

Medical Stereo Dataset (v 1.0)

Andrew Speers

January 11, 2019

1 Introduction

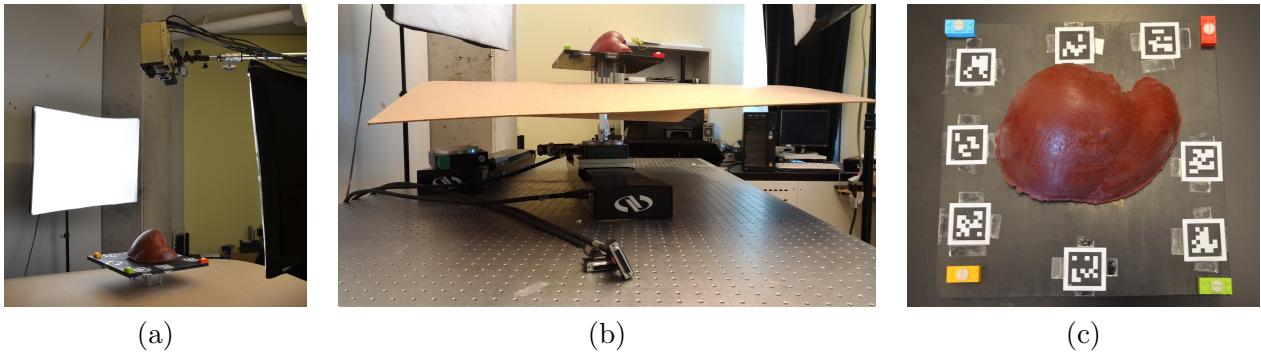


Figure 1: Experimental setup. Figure (a) depicts the general experimental setup (including illumination, plywood backdrop, liver phantom and fiducial board and the stereo camera hovering above the target liver phantom. Figure (b) shows the same setup where the linear (bottom) and rotation (top) stages of the motion platform are visible as well as the linkages connecting the stages directly to the liver phantom and fiducial board. Figure (c) shows an overhead closeup of the fiducial board and liver phantom.

The dataset consists of calibrated stereo image sequences of a silicone liver phantom attached to a motion control platform (one linear and one rotational stage) for precise motion control, see Figure 1. The phantom is rigidly affixed to a fiducial board containing 2D AR-tag fiducials as well as 3D Lego fiducials in the corners of the fiducial board. These fiducials are used to provide an initial alignment of the reconstructed point cloud to the corresponding CT scan and for verification of the provided motion commands. The CT scan is also included as part of the dataset. Note: All fiducials should be masked out and not processed by a vision algorithm using this data for verification / evaluation of your system.

This dataset can be used to help evaluate surgical augmented reality / image guided surgery systems allowing for the reporting of both surface registration error and motion estimation errors.

2 Dataset organization

The sequences contained within the Medical Stereo Dataset test three conditions: translation only, rotation only and translation and rotation combined. Each test sequence consists of 31 frames evenly spaced with one degree / one mm motion increments. The image sequences are located in the “images” subfolder of the dataset. Calibration information is found in plain-text format within the “calibration” subfolder. Naming conventions of the matrices follow Open CV standards.

The liver and fiducial board have been scanned using a CT scanner and processed to create a volumetric model of the liver and the fiducial board. Both of these models are available, in vtk and ply formats, in the “ct_scans” subfolder. These models can be used to aid in an initial alignment between your 3D surface reconstruction and the CT scan as well as to report measures of surface registration error (comparing your registered reconstruction to the CT scan of the liver).

The center of rotation is provided for each frame of each dataset. The additional data allows for an accurate decoupling of registration estimates into a translation and rotation estimate (about the center of rotation of the platform). After this, motion estimates can be compared to the ground truth values of 1 mm / 1 degree of motion per frame allowing for an accurate evaluation of the motion recovery portion of your system. The center of rotation locations are available in the “center_of_rotation” subfolder in plain-text format.

3 Computing error metrics from transformation estimate

3.1 Computing the angle of rotation

Assume the input is a 4x4 matrix representing the rigid transformation estimate between the current frame and the keyframe (frame 1), T .

The rotational component of T can be represented as a quaternion, T_q , and the angle of rotation can then be described as

$$\text{angle of rotation} = 2 * \arccos(T_q(1)) \quad (1)$$

where $T_q(1)$ is the first element of the vector representation of the quaternion.

```

1 % Assume 4x4 rigid transformation in variable Ta
2
3 % Compute quaternion representation of the rotation portion of transformation Ta
4 dq = qm(Ta);
5
6 % Compute the angle of rotation from dq
7 angle_of_rotation = 2 * acosd(dq(1));
8
9 function q = qm(M)
10 % QM Quaternion from affine matrix.
11 %   Q = QM(M)
12
13 q = zeros(1,7);
14 q(1,1) = 0.5*sqrt(trace(M));
15 q(1,2) = (M(3,2) - M(2,3))/(4*q(1));
16 q(1,3) = (M(1,3) - M(3,1))/(4*q(1));
17 q(1,4) = (M(2,1) - M(1,2))/(4*q(1));
18 q(1,5) = M(1,4);
19 q(1,6) = M(2,4);
20 q(1,7) = M(3,4);
21 end

```

Listing 1: Matlab example: Angle of rotation computation

3.2 Computing the translation magnitude

Before computing the translation magnitude the center of rotation has to be removed from the transformation estimate. Then the magnitude of the translation component can be shown as the L2-norm of the translation component of the 4x4 transformation matrix.

```

1 % Assume 4x4 rigid transformation in variable Ta
2 % Assume center of rotation 1x3 vector for current frame in decen
3
4 % Compute translation-only 4x4 transformation matrix from decen
5 Ddecen = [eye(3) -decen; 0 0 0 1];
6
7 % Compute rotation-only 4x4 transformation matrix from Ta
8 R = Ta;
9 R(1:3, 4) = [0 0 0]';

```

```

10
11 % Compute residual translation , dVec, after removing center of rotation
12 Ddelta = Ta * inv(inv(Ddecen) * R * Ddecen);
13 dVec = Ddelta(1:3, 4);
14
15 % Compute magnitude of translation component
16 dmag = sqrt(sum(dVec .* dVec)) * S;

```

Listing 2: Matlab example: Translation magnitude computation

The center of rotation values are stored in the “center_of_rotation” folder and are labelled cor_1 - cor_30. In general, cor_n should be used in the computation of the translation component of the transformation estimate between frames n and n+1.

4 References

Please site our paper [1] whenever presenting work that makes use of this dataset.

- [1] A. Speers, B. Ma, W. Jarnagin, S. Himidan, A. Simpson and R. Wildes, Fast and accurate vision-based stereo reconstruction and motion estimation for image-guided liver surgery, Healthcare Technology Letters 2018.