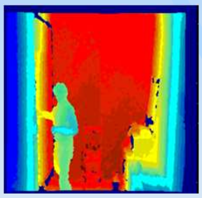
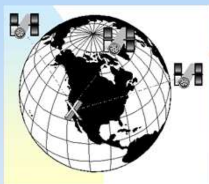
Personal Navigation and Indoor Mapping: Performance Characterization of Kinect Sensor-based Trajectory Recovery

[1]Charles TOTH, [1]Dorota BRZEZINSKA, USA
[2]Allison KEALY, Australia, [3]Guenther RETSCHER, Austria
[1]Department of Civil, Environmental and Geodetic Engineering
Satellite Positioning and Inertial Navigation (SPIN) Lab
The Ohio State University
[2]Melbourne School of Engineering
[3]Vienna University of Technology

toth.2@osu.edu





XXV FIG Congress 2014
Kuala Lumpur, Malaysia ♦ 16-21 June 2014

Outline

- **Introduction**
  - FIG 5.5 Ubiquitous Positioning System (indoor, collab-nav)
- **2D/3D optical ranging and tracking**
  - Concept
  - Feature extraction and matching
    - SIFT (2D – feature based)
    - ICP (3D – shape based)
- **Kinect sensor**
  - Characteristics, test configuration
  - Data collection
- **Performance results and evaluation**
  - 2D image-based trajectory reconstruction
  - 3D image-based trajectory reconstruction
  - Combining 2D and 3D imagery
- **Conclusions**

Introduction

Navigation and mapping trends:

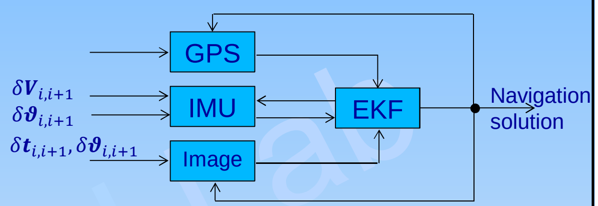
- GPS-level indoor navigation performance is expected
- Sensor integration (multisensory systems)
- Collaborative navigation
- Indoor mapping and personal navigation (growing demand)

Objectives

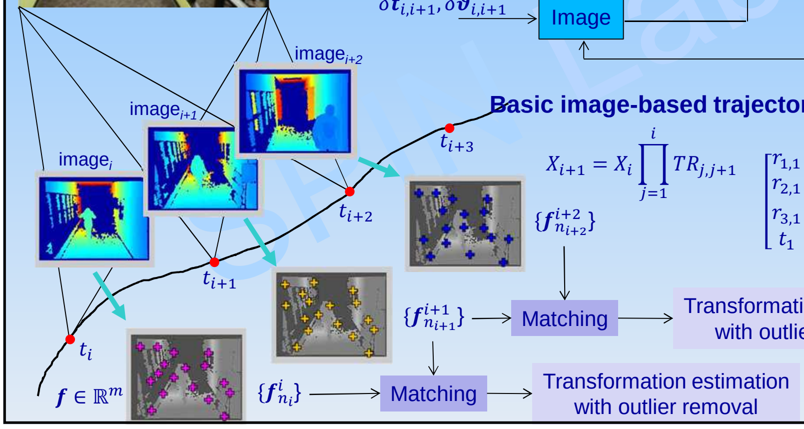
- To use low-cost sensors for indoor navigation and mapping purposes
- To assess the performance of using a low-cost 2D/3D imaging sensor, performance evaluation of a component of a multisensory system (error budget formation)
- To perform simultaneous indoor navigation and mapping of unknown environment (without a priori information of the surveyed environment)

2D/3D optical ranging and tracking

Sensor integration-based trajectory estimation:



Basic image-based trajectory estimation:



$$X_{i+1} = X_i \prod_{j=1}^{i} TR_{j,j+1}$$

$$TR_{i,i+1} = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} & 0 \\ r_{2,1} & r_{2,2} & r_{2,3} & 0 \\ r_{3,1} & r_{3,2} & r_{3,3} & 0 \\ t_1 & t_2 & t_3 & 1 \end{bmatrix}$$

$C_{i+1} = sRC_i + T$

$\delta t_{i,i+1}, \delta \vartheta_{i,i+1}$

2D/3D optical ranging and tracking

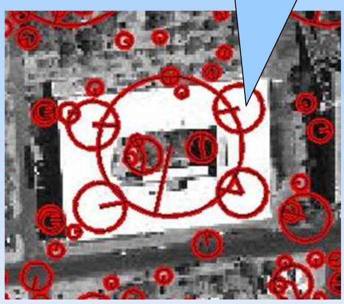
Notes:

