past the scene to be digitized. Data fusion from the sensors is performed to construct an accurate 3D model of the target environment.

Findings – The developed system can acquire accurate models of real-world environments in real time, at resolutions suitable for a variety of tasks.

Originality/value – Treating the individual subsystems of the Mobile Scanning system independently yields a robust system that can be easily reconfigured on the fly for a variety of scanning scenarios.

Keywords Laser Range Scanning, Mobile Scanning, Data Fusion, Pose Estimation, Model Building

Paper type Research paper

INTRODUCTION

Accurate 3D models of existing roadways and structures can provide immense benefit to a variety of applications involving simulation and sensor planning. Quality assessment of roadways can be done more accurately through 3D models of the actual surfaces. With an accurate 3D model of an intersection, sensor placement for traffic monitoring applications can be done in an efficient manner, minimizing the number of sensors by maximizing each sensor’s field-of-view within the target location. Other applications can also benefit greatly from the availability of accurate 3D models of real-world environments: driving simulators can use models of existing intersections to evaluate traffic flow and safety; tire-soil interactions can be measured and the models used to improve transportation vehicles of all types; robots can use these models as a priori information for path planning; etc. In the past, these models were created by graphics
artists from a library of pre-constructed primitives, or from a rough measurement skeleton. Today however, the technology has advanced to the point where we can directly acquire 3D models from real world environments. In effect, we can now “digitize reality”.

As an example, take the problem of pavement inspection. Pavement inspection – investigating roadways for imperfections such as potholes, cracks, etc. – can be an important, yet time consuming task. Manual pavement inspection typically involves someone either walking over the areas to be inspected, or riding on a slow-moving vehicle, taking measurements and/or photographs of the deterioration of the road surface for later evaluation. Some agencies have taken the technological approach, by using triangulation-based laser scanners mounted on vehicle bumpers to create 3D models of the road surface on the fly (see Figure 1). The 3D models from these mobile scanning systems are automatically generated, and can be evaluated using mathematical tools, thus removing the majority of the manual effort involved in road surface inspection.

![Figure 1. INO Laser Rut Measurement System (INO, 2005). Dual laser profilers acquire the structure of the pavement while the vehicle is in motion.](image)

In essence, a mobile scanning system consists of 4 main components: hardware for 3D geometry acquisition; hardware for positioning and orientation (pose) measurement; a mobile platform which moves the sensing package past the environment to be digitized; and software to perform the data fusion necessary to combine the data from all the sensing modalities and to process the resulting model to fit the application at hand. While other researchers (see Section 2) have developed 3D terrain acquisition systems, these tend to be fixed in regards to the hardware and the fusion methods used. In contrast, our Mobile Scanning System treats the 4 sub-systems independently, giving us a system that is designed for:
• **Modularity** – each component in the system can be replaced by another that performs a similar function.

• **Configurability** – each component of the system has a configuration file and the processing software is designed to pick up system specifics from that file, rather than having to be recoded for each piece.

• **Static Processing** – the functionality of the data fusion and post-processing for the various data is independent of the sensors and vehicle used, enhancing the configurability of the system.

• **Robustness** – the modularity inherent in our design allows the system to be as robust to real world environments as the individual components.

A general system pipeline for our Mobile Scanning System can be seen in Figure 2, with an example setup of the sensor suite showing the positioning and orientation devices, 3D geometry scanner, and video for texture overlays and pose estimation. What makes our system different from previous approaches is the way we treat the hardware components as interchangeable and keep the post processing of the data fixed, which makes our system more robust to environmental conditions and allows us to digitize a wider variety of environments.

![Figure 2. General flowchart for the Mobile Scanning System.](image)

The rest of the paper is laid out as follows: Section 2 discusses previous efforts in the digitization of 3D environments, including a relative comparison of techniques; Section 3 lays out the design of our Mobile Scanning System, and goes into brief detail about the processes used for data fusion and processing;
experimental results from our system in digitizing real world environments can be seen in Section 4; and Section 5 wraps up with an overview and a discussion of future enhancements to the system.

PREVIOUS METHODS

In the last several years there has been several groups studying how to best acquire these 3D models of real world environments. These research efforts generally fall into one of two categories: image-based methods, where the 3D geometry is inferred from 2D images; and the laser range-based approach, where the 3D geometry is measured directly through laser range scanners.

Table I summarizes a few of the current systems that are used to develop large-scale 3D terrain models. Image-based methods infer the 3D geometry from a video stream using stereo imaging principle, and have the advantage of being real-time in acquisition and yielding nicely textured models, at the cost of lower sampling resolution on the surface structures. Laser-based methods measure the target geometry directly through physical principles, and have the advantage of more accurate and dense surface sampling, at the cost of higher processing requirements in the form of data fusion.

<table>
<thead>
<tr>
<th>System</th>
<th>Modality</th>
<th>Capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Airborne Video Systems</strong></td>
<td>Image-based</td>
<td>meter-level accuracy; can acquire large swaths of terrain data in a short time, but generates coarse models; data acquired in real time and processed offline</td>
</tr>
<tr>
<td>(Baillard, 1999; Frere, 1998)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Shape from Motion</strong></td>
<td>Image-based</td>
<td>centimeter- to meter-level accuracy; general motion with a few degenerate cases; 3D structure inferred from stereo principles; demonstrated on small scale; acquires in real-time and processes offline</td>
</tr>
<tr>
<td>(Pollefeys, 2000; Zisserman, 1999)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>MIT City Scanning Project</strong></td>
<td>Image-based</td>
<td>meter-level accuracy amongst models; structure inferred from spherical nodules; time consuming to setup and acquire data</td>
</tr>
<tr>
<td>(Antone, 2000)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Radial Laser Scanning</strong></td>
<td>Laser range-based</td>
<td>cm-level accuracy; involves acquiring a large number of ground-based range scans, registering them into a common frame, and merging them together; time consuming in both acquisition and processing</td>
</tr>
<tr>
<td>(El-Hakim, 1997; Sequeira, 1999)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Mobile Laser Scanning</strong></td>
<td>Laser range-based</td>
<td>cm-level accuracy; use orthogonal range scanners for pose estimation, limiting them to semi-planar urban environments; orthogonal scan-line matching makes data fusion fast and accurate; data acquired in real time and processed offline</td>
</tr>
<tr>
<td>(Früh, 2001; Zhao, 2001)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Our Mobile Scanning System</strong></td>
<td>Laser range-based</td>
<td>mm- to cm-level accuracy; uses instrumentation for pose estimation, allowing the system to be used in a variety of environments; makes no assumptions about the environment to be digitized; data acquired in real time and processed offline</td>
</tr>
</tbody>
</table>

One of the main disadvantages of the existing mobile laser range systems is that they assume that the target environment will have regular (manmade) constructs and that the vehicle path will be suitably
smooth and semi-planar. Our Mobile Scanning System was developed to utilize the strengths of these systems, and overcome their limitations by utilizing a more complicated, instrumented system for pose estimation. This makes our system more robust to environmental conditions and allows us to digitize a wider variety of environments. In addition, the system is designed to be a modular package so that the hardware can be changed to meet the specific needs of the application at hand, while the data processing procedures remain the same.

**MOBILE SCANNING SYSTEM**

Digitizing large-scale environments is a task that has inherent constraints, depending on the target application, the environment, the structures to be digitized, the size and expense of the equipment to be used, the time available for data acquisition, the purpose for which the models are to be used, etc. Some of these constraints are constant, and some depend on the application – e.g., what resolution is necessary for the model. Perhaps one of the main constraints is the time on site necessary to do the data acquisition.

We use high-resolution sensors mounted on a mobile platform to acquire mm- to cm-level resolution models. This allows us to quickly digitize the target scene in a variety of environments, under a variety of conditions and varying lighting. As the vehicle moves past the scene to be digitized, the system acquires 3D geometry profiles of the terrain around it while the pose estimation system acquires information about the motion of the platform. Let [X Y Z] be the global coordinate reference for the digitized scene, let [x y z] be the laser scanner’s internal coordinate system, and let the orientation parameters for the scanner be \( [\text{roll} \ \text{pitch} \ \text{yaw}] \). Each point \( P \) in the current profile has a 3D identity of \( zyxvP \), as seen from the scanner, and the scanner has a 3D identity of \( \ell(X, Y, Z) \) in the global reference. Thus, the location of \( P \) in the world reference can be found as:

\[
V_P = \ell + R \cdot v_p,
\]

where \( R \) is the rotation matrix generated from the \( \text{roll} \), \( \text{pitch} \), and \( \text{yaw} \) values determined from the pose sensing instrumentation.
Self-localization

Self-localization, pose estimation, is a critical part of the Mobile Scanning System as the accuracy of the 3D model is directly related to the accuracy of the estimation of the position and orientation at the time of each acquisition. For the example configuration shown in this paper, we have utilized an Inertial Measurement Unit (IMU) to provide orientation and acceleration information for the platform. For positioning, we have integrated a highly accurate differential real-time kinematic (D-RTK) GPS receiver, which is accurate up to 2 cm. To augment the instrumented self-localization process, we have also implemented a Pose from Video (PfV) algorithm that utilizes the high-resolution images acquired by our sensing module.

GPS data gives us the ability to georeference the data acquired by the Mobile Scanning System, giving an absolute frame of reference, allowing us to incorporate all the localization information available from the other measurement systems into a common, fixed reference. The fusion of the localization data thus involves mainly the computation of the fixed rotations and translations if the various sensors with respect to the global reference, the GPS (Grewal, 2001).

Let us denote the position of the platform at a given time as \( \mathbf{p}(t) = [x \ y \ z] \), and its orientation as \( \mathbf{r}(t) = [\phi \ \psi \ \omega] \). Since the acquisition rates are different for the various sensors and we only have discrete time sampled data, interpolation is needed to obtain a continuous path trajectory. For a full 3D motion estimate \( \mathbf{p}_k = [x \ y \ z \ \phi \ \psi \ \omega] \) at time \( k \) in the interval \([t_i, t_{i+1}]\), the displacements and rotations are defined as:

\[
P_k = \mathbf{A} \eta_i^k + \mathbf{B} \eta_i^{k-1} + \mathbf{C} \eta_i^{k-2} + \mathbf{D} \eta_i^{k-3},
\]

where \( A, B, C, \) and \( D \) and the third order polynomial coefficients given by the tridiagonal system, as per Bartels et al (1998).

The orientation parameters for the system are measured by the IMU through a series of gyroscopes and linear accelerometers. The measurements can provide the velocity of the system, as well as the orientation. In a pinch, the IMU information can be used to estimate the position of the system as well, through double integration of the accelerations. Unfortunately, the errors that accrue at each sampling point eventually degrade the estimate beyond the acceptable bounds of accuracy, unless a fixed keystone measurement (such as accurate GPS) is used to anchor the system.
In our system, the GPS is the only device that provides these absolute points of measurement, which sets the bounds on the accuracy of the pose estimation subsystem. Under non-ideal conditions, the degradation of the GPS signal increases the uncertainty in the self-localization process. In this case, we can monitor the uncertainty of the hardware-based pose estimation system, and when it falls below a specified threshold, we can switch over to a video-based method.

The video-based method determines the vehicle’s motion from a sequence of video images by finding corresponding features between the images and using the scene’s epipolar geometry to compute pose between views (Branca, 2000; Chroust, 2004; Rivlin, 2003; Usher, 2003). First, distinctive features in an image pair are identified using a Harris corner detector (Harris, 1988). Next, features in the first image are matched to corresponding features in the second image, using intensity correlation followed by our in-house method for false match removal. Given a good set of correspondences, the motion state of the camera can be calculated, up to scale, using a two stage motion estimation algorithm similar to those described by Johnson (2000) and Hartley (2000). We can then use the onboard laser range finder of our system to get an absolute distance to a known point in the scene, providing the scale, and giving us the full 6 Degree of Freedom (DoF) motion estimate.

**Data Fusion and Model Processing**

For our target application range, the Mobile Scanning System acquires data in realtime, but processes the data offline. This allows us to meet our parallel goals of minimal time on site and maximal model quality. A typical terrain model can consist of over 200,000 geometry profiles, 1,000,000 position and orientation measurements, and 20,000 high-resolution color images. A rough estimate of the storage capacity required for the raw data alone exceeds 10 gigabytes. Thus, we need to combine the data together in a meaningful fashion, while reducing the amount of memory required to deliver the model’s information to the target application. This procedure consists of the following stages: (1) fusion of the raw data into a single model, and (2) geometry processing to remove the effects of noise, fill holes and remove duplicate data, and provide and adaptive simplification of the model.

Data fusion is carried out in two main stages: combining the localization information to provide position and orientation of the sensor package at each scanning step, and using that pose information to relate the individual laser range profiles and digital images. The localization and scanning instruments are synchronized by a common timing signal. Then, for range profiles we find the corresponding localization
information by combining interpolated values for the position and orientation. As mentioned previously, we use the localization information provided by the video system to provide position and orientation information when the instrumented systems’ quality falls below a given level.

Once the data fusion of the localization information has been done, the alignment of the individual laser range profiles and color images is a straightforward process. Pose estimates local to the individual range scans are interpolated to find the best estimate of the position and orientation of the scanner at the time of acquisition. A rigid transformation is then applied to the coordinate system of the scanner to bring the data into the world reference frame, as defined by the pose instrumentation.

As with any system that measures real data, our Mobile Scanning System has a noise component associated with it, derived from the inherent inaccuracies of the laser range scanners as well as the uncertainty associated with each pose estimate. The filter that we use to remove the noise is based on the assumption that the system noise is an additive Gaussian noise and the characteristics are based on those of the scanner, experimentally determined within our lab. In addition, while the vehicle is stopped or undergoing turns the scene geometry may be sampled more than once, yielding redundant data. These data are identified through motion analysis and removed.

Occasionally, the model created may be used by more than one application, with each task requiring a different level of resolution and accuracy. In order to have the generality of a system that can acquire high-resolution geometry information and provide low-resolution models as needed, we have developed a multi-resolution processing scheme (Roy, 2003) that defines the operations needed to display/store/process the generated 3D model in various levels-of-detail. Figure 3 shows this process on a building model at the original resolution, and at 25% and 2.5% of the original resolution. Notice that the geometry is increasingly simplified, and storage/computations reduced, but the overall appearance of the model remains the same.

**EXPERIMENTAL RESULTS**

We have used our Mobile Scanning System to acquire 3D models for a number of different environments, in a variety of configurations. For this paper we will discuss 2 different configurations of the Mobile Scanning System. The first will be the system we use to develop models of building and other above-ground-level structures. These models are suitable for populating driving simulators, aiding robotic path
planning and manipulation tasks, etc. The second configuration is a working example of how the Mobile Scanning System can acquire detailed models of the terrain over which it moves – i.e., 3D models of the ground itself. Figure 4 shows the instrumentation for both systems, with the downward-looking laser range scanner for the road scanning setup, as well as the long-range side-scanning laser for the building façade generation (inset).

Figure 3. A multi-resolution representation of a building model with the textured model shown above the wire frame and point cloud representations. The level of detail decreases from left to right.

Figure 4. Experimental setup for both the building façade scanning system and the roadway inspection and modeling system.
For the structure scanning configuration, we have used a mobile platform – in this case a van – with the sensor package mounted on the roof for maximum field of view and minimal pedestrian impact. The sensor package consists of: a Leica GS500 D-RTK GPS system, an Xsens MT9 IMU, a JVC GR-HD1 high definition, and a Riegl LMS-Z210 laser range scanner (see Table II for sensor details).

<table>
<thead>
<tr>
<th>Component Name</th>
<th>Data Type</th>
<th>Accuracy</th>
<th>Acquisition Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leica GS500 D-RTK GPS</td>
<td>Position</td>
<td>up to 2 cm</td>
<td>10 samples/sec</td>
</tr>
<tr>
<td>Xsens MT9 IMU</td>
<td>Orientation/Accelerations</td>
<td>&lt; 1°</td>
<td>100 samples/sec</td>
</tr>
<tr>
<td>JVC GR-HD1 Hi-Def Camcorder</td>
<td>Video Navigation/Texture</td>
<td></td>
<td>1280x720 @ 30 frames/sec</td>
</tr>
<tr>
<td>Riegl LMS-Z210</td>
<td>Laser Range</td>
<td>5 cm @ up to 350 m</td>
<td>10,000 points/sec (20 profiles/sec)</td>
</tr>
<tr>
<td>IVP Ranger</td>
<td>Laser Range</td>
<td>5 mm @ up to 0.5 m</td>
<td>50,000 points/sec</td>
</tr>
<tr>
<td>SICK LMS200</td>
<td>Laser Range</td>
<td>1 cm @ up to 8 m</td>
<td>12,600 points/sec (35 profiles/sec)</td>
</tr>
</tbody>
</table>

Figure 5 shows the digitization of a small, but complex scene. In this case, the Mobile Scanning System was driven around a shopping strip, acquiring the 3D geometry from all 4 sides of the building, to generate a more complete model. This model consists of 500,000 triangles and was acquired in 5 minutes.

Figure 6 is a large-scale structure model of the Downtown West Shopping Mall in Knoxville, TN. The Mobile Scanning System was driven around the entire mall, acquiring the geometry and color information to combine into a detailed 3D model. The dataset is rather large, consisting of over 5,000,000 data points, 5,000 color images, and 48,000 pose measurements. However, only 18 minutes of scanning time was required to acquire the data, which encompassed over 1.5 km of scanning. The textured 3D model shown in Fig. 6 is a small subset of the entire dataset, which is shown in reduced point cloud form.

In addition to the structure scanning system, we are also currently working with a scanning platform designed to acquire the terrain (ground) the system is moving over. This system uses a downwards looking laser range scanner to digitize the terrain. The process for data fusion and model processing is the same as the previous system due to the modularity that went into the system design process.

There are two versions of the downward looking system that we are currently using. One is a micro-scale system that is capable of measuring the geometry very densely (approximately 1 mm between data points) with an accuracy on the order of 5 mm, at the cost of having a very limited field-of-view and a high amount of raw data to process. The other system uses a laser range scanner designed to scan larger objects, similar to the Riegl scanner used in the previous discussion.
The micro-scale system utilizes an IVP Ranger to acquire high-resolution, high-accuracy 3D models of roads and other surfaces. This level of resolution is useful for tasks such as pavement inspection. The macro-scale terrain scanning system uses a SICK LMS200 system to acquire terrain models on a larger scale. This scanner provides terrain models appropriate for terrain-tire interaction, vehicle dynamics simulation, etc. See Table II for further specifications on these sensors.

Figure 7 shows an example of the Mobile Scanning System with the LMS200 laser range scanner providing the 3D terrain geometry. The 1.2 km path scanned is on a gravel road and parking lot, with a number of cars, ditches, and railroad tracks to provide obstructions and details on the ground. Figure 7a shows an aerial view of the region that was digitized, with the scanning platform’s path superimposed. Figure 7b shows the digitized surface, with a zoomed in view of the raw data in point cloud form shown in Figure 7c, displaying the geometry of the road as well as the ditches, cars, and other objects of interest in the terrain model. The model contains over 3,000,000 data points, and took approximately 6 minutes to acquire.
Figure 7. Large-scale terrain digitization of a “double-U” loop on a gravel road and parking lot. (a) Aerial view of the region that was digitized, with the scanning path superimposed. (b) 3D surface of the digitized terrain with (c) a zoomed in view of the raw data in point cloud form.

CONCLUSIONS

In this paper we have presented a Mobile Scanning System for the fast digitization of 3D terrain models that is modular in nature. The system consists of laser range scanners, high-resolution digital cameras, and pose estimation instrumentation, and a mobile platform. This system can digitize kilometers’ worth of data in a short amount of time, and the modularity of the equipment allows it to be easily configured to meet a wide variety of applications. The results presented show the ability of our system to handle a variety of applications and environments, yielding accurate 3D models in a robust fashion. Switching out the laser range scanners allows the system to quickly switch between macro structure scanning (buildings) and micro geometry scanning (inspection tasks).

Future efforts for extending the capabilities of the Mobile Scanning System include: improving the pose estimation algorithms, adaptive regularization of the surface for noise removal, software enhancement of small-scale details, and improving the automation and robustness.

ACKNOWLEDGMENTS

REFERENCES


Model building for simulation and testing under uncertain conditions
Brad Grinstead*, Andreas Koschan, David Page, and Mongi A. Abidi
Imaging, Robotics, and Intelligent Systems Laboratory, The University of Tennessee,
334 Ferris Hall, 1508 Middle Dr, Knoxville, TN 37865

ABSTRACT
3D models of real world environments are becoming increasingly important for a variety of applications: Vehicle simulators can be enhanced through accurate models of real world terrain and objects; Robotic security systems can benefit from as-built layout of the facilities they patrol. Vehicle dynamics modeling and terrain impact simulation can be improved through validation models generated by digitizing real tire/soil interactions. Recently, mobile scanning systems have been developed that allow 3D scanning systems to undergo the full range of motion necessary to acquire such real-world data in a fast, efficient manner. As with any digitization system, these mobile scanning systems have systemic errors that adversely affect the 3D models they are attempting to digitize. In addition to the errors given by the individual sensors, these systems also have uncertainties associated with the fusion of the data from several instruments. Thus, one of the primary foci for 3D model building is to perform the data fusion and post-processing of the models in such a manner as to reconstruct the 3D geometry of the scanned surfaces as accurately as possible, while alleviating the uncertainties posed by the acquisition system. We have developed a modular scanning system that can be configured for a variety of application resolutions, as well as the algorithms necessary to fuse and process the acquired data. This paper presents the acquisition system and the tools utilized for constructing 3D models under uncertain real-world conditions, as well as some experimental results on both synthetic and real 3D data.

Keywords: 3D modelling, system error, kriging, geometry refinement

1. INTRODUCTION
Accurate 3D models of real-world environments often provide useful information that increases the effectiveness of a variety of tasks. Increasingly, static 3D imaging systems – those where the imaging sensor remains fixed in place as it digitizes a scene – are being replaced by mobile 3D imaging systems, in order to provide the full range of motion necessary to acquire accurate 3D models in a short amount of time. Realistic urban models can be used to enhance driver performance analysis in vehicle simulations. Path planning for robotic security systems and couriers can be enhanced by accurate as-built models of the facilities they patrol. Terrain impact modeling and vehicle/terrain interaction simulations can have their results verified and validated through 3D models acquired by digitizing actual tire/soil interactions. These dynamic 3D imaging systems can quickly yield accurate 3D models of their target environments, but the very act of motion that gives them their flexibility also introduces an additional source of measurement uncertainty (error).

Frequently, within the environment that is being modeled there are Regions of Interest (ROIs) containing small-scale details – defined as areas whose geometric features are much smaller than the overall size of the scene as well as being at or near the lowest level of resolution that the digitization system can record. Occasionally, during the scanning process, these ROIs are not digitized at a high enough resolution or accuracy to be accurately represented for further analysis. Enhancing the details of an acquired dataset involves additional sampling of the underlying surface, taking into account the uncertainties involves in the digitization process. Physically re-acquiring the data may not be feasible,
due to the costs involved in performing the acquisition again. In that case, software-based geometry refinement is the only viable option of enhancing the details present in the existing 3D data.

Software-based geometry refinement has a number of benefits, including: the ability to recover details that are near, or below, the low end of the scanning system’s resolution from an existing dataset; the ability to set up the acquisition to acquire at a lower resolution, based on the assumption that the refinement method will recover the details of interest from a lower resolution scan, yielding faster acquisition and smaller raw data files. As an example of this type of data, Figure 1 shows a laser range dataset that was acquired at 5cm/profile resolution. This resolution is not fine enough to accurately represent the ideal (synthetic) 3D geometry seen in the bottom image of Figure 1.

The number of applications that can benefit from small-scale detail reconstruction is quite large. For tire/soil interaction simulation validation, models of actual tire/soil interactions may not always be digitized at an appropriate resolution to determine the small-scale interaction of, for instance, a tire tread and the terrain it was moving over. An algorithmic geometry refinement method could be used to refine the acquired data, rather than getting the acquisition system back out in the field and doing the experiments again. Geometry refinement methods can also improve the results of crack detection from a mobile platform, and even the quality of a priori models used for robotic manipulation tasks. In fact, many 3D modeling applications that have regions of important details that are near, or below, the low end of the scanning system’s resolution and are much smaller than the overall dataset’s dimensions can benefit from this method.

In the majority of these applications, critical information is contained at the edges of the 3D structure. Preservation of these edges, or sharp features, is therefore the priority when implementing any geometry refinement method. Classical approaches for resampling range data are linear and cubic interpolation, and Inverse Distance Weighting (IDW). Statistical methods such as kriging have also been applied to 3D datasets. Despite the fact that 3D data tends to be sampled at very small separation distances, the uncertainty of resampling using these methods tends towards oversmoothing or deformation of the sharp features present in the data.

In this paper, we discuss a 3D acquisition and modelling process that takes into account the various sensor errors, and constructs an accurate model from the data measured. Then, assuming that the data contains ROIs with enhanceable details, we segment out the ROIs and use a modified version of kriging that uses the spatial relationships present within the data to provide estimates of the underlying surface. This geometry refinement process also utilizes a regularization component whose parameters are tailored towards preserving the edge information contained within the data. This model...
geometry refinement process is fully automatic and is demonstrated on several synthetic datasets, showing that our modified kriging method outperforms its common competitors in terms of accuracy and feature enhancement.

Section 2 of this paper describes the basics of a mobile 3D scanning system and the errors associated with such a system. The fundamentals of geometry refinement are discussed in Section 3. Section 4 describes our method for dealing with uncertainty in the models, as well as our modifications made to automate the kriging process and tailor the output towards preserving edge information. Section 5 covers the experimental process used to validate this method, a comparison to existing methods, and the application to real-world problems through synthetic data. The paper concludes with a summary of the work presented, the results obtained, and a brief discussion of the future efforts of this research area.

2. MOBILE SCANNING SYSTEMS

Digitizing real-world environments using any 3D scanning system necessarily imposes certain application-specific constraints. These constraints vary from loose to rigid, and involve things such as: the acquisition environment, the size and shape of the structures to be scanned, the time available for acquisition, the expense and quality of the instrumentation to be used, etc.

Mobile scanning systems were developed in order to reduce the time-on-site of the data acquisition process. They typically involve a 3D scanner setup – laser or stereo imaging – as well as a sensor suite for determining the position and orientation (self-localization) of the scanning equipment at every location. A typical system pipeline for a mobile 3D scanning system can be seen in Figure 2. Here, a laser range scanner acquires the surface geometry of the surrounding environment, a Global Positioning System (GPS) measures the position of the sensor package, an Inertial Measurement Unit (IMU) measures the orientation, and a video camera collects a 2D image sequence that can be used for texture overlays as well as self-localization. The sensor suite is mounted upon a mobile platform – suited for the environment to be scanned – which is moved past the scene to be digitized in order to acquire the 3D model of the real-world surface. More information about this type of mobile 3D scanning system can be seen in the work of Grinstead, et al.5,6

![Figure 2. Basic pipeline of a mobile 3D scanning system. The instrumentation is placed upon a mobile platform during the data acquisition process.](image-url)
It was mentioned previously that the free motion undergone by a mobile scanning system that gives it the flexibility to reconstruct a variety of complex scenes in a short amount of time also contributes an additional source of measurement uncertainty (noise). This is due to the fact that the measurements are now being corrupted not only by the 3D scanner’s estimation errors, but also by the errors involved in measuring the position and orientation of the scanner at each acquisition, as well as the uncertainty involved in the fusion of the data from multiple sensors into a common framework.

Figure 3 demonstrates the effect the motion uncertainty has on the reconstructed surface. On the left is a 3D model of the front of a grocery store, as taken by a stationary rotating laser scanning system. On the right is a 3D model of the same store, taken by the same 3D laser scanner now mounted on a mobile platform undergoing motion past the scene. The mobile scanning data has accurately modeled the overall shape of the structure, but the fine details are missing or distorted due to the motion undergone by the sensor package during the scanning process. Specifically, the vertical “slats” in the grid around the letters are missing, and the letters themselves are oversmoothed.

We have stated previously that the act of motion itself causes the additional uncertainty in the 3D reconstruction. Some of the sources of this error are: sensor measurement error in the positioning and orientation measurement devices; measurement error in relating the motion measured by the pose instrumentation to that undergone by the scanner during acquisition; surface errors due to not sampling the geometry at a high enough resolution to accurately recover small-scale details. The larger error sources can be eliminated, or their effects ameliorated through a data fusion process that uses statistical models of the error sources in order to minimize their effects, such as Kalman filtering4 or Thrun’s Probabilistic Robotics methods.13 These methods combine the measurements in the data fusion process to yield 3D models that are accurate on the gross scale, but less so on the micro scale. Specifically, when the object or area of interest has features that are at or near the lowest level of resolution of the scanning system, often these areas are undersampled due to the motion undergone by the platform, as was the case of the vertical slats in Figure 3.

3. GEOMETRY REFINEMENT METHODS

Spatial interpolation has been defined as the procedure of estimating the value of a field variable at unsampled locations within the area covered by sample locations.15 For data acquired from a mobile scanning system, the geometry refinement process involves estimating the underlying surface from a non-uniformly sampled set of measurements. The underlying assumption of any interpolation algorithm is that samples close to each other will likely be more similar than points that are further apart.

Linear and cubic interpolation on a non-uniformly sampled field variable is typically done by performing a Delaunay triangulation to optimally connect the sampled point into a network of triangles. Interpolation is then done by searching the network for the closest triangle and computing the linear or cubic interpolant from that triangle. This method of interpolation is quite fast, but is very dependant on the uniformity of the sampling of the data.
Inverse distance weighting is a commonly used weighted average interpolator, where the influence of a measured point on the new estimate decreases as the distance from to the location of the estimated point increases. The weighting power $p$ controls how the influence tapers off with respect to distance $d$:

$$Z_e = \frac{\sum_i w_i Z_i}{\sum_i w_i},$$

$$w_i = d_i^{-p},$$

where $w_i$ is the weight associated with the measured value $Z_i$, and $Z_e$ is the estimation of the new value. Frequently the power used for performing the interpolation is attached to the IDW abbreviation, thus inverse distance weighting using a power of 2 is designated as IDW2. In its native state, IDW uses all measured data points in the weighting process, but more frequently only the points within a user-specified radius are used. One of the characteristic effects of IDW is that it tends to generate a “bulls-eye” effect around measured points. Smoothing parameters are sometimes introduced to alleviate this effect.

Kriging is a spatial interpolation method that gives estimates of new surface values based on a statistical model of the spatial correlation between measured values. Kriging uses observed data to estimate the values of the underlying surface, optimized to yield a minimal error variance in the reconstructed values. The acquired data provide the ‘support’ for the estimation, and the quality of the measured data affect the ability to accurately reconstruct the surface. Kriging is often considered to be the most accurate predictor of all other interpolation types – given the proper circumstances – due to the way it models the data both as an underlying trend and a set of residuals.

Variograms are typically used to model the spatial correlation of measured surface data. The variogram is a function which characterizes the dependence of data points measured in a region of interest. For every sampling interval $d$, the semivariance for distances equal to multiples of $d$ can be computed as

$$\gamma(d) = \frac{1}{2N(d)} \sum_{i=1}^{N(d)} (z_i - z_{i+h})^2,$$

where $\gamma(d)$ is the semivariance at lag $d$, $z_i$ is the measurement of a regionalized variable taken at location $i$, $z_{i+h}$ is another measurement taken $h$ intervals away, and $N(d)$ is the number of separating distances: number of points - number of lag intervals. After the experimental variogram has been calculated, an ideal parametric model is fit to the experimental variogram, which is used to simplify the estimation process and provide some robustness to measurement error.

Kriging is the process of using the parametric variogram model to estimate the surface value at the specified location. The most common form of kriging used in engineering applications is ordinary kriging (OK). In ordinary kriging, the estimate of an unknown surface value uses a weighted estimate of other nearby points:

$$Z_e = \sum_i w_i Z_i.$$

Ideally, kriging attempts to minimize the error between the estimated point and the actual value. The variance of this error is the amount of scattering of the estimates $Z_i$ about their true values $Z$:

$$\sigma^2 = \frac{\sum_i (Z_i - Z_a)^2}{n}.$$  

The estimation and its error are dependent on the weights chosen in (3). Optimal weights, therefore, would be those that produce the minimum estimation variance. These are found by solving a system of equations consisting of the
weighted semivariances between measured points, and the estimated semivariances between the unknown point and the
known values

\[
\begin{bmatrix}
\gamma(d_{11}) & \cdots & \gamma(d_{1n}) \\
\vdots & \ddots & \vdots \\
\gamma(d_{m1}) & \cdots & \gamma(d_{mn})
\end{bmatrix}
\begin{bmatrix}
w_1 \\
\vdots \\
w_n
\end{bmatrix}
= \begin{bmatrix}
\gamma(d_{1p}) \\
\vdots \\
\gamma(d_{np})
\end{bmatrix},
\] (5)

where \( \gamma(d_{ij}) \) is the semivariance between the \( i^{th} \) and \( j^{th} \) measured values and \( \gamma(d_{ip}) \) is the estimated semivariance between the \( i^{th} \) observation and the location of the evaluation point.

4. AUTOMATED GEOMETRY REFINEMENT VIA REGULARIZED KRIGING

Standard kriging has a number of issues that cause it to be less than ideal for geometry refinement in the presence of small-scale details. First, the global nature of the kriging process tends towards smooth surface reconstruction, that obscures, rather than enhances, the small-scale details present within the data. Second, there are a number of parameters in the OK process that require manual selection and initialization. Also, OK typically uses the entire set of measured data for training (variogram estimation). However, the small-scale details we are trying to enhance, usually constitute only a fraction of the overall dataset. Using the entire dataset for training, therefore is not beneficial when the geometry refinement is only going to be performed on relatively small ROIs. Finally, although OK itself leads to a robust solution with a minimum of error variance, the results tend to be oversmoothed, without taking into consideration that the measurement errors may be different at every sampling location or region. The rest of this paper presents our solutions to these issues, followed by experimental evidence that verify and validate these solutions.

The ROIs chosen for geometry refinement typically stand out from the rest of the data by being different – in terms of curvature, density, shape, spatial frequency – than the surrounding areas, and also tend to be much smaller than the dimensions of the overall dataset. These regions can be picked out quite easily by the human observer, but to have a computer vision system perform the task requires a mathematical or heuristic description of what constitutes a region of interest. We have chosen to follow the work of Page, et al., where the surface curvature is used to segment areas of common curvature, and to look only at those regions that contain uncommon curvature.

The ROI segmentation process begins with by randomly selecting seed triangles on the digitized surface, as seen in the by the highlighted triangles of Figure 4a. Region growing is then performed, enlarging the seed patches by assimilating adjacent triangles if the curvature is similar to that of the patch. Patches that come in contact with each other are merged together if their curvatures are similar. Figure 4b shows the highlighted seed patches after several iterations of region growing and merging. Once every patch has reached its maximum size – i.e., no more triangles/patches can be merged together on the basis of common curvature – a patch culling algorithm is performed to remove those regions with fewer than a pre-specified number of triangles. What remains is a set of surface patches that have a relatively

![Figure 4. Segmentation process, showing the (a) seeding, (b) region growing and merging, and (c) small region culling.](image-url)
constant curvature, as seen in Figure 4c. The inverse selection yields regions that do not have a common curvature, and are thus our ROIs. These ROIs are not only used to define the areas that need refinement, they are also used as the training data for the kriging process. By restricting the training data to lie only in the ROIs, we get an improvement in surface fit due to the removal of the influence of the data that is “not of interest”.

Variogram selection is another area in which some effort was necessary in order to automate the geometry refinement. Typically, variograms are generated by manually fitting a model selected from a library of functions to the estimated variogram values calculated from Equation (2). In order to automate this process, we perform a least-squares fitting of a number of ideal models, and the one with the best match to the data is used as the ideal variogram model for the kriging process. The fitting is constrained to closely fit the model to those lags near the source, since we are interested in the best local fit of the data. This selection process automatically selects the model appropriate for the measured data, with the added benefit of a closer fit to the underlying surface near the point of estimation.

In experimental systems, it is not possible to sample data at one location over and over without incurring some variation in the samples. This variation is due to system noise, environmental shifts, and other factors. Traditional kriging treats this micro-scale variation in a global fashion, adding a “nugget” effect to the variogram at $d=0$. However, this is inappropriate when it is known that the measurement variation in every region is likely to be different. Instead, we employ a regularization procedure that attempts to model the micro-scale variations independently. This regularized approach accounts for local measurement errors and helps to reduce the smoothing effect that the nugget parameter introduced.

Given that Equation (5) can be written as $Ax=b$, we can solve the system using Ordinary Least Squares

$$x = (A^T A)^{-1} A^T b. \tag{6}$$

This is equivalent to zero-order Tikhonov regularization. Since $(X^T X)^{-1}$ is ill-conditioned, we introduce a regularization parameter $\lambda$ to increase stability of the system

$$x = (A^T A + \lambda I)^{-1} A^T b. \tag{7}$$

Here $\lambda$ can be chosen to be either a scalar or a vector. It can be automatically determined through validation methods such as autocorrelation, or empirically chosen through experimentation. We suggest choosing a $\lambda$ that reflects the amount of uncertainty present in the data acquisition system. Our $\lambda$ was chosen using the L-curve analysis technique of Hansen.8

5. EXPERIMENTAL RESULTS

In order to demonstrate that the augmentations made to the original kriging algorithm actually yield a better geometry refinement, a number of experiments were performed on synthetic datasets. We have developed a simulator that mimics the performance of a mobile 3D scanning system to test our processing algorithms. The user can specify the parameters of the simulator to control the behavior of the system, from a completely noise-free scan of a specified model, to a very noisy dataset where a large amount of uncertainty has been introduced. These parameters include the scanning resolution along a laser profile, the resolution between profiles, the orientation of the “scanner” at each sampling stage, and the noise parameters of the system. The simulator reads in a triangulated 3D model, performs the “scanning” using the specified parameters, and outputs a 3D model built by fusing the 3D range data with the computed pose information, in exactly the same manner as we do for the physical system.

Figure 5 shows a 2m x 2m synthetic model developed for tire/soil interaction. The model is an impression of a repeated tire tread pattern embedded into a varying terrain surface. This model contains regions of high detail, as well as smooth regions, and is useful in demonstrating the effects of a laser scanning system on such data, as well as providing a ground truth model for comparison of geometry refinement results. We set the simulator parameters to be:
Figure 5. Results of geometry refinement methods applied to a synthetic test dataset. A synthetic reference model was scanned using a simulator that replicates the effects of a mobile 3D scanning system, and the output was used as the data for a selection of geometry refinement algorithms. Variations from the reference model are color coded.
a vertical resolution of 5 mm, a horizontal resolution of 2 cm, and a range/orientation uncertainty of 0.5%. This type of non-uniform sampling is consistent with many laser scanning systems, and is the cause of the majority of the undersampling problems of interest in this paper. The output of this system was then used as the input data for 4 different interpolators: linear, cubic, IDW, and our modified kriging, with the surface being reconstructed a resolution of 5 mm in both the horizontal and vertical directions.

The results of the interpolation methods applied to the synthetic model are also shown in Figure 5. The refined surfaces are color coded according to each reconstructed point’s distance to the reference model. Surface points close to the ground truth are shown in blue moving through the spectrum to green and finally to red as the deviation grows large. It can be seen that our modified kriging algorithm has fewer areas of red, and more of blue and green, than the other methods, indicating that the interpolated surface follows closer to our ground truth model. The density of the signed distances from the reconstructed surfaces for each of the 4 methods is shown in Figure 6. Note that our automated Kriging refinement had a tighter distribution about 0, indicating that overall it better represents the underlying surface. The IDW2 model had a much worse distribution of errors (as seen by its mostly red surface in Figure 5), due to the characteristic “bulls eye” effect of IDW on our irregularly sampled data. Table 1 gives more complete information about the global performance of the 4 interpolations methods. Table 1 lists the minimum, maximum, mean, and median deviation (error) of the refined surface from the ground truth model, the variance of the errors, and the Root Mean Square error for each of the refinement methods tested. Note that our kriging algorithm outperforms all of the other tested methods.