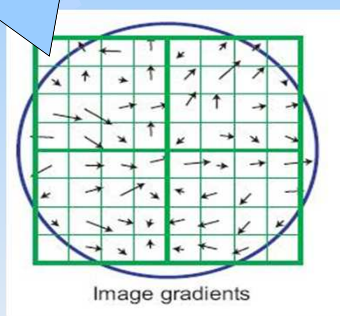
- Images can be 2D and/or 3D, typical matching combinations
  - 2D image to 2D image (image-based navigation)
  - 3D image to 3D image (terrain-based navigation)
  - Combined 2D and 3D image matching
- Features can be points, linear features, surfaces, volumes, etc., typically characterized by a higher dimensional feature vector
- Matched features can be
  - Tie or conjugate features
  - Landmarks, targets; position information of the feature is known in some frame
- Full transformation estimation from 2D images is not possible, as the scale is unknown; note using stereo camera configuration, the scale is known
- Matched features must be filtered for outliers
- Navigation solution (estimate) allows for limiting search space for feature matching (also, helps GPS processing after an outage, etc.)
- There are many ways to include image-based information in EKF, including full or partial changes between epochs, using landmark coordinates in the EKF state vector, etc.

SIFT

SIFT Feature
(4 parameter vector)
1. (X) Location
2. (Y) Location
3. Orientation
4. Scale

SIFT Descriptor
(128 parameter vector)
- Sum of Magnitudes in region of the SIFT Feature
- 4x4 bins reflect sum in each region
- Taken at 8 orientations
- 4x4x8 = 128 parameter descriptor

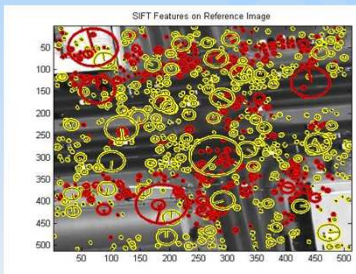




D. G. Lowe, 'Distinctive image features from scale-invariant key points,' IJCV, vol. 2, no. 60, pp. 91-110, 2004.

SIFT matching of various imagery

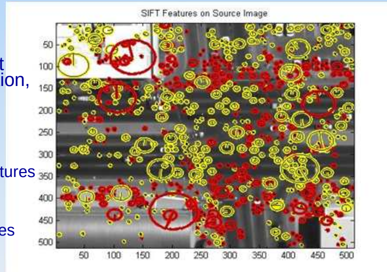
Left image: 199 keypoints **35 matched points** **Right image: 224 keypoints**



Circles represent location, orientation, and “size” of descriptor

Yellow circles
- ❖Unmatched features

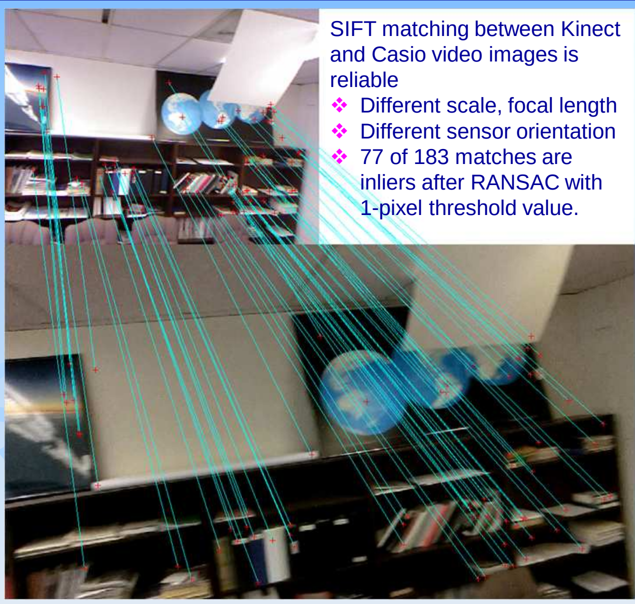
Red circles
- ❖Matched features



Kinect and Casio video image matching

Kinect 2D

Casio P&S



SIFT matching between Kinect and Casio video images is reliable
- ❖ Different scale, focal length
- ❖ Different sensor orientation
- ❖ 77 of 183 matches are inliers after RANSAC with 1-pixel threshold value.

Iterative Closest Point method (ICP)

The ICP algorithm finds the best correspondence between two surfaces (point sets, point clouds) in 3D by iteratively determining the translations and rotation parameters of a 3D rigid body transformation

The ICP algorithm works in three phases:

1) Establish correspondence between pairs of features based on proximity (for each point in *D* compute the closest point in *M*)
2) Estimate the rigid transformation that best maps the first member of the pair onto the second and then

$$\min_{(R,T)} \sum_i \left\| M_i - (RD_i + T) \right\|^2$$

http://en.wikipedia.org/wiki/Point_set_registration

   where *R* is a 3*3 rotation matrix, *T* is a 3*1 translation vector and subscript *i* refer to the corresponding points of the sets M (model) and D (data).

3) Apply that transformation to all features in the first structure, repeat steps 1-2 until convergence is reached

Kinect™ sensor

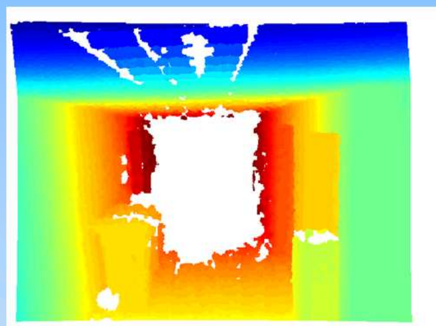
RGB-D camera: passive RGB + active IR

Properties	Parameters
Interface	USB (12 VDC)
2D sensor (RGB camera)	VGA. 640x480; SXGA. 1280x1024
3D sensor (structured IR)	640x480 and 320x240
Frame rate	30 Hz
Operating range	0.8 to 4 m [a]
Range resolution	12 bits
FOV	57° x 43°
Software tools (Microsoft)	SDK. Windows 7. Visual Studio 2010 Enterprise. and DirectX
Open source	Large user group. variety of drivers and tools in C++ and Matlab

[a] It can be extended by ambiguity resolution up to 10 m

Kinect™ data

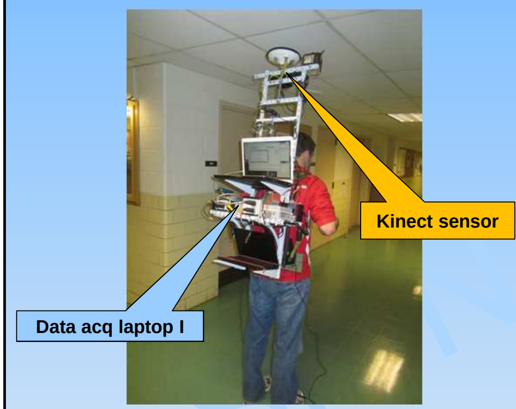
RGB image

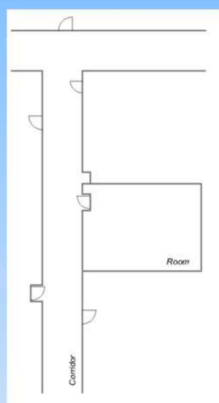


Depth image

- IR camera has a smaller field of view than RGB camera
- Depth image has voided areas (where 3D reconstruction failed due to object characteristics and range)
- Ranging accuracy is around a 1% of the range, rather stable
- Both sensor are calibrated (individual and inter sensor)

Test area





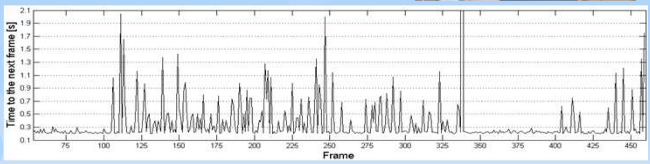
Approaches to sensor Kinect trajectory reconstruction:

- ICP matching of 3D images
- Matching 2D SIFT features from 2D images
- Combining 2D and 3D methods
- Based on sensor trajectory reconstructing object space (colored point cloud)

Data acquisitions

- Nominal data acquisition frequency: 5/30Hz
- Images resolution (RGB and D): 640 x 480
- UWB network used as a reference
- Several runs by different persons
- Office room and hallway scenarios
- Generally straight path with several turns including U-turns

Test 1:
- Average rate: 4.5 FPS
- Frames analyzed: 460
- Significant gaps in image coverage



Test 1:
- Average rate: 26.8 FPS
- Frames analyzed: 3416
- No significant gaps in image coverage