Now that we have shown that our modified kriging method can outperform its competitors in a general sense, we demonstrate the effects of training sets on the geometry refinement process – specifically, how the choice of training

![Figure 6. Distribution of signed distances from the refined surfaces to the reference model.](image)

<table>
<thead>
<tr>
<th>Method</th>
<th>Min (mm)</th>
<th>Max (mm)</th>
<th>Mean (mm)</th>
<th>Median (mm)</th>
<th>Variance</th>
<th>RMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>9.59 e-5</td>
<td>25.95</td>
<td>2.52</td>
<td>1.52</td>
<td>1.01 e-2</td>
<td>4.04</td>
</tr>
<tr>
<td>Cubic</td>
<td>9.61 e-5</td>
<td>25.80</td>
<td>2.58</td>
<td>1.63</td>
<td>8.41 e-3</td>
<td>3.84</td>
</tr>
<tr>
<td>IDW2</td>
<td>13.42 e-3</td>
<td>44.83</td>
<td>9.84</td>
<td>8.19</td>
<td>5.76 e-2</td>
<td>12.43</td>
</tr>
<tr>
<td>Modified Kriging</td>
<td>6.19 e-4</td>
<td>25.18</td>
<td>2.50</td>
<td>1.17</td>
<td>7.59 e-3</td>
<td>3.78</td>
</tr>
</tbody>
</table>

Table 1. Quantitative comparison of interpolation results
data affects the accuracy of refinement in the ROIs. Six training regions were selected, as seen in Figure 7, three each from the smooth and detailed surface area of the tire tread model. Semivariograms were trained on the Scanned Geometry (from Figure 5) of each of these regions independently, and the model was reconstructed at a resolution of 5 mm. A graph relating the median distance of several reconstructed surface patches back to the original model is shown in Figure 8. Each of the 7 reconstructed surfaces – one for each of the 6 training patches, plus the reconstruction using the whole dataset for training – was compared to the reference model as a whole, as well as to each of the regions corresponding to the training patches of Figure 7. The graph shows that the reconstructions using training sets from the detailed regions compare favorably with those using the smooth and whole training sets, and significantly outperform those other refinements within the detailed regions. This supports our claim that restricting the training set to only our specified ROIs is beneficial to the geometry refinement.

Figure 9 gives a visual representation of the location and magnitude of the errors in the surface reconstructed from region Detail3. The larger the dot, the larger the error in the reconstructed surface. Surface elevation contours have been overlain on the image for context. From this figure, it can be seen that the highest magnitude of errors occurs, unsurprisingly, in those areas that have the largest changes in curvature. Also, the largest errors occur in a vertical striping pattern that lies along the scan lines from the input data.

**Figure 7.** Six regions chosen to test the effect of training regions on the kriging refinement. To the right are the estimated variograms from each region.

**Figure 8.** Median distance to each specified region of interest, for each of the training sets used.

**Figure 9.** Spatial location of the errors from the Detail3 interpolation, with the height contours of the data overlain for context. The larger the dot, the larger the error.
The next set of experiments involves downsampling an existing 3D model in order to determine the effects of a reduced sampling on a less symmetric dataset. Figure 10 shows a 400 mm x 300 mm model of a stretch of pavement, imaged at a 1 mm resolution in the horizontal and vertical directions. This high-resolution model shows a large crack that runs in a loose arc through the model, with an average depth of 14 mm. To demonstrate the reconstruction ability of our method on a dataset that has been resampled, or sampled at a lower resolution originally, we resampled the original data at intervals of 3 mm and then performed the geometry refinement on the resampled data. Our automated kriging method provides a close reconstruction, as seen by the color-coded error map to the right of the Automated Kriging reconstructed model. In comparison, the IDW2 reconstruction has more error, and fails to adequately represent the small-scale details present in the Reference Model.

![Reference Model](image1)
![Subsampled Model](image2)
![Automated Kriging](image3)
![IDW2](image4)

**Figure 10.** Comparison of geometry refinement methods on a resampled dataset. A high-resolution model of a patch of pavement is resampled at 1/3 the original sampling and then geometry refinement via our automated kriging and IDW2 are compared. Variations from the reference model are shown color coded to the right of each reconstructed model.

### 6. DISCUSSION

In this paper, we have presented work done on an automatic method for identifying regions within a large 3D model of a real-world environment that contain small-scale details and performing a surface geometry refinement on those regions. This method was shown to be valid by experimentation on synthetic models in a controlled setting. The experiments showed the ability of our automatic geometry refinement method to retrieve data “hidden” by undersampling and measurement uncertainties. Experimental models used for validating our geometry refinement method include tire/soil interaction and pavement inspection, and the results shown are typical for the method.
At this point, it should be stated that our geometry refinement method is computationally intensive, forcing a trade-off between accuracy of reconstruction and time performance. The decision on whether the extra computation is an acceptable cost for the improvements in the geometry refinement will need to be weighed carefully, balancing the cost/accuracy ratio on the needs of the specific application.

The future for this research involves continued improvements in the reconstruction accuracy, enhancements in speed, and further efforts into removing pre-specified parameters from the system. One of the issues of current interest is investigating nonparametric variogram estimation methods such as that presented by Gorsich and Genton. Also of interest is a rigorous study of the minimal measurement sampling requires to reconstruct geometric details of a given dimension.

ACKNOWLEDGEMENTS

This work was supported by the DOE University Research program in Robotics under grant DOE-DEFG02-86NE37968 and the DOD/RDECOM/NAC/ARC Program under grant W56HZV-04-2-0001.

REFERENCES

Improving Video-Based Robot Self Localization Through Outlier Removal

Brad Grinstead, Andreas Koschan, Andrei Gribok, and Mongi A. Abidi
The University of Tennessee
334 Ferris Hall
1508 Middle Drive
{bgrinste, akoschan, agribok, abidi}@utk.edu

Abstract – The purpose of this paper is to present a method for rejecting false matches of points from successive views in a video sequence – e.g., one used to perform Pose from Motion for a mobile sensing platform. Invariably, the algorithms used to determine point correspondences between two images output false matches along with the true. These false matches negatively impact the calculations required to perform the pose estimation from video. This paper presents a new algorithm for identifying these false matches and removing them from consideration in order to improve system performance. Experimental results show that our algorithm works in cases where the percentage of false matches may be as high as 80%, providing a set of point correspondences whose true/false match ratio is much higher than the mutual best match method commonly used for outlier filtering, resulting in comparable or better outlier rejection – increasing the true/false match ratio by 2-3 times – in only a fraction of the time.

Keywords: Robot self-localization, pose from motion, outlier rejection, feature matching, epipolar geometry.

1. INTRODUCTION

Performing tasks in a variety of environments is an increasing demand in DOE applications for robotic systems. This requires that the robots have the ability to navigate within several types of environments including those where GPS signals cannot be received or are not reliable. With the advent of inexpensive video cameras, optical navigation is being increasingly utilized on robotic vehicles. Using video for navigational purposes involves determining the interframe motion between successive images, and often relies on the matching of feature points from one image to another. Accurate and timely estimation of the robot’s motion is dependant on the quality of these feature matches.

It is known that two images of a static scene are related to each other through their epipolar geometry and epipolar geometry constraints have been used to perform Structure and Pose from Motion tasks. A robotic vehicle’s pose can be determined from a video sequence by finding corresponding features between adjacent images in the video sequence and using the scene’s epipolar geometry to calculate the position and orientation changes between the two images. In the scope of this paper, we refer to pose estimation as the process of determining the vehicle’s motion and orientation, also known as ego-motion. Henceforward in this paper, we will use the terms ego-motion and pose estimation interchangeably.

Figure 1 shows how the epipolar geometry plays a fundamental role in the determination of the rigid transformation relating two separate camera systems. Given two images p and p’ of a point P with the optical centers of each image located at O and O’, the five points define an epipolar plane that is formed by the baseline OO’ and the imaging rays OP and O’P. The intersection of this plane with the image planes ? and ?’ form the epipolar lines l and l’. The epipole e’ is the projection of the optical center O in the image plane ?’ and vice versa. The epipolar line l’ is associated with p and vice versa. Thus, it can be seen that if p and p’ are images of the same point P, then p’ must lie on the epipolar line l’ associated with p. Thus, with the knowledge of p and p’ and the optical centers O and O’, the relative transformation between the images can be determined.

The epipolar geometry estimation is directly dependant on the quality of the feature matches input to the algorithm. The feature matching task is difficult enough that even sophisticated methods will output false matches in addition to the true. These false matches are not consistent with the scene’s epipolar geometry, and are known as ‘outliers’, while those true feature matches are known as ‘inliers’. For naïve motion estimation algorithms (e.g., the least-squares algorithms), even a single outlier would significantly affect the pose estimation results.

Figure 1: Epipolar geometry

The purpose of this research is to present a method for rejecting false matches of points from successive views in a video sequence – e.g., one used to perform Pose from Motion for a mobile sensing platform. Invariably, the algorithms used to determine point correspondences between two images output false matches along with the true. These false matches negatively impact the calculations required to perform the pose estimation from video. This paper presents a new algorithm for identifying these false matches and removing them from consideration in order to improve system performance. Experimental results show that our algorithm works in cases where the percentage of false matches may be as high as 80%, providing a set of point correspondences whose true/false match ratio is much higher than the mutual best match method commonly used for outlier filtering, resulting in comparable or better outlier rejection – increasing the true/false match ratio by 2-3 times – in only a fraction of the time.

**Keywords:** Robot self-localization, pose from motion, outlier rejection, feature matching, epipolar geometry.
can cause problems. A typical set of correspondences may contain over 50% false matches, which causes naïve estimators to fail completely.

There has been much effort put into outlier rejection for epipolar geometry estimation, including robust epipolar geometry estimators as well as pre-filtering techniques. Robust estimators use statistical sampling and modeling techniques to alleviate the effects of outliers. However, the computational cost of these algorithms greatly increases with the percentage of outliers present in the data. Shi and Tomasi proposed a feature tracker that discards matches based on their image residual. Tommasini et al. extended this concept to include robustness to illumination changes. Fusiello, et al. compute the epipolar parameters using all available feature matches, filtering out those that do not correspond. Chua et al. compute the epipolar parameters using all available feature matches, filtering out those that do not correspond. In other words, a feature pair \( f_1, f_2 \) is only considered as a potential correspondence if \( f_2 \) is the best match for feature \( f_1 \) in image 2, and \( f_1 \) is the best match for feature \( f_2 \) in image 1. The MBM approach greatly reduces the number of outliers present in a set of feature correspondences, but often, the results still contain a significant amount of outliers. In addition, the very nature of the mutual best match approach means that an intensity correlation is done – at minimum – twice for every feature. This paper explores an alternative outlier rejection method that avoids the multiple intensity correlation passes through the images’ features by taking advantage of the “uni”-directional nature of the motion of our scanning system.

Our application involves recovering the motion of a mobile scanning system whose main imaging device is orthogonal to the main direction of motion. This system is commonly seen in airborne stereo applications, and in ground level urban scanning and modeling. For this scenario, we have developed a new technique for outlier rejection, based on the motion trajectories of matched points, that improves on the results generated by the mutual best match method found in the literature in two major areas: improved true/false correspondence ratio, and significant increase in speed.

The results presented in this paper show that our method decreases the number of outliers significantly, while reducing the number of computations needed for both outlier rejection and for the robust estimation of the epipolar geometry. Our algorithm provides outlier rejection that provides a true/false match ratio of up to 95%, leading to better epipolar geometry estimation in a fraction of the time required for previous outlier rejection algorithms.

The rest of this paper is laid out as follows: Section 2 discusses the Oriented Tracks algorithm, including the assumptions made and the selection of the various parameters, as well as a discussion of the parameter reduction from Adam’s method. In Section 3, experimental results are presented, comparing our algorithm with the MBM approach, and showing that our algorithm results in faster processing time and better outlier rejection. Finally, Section 4 concludes with a summary of the algorithm and its performance, along with a discussion on future extensions.

2. ORIENTED TRACKS

For our application, the camera is mounted on a rear-wheel drive vehicle so that the camera’s optical axis is perpendicular to the vehicle’s main direction of motion. We assume that the acquisition rate of the camera is high enough that the interframe motion is relatively small (\(<50\) pixels). We also assume that the environment is such that the vehicle motion along or about the optical axis of the camera is relatively small. For a static environment under these constraints, true interframe correspondences will exhibit similar behavior, in regards to their motion trajectories.

The feature tracks for a typical sequence can be seen in Figure 2b. This set of correspondences contains both true and false matches, as can be seen in the feature tracks – e.g., on the right side of the scene, there are feature tracks that are nearly vertical due to mismatches in the intensity correlation process. These mismatches are outliers that limit the efficacy of standard epipolar geometry estimation methods. Filtering out these mismatches allows us to...
achieve correspondence sets such as that shown in Figure 2c, reducing the input data to a smaller set of correspondences with a higher true/false match ratio and increasing the efficacy of the epipolar geometry estimator.

To show that our observation is valid, let us assume that a single camera, with intrinsic parameters $K$, undergoes a translation. Let an image point in the first image be normalized as $x = [x_1, x_2, 1]$. Then, the space point’s inhomogeneous coordinates are $X = [X_1, X_2, X_3] = K^{-1}x$. From $x = K' [R|t]X$, the mapping of $x$ to its corresponding location in the second image $x'$ is

$$x' = K'R^{-1}x + K'Rt / X_3. \quad (1)$$

In our case, we are assuming that the motion is constrained to the image plane, thus $R=I$, $K=K'$, and $t=[t_1, t_2, 0]^T$. Under these conditions, (1) reduces to

$$x' = x + K't / X_3. \quad (2)$$

The feature track for this pair is $x' - x = K't / X_3$. Similarly, the feature track for a second corresponding pair $y' - y = K't / Y_3$. Thus, the feature tracks are shown to follow the same direction, and are parallel, scaled by the depth values $X_3$ and $Y_3$. System noise and inaccuracies in the calibration of the camera – as well as perturbances outside of the image plane that are much smaller than the motion within the plane – may lead to correspondence tracks that are not quite parallel, but this can be handled by defining an acceptance region for which the matches are considered to be correct.

Adam, et al. proposed an algorithm – Rejection of Outliers by Rotations (ROR) – to reject false matches from any two views of the same scene that involves virtual rotations of one of the images to identify some common behavior among the correct feature matches. The basic concept is that under a correct rotation, the inliers will have a common feature track direction, and the outliers will be “shaken away”. A user-defined number of random rotations are performed and for each rotation, the algorithm looks for feature tracks that lie in a common direction. The correct feature tracks are those that lie within a specified angular distance from the common direction, averaged over the best rotations. This algorithm has proven to be effective when applied to general camera motion viewing a static scene.

Under our application, the motion constraints of the mobile sensor platform mean that the correct matches will share a common behavior in their feature tracks for the original image pair, with no rotations necessary. With this knowledge we have designed an algorithm – similar to ROR – that requires no virtual rotations of the images and relies only on a single user-specified parameter, as opposed to the five required for the ROR algorithm. Our proposed algorithm is outlined as follows:

**Oriented Tracks Algorithm**

- Begin with a set of potential correspondences $c_i = (a_i, h_i)$ between images A and B.
- Calculate each feature track direction from the line joining the locations of the corresponding pair.
- Estimate the probability density of the feature track directions.
- Find the main orientation as the mode of the distribution of feature track directions $m$.
- Find the angular distance $d_i$ from each feature track to the mode $m$.
- A feature track is considered consistent with the main orientation if for some acceptance region $e$, $d_i < e$.

**2.1. Estimating the Distribution**

There are many ways to estimate a probability density function of a random variable. Generating a histogram of the data can give a quick estimate of the distribution, but histograms are subject to errors due to the selection of bin width and interval, requiring additional user-specified parameters. Thus, we have chosen to estimate the...
probability density of the feature track directions automatically, using a kernel density estimator (KDE). A kernel density estimator with a given kernel $K$, is defined by

$$p_{\text{pdf}}(x) = \frac{1}{nh} \sum_{i=1}^{n} K \left( \frac{x - X_i}{h} \right),$$

where $h$ is the kernel width, $n$ is the number of sampled data points, and $X_i$ is the $i^{th}$ observation of the random variable $X$. The KDE we chose to use is a weighted summation of Gaussian kernels, which are placed at the observed data. The width of the Gaussian kernels can be chosen empirically, or optimized with respect to the data.

We have chosen to use an L-stage Plug-in method to automatically determine the optimal kernel width, because it performs an asymptotic minimization of the mean integrated square error between the estimated probability density and the actual density function. Silverman’s solution for the optimal kernel width assumes that the true distribution function can be represented by a normal distribution. Using this assumption in combination with a Gaussian kernel, the optimal bandwidth for kernel density estimation can be computed as

$$h_{\text{opt}} = \left[ \frac{4 \sigma}{3n} \right]^{1/5},$$

where $\sigma$ is either the sample standard deviation or the standardized inter-quartile range. In practice the smaller of the two is used.

### 2.2. Choosing the Acceptance Region

The parameter controlling the width of the “consistent” region is the only user-selected parameter in this outlier rejection method. If this region is too large, feature tracks inconsistent with the epipolar geometry may be kept in the correspondence set. If this region is chosen to be too small, the noise inherent in digital acquisition systems will cause consistent feature matches to fall outside the threshold, and thus will remove them from the set of correspondences.

An acceptance region of 0.17 radians has been empirically determined from our test data to provide the best tradeoff between outlier rejection and maximum inliers for the epipolar geometry estimation procedure. As another option, since the window width choice is used to account for errors in the vision system, it could be predetermined by some foreknowledge of the system noise.

Tracks inconsistent with the main orientation - i.e., those feature tracks whose angular distance from the main orientation does not lie within the acceptance window – are determined to be outliers and are filtered out, with the remaining correspondences retained as our new, concise feature set.

### 3. EXPERIMENTAL RESULTS

To demonstrate the efficacy of our outlier rejection method, we compared its filtering results to those of the typical mutual best match method for a variety of image pairs (~50) taken from our test platform, using a high-definition camcorder (JVC GR-HD1). Figure 3 shows three of these test pairs, which contain several effects that
cause outliers in the estimation of the epipolar geometry, including: digitization noise, resampling noise, motion blur, parallax effects, glare, shadows, etc. The scenes in Figure 3 increase in complexity from left to right.

For each image pair, a standard Harris detector was used to select approximately 1000 critical points, chosen for its prevalence in the literature and because it has been shown to have good stability properties. Matches between the images were found using a standard intensity correlation method. MBM was used as the baseline outlier rejection method. Feature track orientations were determined via the method discussed in the previous section, yielding the Oriented Tracks (OT) correspondences. Epipolar geometry estimation was performed as discussed in and the number of inliers was determined for both the MBM and the OT correspondences. For our experiments, inliers are determined as those features that lie within 2 pixels of the computed epipolar line. The procedures for feature detection, outlier rejection, and epipolar geometry estimation were implemented in Matlab and run on a 2.4 GHz Pentium processor.

In order to evaluate the pre-filtering results, we selected 50 true matches by hand for each of our example image pairs. Then, we used RANSAC to robustly estimate the epipolar geometry from these matches, yielding a set of ‘true matches’ that we then used as ground truth for comparison purposes. Following this, both MBM and our method were used to filter all of the feature pairs and a comparison of the results is shown in Table I, using the same matching process – intensity correlation, with a specified search radius of 50 pixels – for the initial matching of OT as well as the 2-pass matching of MBM. A comparison of the results obtained with our outlier rejection method and those obtained with MBM is shown in Table 1.

The results show that even for datasets that contain fewer than 40% true matches, our filtering method improved the true/false match ratio to better than 90%. This improvement in the quality of the feature correspondences leads directly to an improvement in the calculation of the epipolar geometry of the scene from feature matching, using RANSAC, in terms of both efficiency (time of execution) and accuracy (more inliers).

The RANSAC procedure used to estimate the epipolar geometry from the filtered set of correspondences does so through a randomly selected sample of feature matches, the choice of which can greatly affect the outcome in terms of iterations and inliers. In order to present an “average performance”, we ran the geometry estimation process 1000 times for each test image pair for both MBM and OT filtered correspondences.

Table I. Comparison of OT outlier rejection and MBM on real images

<table>
<thead>
<tr>
<th></th>
<th>Tasks</th>
<th>Intersection</th>
<th>Chemical Plant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total # of Features</td>
<td>910</td>
<td>868</td>
<td>949</td>
</tr>
<tr>
<td>True Matches</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oriented Tracks</td>
<td>463 (51%)</td>
<td>524 (60%)</td>
<td>696 (73%)</td>
</tr>
<tr>
<td>Mutual Best Match</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filtered Matches</td>
<td>416</td>
<td>458</td>
<td>503</td>
</tr>
<tr>
<td>False Matches Accepted (Misses)</td>
<td>15 (4%)</td>
<td>132 (12%)</td>
<td>23 (5%)</td>
</tr>
<tr>
<td>RANSAC Iterations</td>
<td>6</td>
<td>30</td>
<td>6</td>
</tr>
<tr>
<td>Outlier Rejection (s)</td>
<td>0.27</td>
<td>133.66</td>
<td>0.22</td>
</tr>
<tr>
<td>RANSAC (s)</td>
<td>0.16</td>
<td>0.77</td>
<td>0.19</td>
</tr>
<tr>
<td>Total (s)</td>
<td>0.33</td>
<td>0.41</td>
<td>0.26</td>
</tr>
</tbody>
</table>

Table I shows that increase for our demonstration data sets. By presenting the geometry estimation algorithm a better set of correspondences, we reduce the number of iterations by at least half, and often the improvement is much greater – on the order of 1/10. This decrease in iterations corresponds to a decrease in the amount of time the geometry estimation process takes. Overall, because we deliberately avoid duplicating the intensity correlation process – required by MBM – the amount of total time for pre-filtering and epipolar geometry estimation for our method is a mere fraction of that necessary for MBM. In total, our OT method for outlier rejection reduces the computation time by 2 minutes (a speedup of over 500 times) per image pair, which is a significant savings when the goal is to process an entire video sequence.

* Median number of RANSAC iterations taken from 1000 trials.
† Median time for RANSAC iterations taken from 1000 trials.
These results show that under the assumed motion constraints, our oriented tracks outlier rejection algorithm outperforms the mutual best match method in terms of reduced computations (improved speed) and has a more accurate epipolar geometry estimation – as ascertained by a greater number of inliers for the epipolar geometry estimation.

4. SUMMARY AND CONCLUSIONS

In this paper, we have discussed a method for the rejection of outliers as a processing stage between feature matching and epipolar geometry estimation in the scenario of estimating a vehicle’s motion from a video sequence. Our outlier rejection is based on the observation that for video captured from a mobile platform, true feature matches evince a common orientation in their frame-to-frame trajectories. We have demonstrated experimentally that this outlier rejection method provides more inliers – features consistent with the epipolar geometry – than the typical mutual best match filtering technique, while reducing the number of computations and halving the expected processing time. Our filtering method can provide true/false match ratios of up to 95% from data whose original ratios were less than 50%. Under certain conditions, ratios of 80% have been developed from feature matches that originally contained less than 20% of correct matches.

This algorithm is dependant on assumptions of motion common to Pose from Motion tasks. Specifically, that interframe motion cannot have significant motion along or about the camera’s optical axis. In the future, efforts will be made to extend the algorithm so that such motion is allowed. Also, an extension is underway to accommodate scenes where more than one motion is present.

ACKNOWLEDGEMENTS

This work was supported by the DOE University Research Program in Robotics under grant DOE-DE-FG02-86NE37968, by the DOD/RDECOM/NAC/ARC Program, R01-1344-18, and by FAA/NSSA Program, R01-1344-48/49.

REFERENCES

Outlier rejection by oriented tracks to aid pose estimation from video

Brad Grinstead a,*, Andreas Koschan a, Andrei Gribok a, Mongi A. Abidi a, David Gorsich b

a Imaging, Robotics, and Intelligent Systems Laboratory, Department of Electrical and Computer Engineering, The University of Tennessee, 334 Foris Hall, Knoxville, TN 37996, United States
b U.S. Army RDECOM Tank-Automotive Research, Development and Engineering Center, Warren, MI 48397-5000, USA

Received 12 April 2004; received in revised form 27 May 2005
Available online 31 August 2005
Communicated by T.K. Ho

Abstract

This paper introduces a method for rejecting the false matches of points between successive views in a video sequence used to perform Pose from Motion for a mobile sensing platform. Typical methods for pose estimation require point correspondences to estimate the epipolar geometry between the two views. Algorithms for determining these correspondences invariably output false matches along with the good. We present an algorithm for identifying and removing these mismatches for scenes generated by a mobile scanning platform. The algorithm utilizes the motion characteristics of a rear-wheel drive sensing platform to identify correct point matches through their common motion trajectories. Our algorithm works in cases where the percentage of false matches may be as high as 80%, providing a set of correspondences whose correct/incorrect match ratio is higher than the mutual best match approach found in the literature. This algorithm is intended as a post-processing step for any point correspondence algorithm and its output can be used in standard pose estimation algorithms to enhance their speed and accuracy. Experimental results show the computational savings of our approach over the mutual best match method, resulting in comparable or better outlier rejection—increasing the true/false match ratio by 2–3 times—in only a fraction of the time.

© 2005 Elsevier B.V. All rights reserved.
Keywords: Outlier rejection; Feature correspondence; Feature matching; False matches; Epipolar geometry; Robust estimation

1. Introduction

The last decade has seen video cameras being increasingly utilized on robotic vehicles for navigation purposes—as well as object identification and construction of 3D scene models (Shaprio et al., 1994; Johnson et al., 2000; Pollefeys et al., 2002; Armangue et al., 2003). Optical navigation involves the determination of frame-to-frame motion, which often relies on the determination of corresponding feature pairs between subsequent images. Accurate and timely estimation of the vehicle’s motion is dependent on the quality of these feature matches. It is known that two images of a static scene are related to each other through their epipolar geometry (Barnard and Fischler, 1982; Faugeras, 1993). More recently, epipolar geometry constraints have been used in Structure and Pose from Motion tasks using uncalibrated cameras (Forsyth and Ponce, 2003; Chua et al., 2000; Hartley and Zisserman, 2000). Determining a robotic vehicle’s pose from video (PV) typically involves finding corresponding features between adjacent images in the video sequence, estimating the scene’s epipolar geometry, and calculating the position and orientation changes between the two images (Branca

* Corresponding author. Tel.: +1 865 974 0262; fax: +1 865 974 5459.
E-mail address: bgrinste@utk.edu (B. Grinstead).
et al., 2000; Rivlin et al., 2003; Usher et al., 2003; Chroust and Vincze, 2004).
In the scope of this paper, we refer to pose estimation as the process of determining the vehicle’s motion and orientation, also known as determining its ego-motion. Henceforward in this paper, we will use the terms ego-motion and pose estimation interchangeably.

The results of the procedures used to estimate the epipolar geometry are dependent on the quality of the feature correspondences. The feature matching task is difficult enough that even the most sophisticated methods output false matches in addition to the good. These false matches are not consistent with the scene’s epipolar geometry and are known as outliers, while those matches that are consistent with the epipolar geometry are called inliers. For naive motion estimation algorithms—e.g., the least-squares algorithm (Hartley, 1995)—even a single outlier can cause problems. Now, a typical set of correspondences may contain over 50% false matches, which causes such naive estimators to fail.

Much work has been done on outlier rejection for epipolar geometry estimation. These techniques—designed to increase the robustness of the epipolar geometry estimation—include robust geometry estimators as well as pre-filtering techniques. The effects of outliers on the epipolar geometry estimation can be alleviated through robust algorithms (Zhang, 1998; Pollefeys et al., 1999; Salvi et al., 2001). However, the computational cost of these algorithms greatly increases with the percentage of outliers present in the data.

Shi and Tomasi (1994) proposed a feature tracker that discards matches based on their image residual. Tommasini et al. (1998) extended this concept to include robustness to illumination changes. In addition to the robustness to illumination changes, Fusselio et al. (1999) also developed an outlier rejection method that is linked to their feature tracker. Feature matches are rejected if their computed image residual is greater than a certain threshold. Chiu et al. (2000) compute the epipolar parameters using all available feature matches, filtering out those that do not behave consistently across a triplet of images.

A typical approach to reducing the number of outliers is to limit the set of putative correspondences by only keeping those features that are mutual best matches (Pollefeys et al., 2000; Hartley and Zisserman, 2000; Corke, 2004)—also known as left-right consistency checking to those in the stereo field. In other words, a feature pair \( (f_i, f_j) \) is only considered as a potential correspondence if \( f_i \) is the best match for feature \( f_j \) in Image 2, and \( f_j \) is the best match for feature \( f_i \) in Image 1.

The mutual best match approach greatly reduces the number of outliers present in a set of feature correspondences. However, the set of correspondences generated still contains a significant amount of outliers. In addition, the very nature of the mutual best match approach means that an intensity correlation is done—at minimum—twice for every feature. This paper explores an alternative outlier rejection method that avoids the multiple intensity correlation passes through the images’ features.

Our application involves recovering the ego-motion parameters for a mobile scanning system whose imaging device is orthogonal to the main direction of motion. This system is commonly seen in airborne stereo applications (Jaeger and Bers, 2001), and in ground level urban scanning and modeling (Fruh and Zakhor, 2001; Zhao and Shibasaki, 2001).

For this system, we have developed a new technique for outlier rejection in the set of correspondences based on the motion trajectories of matched points that improves on the results generated by the mutual best match method found in the literature in two major areas: improved correct/incorrect correspondence ratio, and significant increase in speed. The results presented in this paper show that our method decreases the number of outliers significantly, while reducing the number of computations needed for both outlier rejection and for the robust estimation of the epipolar geometry. Our algorithm provides outlier rejection that results in a true/false match ratio of up to 95%, leading to better epipolar geometry estimation in a fraction of the time required for previous outlier rejection algorithms.

The rest of this paper is organized as follows: In Section 2, we discuss the Oriented Tracks algorithm, including the assumptions leading up to the procedure and how the various parameters are chosen. Section 3 presents the experimental results of our algorithm based on real data taken from a mobile scanning system in the field. These results are compared to those obtained by the mutual best match method. The paper is concluded in Section 4 with a summary of the algorithm and experiments, along with a discussion on future extensions.

2. Oriented tracks

In our application, a sensor package is mounted on a mobile platform—either a vehicle, or a 2 driving-, 2 steering-wheeled robot—such that the recording camera’s optical axis is orthogonal to the platform’s main direction of motion. We assume that the frame rate of the camera is high enough so that inter-frame motion is relatively small (<50 pixels). We also assume that the vehicle’s motion will be restricted by the environment such that motion along or about the camera’s optical axis will be relatively small. For a static scene under these conditions, correspondences that are consistent with the epipolar geometry of the scene will have similar behavior with regards to their motion trajectories—the lines connecting the feature locations in the two images—also called feature tracks.

Fig. 1b shows the feature tracks for a typical sequence. The feature tracks display motion that is consistent with the epipolar geometry as well as trajectories due to mismatches that are not consistent with the camera motion. For example, in the center of the image are motion tracks that are nearly vertical due to mismatches in the intensity
correlation process. In effect, feature pairs of this type are the outliers that limit the efficacy of standard epipolar geometry estimation algorithms. By filtering out all these outlying feature tracks, we can generate a data set similar to that shown in Fig. 1c. In doing so, we will have reduced the input to a smaller set of feature correspondences whose track orientations are consistent with the epipolar geometry—which will increase the efficacy of the epipolar geometry estimation algorithm.

To show that our observation is valid, let us assume that a single camera, with intrinsic parameters \( K \), undergoes a translation. Let an image point in the first image be normalized as \( x = [x_1, x_2, 1] \). Then, the space point's inhomogeneous coordinates are \( X = [X_1, X_2, X_3] = K^{-1}x/X_3 \). From \( x = K[RK']X \), the mapping of \( x \) to its corresponding location in the second image \( x' \) is

\[
x' = K'RK^{-1}x + K't/X_3.
\]

In our case, we are assuming that the motion is constrained to the image plane, thus \( R = I \), \( K = K' \), and \( t = [t_1, t_2, 0]^T \). Under these conditions, (1) reduces to

\[
x' = x + K't/X_3.
\]

The feature track for this pair is \( x' - x = K't/X_3 \). Similarly, the feature track for a second corresponding pair \( y' - y = K't/Y_3 \). Thus, the feature tracks are shown to follow the same direction, and are parallel, scaled by the depth values \( X_3 \) and \( Y_3 \). System noise and inaccuracies in the calibration of the camera—as well as perturbances outside of the image plane that are much smaller than the motion within the plane—may lead to correspondence tracks that are not quite parallel, but this can be handled by defining an acceptance region for which the matches are considered to be correct.

Adam et al. (2001) proposed an algorithm—Rejection of Outliers by Rotations (ROR)—to reject false matches from any two views of the same scene that involves virtual rotations of one of the images to identify some common behavior among the correct feature matches. The basic concept is that under a correct rotation, the inliers will have a common feature track direction, and the outliers will be “shaken away”. A user-defined number of random rotations are performed and for each rotation, the algorithm looks for feature tracks that lie in a common direction. The correct feature tracks are those that lie within a specified angular distance from the common direction, averaged over the best rotations. This algorithm has proven to be effective when applied to general camera motion viewing a static scene.

Under our application, the motion constraints of the mobile sensor platform mean that the correct matches will share a common behavior in their feature tracks for the original image pair, with no rotations needed. With this knowledge, we have designed an algorithm similar to ROR, that requires no virtual rotations of the images and relies only on a single user-specified parameter, as opposed to the five required for the ROR algorithm. Our proposed algorithm is outlined as follows:

---

Fig. 1. Example of track orientations. (a) First image in a video sequence. (b) Second image, showing matched features and their respective track orientations. (c) The same image showing only those tracks with orientation consistent with the epipolar geometry.
Oriented Tracks Algorithm

1. Begin with a set of potential correspondences \( c_i = (a_i, b_i) \) between images A and B.
2. Calculate each feature track direction from the line joining the locations of the corresponding pair.
3. Estimate the probability distribution of the feature track directions.
4. Find the main orientation as the mode of the distribution of feature track directions \( m \).
5. Find the angular distance \( d_i \) from each feature track to the mode \( m \).
6. A feature track is considered consistent with the main orientation if for some acceptance region \( e \), \( d_i < e \).

2.1. Estimating the probability distribution

There are many ways to estimate a probability density function of a random variable. Generating a histogram of the data can give a quick estimate of the distribution, but is subject to errors due to the selection of bin width and interval, and is mathematically unsuitable for a variety of analysis tasks. A better method for estimating the probability density function is to use a kernel density estimator.

The kernel density estimator uses a weighted summation of kernel functions, placed at the observation data, to estimate the density. These kernels can be chosen to have all of the desired mathematical properties (e.g., continuous, differentiable, etc.) and their parameters can be chosen to fit the data in some optimal sense (Silverman, 1986). The kernel estimator with a given kernel \( K \) is defined by

\[
\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K \left( \frac{x - X_i}{h} \right),
\]

where \( h \) is the kernel width, \( n \) is the number of sampled data points, and \( X_i \) is the \( i \)th observation of the random variable \( X \). As the kernel width tends to zero the estimated density becomes a sum of delta functions—the under-smoothed case. As the width becomes large the probability density function is over-smoothed and all detail is obscured.

![Fig. 2. Comparison of probability density estimation using histograms and kernel density estimators for the feature tracks orientations shown in Fig. 1a.](image-url)
Fig. 2 shows the comparison of the probability density estimation of the feature track directions from Fig. 1. A standard histogram (Fig. 2a) is shown beside density functions determined from a weighted sum of Gaussian kernels. Fig. 2b shows the case where the kernel width is determined empirically, while in Fig. 2c the kernel width is optimally chosen using an L-Stage Plug In method (Wand and Jones, 1995).

Recall that we are interested in the main orientation as defined by the feature track direction that has the greatest probability. Thus, we are more interested in the location of the PDFs maximum than its shape. We experimented with a number of kernel width optimization algorithms—e.g., L-Stage Plug-In, Empirical, Bootstrap—and the results showed that for every case, the maxima occurred at the same location—the main orientation. Thus, any of the above kernel width optimization procedures will do.

For our work we have chosen to use the L-Stage Plug In method to estimate the bandwidth of our Gaussian kernel function. This method was chosen for its ease of implementation and its asymptotic minimization of the Mean Integrated Square Error (MISE) between the estimated probability density and the actual density function. The MISE can be approximated as

$$\text{AMISE}(\hat{f}) = \frac{R(K)}{nh^2} + \frac{1}{4h^4} \| \mu_2(K)^2 \| \text{R}(f'').$$

where $R(K)$ and $\mu_2(K)$ are constants that depend on the selection of kernel function $K$, and $R(f'')$ is a constant depending on the unknown density function of the random variable.

The optimal bandwidth w.r.t. this criteria has the closed form solution

$$h_{opt} = \left( \frac{R(K)}{\mu_2(K)^2 \| R(f'') \|^{1/5}} \right)^{1/5}. \tag{5}$$

Silverman’s solution for the optimal bandwidth assumes that the true distribution function can be represented by a normal distribution. Using this assumption in combination with a Gaussian kernel, the optimal bandwidth for kernel density estimation can be computed as

$$h_{opt} = \left[ \frac{4\hat{\sigma}^2}{3} \right]^{1/5}, \tag{6}$$

where $\hat{\sigma}$ is either the sample standard deviation or the standardized inter-quartile range. In practice the smaller of the two is used.

We also investigated the effect of the kernel width optimizer on the choice of the main orientation. Our experiments showed that the differing kernel widths chosen by the various optimizing algorithms cause the peak of the density function to shift minutely. However, in practice, the acceptance window specified for track consistency is significantly larger than the shift in the maxima location between various estimators. Thus, the choice of methods for density estimation has no real impact on the outlier rejection technique.

2.2. Choosing the acceptance region

The parameter controlling the width of the “consistent” region is the only user-selected parameter in this outlier rejection method. If this region is too large, feature tracks inconsistent with the epipolar geometry may be kept in the correspondence set. If the region is too small, the noise inherent in digital acquisition systems will cause consistent feature matches to fall outside the threshold, and thus will remove them from the set of correspondences.

To test the effect the acceptance window width has on the number of track correspondences chosen to be consistent with the main orientation, the window size was allowed to vary from 0 to $\pi/2$ and the number of accepted feature correspondences was recorded. Fig. 3 shows the results of this experiment.
experiment. In Fig. 3a, the results for this experiment are shown for the scene given in Fig. 1. Fig. 3b shows the results from a randomly selected group of 10 scenes.

For a clean data set—one whose density function has a sharp peak at the main orientation and a fairly uniform distribution otherwise—such as the scene shown in Fig. 1, the graph exhibits a sharp increase in accepted feature pairs for a short time, and then the number of additional matches begins to slow down. Fig. 3a shows this behavior, with a “knee” clearly evident. Fig. 3b shows that even scenes imaged with a variety of noise (motion blur, resampling, pixelation, etc.) exhibit a knee-type structure.

Taking the acceptance window width to be just above the knee allows us to retain the most consistent feature tracks, while providing for some system error. The vertical line shown in Fig. 3(b) represents the window size chosen for our experiments in the next section. It has been empirically determined from our test data to provide the best tradeoff between outlier rejection and maximum inliers for the epipolar geometry estimation procedure. As another option, since the window width choice is used to account for errors in the vision system, it could be predetermined by some foreknowledge of the system noise.

Tracks inconsistent with the main orientation—i.e., those feature tracks whose angular distance from the main orientation does not lie within the acceptance window—are determined to be outliers and are filtered out, with the remaining correspondences retained as our new, concise feature set.

3. Experimental results

In order to demonstrate the efficacy of outlier rejection by oriented tracks, we compared the results of our outlier rejection method with the typical mutual best match approach. We chose as data image pairs from a variety of image sequences as shown in Fig. 4. These include images taken from both high definition- and standard video, from our mobile sensing platform. These images contain several effects that cause outliers in the epipolar geometry estimation including: digitization noise, resampling noise, motion blur, motion parallax, glare, shadows, highlights, etc. A Sony Handycam and a JVC GRHD1 were used to collect the standard- and high-definition video, respectively.

The scenes shown in Fig. 4a–c were obtained with a Sony Handycam camcorder. The complexity of the scenes increases from left to right. In Fig. 4a, the Women’s Basketball Hall of Fame (Knoxville, TN) building with strong, regular features is shown. In this image, the motion blur is minimal and there are no shadow or glare effects. Fig. 4b, the Restaurant, introduces large parallax effects, shadows, and significant motion blur. The “For Lease” pair in 4c is further complicated by the addition of glare effects.