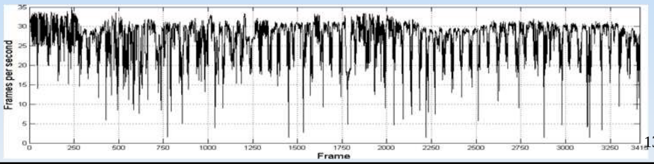
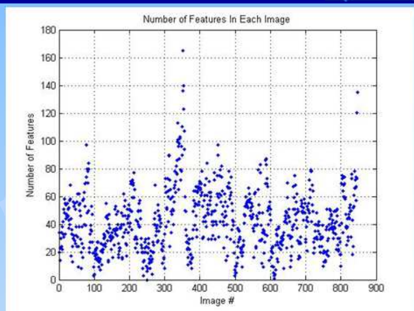
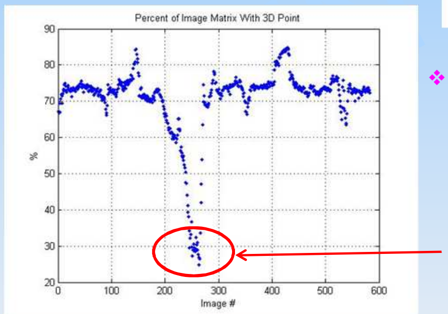


Image characterization

- Image-to-image feature matching using SIFT from the Kinect's 2D camera
  - Low quality sensor
  - Small number of SIFT features (~42)
  - Features may be unreliable or incorrect
  - The distribution of SIFT features may provide bad geometry
  - Incorrect/unstable calibration can lead to unreliable SIFT points



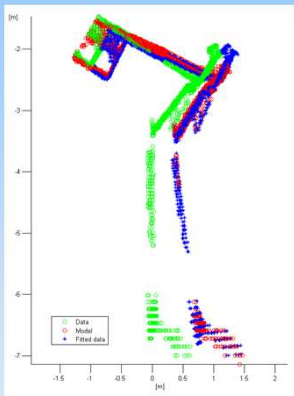


- Image-to-image matching using ICP based on the Kinect's 3D camera
  - Modest quality sensor
  - About 70% of image matrix contains a corresponding 3D point (640 x 480)
  - The distribution of 3D points varies
  - Areas with high levels of ambient IR caused by large windows in the hallway have poor reconstruction

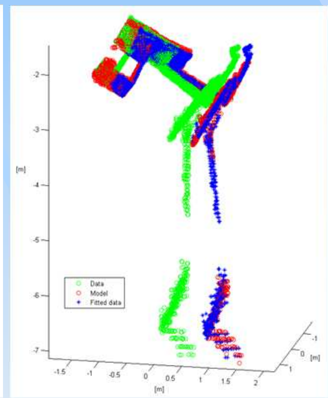
3D image based trajectory reconstruction (ICP)

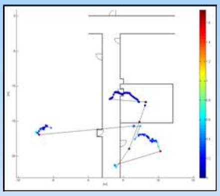
Iterative closest point (ICP) algorithm was used for determining rotation and translation (3+3) parameters