Likewise, a progression of complexity for images taken by the High Definition camera is shown in Fig. 4d–f. The Parking Garage shown in 4d has well-distributed features over a rectangular structure, with only a few trees.
Fig. 4e shows a circular Storage Tank with more parallax, motion blur, and vegetation. The University of Tennessee Hill shown in 4f combines motion blur, significant vegetation, minimal regular structures, shadows, and variations in illumination for a very complex scene.

For each image pair, a standard Harris feature detector (Harris and Stephens, 1988) was used to select approximately 1000 critical points. The Harris detector was chosen for its prevalence in the literature and because it has been shown to have good stability properties (Schmid et al., 1998). Then, the best match features in image two were found for the features in the first image. Mutual best match (MBM) was used as the baseline outlier rejection method. Feature track orientations were calculated from the Image 1 → Image 2 feature matches, and the main orientation was determined via the method discussed in the previous section. From this main orientation, the Oriented Tracks (OT) correspondence set was obtained. Epipolar geometry estimation was performed as discussed in (Hartley and Zisserman, 2000) and the number of inliers was determined for both the MBM and the OT correspondence sets. For our experiments, inliers are determined as those features that lie within 2 pixels of the associated epipolar line determined by the geometry estimation procedure. The procedures for feature detection, outlier rejection, and epipolar geometry estimation were implemented in Matlab and run on a 2.4 GHz Pentium processor.

In order to compare the filtering results between our Oriented Tracks method and the Mutual Best Match standard, we performed Harris corner detection and intensity correlation matching for each image pair, and then selected 50 true matches by hand from each image pair. RANSAC was used to robustly estimate the epipolar geometry from these picked matches, and all feature pairs that fell within 2 pixels of their associated epipolar line made up the “True Matches”. Then, both MBM and our method were used to filter all of the feature pairs and a comparison of the results is shown in Table 1, using the same matching process—intensity correlation, with a specified search radius of 50 pixels—for the initial matching of OT as well as the 2-pass matching of MBM. The results show that even for datasets that contain fewer than 40% true matches, our filtering method improved the true/false match ratio to better than 90%. This improvement in the quality of the feature correspondences leads directly to an improvement in the calculation of the epipolar geometry of the scene from feature matching, using RANSAC, in terms of both efficiency (time of execution) and accuracy (more inliers).

The RANSAC procedure used to estimate the epipolar geometry from the filtered set of correspondences does so through a randomly selected sample of feature matches. The random samples chosen can greatly affect the outcome in terms of iterations and inliers. In order to present an “average performance”, we ran the geometry estimation process 1000 times for each test image pair for both MBM and OT filtered correspondences.

Fig. 5 shows the results of those trials. For every image pair, a probability density function of the number of inliers was computed, using the same plug-in method discussed previously. Fig. 5 shows that for every case, our OT method gives a higher number of inliers for the epipolar geometry estimation than the mutual best match approach.

In addition, the plots also show that for the most part the distributions are disjoint, which means a substantial improvement in the number of inliers for our algorithm as compared to mutual best match.
In addition to a higher number of inliers, the OT filtered correspondences also yielded fewer RANSAC iterations, providing a performance increase in speed. Table 2 shows that increase for our demonstration data sets. By presenting the geometry estimation algorithm a better set of correspondences, we reduce the number of iterations by at least half. In the majority of our data sets, this reduction in iterations is at least half, and is often much greater—on the order of 1/10.

Table 2 also shows the comparison of timing results for both our outlier rejection method and the mutual best match method. Our algorithm takes less than a half of a second to reject the outliers, while the MBM method takes over 2 min—due to the fact that an additional batch of intensity correlations must be performed. Also, because our method preserves those correspondences that are consistent with the camera motion, the number of RANSAC iterations required to estimate the epipolar geometry is reduced, providing a speed-up of up to 1 second (4.6 times faster) as compared to the MBM features. In total, our OT method for outlier rejection reduces the computation time by 2 min (a speed-up of over 500 times) per image pair, which is a significant savings when the goal is to process an entire video sequence.

These results show that under the assumed motion constraints, our oriented tracks outlier rejection algorithm outperforms the mutual best match method in terms of reduced computations (improved speed) and has a more accurate epipolar geometry estimation—as ascertained by a greater number of inliers for the epipolar geometry estimation.

3.1. Evaluation of results

Tables 1 and 2 showed that our oriented tracks rejection algorithm outperforms the mutual best match rejection procedure in both the number of outliers removed from the initial dataset, and in the amount of time necessary for the outlier rejection process. Why is this the case?

Mutual best match works by performing two intensity correlation stages. For a feature $P$ in Image 1, a potential match is found by looking for a feature in Image 2 that is the most similar, in terms of intensity correlation—the first correlation stage. Let us call that best match feature $N$. This search can be limited by restricting the search region to a defined radius around the location of $P$. This is how we obtain our original set of feature correspondences.

MBM outlier rejection involves taking the $N$, and searching all features in Image 1 for the best match—the second intensity correlation stage. Now assume that within the search region in Image 1, three features $P$, $Q$, and $R$ are found. If the best match for $N$ is $P$, then the pair of features $(P, N)$ is determined as a correspondence. This can be seen in Fig. 6c. If the best match is not $P$, then the feature pair is rejected, as seen in Fig. 6e.

For an image containing many features, this can be a very costly process in terms of computation time. For each feature a minimum of two intensity correlations has to be performed—one to find the Image 1 — Image 2 matches, and one for the Image 2 — Image 1 matches. In practice, it is not unusual to need five or more intensity correlations per feature in order to accept/reject using MBM.

In contrast, our Oriented Tracks algorithm requires only a single intensity correlation stage—to provide the initial putative correspondences—followed by a linear algorithm for filtering out the mismatches. Fig. 6d and f show the rejection process as a simple comparison of track direction to the main orientation calculated by our algorithm. This leads to a false match rejection method that is computationally efficient. In addition, our method is designed to only accept those features that are consistent with the motion of the vehicle, thereby keeping only those features that best represent the epipolar geometry of the scene. In other words, while both MBM and our method require a single intensity correlation stage to provide possible feature correspondences, the OT algorithm outperforms MBM in terms of speed because it does not need to do the second...
set of intensity correlations, which is a computationally expensive process.

Table 3 presents the computational comparison between our OT false match rejection procedure and MBM side by side. The first two stages of feature detection and feature matching are identical between the two methods. Where they differ is in how they perform the filtering on the potential correspondence output from stage 2. Our OT outlier rejection algorithm is a sequence of linear processes, whereas the MBM approach has another intensity correlation stage, which is an $O(mn)$ process. Even if we limit the correlation to features found within a specified radius of the target pixel, this process is of non-linear complexity. This is why the OT algorithm performs so much faster than that of MBM—it is a simpler process.

4. Summary and conclusions

In this paper, we have discussed a new method for the rejection of outliers as a processing stage between feature matching and epipolar geometry estimation in the scenario of estimating a vehicle's motion from a video sequence. Our outlier rejection is based on the observation that for video captured from a mobile platform, true feature matches evince a common orientation in their frame-to-frame trajectories. We have demonstrated experimentally that this outlier rejection method provides more inliers—features consistent with the epipolar geometry—than the typical mutual best match filtering technique, while reducing the number of computations and halving the expected processing time. Our filtering method can provide true/
Table 2
Timing and iteration results for test image pairs shown in Fig. 4

<table>
<thead>
<tr>
<th>Scene</th>
<th>Algorithm</th>
<th># RANSAC iterations</th>
<th>Outlier rejection (s)</th>
<th>RANSAC (s)</th>
<th>Total (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Women’s Basketball Hall of Fame</td>
<td>Oriented tracks 5</td>
<td>0.22</td>
<td>0.18</td>
<td>0.40</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mutual best match 8</td>
<td>90.74</td>
<td>1.18</td>
<td>92.96</td>
<td></td>
</tr>
<tr>
<td>Restaurant</td>
<td>Oriented tracks 3</td>
<td>0.27</td>
<td>0.08</td>
<td>0.35</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mutual best match 6</td>
<td>97.38</td>
<td>0.18</td>
<td>97.76</td>
<td></td>
</tr>
<tr>
<td>For Lease Sign</td>
<td>Oriented tracks 25</td>
<td>0.20</td>
<td>0.25</td>
<td>0.45</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mutual best match 64</td>
<td>129.64</td>
<td>1.16</td>
<td>131.00</td>
<td></td>
</tr>
<tr>
<td>Parking Garage</td>
<td>Oriented tracks 8</td>
<td>0.33</td>
<td>0.19</td>
<td>0.52</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mutual best match 53</td>
<td>112.08</td>
<td>1.18</td>
<td>113.26</td>
<td></td>
</tr>
<tr>
<td>Storage Tank</td>
<td>Oriented tracks 7</td>
<td>0.33</td>
<td>0.16</td>
<td>0.49</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mutual best match 36</td>
<td>128.88</td>
<td>0.82</td>
<td>130.70</td>
<td></td>
</tr>
<tr>
<td>UT Hill</td>
<td>Oriented tracks 4</td>
<td>0.38</td>
<td>0.09</td>
<td>0.47</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mutual best match 21</td>
<td>195.03</td>
<td>0.53</td>
<td>195.56</td>
<td></td>
</tr>
</tbody>
</table>

* Median time for RANSAC iterations taken from 1000 trials.

Fig. 6. Visual comparison of how MBM and OT work: (a) Features in Image 1, with a region around the target feature \( P \) circled. (b) Features in Image 2, including \( N \), the potential correspondence for \( P \). (c) MBM matching of two features \( P \) and \( N \). (d) OT matching of two features \( P \) and \( N \). (e) MBM rejection of two features \( P \) and \( N \). (f) OT rejection of two features \( P \) and \( N \).
false match ratios of up to 95% from data whose original ratios were less than 50%. Under certain conditions, ratios of 80% have been developed from feature matches that originally contained less than 20% of correct matches.

This algorithm is dependant on assumptions of motion common to Pose from Motion tasks. Specifically, that inter-frame motion cannot have significant motion along the optical axis. In the future, efforts will be made to extend the algorithm so that such motion is allowed. Also, an extension is underway to accommodate scenes where more than one motion is present. Another area for possible improvement is to make the selection of the acceptance window width automatic through data or system analysis.

Acknowledgements

This work was supported by the DOE University Research Program in Robotics under grant DOE-DE-FG02-86NE37968, by the DOD/RDECOM/NAC/ARC Program, R01-1344-18, and by FAA/NSSA Program, R01-1344-48/49.

References


Table 3
Computational complexity of OT and MBM algorithms

<table>
<thead>
<tr>
<th>Feature detection</th>
<th>Feature matching</th>
<th>Filtering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corner detection (Harris)</td>
<td>Image 1 — Image 2 intensity correlation to provide a set of potential correspondences</td>
<td>• Compute feature tracks—(O(n))</td>
</tr>
<tr>
<td>Generate feature track direction PDE—(O(n))</td>
<td>• Cull out feature tracks that do not fall in the acceptance region—(O(n))</td>
<td>• Generate feature track direction PDE—(O(n))</td>
</tr>
<tr>
<td>Oriented tracks</td>
<td>Mutual best match</td>
<td>Left-right consistency checking—(O(n))</td>
</tr>
</tbody>
</table>

ABSTRACT

Three-dimensional models of real world terrain have application in a variety of tasks, but digitizing a large environment poses constraints on the design of a 3D scanning system. We have developed a Mobile Scanning System that works within these constraints to quickly digitize large-scale real world environments. We utilize a mobile platform to move our sensors past the scene to be digitized – fusing the data from cm-level accuracy laser range scanners, positioning and orientation instruments, and high-resolution video cameras – to provide the mobility and speed required to quickly and accurately model the target scene.

INTRODUCTION

The availability of accurate 3D terrain models can be a benefit to a number of applications, including: driving simulators [1], robotic path planning [2-4], terrain-tire interaction [5], and more. In the past, these models were created by graphics artists from a library of pre-constructed primitives, or from a rough measurement skeleton. However, technology has advanced enough that we can now directly acquire 3D models from real world environments. In effect, we can now “digitize reality”.

In the last several years there has been several groups studying how to best acquire these 3D models of real world environments. These research efforts generally fall into one of two categories: image-based methods, where the 3D geometry is inferred from 2D images; and the laser range-based approach, where the 3D geometry is measured directly through laser range scanners.

Image-based methods have been around the longest, and have been utilized for many modeling tasks. The standard method for developing large-scale terrain models is to use stereo techniques with aerial imagery to develop coarse models of the ground below [6, 7]. Shape from Motion techniques such as those used by Pollefeys et al. [8] and Zisserman et al. [9] use the calculated motion of a single moving camera to infer the geometry of the imaged scene, while the MIT City Scanning Project [10] uses spherical cameras and Geographic Information System (GIS) localization to develop models of the environment around them. In general, these techniques give results that have centimeter- to meter-level accuracy.

Recent efforts have also been made in utilizing a more direct method of acquiring 3D models of real world environments user ground-based laser scanning systems [11-13]. These approaches require the acquisition, registration, and merging of a large number of individual scans, and thus are only effective for a small environment, say a block’s worth of data. In order to alleviate this limitation, laser scanning has been combined with mobile platforms [14, 15]. These efforts use orthogonal laser range scanners and scanline matching techniques to identify the motion of the platform during scanning. These techniques work well under semi-planar motion constraints and a scene with regular (i.e., manmade) constructs, but fail in the general case.

We have developed a Mobile Scanning System similar to these methods that utilizes an approach for identifying the vehicle’s motion, similar to that used in aerial terrain scanning [16, 17], that is more flexible in the types of environments and structures that can be digitized. This system combines laser range scanners, high-resolution video cameras, and Global Positioning System (GPS) and Inertial Navigation System (INS) instrumentation to measure the system’s position and orientation (pose). This sensory package is then mounted on a vehicle stretch.
Figure 1, which is moved past the scene to be digitized to provide a geometrically accurate 3D model of the terrain, with the added benefit of high quality texture overlays. The system is designed to be a modular package so that the hardware can be changed to meet the specific needs of the application at hand, while the data processing procedures remain the same.

MOBILE SCANNING SYSTEM

Digitizing large-scale environments is a task that has inherent constraints, depending on the target application. These constraints have to do with the target environment, the structures to be digitized, the size and expense of the equipment to be used, the time available for data acquisition, the purpose for which the models are to be used, etc. Some of these constraints are constant, and some depend on the application — e.g., what resolution is necessary for the model. Perhaps one of the main constraints is the time on site necessary to do the data acquisition. Closing down the Aberdeen Proving Grounds for 3 months in order to digitize a single course is not an ideal solution.

We use high-resolution sensors mounted on a mobile platform to acquire cm-level resolution models. This allows us to quickly digitize the target scene in a variety of environments, under a variety of conditions and varying lighting. Texture overlays can be provided as necessary through the imagery acquired by our high-resolution video cameras. Additional texture overlays can be provided through the use of thermal cameras or chemical sensors. Meanwhile, pose estimation hardware and video cameras are acquiring and calculating the position and orientation information for the mobile platform. This information is used to align the range profiles and texture images and bring them into a common coordinate system.

Figure 2 shows how our Mobile Scanning System digitizes a target scene. As the vehicle moves past the scene to be digitized, the system acquires 3D geometry profiles of the terrain around it while the pose estimation system acquires information about the motion of the platform. Let \([X \ Y \ Z]\) be the global coordinate reference for the digitized scene. This can be either a georeferenced coordinate system or a local equivalent. Let \([x \ y \ z]\) be the laser scanner's internal coordinate system and let the orientation parameters for the scanner be \([\text{roll} \ \text{pitch} \ \text{yaw}]\). Each point \(P\) in the current profile has a 3D identity of \(v_p(x, y, z)\), as seen from the scanner, and the scanner has a 3D identity of \(i(X, Y, Z)\) in the global reference. Thus, the location of \(P\) in the world reference can be found as:

\[
v_p = \ell + R \cdot v_p,
\]

where \(R\) is the rotation matrix generated from the roll, pitch, and yaw values determined from the pose sensing instrumentation.

The Mobile Scanning System essentially can be broken down into 3 main sub-systems: the geometry and texture acquisition sensors; the pose estimation module, and the data fusion and processing system. The first two are hardware systems, while the third is software-based.

SELF-LOCALIZATION

Self-localization, or pose estimation, is a critical part of the Mobile Scanning System as the accuracy of the 3D model is directly related to the accuracy of the estimation of the sensor package’s position and orientation at the time of each acquisition. We have utilized an Inertial Measurement Unit (IMU) to provide orientation and acceleration information for the platform. For absolute positioning, we have integrated a highly accurate differential real-time kinematic (D-RTK) GPS receiver, which is accurate up to 2 cm. To augment the self-localization process, we have also implemented a Pose from Video (PfV) algorithm that utilizes the high-resolution images acquired by our sensing module. Figure 3 shows the hybrid pose estimation system block diagram.

Utilizing the GPS system for positioning occasionally results in lost positional information, due to the loss of GPS signal that occurs when vegetation or manmade structures block or attenuate the satellite signals. Thus, we have incorporated the video self-localization, which
allows us to take advantage of the onboard color texture cameras and computer vision techniques to provide pose estimates in areas where GPS is either not available, or just has a low accuracy.

Hardware (Direct) Self-Localization

GPS data gives us the ability to georeference the data acquired by the Mobile Scanning System. This means that we have an absolute frame of reference to apply to the data — even to data acquired using different scanners on different days. The majority of the localization instrumentation available only provides a relative reference, e.g., using the start location as a zero point and measuring from there. The GPS data allows us to incorporate all the localization information available from the INS, PF, and other systems into a common reference frame, and one that has an absolute meaning. The fusion of the localization data thus involves mainly the computation of the fixed rotations and translations if the various sensors with respect to the global reference, the GPS [18].

Let us denote the position of the platform at a given time as \( \mathbf{p}(t) = [x \ y \ z] \), and its orientation as \( \mathbf{r}(t) = [\psi \ \theta \ \omega] \). Since the acquisition rates are different for the various sensors and we only have discrete time sampled data, interpolation is needed to obtain a continuous path trajectory. For a full 3D motion estimate \( \mathbf{p}_i = [x \ y \ z \ \psi \ \theta \ \omega] \) at time \( t_i \) in the interval \( [t_{i-1}, t_i] \), the displacements and rotations are defined as:

\[
\Delta \mathbf{h}_i = (t_i - t_{i-1}) / (t_i - t_{i-1}) \mathbf{p}_i \\
\mathbf{p}_i = \mathbf{A} + \mathbf{B} \Delta \mathbf{h}_i + \mathbf{C} \Delta \mathbf{h}_i^2 + \mathbf{D} \Delta \mathbf{h}_i^3,
\]

where \( A, B, C, \) and \( D \) and the third order polynomial coefficients given by the tridiagonal system, as per Bartels et al. [19].

The roll, pitch, and yaw parameters for the system are measured by the IMU through a series of gyroscopes and linear accelerometers, as seen in Figure 4. These measurements can provide the instantaneous velocity of the system, as well as the orientation. Because the accelerations are being measured, in a pinch the IMU information can be used to estimate the position of the system as well, through double integration of the measurements. Unfortunately, the errors that accrue at each sampling point eventually degrade the estimate beyond the acceptable bounds of accuracy. When a fixed reference is available — e.g., using the GPS points (when available) as keystones — the positional estimates from the IMU can be accurate, as long as the system is not allowed to drift without a reference for too long.

In our system, the GPS is the only device that provides these absolute points of measurement, bounding the accuracy of the pose estimation subsystem to the accuracy of the GPS data. Under non-ideal conditions, the degradation of the GPS signal increases the uncertainty in the self-localization process. In this case, we can monitor the uncertainty of the hardware-based pose estimation system, and when it falls below a specified threshold, we can switch over to a video-based method as described in the next section.

![Figure 4. Basic Inertial Measurement Unit components and navigation algorithm (adapted) [18].](image-url)

Video (Indirect) Self-Localization

With the advent of inexpensive video cameras, optical navigation is being increasingly utilized on robotic vehicles [20-23]. Using video for navigational purposes involves determining the interframe motion between successive images, and often relies on the matching of feature points from one image to another.