2D view

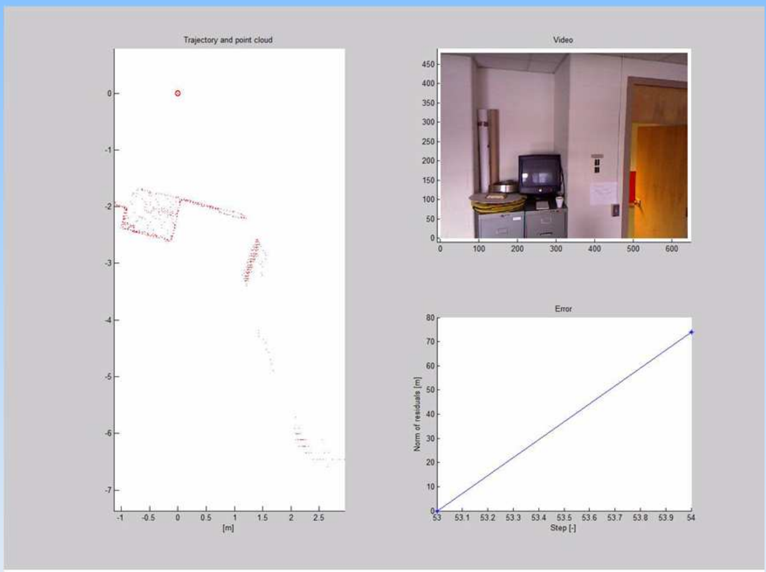


3D view

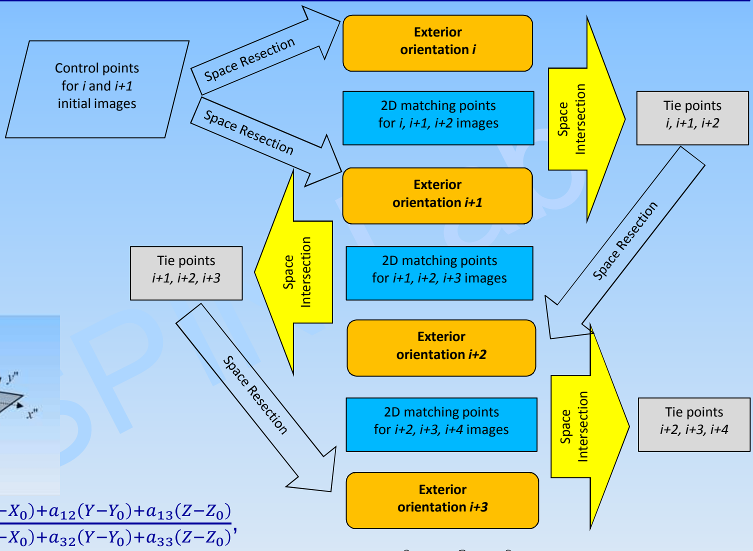


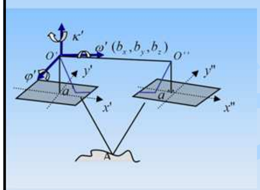


3D image based trajectory reconstruction (ICP)



2D image based trajectory reconstruction (SIFT)





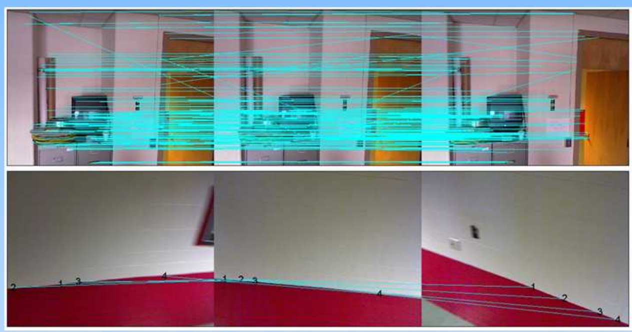
$$x - x_0 = -c\frac{a_{11}(X-X_0)+a_{12}(Y-Y_0)+a_{13}(Z-Z_0)}{a_{31}(X-X_0)+a_{32}(Y-Y_0)+a_{33}(Z-Z_0)},$$

$$y - y_0 = -c\frac{a_{21}(X-X_0)+a_{22}(Y-Y_0)+a_{23}(Z-Z_0)}{a_{31}(X-X_0)+a_{32}(Y-Y_0)+a_{33}(Z-Z_0)},$$

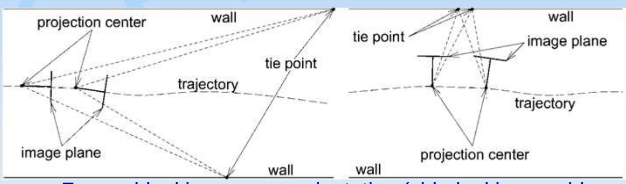
and so forth

Reference on slide 119

Image orientation and performance of 2D imaging



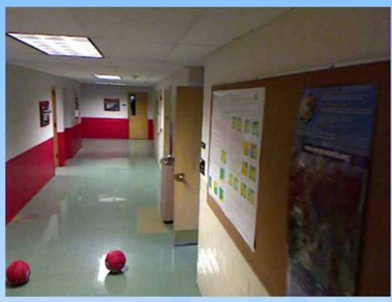
Matched SIFT features in different scenarios



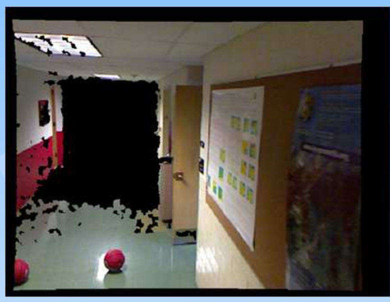
Forward-looking camera orientation (side-looking would give better geometry)

Fusing 2D and 3D imagery

2D image



Reconstructed 2D image



- Images can be reconstructed from point cloud (3D) on pixel-based correspondence
- Reconstruction has some issues at gaps (max range, etc.) and bright objects

Method

1. Apply the SIFT process to reconstructed images
2. Keypoints are extracted from reconstructed 2D images
3. In addition, SIFT was used on all bands (RGB) to find more features
4. Robust estimation of 3D transformation parameters
5. Use the 3D parameters as initial values for ICP

Extended ICP method

Use RGB information, photo registration, and previous frames

TR = identity

Push the first cloud into *S* stack

Pop the last 5 known clouds from stack *S*, it is model *M*.

Transform *D* into first frame CS with *TR*

Read *D* point cloud from sensor

Initial parameters from SIFT based matching

Calculate rotation and transformation parameters by minimizing

$$\min_{R,T}[e_d(R,T) + e_c(R,T)],$$

$$e_d(R,T) = \sum_{v \in D} \|match_M(Rv+T) - (Rv+T)\|_2^2,$$

$$e_c(R,T) = \sum_{v \in D} \|rgb(match_M(Rv+T)) - rgb(Rv+T)\|_2^2,$$

Transform *D* into *M*.

Error budget analysis

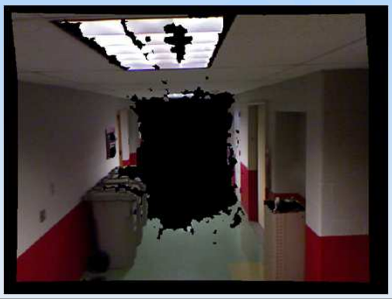
Push down *D* into *S* stack

algorithmic step

initialization

data

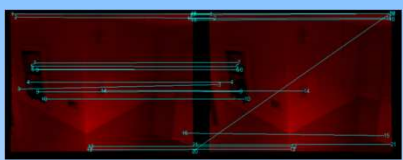
Image reconstruction



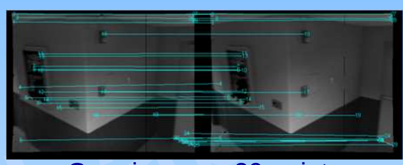
- Each point in the point cloud (on the depth image) has an unique corresponding point on RGB image
- Matching 3D points in consecutive frames can be found using 2D matching techniques, e.g. SIFT, SURF
- 2D matching applied on reconstructed images created from colored point clouds
- No features descriptors are returned for blank image parts (different from using the original 2D images)

SIFT-based matching in RGB

SIFT keypoints on two consecutive frames

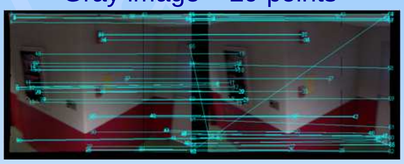


R channel – 21 points

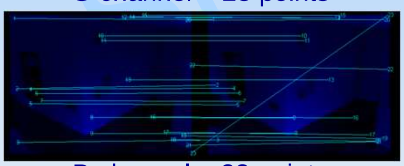


Gray image – 29 points

G channel – 23 points



R + G + B channels – 66 points



B channel – 23 points

R + G + B + Gray – 87 points

Outlier removal

Similarity transformation

- Applying filtering mask to maintain even (as possible) distribution of keypoints over image area
- 6 transformation parameters (scale equal to 1) estimated on the basis of 3D matching points
- Outliers filtered during transformation parameters estimation (robust) – outliers get weight close to 0

$$w_i^{(k)} = \begin{cases} e^{-a\left(\left|v_i^{(k-1)}\right|-s\right)^b}, & \left|v_i^{(k-1)}\right| > s \\ 1, & otherwise \end{cases}$$

where:

$w_i^{(k)}$ – weight of point in *k* step of iteration,

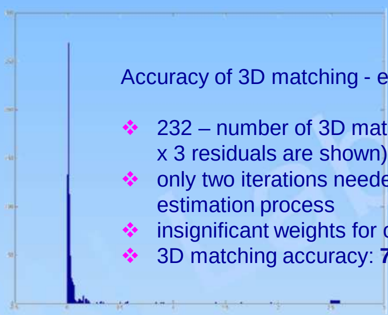
$v_i^{(k-1)}$ – point coordinate residual in *k*-1 step of iteration, and

a, b, s – damping function parameters, empirically chosen as 1, 2, and 0.01, respectively.