It is known that two images of a static scene are related to each other through their epipolar geometry [24, 25] and epipolar geometry constraints have been used to perform Structure and Pose from Video tasks [26-28]. A robotic vehicle’s pose can be determined from a video sequence by finding corresponding features between adjacent images in the video sequence and using the scene’s epipolar geometry to calculate the position and orientation changes between the two images [29-32].

The pose estimation from video process can be seen in Figure 5. First, distinctive features in an image pair are identified. Next, the features in the first image are matched to corresponding features in the second image. Given a good set of correspondences, the motion state of the camera can be calculated, up to scale, using a two stage motion estimation algorithm. We can then use the onboard laser range finder of our system to get an absolute distance to a known point in the scene, providing the scale, and giving us the full 6 Degree of Freedom (DoF) motion estimate.

The feature detection and matching is an important part of the video self-localization process. There are a number of feature types, detection algorithms, and
Matching schemes available for this task. We have chosen to use the Harris corner detector [33] because of its robustness to noise and its stability performance [34, 35], as well as its prevalence in the literature. Intensity correlation is used for the matching, and combined with our own in-house algorithm for false match removal to improve the performance of the pose computation.

The pose estimation is performed using a robust algorithm similar to that described by Johnson [21]. The approach is based on a probabilistic solution following the well-known RANSAC procedure [24]. A random selection of 7 feature correspondences are used to define the Fundamental Matrix \( F_i \) for the image pair, using Hartley’s 7-point Algorithm [28]. The epipolar error \( e_i \) is computed for all feature tracks, measuring the distance of each feature from its corresponding epipolar line, defined by the computed \( F_i \). If this error is less than that from previous iterations, the current fundamental matrix and its associated mean epipolar error become the best estimate for this two-frame motion. The process is then iterated until convergence.

When the procedure is complete, the best estimate of \( F \) is used to compute the robust \( E \) by removing all features that were considered outliers – epipolar error greater than 2 pixels – and recomputing \( F \) using all the inlying feature matches. Then, the camera’s pre-computed calibration matrix \( K \) is used to calculate the essential matrix \( E \) via

\[
E = K^T F K^T .
\]

The translation and rotation parameters are extracted from \( E \). The output of this motion estimation system is a 5 DoF motion state, with an unknown scale factor \( y \). We use an absolute distance measurement from the onboard laser range scanner to provide \( y \).

**DATA FUSION AND MODEL PROCESSING**

For our target application range, the Mobile Scanning System acquires data in real-time, but processes the data offline. This allows us to meet our parallel goals of minimal time on site and maximal model quality. A typical terrain model can consist of over 200,000 geometry profiles, 1,000,000 position and orientation measurements, and 20,000 high-resolution color images. A rough estimate of the storage capacity required for the raw data alone exceeds 10 gigabytes. Thus, we need to combine the data together in a meaningful fashion, while reducing the amount of memory required to deliver the model’s information to the target application. This procedure consists of the following stages: (1) fusion of the raw data into a single model, and (2) geometry processing to remove the effects of noise, fill holes and remove duplicate data, and provide and adaptive simplification of the model.

**Data Fusion**

Data fusion is carried out in two main stages: combining the localization information to provide position and orientation of the sensor package at each scanning step, and using that pose information to relate the individual laser range profiles and digital images. The localization and scanning instruments are synchronized by a common timing signal. Then, for range profiles we find the corresponding localization information by combining interpolated values for the position and orientation.

As mentioned previously, we use the localization information provided by the video system to provide position and orientation information when the instrumented systems’ quality falls below a given level. Figure 6 shows an experimental comparison of the direct and indirect localization methods. The experimental subject shown here is a building in a small strip shopping center. The dotted line is the position as measured by the GPS. The solid line is the position estimate as made by the video-based system. The breaks in the GPS signal were deliberate, in order to simulate a loss of signal condition. The GPS data available was high-quality, with an accuracy of 2-3 cm.

**Combining the Raw Data**

The pose estimation is performed using a robust algorithm similar to that described by Johnson [21]. The approach is based on a probabilistic solution following the well-known RANSAC procedure [24]. A random selection of 7 feature correspondences are used to define the Fundamental Matrix \( F_i \) for the image pair, using Hartley’s 7-point Algorithm [28]. The epipolar error \( e_i \) is computed for all feature tracks, measuring the distance of each feature from its corresponding epipolar line, defined by the computed \( F_i \). If this error is less than that from previous iterations, the current fundamental matrix and its associated mean epipolar error become the best estimate for this two-frame motion. The process is then iterated until convergence.

When the procedure is complete, the best estimate of \( F \) is used to compute the robust \( E \) by removing all features that were considered outliers – epipolar error greater than 2 pixels – and recomputing \( F \) using all the inlying feature matches. Then, the camera’s pre-computed calibration matrix \( K \) is used to calculate the essential matrix \( E \) via

\[
E = K^T F K^T .
\]

The translation and rotation parameters are extracted from \( E \). The output of this motion estimation system is a 5 DoF motion state, with an unknown scale factor \( y \). We use an absolute distance measurement from the onboard laser range scanner to provide \( y \).

**Diagram**

![Block diagram of self-localization from a video sequence.](image)

**Figure 5.** Block diagram of self-localization from a video sequence.

The pose estimation is performed using a robust algorithm similar to that described by Johnson [21]. The approach is based on a probabilistic solution following the well-known RANSAC procedure [24]. A random selection of 7 feature correspondences are used to define the Fundamental Matrix \( F_i \) for the image pair, using Hartley’s 7-point Algorithm [28]. The epipolar error \( e_i \) is computed for all feature tracks, measuring the distance of each feature from its corresponding epipolar line, defined by the computed \( F_i \). If this error is less than that from previous iterations, the current fundamental matrix and its associated mean epipolar error become the best estimate for this two-frame motion. The process is then iterated until convergence.

When the procedure is complete, the best estimate of \( F \) is used to compute the robust \( E \) by removing all features that were considered outliers – epipolar error greater than 2 pixels – and recomputing \( F \) using all the inlying feature matches. Then, the camera’s pre-computed calibration matrix \( K \) is used to calculate the essential matrix \( E \) via

\[
E = K^T F K^T .
\]

The translation and rotation parameters are extracted from \( E \). The output of this motion estimation system is a 5 DoF motion state, with an unknown scale factor \( y \). We use an absolute distance measurement from the onboard laser range scanner to provide \( y \).

**DATA FUSION AND MODEL PROCESSING**

For our target application range, the Mobile Scanning System acquires data in real-time, but processes the data offline. This allows us to meet our parallel goals of minimal time on site and maximal model quality. A typical terrain model can consist of over 200,000 geometry profiles, 1,000,000 position and orientation measurements, and 20,000 high-resolution color images. A rough estimate of the storage capacity required for the raw data alone exceeds 10 gigabytes. Thus, we need to combine the data together in a meaningful fashion, while reducing the amount of memory required to deliver the model’s information to the target application. This procedure consists of the following stages: (1) fusion of the raw data into a single model, and (2) geometry processing to remove the effects of noise, fill holes and remove duplicate data, and provide and adaptive simplification of the model.

**Combining the Raw Data**

Data fusion is carried out in two main stages: combining the localization information to provide position and orientation of the sensor package at each scanning step, and using that pose information to relate the individual laser range profiles and digital images. The localization and scanning instruments are synchronized by a common timing signal. Then, for range profiles we find the corresponding localization information by combining interpolated values for the position and orientation.

As mentioned previously, we use the localization information provided by the video system to provide position and orientation information when the instrumented systems’ quality falls below a given level. Figure 6 shows an experimental comparison of the direct and indirect localization methods. The experimental subject shown here is a building in a small strip shopping center. The dotted line is the position as measured by the GPS. The solid line is the position estimate as made by the video-based system. The breaks in the GPS signal were deliberate, in order to simulate a loss of signal condition. The GPS data available was high-quality, with an accuracy of 2-3 cm.

**Figure 6.** Comparison of the paths determined by GPS (dotted line) and video (solid line) localization. Note that the gaps in the GPS signal are intentional loss of satellite lock, in order to demonstrate the need for a hybrid localization system. The inner rectangle (dashed line) is a building to be scanned by the system.
Unsurprisingly, the video system deviated from this accurate path, and that deviation was the strongest in the areas of turning — as the feature tracks become less coherent, and yaw-turning and y-translation are highly coordinated in PfV. However, using the GPS data as keystones when available, the pose estimates from video are accurate enough in the short term to be used until the GPS system can come back online.

Once the data fusion of the localization information has been done, the alignment of the individual laser range profiles and color images is a straightforward process. Pose estimates local to the individual range scans are interpolated to find the best estimate of the position and orientation of the scanner at the time of acquisition. A rigid transformation is then applied to the coordinate system of the scanner to bring the data into the world reference frame, as defined by the pose instrumentation.

Processing the Data

Once all the raw data has been fused, we have a rough 3D model available for processing. There are a wide variety of processing algorithms available for 3D models. We have focused our efforts on 3 main areas: hole filling and redundant geometry removal; noise removal; and adaptive simplification. The strategies followed were designed to remove as many of the effects of system noise on the scanned data as possible, without degrading the information that is present in the scanned data.

Redundancy comes from areas that are scanned multiple times due to the motion of the mobile platform. Figure 7 shows another case of redundant scans, where the mobile platform is turning into the laser scanner’s field of view. In this case, the vehicle’s turning angle is large enough so that the scanning planes for sequential range profiles cross each other before they reach the target object. This causes geometry that was scanned in a previous profile to be scanned again. These cases are identified by determining when the object in question is beyond the intersection distance $d_{cross}$ and removing those crossing scans. The critical distance $d_{cross}$ is calculated as

$$d_{cross} = \frac{\Delta t}{\sin(\Delta \theta)}$$

where $\Delta t$ is the inter-frame translation distance and $\Delta \theta$ is the turning angle.

As with any system that measures real data, our Mobile Scanning System has a noise component associated with it. This noise corrupts the data, and derives from the inherent inaccuracies of the laser range scanners as well as the uncertainty associated with each pose estimate. For many applications the noise reduction stage may not be necessary, but for most it is desirable to remove or at least reduce these noise effects — especially in areas where the noise effects are on the same order of magnitude as the local detail, obscuring the desired information.

The filter that we currently use to remove the noise is based on the assumption that the system noise is an additive Gaussian noise and the characteristics are based on those of the scanner. These characteristics were determined experimentally within our lab. Ongoing research is geared towards a more elegant and automated solution to removing system noise from the model, while preserving and even enhancing the regions of small-scale details.

For many applications, the full-resolution 10GB model may not be necessary. For example, you are creating an a priori model of terrain that will be driven over by a small robot. The robot will only operate in a small region of that terrain, but for other simulation reasons you wish to have at least a gross model of the surrounding areas. In this instance, you would be satisfied to have all objects outside the mission area processed at a lower resolution than those interacting with the robot. In order to have the generality of a system that can acquire high-resolution geometry information and provide low-resolution models as needed, we have developed a multi-resolution processing scheme [36]. This scheme defines the operations needed to display/store/process the generated 3D model in various levels-of-detail. Figure 8 shows this process on our test model at the original resolution, and at 25% and 2.5% of the original resolution. Notice that the geometry is increasingly simplified, and storage/computations reduced, but the overall appearance of the model remains the same.

EXPERIMENTAL RESULTS

We have used our Mobile Scanning System to acquire 3D models for a number of different environments, in a variety of configurations. For this paper we will discuss 2 different configurations of the Mobile Scanning System. The first will be the system we use to develop models of building and other above-ground-level structures. These models are suitable for populating
driving simulators, aiding robotic path planning and manipulation tasks, etc. The second configuration is a working example of how the Mobile Scanning System can acquire detailed models of the terrain over which it moves – i.e., 3D models of the ground itself.

For the structure scanning configuration, we have used a mobile platform – in this case a van – with the sensor package mounted on the roof for maximum field of view and minimal pedestrian impact. The sensor package consists of: a Leica GS500 D-RTK GPS system, with a 2 cm positional accuracy and an acquisition rate of 10 Hz; an Xsens MT9 IMU, which gives orientation and acceleration information at a rate of 100 Hz with sub-degree angular accuracy; a JVC GR-HD1 high definition camera that acquires color images at a rate of 30 frames per second; and a Riegl LMS-Z210 laser range scanner that acquires the 3D geometry at 10,000 points per second (corresponding to 20 profiles/sec), and has a 5 cm accuracy at up to 350 m.

Figure 9 shows the results of the Mobile Scanning System along a 2-block subset of a larger 2 mile stretch of data. Figure 9a shows an aerial view of the area, with the vehicle’s path superimposed. Figure 9b shows the 3D geometry in point cloud form for this subset. The focus of this model is the Women’s Basketball Hall of Fame, a Knoxville, TN landmark, whose textured 3D model is shown in Figure 9c. This dataset contains over 700,000 triangles but required only 3 minutes to acquire.

Figure 10 shows the process on a smaller, but more complex scene. In this case, the Mobile Scanning System was driven around a shopping strip, acquiring the 3D geometry from all 4 sides of the building, to generate a more complete model. This model consists of 500,000 triangles and was acquired in 5 minutes.

Figure 11 is a large-scale structure model of the Downtown West Shopping Mall in Knoxville, TN. The Mobile Scanning System was driven around the entire mall, acquiring the geometry and color information to combine into a detailed 3D model. The dataset is rather large, consisting of over 5,000,000 data points, 5,000 color images, and 48,000 pose measurements. However, only 18 minutes of scanning time was required to acquire the data, which encompassed over 1.5 km of scanning. The textured 3D model shown in Figure 11 is a small subset of the entire dataset, which is shown in reduced point cloud form.

Figure 12. A multi-resolution representation of a building model with the textured model shown above the wire frame and point cloud representations. The level of detail decreases from left to right.
In addition to the structure scanning system, we are also currently working with a scanning platform designed to acquire the terrain (ground) the system is moving over. This system uses a downwards looking laser range scanner to digitize the terrain. The process for data fusion and model processing is the same as the previous system due to the modularity that went into the system design process.

The micro-scale system utilizes an IVP Ranger to acquire high-resolution, high-accuracy 3D models of roads and other surfaces. This level of resolution is useful for tasks such as pavement inspection. Figure 12a shows an example patch of road surface, with the corresponding path the scanning system followed superimposed. Figure 12b shows the corresponding digitized 3D surface for that surface patch. Notice the highly detailed nature of the data, where even the texture of the asphalt is evident. It took approximately 45 seconds to acquire this 25 cm x 50 cm dataset, which contains over 50,000 data points.

The macro-scale terrain scanning system uses a SICK LMS200 system to acquire terrain models on a larger scale. The LMS200 has a 180° field-of-view and a depth accuracy on the order of 1 cm. With this field-of-view and a working range of up to 8 m, this scanner provides terrain models appropriate for terrain-tire interaction, vehicle dynamics simulation, etc.

There are two versions of the downward looking system that we are currently using. One is a micro-scale system that is capable of measuring the geometry very densely (approximately 1mm between data points) with an accuracy on the order of 5 mm, at the cost of having a very limited field-of-view and a high amount of raw data to process. The other system uses a laser range scanner designed to scan larger objects, similar to the Riegl scanner used in the previous discussion.
well as the ditches, cars, and other objects of interest in the terrain model. The model contains over 3,000,000 data points, and took approximately 6 minutes to acquire.

Figure 13. Large-scale terrain digitization of a ‘double-U’ loop on a gravel road and parking lot. (a) Aerial view of the region that was digitized, with the scanning path superimposed. (b) 3D surface of the digitized terrain with a zoomed in view of the raw data in point cloud form.

CONCLUSION

We have presented experimental results from the UTK Mobile Scanning System, as applied to terrain modeling. The system we have developed uses laser range scanners, high-resolution digital cameras, GPS, and INS mounted on a mobile platform to quickly digitize 3D models of real world environments. Our scanning system can acquire kilometers’ worth of data in real time, and process the data offline to obtain accurate 3D models of the target scenes. It has been developed modularly, so that the system can be reconfigured according to the application with minimal amount of effort and changes to the processing.

The results we presented showed that our system is capable of scanning both terrain and the accompanying structures, at the desired level of resolution. The structure scanning configuration is appropriate for populating simulators and providing a priori models for robotic tasks. The micro-scale terrain scanning configuration is appropriate for inspection tasks of relatively small region, while the macro-scale terrain scanning configuration is appropriate for developing models for use in simulation, dynamics testing, etc. The configurability of the system is such that a larger scale system could be implemented using an aerial platform to extend the working range of the system even more, if desired.

Current work is underway to extend the capabilities and robustness of our Mobile Scanning System. Some of these efforts include: improving the quality and robustness of the pose estimation algorithms, adaptive regularization of the surface for noise removal, algorithmic enhancing of small-scale detail regions in the presence of noise, and improving the automation of the data processing tasks. Also under consideration is the development of a fore-and-aft scanning system, that can model vehicle-soil interaction by providing before and after models and comparing the two to validate existing and upcoming theoretical terrain impact simulations.

ACKNOWLEDGMENTS

This work was supported by the DOE University Research Program in Robotics under grant DOE-DE-FG02-86NE37968, by the DOD/RDECOM/NAC/ARC Program, R01-1344-18, and by FAA/NSSA Program, R01-1344-48/49.

We would also like to thank Ph.D. students Sreenivas Sukumar and Sijie Yu for their data on road profiling.

REFERENCES


Hybrid Self Localization for a Mobile Robotic Platform for Indoor and Outdoor Environments

Brad Grinstead, Andreas Koschan, Mongi A. Abidi

The University of Tennessee
334 Ferrie Hall
1508 Middle Dr.
Knoxville, TN 37996, USA
{bgrinstein, akoschan, abidi}@utk.edu

INTRODUCTION

There is an increasing demand in DOE applications for robotic systems to perform tasks in a variety of environments. This requires that the robots have the ability to navigate within several types of environments. Here work is presented on a self-localization system that allows a robot to move between indoor and outdoor environments.

DESCRIPTION OF ACTUAL WORK

Traditionally, robots operating in outdoor environments use direct pose estimation – measuring their location and orientation directly through the use of instruments – while robots operating in an indoor environment commonly use pose estimation from video, an indirect method. For a system that may be traveling back and forth between indoor and outdoor environments, neither of these systems in optimal. Hardware solutions can provide extremely accurate information, but some components are restricted in their avenues of operation – for example, GPS does not get accurate data indoors. Pose from video solutions are often degraded by noisy images, the absence of recognizable features, and moving objects.

This paper describes a hybrid approach, where the accuracy of the individual self-localization estimates are evaluated and a mixture of the two models is used to provide better localization of the robot.

The system we are using to test and evaluate the hybrid self-localization package is The University of Tennessee’s Mobile Scanning System, which consists of a variety of sensors – laser range scanners, video cameras, GPS, and INS – mounted on a mobile platform, in this case a van, and used to digitize large scale environments, as seen in Fig. 1.

The intended use of the Mobile Scanning System is to digitize a variety of environments, both indoors and outdoors. Among the outdoor environments are urban sites, which degrade GPS positioning capability to a point where it is no longer useful, similar to indoor environments.

The direct self-localization for the Mobile Scanning System is performed using a high-end Leica GS500 Differential GPS with an accuracy of up to 2 cm, and an XSens MT9 IMU with sub-degree accuracy and a measurement rate of 100 Hz. The data from these devices are combined via a filter that utilizes the redundancies present to provide an optimal estimate of the robot’s pose. Since the GPS is the only sensor that makes absolute position measurements, the overall accuracy of the instrumented approach is dependant on the accuracy of the GPS. In the case of the platform moving under vegetation, near buildings, or indoors, the GPS positional quality degrades to the point where the uncertainties involved in estimating the robot’s pose become too large. When this happens, the system switches over to the video localization mode.

In this navigation mode, the robot uses features present in the image sequence captured by the system’s onboard video camera to perform pose estimation from video (PfV). A standard PfV technique is used to develop the Euclidian transformation between 2 subsequent images, with the laser range scanner providing the scale factor.

A diagram of the hybrid self-localization system can be seen in Fig 2. As the motion state estimate from the direct localization filter is over the PfV (indirect) block. The switchover decision is based on the variance associated with the motion state estimate. As
this value gets larger than a pre-set threshold, the system switches over to the visual method of localization.

Fig. 2. Hybrid self-localization system, based on GPS, INS, and visual data.

RESULTS

To demonstrate the efficacy of this hybrid self-localization system, a number of tests were performed. The results are being evaluated in terms of numerical accuracy, but Fig. 3 shows a visual example of the data from a sequence taken in downtown Knoxville, TN. Fig 3a shows an aerial view of the scanning area, with the vehicle’s localization estimate superimposed.

Fig 3. Women’s Basketball Hall of Fame sequence. (a) Aerial view of scanned area with localization information superimposed. (b) Scanned geometry (shown in low resolution) for the road segment highlighted in blue. (c) 3D model of Hall of Fame complete with texture.

ACKNOWLEDGEMENTS

This work was supported by the DOE University Research Program in Robotics under grant DOE-DE-FG02-86NE37968, by the DOD/RDECOM/NAC/ARC Program, R01-1344-18, and by FAA/NSSA Program, R01-1344-48/49.
A comparison of pose estimation techniques: Hardware vs. video

Brad Grinstead¹, Andreas Koschan, Mongi A. Abidi

Imaging, Robotics, and Intelligent Systems Laboratory
334 Ferris Hall
Department of Electrical & Computer Engineering
The University of Tennessee
Knoxville, TN 37996

ABSTRACT

Robotic navigation requires that the robotic platform have an idea of its location and orientation within the environment. This localization is known as pose estimation, and has been a much researched topic. There are currently two main categories of pose estimation techniques: pose from hardware, and pose from video (PIV). Hardware pose estimation utilizes specialized hardware such as Global Positioning Systems (GPS) and Inertial Navigation Systems (INS) to estimate the position and orientation of the platform at the specified times. PIV systems use video cameras to estimate the pose of the system by calculating the inter-frame motion of the camera from features present in the images. These pose estimation systems are readily integrated, and can be used to augment and/or supplant each other according to the needs of the application. Both pose from video and hardware pose estimation have their uses, but each also has its degenerate cases in which they fail to provide reliable data. Hardware solutions can provide extremely accurate data, but are usually quite pricey and can be restrictive in their environments of operation. Pose from video solutions can be implemented with low-cost off-the-shelf components, but the accuracy of the PIV results can be degraded by noisy imagery, ambiguity in the feature matching process, and moving objects. This paper attempts to evaluate the cost/benefit comparison between pose from video and hardware pose estimation experimentally, and to provide a guide as to which systems should be used under certain scenarios.

Keywords: Mobile mapping, data fusion, pose estimation, GPS, inertial measurement, self-localization, video pose estimation

1. INTRODUCTION

It is difficult to develop an autonomous robot that can successfully navigate and interact with real-world environments. The problem encompasses a wide spectrum of issues: chassis design, sensor selection, path planning, obstacle avoidance, etc. The problem's complexity increases with the complexity or variety of environments the robot needs to operate in. The predictability of indoor environments has made them the focus of a majority of the research efforts on robot self-localization¹⁻⁶ - with their planar floors, regular features, etc. However, there is also a need for robot pose estimation in outdoor environments. Even more complex are those tasks that require the robot to operate in both indoor and outdoor environments.

There are two methods for robotic self-localization that are typically used. Direct methods use sensors to directly measure the robot's position and orientation to the world around it. These sensors often include Global Positioning Systems (GPS) for determining absolute position, and Inertial Navigation Systems (INS) with bearing sensors for absolute orientation. Indirect methods infer the robot's pose from the change in observance of the world around it. The most common indirect method is Pose from Video (PIV), where the robot's pose is determined by observing the change in image features from subsequent frames imaged by an onboard video camera.

¹ bgrinste@utk.edu  phone: (865) 974-9213; fax: (865) 974-5459
Typically, indoor applications utilize indirect methods, combining PIV with dead reckoning and other methods to navigate the robot. Outdoor applications often involve direct pose estimation, as GPS can give extremely positioning information with and accuracy higher than other localization methods\(^5\). For applications that involve moving between indoors and outdoors environments, neither of these systems is optimal. Hardware solutions can provide extremely accurate information, but some components are restricted in their avenues of operation - for example, GPS does not get accurate data indoors. Pose from Video solutions are often degraded by noisy images, the absence of recognizable features, and moving objects.

This paper describes an experimental approach to comparing these self-localization techniques in an attempt to quantify when one outperforms the other. This information can be used in the development of a hybrid-self localization scheme, that combines direct and indirect methods to improve the pose estimate of the mobile robot.

The system we are using to test and evaluate the hybrid self-localization package is the Mobile Scanning System developed here at The University of Tennessee, which consists of a sensor package - laser range scanners, video cameras, GPS, and INS - mounted on a mobile platform, in this case a van, and used to digitize large scale environments, as seen in Fig. 1. The intended use of the Mobile Scanning System is to digitize a variety of environments, both indoors and outdoors. Among the outdoor environments are urban sites, which degrade GPS positioning capability to a point where it is no longer useful, similar to indoor environments.

![Fig. 1. Mobile scanning system consisting of vehicle, range scanner, GPS, INS, and video camera](image)

2. SELF-LOCALIZATION

For navigation, we have equipped our mobile platform with an XSens MT9 Inertial Measurement Unit (IMU), which provides orientation and acceleration information. For absolute positioning, we have included a Leica GS500 series differential real-time kinematic (D-RTK) GPS receiver, which is accurate up to 1 cm. We have also included a JVC GR-HD1 high definition color video camera for texturing the acquired 3D models and performing pose estimation from video. The hybrid pose estimation system block diagram can be seen in Fig. 2.

The video self-localization is needed to address the limitations of GPS performance in certain environments, notably indoors and urban canyons, where tall building and steel structures may block or attenuate the GPS signals, leading to inaccurate positioning. The addition of a video camera to the localization module allows us to take advantage of the regular features present in such environments, and use computer vision techniques to provide pose estimation in areas where the GPS is not accurate.
2.1 Hardware Self-Localization

The advantage of using GPS data (when available) is the ability to geo-reference the data acquired by the mobile scanning platform. Most of the localization instrumentation available is INS, odometry, compasses, etc. They measure data with respect to their own internal reference frames. GPS allows us to collate all of these various data into a common reference frame, and one that has physical meaning - i.e., grid location on the global surface. Thus, self-localization using GPS involves integrating all the various data into a common coordinate system, which involves mainly the computation of the fixed rotation and translations of the different sensors with respect to the global reference, the GPS1.

Let us denote the position of the platform at a given time as \( p(t) = [x, y, z] \), and its orientation as \( \psi(t) = [\varphi, \theta, \psi] \). Since the acquisition rates are different for the various sensors and we only have discrete time sampled data, interpolation is needed to obtain a continuous path trajectory. For a full 3D motion estimate \( P_k = [x, y, z, \varphi, \theta, \psi] \), at time \( k \) in the interval \([t_i, t_{i+1}]\), the displacements and rotations are defined as:

\[
\eta_k = (k - t_i)/(t_{i+1} - t_i)
\]

\[
P_k = A + B \eta_k + C \eta_k^2 + D \eta_k^3,
\]

where \( A, B, C, \) and \( D \) are the third order polynomial coefficients given by the tridiagonal system, as per Bartels et al.16.

The INS measurements of angle and acceleration are prided through a series of linear accelerometers and gyroscopes as seen in Fig. 3, and are used to provide an initial velocity vector reference and to coordinate the pose estimation instruments in a common fashion. They can even be used to provide positioning information in and of themselves. However, the errors which accrue at each sampling point eventually grow outside the acceptable bounds of accuracy. The GPS measurements (when available) act as keystones, holding the pose estimation system within acceptable accuracy.

The GPS sensor is the only sensor that provides these fixed points of measurement, thus the accuracy of the entire system is bounded by the quality of the GPS data. As the GPS signal quality deteriorates, the pose estimation uncertainty increases. Thus, additional pose estimation data is needed in these cases. The uncertainty of the system can be monitored, and the self-localization can be switched over to a video-based method as the uncertainty crosses over a specified threshold.

---

168 Proc. of SPIE Vol. 5804
2.2 Video self-localization

Video-based pose estimation has been used frequently to provide pose estimation for mobile robots in a variety of indoor and outdoor environments\textsuperscript{13,14}. There are a number of motion estimation methods available for video systems. Here we have chosen to use a two-frame method similar to that used by Johnson and Matthies\textsuperscript{15}. To obtain the 6 degrees of freedom (DOF) position and orientation estimates, our PIV algorithm incorporates an absolute distance measurement from the laser scanner present on the mobile scanning system. The block diagram for the PIV system can be seen in Fig. 4.

![Diagram of PIV system](image)

**Fig. 3.** Basic inertial system components and navigation algorithm (adapted\textsuperscript{9}).

**Fig. 4.** Block diagram for video self-localization.

For each image pair in the sequence, discrete features were detected in the images, sifted to find the corresponding matches between the images, and then used to determine the motion estimate of the platform between the views. There are a number of feature detectors available for this task, and we have chosen to use the standard Harris corner detector\textsuperscript{16} for its robustness to noise and stability performance\textsuperscript{16,17}.

The Harris features are used as the starting locations for window-based intensity correlation matching. This matching process is typically an $O(N^2)$ operation, but it can be ameliorated by reducing the search space. We do this by restricting the range of search to those features in the second that lie within $n$ pixels of the given feature in the first image. This radius is determined based on the assumed range of velocities of the mobile platform, as compared to the acquisition rate of the camera. The resulting correspondences are then filtered using an algorithm based on common behavior of the correct matches, similar to that developed by Adam, et al.\textsuperscript{18}.

Fig. 4 shows a subset of an image with the motion tracks of observed features superimposed. Notice that while the majority of the feature tracks move in the same direction, some of them exhibit completely different behaviors. Experience has taught us that the different behavior of these features is due to the presence of noise - either through motion in the scene or by false matches from the correlation stage. Removal of these features increases the video localization system's robustness to noise, and provides a more accurate estimate of the platform's pose.
The pose computation of the camera is performed by using a robust motion estimation algorithm similar to that described in Johnson's work. The approach is a probabilistic solution following the well-known RANSAC procedure, introduced by Fischler and Bolles. Seven feature correspondences are randomly selected from the set of all feature tracks. This subset defines a Fundamental Matrix $F_i$ for the image pair, using Hartley's 7-point Algorithm. Next, the epipolar error $e_i$ is computed for all feature tracks, measuring the distance of each feature from its corresponding epipolar line, defined by the computed $F_i$. The mean epipolar error for this subset is less than that from previous iterations, the current fundamental matrix and its associated mean epipolar error become the best estimate for this two-frame motion. The process is then iterated until convergence.

When the procedure is complete, the best estimate of $F$ is used to compute the robust $F$ by removing all features that were considered outliers - epipolar error greater than 2 pixels - and recomputing $F$ using all the inlying feature matches. Then, the camera's pre-computed calibration matrix $K$ is used to calculate the essential matrix $E$ via

$$E = K^T FK,$$  

(2)

The next stage of motion calculation is to extract the translation and rotation parameters from $E$. It can be shown that the translation vector $T_e$ is the solution to $\min\|E T_e\|$ - the unit eigenvector with the smallest eigenvalue of the matrix $EE^T$. The sign of the translation vector can be determined using the constraint that the imaged scene must lie in front of the camera.

Determined the solution to the rotation matrix $R$ involves solving

$$\min\|R^T [-T_e] - E^T\|,$$  

(3)

which can be efficiently solved using the quaternion form. The output of this motion estimation system is a 5 DOF motion state, with an unknown scale factor. We can then use an absolute distance measurement from the onboard laser range scanner to provide $\gamma$.

### 3. PRELIMINARY RESULTS

As previously mentioned, our experimental setup consists of a mobile platform - in this case a van - with a sensor package mounted on top, consisting of: GPS, INS, video, and a laser range scanner. The GPS system we are using is a Leica GS500, in differential RTK mode, with a positional accuracy of up to 2 cm and an acquisition rate of 10 Hz. The inertial sensor is an Xsens MT9 IMU, with a sub-degree accuracy and an acquisition rate of 100 Hz. The laser range scanner is a Leica LMS-Z210, with a 350 m operating range, and is capable of acquiring range data at a rate of 10000 points/sec (corresponding to a 21 Hz profile acquisition rate). A JVC GR-HD1 high definition camcorder acquires color data at a rate of 30 frames/sec.

The experimental subject shown here is a building in a small strip shopping center. This building is ideally situated to get the best GPS data possible, with nothing in the area to block or attenuate the GPS signals. To simulate a loss of lock in this environment, we artificially blocked antenna access for the GPS, reducing the confidence in the GPS position estimate below acceptable thresholds. Fig. 5 show the resulting path of vehicle traversal, estimated by both the instrumented, and the video-based localization approaches. The vehicle traveled around the building (blue dashed lines), and acquired 3D geometry form the laser scanner. Using the GPS system (red dotted line), the data was good - ~2.3 cm accuracy in the areas of clear signal, and no data when the antenna was cut off - with a noticeable trend to the positioning by video data (black solid line). Unsurprisingly, the video data was off by the most in the areas of turning, as feature tracks became less coherent, and more noise was introduced in the feature matching process.
Fig. 5. Comparison of the paths determined by GPS (red dotted) and video (solid black) localization. Note that the gaps in the GPS signal are intentional loss of satellite lock, in order to demonstrate the need for a hybrid localization system. The inner rectangle (dashed blue line) is a building to be scanned by the system.

Fig. 6 shows the orientation angles (pitch, roll, and yaw) determined by the video system (light blue) as compared to the measured values from the inertial sensor (dark blue), georeferenced to the GPS system. The rotations about the cameras’ x- and y-axes are very small, as expected, and the differences between the video estimates and the measured values are fairly small. However, for the yaw angles, the drifting trend in the video data is again evident, showing that for a naive system such as the one implemented here, video self-localization over large distances exhibits a strong accumulated error effect.

Using the path estimates from the GPS/INS/Video system, we can align the laser range data acquired by the system to get the 3D model shown in Fig. 7. Here, the path estimates from the video were used in place of the missing data 6 as opposed to augmenting the results, as will be discussed later. The geometry was then processed with the techniques described in a previous paper. It is noteworthy that the video data we used for the pose estimation process can also be used to provide a texture overlay for the 3D model, yielding a model that is accurate not only in geometry and scale, but also in color.

4. CONCLUSIONS

We have presented some preliminary experimental results on quantifying the difference in accuracy between high-end hardware based motion estimators and localization efforts from video streams. Our preliminary results show that under the best of cases, the instrumented approach will easily outperform the video localization. However, in those cases where the GPS drops out, there is a definite need for an auxiliary pose estimator, and the video data provides localization that is accurate enough until the GPS system’s accuracy comes back into acceptable bounds.

It perhaps a bit unfair to do a quantitative analysis of the two systems at this time, due to our naive video pose estimator. Using only the two-frame approach simplifies matters, but the accumulated errors become increasingly large over time. Incorporating a bundle adjustment module into the video localization procedure can alleviate these effects. Additionally, the PIV approach is computationally intensive, but improvements in performance can come from a number of sources,
including code optimization, region of interest searching, and processing only when the PIV system is needed - i.e., after the GPS system has dropped out due to accuracy concerns.

Further work on this project will involve improving the PIV system, and providing a quantitative analysis of the accuracies inherent to each system. Additionally, further experiments will be done, providing a wider range of operating areas and the systems performance in these areas.

Fig. 6. Comparison of orientation angles from video and instrumented approaches. Note that the orientations about the x- and y-axes are nearly identical between the two approaches, but the yaw angles vary quite a bit due to inaccuracies in the video processing.

Fig. 7. Qualitative results of the self-localization of our Mobile Scanning System, showing the system output when the data from the laser range scanner is combined with the localization modules. On the left is the raw geometry in point set form, while the textured model is shown on the right.
ACKNOWLEDGEMENTS

This work was supported by the DOE University Research Program in Robotics under grant DOE-DE-FG02-86NE37968, by the DOD/DECOM/NAC/ARC Program, R01-1344-18, and by FAA/NSSA Program, R01-1344-48-49.

REFERENCES

Fast Digitization of Large-Scale Hazardous Facilities

Brad Grinstead, Andreas Koschan, Mongi A. Abidi

Imaging, Robotics, and Intelligent Systems Laboratory
334 Ferris Hall
Department of Electrical & Computer Engineering
The University of Tennessee
Knoxville, TN 37996
{bgrinste, akoschan, abidi}@utk.edu

Abstract – Robotics technology has become a major interest for groups that deal with hazardous environments. Of special interest is the conversion from telepresence robotics to more automated approaches. In order to facilitate this conversion, we have developed a system for the fast acquisition of large-scale environments to develop the a priori information needed for prediction and performance optimization. This system collects ground-level range scans and combines them with high-resolution digital imagery and positioning and orientation information to provide detailed 3D models of the target environment. The fusion of these data provides models suitable for a variety of robotics-oriented tasks, such as visualization, sensor placement, and robotic path planning.

I. INTRODUCTION

Over the last 10 years, there has been a large push to apply robotics technology to problems involving hazardous environments. These environments contain chemicals or radiation that make having a human presence a risky endeavor. Currently, many of these facilities employ mobile or fixed robotics with telepresence capability. In other words, the robots examine and manipulate objects in the hazardous area, with a human being running the controls from a safe location. However, there is currently a push to further automate these systems by deploying robotic systems with some form of innate intelligence. This intelligence is usually problem specific and depends greatly on the a priori information available. The more information we have about an environment, the more effective we can build our robotic systems.

Consider, for example, a small mobile robot designed to navigate a decommissioned nuclear facility and perform some decontamination tasks. If we have information beforehand about expected obstacle location and terrain type, we can more effectively design the navigation software. Or how about remotely monitoring a warehouse where certain chemicals are stored. With a complete 3D model of the environment, we can place our sensors to get maximum coverage of the facility with minimal equipment. Also, if there is a breach we can provide the 3D location to monitoring personnel. These are just a few of the many applications that can be enhanced with a complete model of the hazardous environment in question.
mobile platform available, the process of acquiring the data and combining them to form a complete representative model is the same.

![Figure 1. Mobile scanning system consisting of vehicle, range scanner, GPS, INS, and video camera](image1)

Our approach is to acquire cm-level resolution models from laser range scans, with the scanner mounted on a moving platform. This allows us to acquire 3D geometry in a variety of environments with varying lighting conditions. If textured models benefit the application, we also capture high-resolution digital imagery in conjunction with the laser range data. Meanwhile, pose estimation hardware is acquiring positioning and orientation information for the vehicle. This information is used to align the individual profiles and bring them into a common coordinate system.

The mobile scanning system can be decomposed into two parts: the geometry sensing module, and the pose sensing module. The geometry sensing is accomplished through the use of laser range scanners, which acquire 3D geometry in a profile fashion as the mobile platform is moved past the scene. The pose is sensed by a number of instruments, including a Global Positioning System (GPS) and an Inertial Measurement Unit (IMU). The initial experimental system consisted of a Riegl LMS-Z210 laser range scanner, a Leica GPS, and an Xsens IMU. The entire sensing package is mounted on a vehicle that moves the equipment past the scene to be digitized.

![Figure 2. System diagram for the mobile scanning system showing profile acquisition and vehicle motion.](image2)

II. DATA ACQUISITION

Digitizing a large environment – such as a large building complex – poses a number of constraints in the design and implementation of a 3D scanning system. Some of the constraints are application dependent (e.g., what resolution is required of the model?), while others apply to any application. Perhaps the main concern is to be able to acquire representative data with minimal impact to the environment being scanned. After all, how many program directors would want a 3D model of their facilities, at the cost of having to completely close down the entire area for two weeks?

Figure 2 shows how a facility might be scanned using this system. During motion the laser range scanner acquires a profile of the geometry present in the scene, while the pose hardware captures the vehicle’s motion. Let \([X \ Y \ Z]\) be the global coordinate reference for the digitized scene. This can be either a georeferenced coordinate system or a local equivalent. Let \([x \ y \ z]\) be the laser scanner’s internal coordinate system and let the orientation parameters for the scanner be \([\text{roll} \ \text{pitch} \ \text{yaw} \ ]\). Each point \(P\) in the current profile has a 3D identity of \(v_P(x, y, z)\), as seen from the scanner, and the scanner has a 3D identity of \((X, Y, Z)\) in the global reference. Thus, the location of \(P\) in the world reference can be found as:

\[
V_P = \ell + R \cdot v_P, \tag{1}
\]

where \(R\) is the rotation matrix generated from the roll, pitch, and yaw values determined from the pose sensing instrumentation.