Trajectory reconstruction performance

Mismatch removal

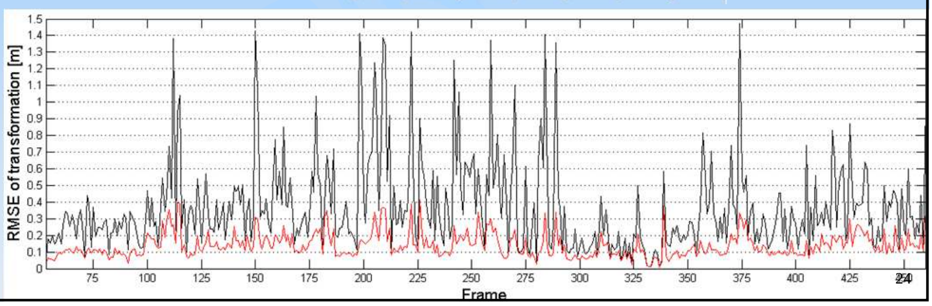
- Point weighting
- Iterative solution



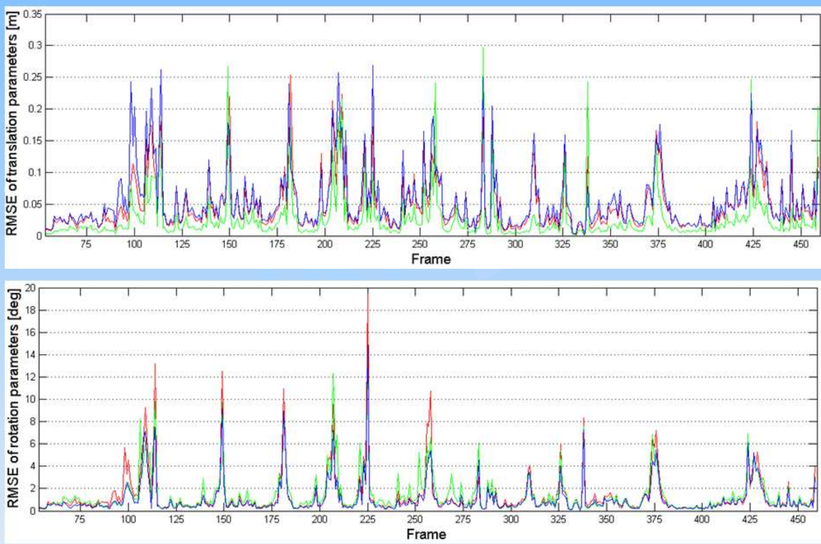
Accuracy of 3D matching - example

- 232 – number of 3D matching points (232 x 3 residuals are shown)
- only two iterations needed in the robust estimation process
- insignificant weights for outliers
- 3D matching accuracy: **7.8cm**

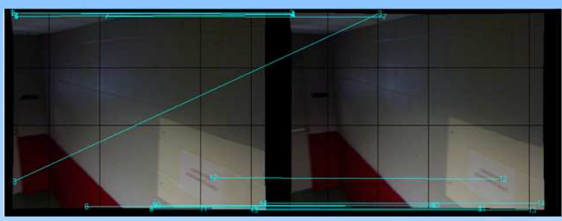
Black: SIFT solution
Red: Outliers removed



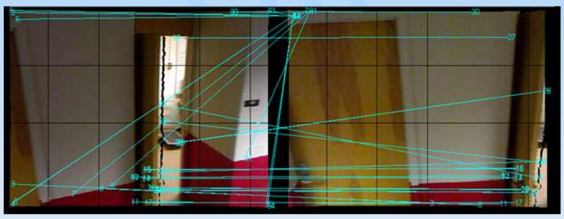
Estimated accuracy



Error identification

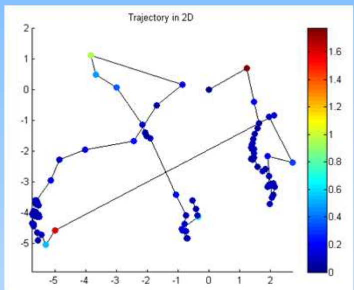


Low image variability scenario:
- insufficient number of inliers
- uneven distribution of inliers