In our setup, the scanner’s location in the global reference is determined via a Leica differential GPS system that is capable of positional accuracy of up to 2 cm and an update rate of 10 Hz. The orientation parameters for the system are acquired using an Xsens IMU, with an update of 100 Hz and sub-degree accuracy. IMUs utilize gyroscopes and linear accelerometers for acquiring orientation and positioning information, as seen in Figure 3. Due to the drift errors inherent in inertial systems, the positional accuracy degrades significantly over long periods of time, but using the GPS position as a...
keyframe where available, we can provide positional information with the IMU even in areas where GPS fails (e.g., urban canyons, under foliage, etc).

III. DATA PROCESSING

When the acquisition is complete, a large amount of data has been acquired by our system. A typical 1 mile stretch of city streets generates a 3D model with approximately 50 million triangles, 1 million position and orientation measurements, and 20 thousand high-resolution color images. This comes to a rough, uncompresed, total of approximately 10 gigabytes of data for a single mile’s worth of models. Obviously, some data processing is necessary in order to make the models manageable for robotic tasks.

The first step is to fuse the geometric and pose information. Recall that each data type is acquired at a different sampling rate. The range data is acquired at 10000 points/sec, the positional data is received at 10 Hz, and the orientation information at 100 Hz. In initial system the data fusion is accomplished through an initial synchronization method and then cubic spline interpolation is used to find the pose for every range profile. In the future, this method may be upgraded to an Extended Kalman Filter to improve robustness and accuracy.

Next, the geometry data is processed in order to remove the effects of noise, duplicate data, etc. Noise is removed by using an averaging filter on the range values, based on the known accuracy of 5 cm in the scanning direction. Where the vehicle has stopped (at stop signs or red lights), and as it turns corners, the scene geometry may be sampled more than once, causing duplicate data. These areas are identified and the duplicate points are removed from the dataset.

Many applications benefit from having photorealistic models of the target environment. To meet these needs we have incorporated high-resolution color imagery as an optional sensing modality. Within each color image, the appropriate texture coordinates are assigned based on the fixed transformation between the color camera(s) and the laser range scanner. The resulting model is a good approximation of the 3D scene.

For many applications, the full-resolution 10GB model may not be necessary. For example, you are creating an a priori model of an environment that will be navigated by a small robot. The robot will only interact with objects no more than 1 m from the ground. In this instance, you would be satisfied to have all objects greater than 1 m from the ground processed at a lower resolution than those interacting with the robot. In order to have the generality of a system that can acquire high-resolution geometry information and provide low-resolution models as needed, we have developed a multi-resolution processing scheme. This scheme defines the operations needed to display/store/process the generated 3D model in various levels-of-detail.

IV. RESULTS

For our experimental system, we used a Riegl LMS-Z210 laser range scanner to acquire the geometry. This scanner has a maximum range of 350 m and an 80° vertical field of view, with a range accuracy of ± 5 cm and an acquisition rate of 10,000 points/sec. The vehicle position was determined via a differential GPS system with an accuracy of 2 cm at 10 points/sec, and the orientation was measured with an XSens MT9 IMU at 100 Hz. For each of the presented results, the vehicle was driven past the scene at normal driving speeds.

Figure 4 shows the results of acquisition for a small strip shopping center. The complete model was acquired in 2 minutes and consists of over 3 million triangles. Color imagery was combined with the geometry to provide a photorealistic model. Figure 4a shows a sample digital image of the scene that was digitized (Figure 4b). Figure 4c is a closer view of the digitized model.

Figure 5 demonstrates the use of the system on a larger scale. The data shown here is a 2-block subset of a dataset that encompassed 2 miles of scanning. The data shown here was acquired in 3 minutes and contains over 5
million triangles. The focus of these results is on the Women’s Basketball Hall of Fame, a prominent landmark in Knoxville, TN. Figure 5b shows the geometry acquired for this subset (as a cloud of points).

Figure 4. Strip mall sequence (a) High-resolution digital image of a portion of the building. (b) Low resolution 3D geometry in point cloud form. (c) High resolution textured 3D model of one side of the complex.

Figure 5. Women’s Basketball Hall of Fame sequence. (a) Aerial view of scanned area. (b) Scanned geometry (shown in low resolution) for the road segment highlighted in blue. (c) 3D model of Hall of Fame complete with texture.

Figure 6 shows how this level-of-detail representation works. Here we see the example model rendered at various resolutions. On the left, the model is rendered at full detail – typical of center view, close range objects. The center model shows a resolution appropriate to objects that fall in the mid-range of viewpoint and distance. Notice that there are a much smaller number of triangles here, but the visual appearance of the model is still close to the original. On the right is a low resolution version of the model, fit for applications requiring only a gross representation of the object. The levels-of-detail are stored in a data structure so that the application-appropriate model can be retrieved on demand.

V. CONCLUSIONS

We have presented an experimental system for quickly acquiring 3D digitized models of large-scale environments. The system – based on laser range scanners, pose estimation hardware, and high-resolution digital cameras all mounted on a mobile platform – has been successfully used to capture models of large buildings and city streets. The system we have demonstrated performs all of the tasks necessary for basic 3D model digitization, including: geometry acquisition, color image acquisition, data processing, texturing, and multi-resolution processing.

Future goals for this project involve extending the pose estimation technique to be able to handle more environment types (rough roads, off-road capability, etc.), as well as improving the generated models by developing methods to handle moving foreground objects (pedestrians), windows, and vegetation. In addition, we intend to add the option for downward-facing laser scanners to capture terrain geometry.

ACKNOWLEDGEMENTS

This work was supported by the DOE University Research Program in Robotics under grant DOE-DE-FG02-86NE37968, by the DOD/TACOM/NAC/ARC Program, R01-1344-18, and by FAA/NSSA Program, R01-1344-48/49.
Figure 6. A multi-resolution representation of a building model with the textured model shown above the wireframe representation. The level of detail decreases from left to right.

REFERENCES


Developing detailed *a priori* 3D models of large environments to aid in robotic navigation tasks

Brad Grinstead*, Andreas Koschan, Mongi A. Abidi

Imaging, Robotics, and Intelligent Systems Laboratory
334 Ferris Hall
Department of Electrical & Computer Engineering
The University of Tennessee
Knoxville, TN 37996

ABSTRACT

In order to effectively navigate any environment, a robotic vehicle needs to understand the terrain and obstacles native to that environment. Knowledge of its own location and orientation, and knowledge of the region of operation, can greatly improve the robot’s performance. To this end, we have developed a mobile system for the fast digitization of large-scale environments to develop the a priori information needed for prediction and optimization of the robot’s performance. The system collects ground-level video and laser range information, fusing them together to develop accurate 3D models of the target environment. In addition, the system carries a differential Global Positioning System (GPS) as well as an Inertial Navigation System (INS) for determining the position and orientation of the various scanners as they acquire data. Issues involved in the fusion of these various data modalities include: Integration of the position and orientation (pose) sensors’ data at varying sampling rates and availability, Selection of “best” geometry in overlapping data cases, Efficient representation of large 3D datasets for real-time processing techniques. Once the models have been created, this data can be used to provide a priori information about negative obstacles, obstructed fields of view, navigation constraints, and focused feature detection.

Keywords: Mobile mapping, laser range scanning, data fusion, pose estimation, GPS, inertial measurement

1. INTRODUCTION

Over the course of the last decade, there has been a large push to apply robotics technology to a variety of autonomous navigation and mapping functions. These applications include: Mapping hazardous environments,1 developing robotic couriers,2 remote and autonomous inspection of parking facilities,3, 4 etc. However, the intelligence of the robot – as pertaining to path determination – is dependant on the initial assumptions about the environment.

In order to effectively navigate through any environment, a robotic system must be aware of its own position and orientation (pose) – where am I? – as well as have knowledge of its environment of operation – what else is out there?. Providing an *a priori* model containing relevant geometry and texture overlays can greatly improve the navigational intelligence of any robotic system.

Methods for creating such models have been studied for many years. In the beginning, 3D maps were created from the designers’ knowledge of the area of interest and basic geometrical shapes – i.e., constructive geometry using blocks, cylinders, cones, etc. Later, stereo methods were used,3 often in combination with aerial imagery for large, outdoor environments.6, 7 Other image-based methods for developing structure maps include Shape from Motion techniques,5, 8

* bgrinste@utk.edu; phone: (865) 974-9213; fax: (865) 974-5459

561
and the MIT City Scanning project. In general, these methods generate results that have centimeter- to meter-level accuracy, often at the cost of extensive manual guidance and intervention.

There have also been recent efforts at acquiring 3D models of large areas from ground-based laser range scans. These approaches require the registration and integration of scans from a number of different viewpoints, and are thus only feasible for a small region – say, a few buildings. In order to overcome this limitation, mobile platforms have been combined with laser range scanning. These mobile systems use a scanline matching technique for pose estimation, which works well in planar environments, but fails in the case of general motion. We have developed a system, based on these technologies, that is more flexible in terms of the type of environments that can be scanned.

In this paper, we describe a system to quickly acquire the data needed to generate a priori 3D models of the environment in question from a ground-level digitization system. The system that we are developing combines the geometry acquisition of laser range scanners, color imagery from high-resolution digital cameras, and accurate pose information from instrumentation. These sensors are mounted on a mobile platform (Fig. 1) that is moved past the scene in order to acquire the data necessary to build the a priori model – thus digitizing the real world environment. The output of our system is a detailed 3D model of the desired environment that is geometrically accurate, with the added benefit of having texture overlays, as needed, that can be used to enhance the performance of the intended navigation algorithm. The system is designed as a modular package so that the specific instruments and sensors can vary according to the needs of the application.

The rest of this paper is organized as follows: Section 2 discusses the mobile scanning system in general, and the specific instruments and methods we have used to acquire the necessary data for model building; Section 3 covers the model building process; Experimental results are shown in Section 4 and the paper concludes in Section 5.

2. DATA ACQUISITION

The digitization of a large environment poses a number of constraints in the design and implementation of a 3D scanning system. These constraints have to do with the size and expense of the equipment used, the time available for data acquisition, the purposes for which the model is to be used, etc. Some of these constraints are application dependant – i.e., what resolution is needed for the model? – and some apply to any application. One of the main concerns is the ability to acquire the necessary data with minimal impact on the environment to be scanned. Closing down a military base or school for 2 weeks in order to create a 3D model for your robotic courier is not exactly an ideal solution.

Our approach is to acquire the necessary 3D data from laser range scans at cm-level resolution from laser range scans, with the scanner mounted on a mobile platform. This provides the ability to quickly acquire the 3D geometry of a
variety of environments with varying lighting conditions. Mounting the scanners on a vehicle matches two of our system constraints: (1) by mounting the scanners on top of a vehicle, we minimize the interference of ground-level moving bodies (e.g., pedestrians, other cars, etc.) in our models, and (2) the moving platform keeps up with traffic flow, thus minimizing our impact on the sampled environment. We can also provide a number of texture overlays, depending on the application, such as: color imagery, radiation maps, thermal signatures, and any other sensing modality that may be applicable. As the scanning devices are acquiring the geometry of an environment, pose estimation hardware is sampling the vehicles position, orientation, speed, etc. to determine where in the global reference the scanned data belongs.

The mobile scanning system can be decomposed into two main parts: the environment scanning module, and the pose sensing module. Environment sensing is performed with laser range scanners to acquire 3D geometry, and cameras and other sensors to provide the texture overlays. Pose estimation is performed by a number of units, including a Global Positioning System (GPS) and an Inertial Measurement Unit (IMU).

Our current experimental setup consists of a Riegl LMS-Z210 laser range scanner, a JVC GR-HD1 High-Definition camcorder, a Leica 500 series Differential Real-Time Kinematic (RTK) GPS system, and an XSens MT9 IMU. The sensor package is mounted on a vehicle that can be driven past the scene to be digitized in order to build the 3D model.

Fig. 2 shows how a facility is scanned using our system. As the vehicle moves past the facility, the laser range scanner acquires profiles of the 3D geometry while the pose estimation hardware determines the vehicle’s motion. Let \([X \ Y \ Z]\) be the global coordinate reference for the digitized scene. This can be either a georeferenced coordinate system or a local equivalent. Let \([x \ y \ z]\) be the laser scanner’s internal coordinate system and let the orientation parameters for the scanner be [roll pitch yaw]. Each point \(P\) in the current profile has a 3D identity of \(v_p(x, y, z)\), as seen from the scanner, and the scanner has a 3D identity of \((X, Y, Z)\) in the global reference. Thus, the location of \(P\) in the world reference can be found as:

\[
V_p = \xi + R \cdot v_p,
\]

where \(R\) is the rotation matrix generated from the roll, pitch, and yaw values determined from the pose sensing instrumentation.

![Fig. 2. System diagram for the mobile scanning system showing profile acquisition and vehicle motion.](image)
Using an RTK GPS system to identify its location in a global reference, the system is capable of positional accuracy of up to 2 cm and has an acquisition rate of up to 10 Hz. GPS, though a common positioning solution, has a few drawbacks. A single system can only attain a real-time kinematic accuracy of a few meters and loss of satellite signals due to canyon (building) shadow can cause loss of positioning data. The accuracy can be improved by using two systems in concert to perform differential positioning, while shadow effects can only be overcome with additional information.

The roll, pitch, and yaw parameters are acquired using an IMU with an update rate of 100 Hz and sub-degree accuracy on orientation. The IMU used in our system acquires orientation and positioning information from a collection of gyroscopes and linear accelerometers, as seen in Fig. 3. Integration of the measured accelerations gives orientation, as well as vehicle velocity and position. However, due to drift errors the positional accuracy of the IMU degrades significantly over time. This can be corrected by using the GPS measurements as key points, providing the ability to obtain positional information even in areas where GPS fails for short amounts of time (i.e., under foliage canopy, in urban canyons, etc.).

Fig. 3. Basic inertial system components and navigation algorithm (adapted).
\[ d_{\text{cross}} = \frac{\Delta t}{\sin(\Delta \theta)}, \]  

(2)

where \( \Delta t \) is the inter-frame translation distance and \( \Delta \theta \) is the turning angle.

Our processing algorithm also attempts to remove the effects of system noise on the generated 3D model. For many cases, this stage may not be necessary, but for those applications that require visualization of the robot’s environment, it is necessary to remove the system noise effects. For instance, see the scanned building segment shown in Fig. 5, where the noisy, textured model is shown along with the denoised version. The filter that we use to remove the noise is based on the assumption that the system noise is an additive Gaussian noise and the characteristics are based on those of the scanner. These characteristics were determined experimentally within our lab.

For many applications, the full-resolution 10 GB model may not be necessary. In fact, for the application of a priori model creation for robotic navigation the full-resolution model is often inhibitive detailed. For example, assume that the robot in question will only interact with objects no more than 2 m off of the ground. In this case, all objects at a height greater than 2 m can be processed at a lower resolution than those at ground level. In order to have a general system that can acquire high resolution 3D models and then provide lower resolution models as needed, we have used the multi-resolution processing scheme of Roy.\(^7\) This scheme provides a data-driven set of operations needed to display, store, and process the acquired 3D model at various levels of detail.

Fig. 4. Redundant geometry scanning case, where the vertical scanning planes of sequential scans intersect.  
Fig. 5. Noise removal process for a building segment.  
(a) Textured 3D model with noise.  
(b) Untextured model with noise removed.  
(c) Denoised 3D model with texture.

4. EXPERIMENTAL RESULTS

Our experimental system consists of a Riegl LMS-Z210 laser range scanner mounted on the roof of a van to acquire the geometry of the target scene. The scanner is mounted as in Figure 2, with the scanning plane perpendicular to both the main direction of motion of the vehicle and the ground. The Z210 has a maximum range of 350 m and an 80° vertical field of view. Its accuracy is ± 5 cm and it can acquire 3D geometry at a rate of 10,000 points/sec. The vehicle’s position was determined by a Leica System 500 differential RTK GPS system. The positional accuracy is approximately 2 cm, and it acquires positional data at a rate of 10 Hz. The laser range scanner orientation was sensed using an Xsens...
MT9 IMU, with an angular accuracy of better than 1°, with an update rate of 100 Hz. For each of the presented results, the mobile platform was driven past the scene to be digitized at normal driving speeds.

Fig. 6 shows the results for a typical robotic systems application where an *a priori* model of the scene has been acquired. The environment is the parking lot of a local supermarket. Suppose that the supermarket’s parking lot is to be monitored robotically, and inspected regularly. Navigation and functionality can be enhanced in this case with an accurate 3D model of the area of operation. In this case, we have scanned the building and parking lot, and have inserted computer generated cars into to acquired 3D model to simulate the robot’s identification and concurrent mapping functionality. The data shown here is from a scan of a 700 m long shopping strip, and was acquired in 2 minutes. The unsimplified model contains 5 million triangles. In this case, color imagery was combined with the 3D geometry to provide a photorealistic model. Fig. 6a shows a single digital image of the scene that was digitized. Fig. 6b shows the acquired geometry, pose corrected using the methods discussed previously. The complete texture 3D model of the front of the complex is shown in Fig. 6c, with an augmented model showing simulated results from a concurrent monitoring and identification robotic system shown in Fig. 6d.

Fig. 7 shows the fast digitization process on a larger scale, for an environment containing more difficult geometry. Present in the scene are: small-scale details such as signs, pedestrians, etc; non-linear scanning path; trees; moving cars; and traffic issues like stoplights, lane shifts, etc. This particular dataset comes from a well-known building in Knoxville, TN – the Women’s Basketball Hall of Fame. The data shown is a 2-block subset of a larger sequence encompassing over 2 miles of scanned geometry. Fig. 7a shows the scanning path for this particular dataset superimposed on an aerial view of the region in question. The geometry in low resolution point cloud form is shown in Fig. 7b, clearly showing the trees, curved roads, elevation changes, signs, etc. that make this a challenging environment to digitize. The full resolution photorealistic model is shown in Fig. 7c.
The multi-resolution processing to create levels-of-detail is shown in Fig. 8. Here, the model first seen in Fig. 5 is processed and rendered at various resolutions. On the left, the full resolution model is suitable for close-up viewing, small scale robot interactions, etc. The middle view shows a lower resolution model that is suitable for mid-range visualization and minimal interactivity. Notice that the geometry is greatly simplified here, with only 25% of the data of the original model, but the visual appearance of the model is very close to the original. The lowest resolution model is shown on the right. This model is only good for applications that require only a gross representation of the object. This resolution is good for providing an initial environment map for a mobile mapping and inspection system that will not be interacting directly with the provided model. These levels of detail are stored in a data structure so that the appropriate model for the application can be retrieved from storage on demand.

5. CONCLUSIONS

We have presented an experimental system for the fast acquisition of 3D digitized models of large-scale environments. Our system uses laser range scanners, high resolution digital cameras, and pose estimation hardware mounted on a mobile platform to acquire 3D models of buildings and terrain. Using our mobile sensing platform, we can acquire kilometers of geometry in just minutes, processing the data offline to obtain 3D models of facilities of a resolution appropriate to the task at hand. The system presented performs all of the basic tasks require to generate 3D models that can be used as the \textit{a priori} input to a robotic system for path planning, navigation, intelligent response, etc.

Future goals for this project involve extending the pose estimation technique to incorporate Pose from Video information as an additional pose estimation modality in a more robust framework — such as an Extended Kalman Filter pose estimator. We also intend to investigate methods of improving the acquired models through such techniques as: adaptive
regularization to improve the geometry, hole filling methods, vegetation modeling, and handling small-scale foreground objects such as signs and pedestrians. Forward and rear facing scanners are currently being added to our system to evaluate terrain effects and to estimate the vehicle’s impact on the terrain – i.e., tread marks left in soft soil.

ACKNOWLEDGEMENTS

The authors would like to thank Michael Roy for lending his implementation of the multi-resolution processing algorithm he developed to generate the level-of-detail models.

This work is supported by the University Research Program in Robotics under grant DOE-DE-FG02-86NE37968, by the DOD/RDECOM/NAC/ARC Program, R01-1344-18, by the US Army under grant Army-W56H2V-04-C-0044, and by FAA/NSSA Program, R01-1344-48/49.

REFERENCES