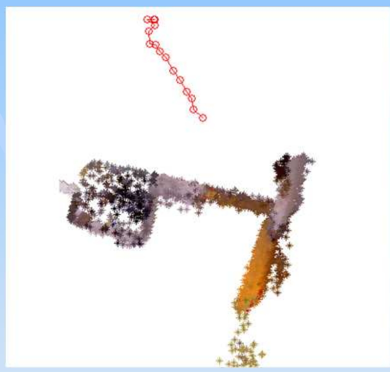


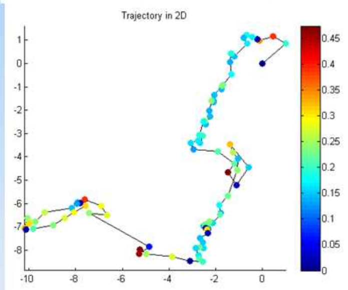
Rapid turn and variation in the frame rate:
- small overlapped area
- low number of inliers
- uneven distribution of inliers

3D Solutions in 2D Projections



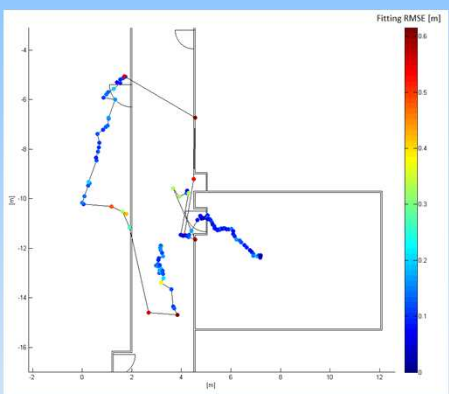
ICP 3D solution

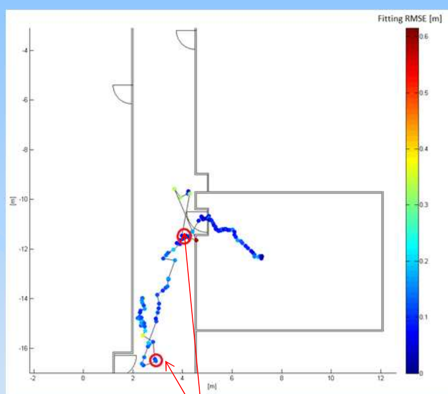




Modified ICP 3D solution

Final Trajectory Reconstruction



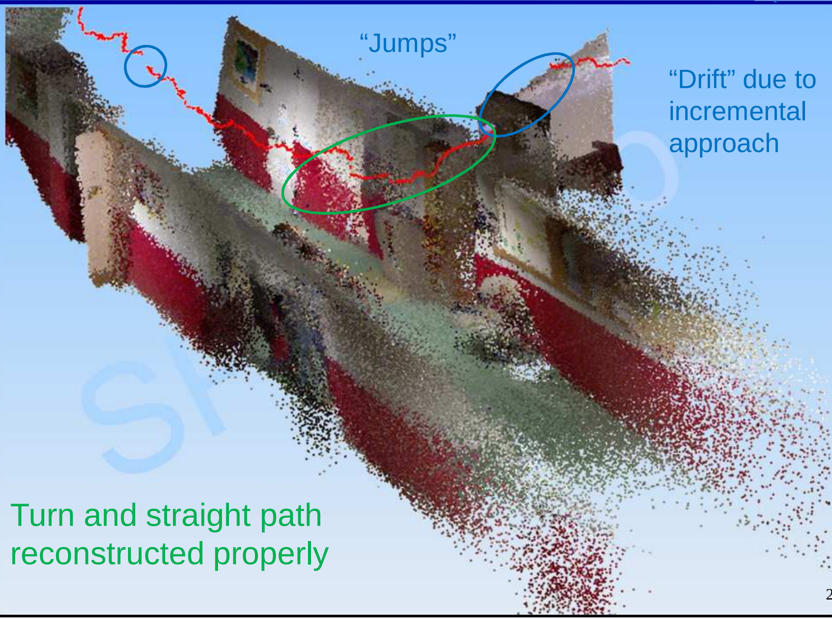


Accuracy assessment:

- UWB reference accuracy: ~ 30 cm per coordinate
- For the properly reconstructed sections of the trajectory: ~ 50 cm per coordinate

Providing “fixes”

Reconstructed trajectory and stitched point cloud



Conclusion

- Transformation parameters between consecutive frames are properly estimated for most of the survey, producing reliable trajectory and stitched point cloud
- Errors are mainly caused by unusual navigation behavior (U-turns) and frame rate variations; note, these situations can be easily detected
- Integration with other sensor data, e.g. IMU may/will improve quality of trajectory reconstruction and mapping
- Drift effects can be reduced introducing key-frames and applying Kalman Filter