Automated 3D Reconstruction of Moving Rigid Specimen from RGB-D Video Input

Erika Harrison
Department of Computer Science
University of Calgary
Calgary, Canada
eharris@ucalgary.ca

Faramarz Samavati
Department of Computer Science
University of Calgary
Calgary, Canada
samavati@ucalgary.ca

Jeffrey Boyd
Department of Computer Science
University of Calgary
Calgary, Canada
boyd@ucalgary.ca

***Abstract*—**
To assist morphometric and behavioural analysis of live animals, we present an automatic process for generating a 3D volumetric representation of a dynamic rigid specimen from RGB-D video. This process uses multi-feature extraction and automatic labelling through a novel distance matrix structure and rigid transformation validation to reduce maximal clique calculations. An adapted SiftFu implementation then incorporates the resulting rigid transformations to perform the final volumetric reconstruction. Validation occurs using an RGB-D sequence from a living leopard tortoise resulting in a smoothly merged and textured volume from the original RGB-D frames.

***Index Terms*—3D reconstruction; rigid transformations; animal RGB-D video capture**

I. Introduction

The biological sciences are tasked with studying animals in laboratory and wildlife settings while constrained to minimize impact and interaction of their specimen. Modern technology is reaching sufficient advancement to assist in live acquisition and computational reconstruction of real-world specimen models for tracking, analyzing and archiving. However, biological specimen vary in shape and form, and their precise morphometric configuration is often unknown.

To assist with non-invasive, automated support for 3D analysis of animals, we present a frame-to-model multi-step process for general specimen reconstruction involving: initial RGB-D frame capture; multi-feature identification through point-pair distance tracking and feature point rigid object labelling; and per-object volumetric representation. This is useful for visual capture and analysis for biological studies where animals are able to move at will, and input is often limited in the number of cameras, and therefore viewing perspectives.

The main contributions of this work is presenting a cohesive, automated process to convert RGB-D input into a final volumetric representation; and its application to live specimen.

II. Related Work

Increasingly sophisticated techniques are being developed for 3D reconstruction using modern RGB-D camera input. Its application to animals, however, is relatively limited. For example, Winter [14] uses a laser scanner to reconstruct a taxidermy owl, while Falkingham [3] explores the low-cost RGB-D Kinect v1.0 to scan fossils. While these examples are valuable at demonstrating post-construction measurement accuracy, they are limited to stationary specimen.

For moving animal scenes, Fernandez et al. [4] explore laboratory rodent segmentation from point cloud information. Ross et al. [10], does not employ depth information, however they are able to identify rigid motion articulations of giraffes using their extraction system. Lastly, while the work of Duo et al. [2] emphasize 3D reconstruction of highly detailed motions from RGB-D video capture, it does demonstrate potential application to animals with their real-time canine reconstruction. However, it requires multiple cameras and its full-surface reconstruction does not support separation of motions, needed for any subsequent analysis.

To perform 3D reconstruction from RGB-D footage, a number of related works must be considered. For other examples of range image registration, see the survey of Salvi et al. [12] For the initial camera input, Khoshelham and Elberink [5] provide technical descriptions and accuracy evaluations of the Kinect 1.0. Registration between frames occurs using different types of feature points, described by Krig [6]. To identify rigid transformations, Perera and Barnes [9] describe an approach using point-pair distances and max-clique finding. Our work employs rigid transformation detection, inspired by the efficient calculations of Sorkine [13], to speed up the process and improve accuracy. Lastly, visualization using a volumetric representation is computed using the truncated signed distance function, as used in other works, such as Newcombe et al. [8] and the SiftFu work of Xiao et al. [15].

Note, whereas Chiari et al. [1] use stereo RGB images as input on stationary tortoise carapace, this work reconstructs a freely moving tortoise using RGB-D sequences.

III. Methodology

Converting a sequence of RGB-D images to a volumetric representation requires a number of stages. Firstly, we establish how frames relate by finding correspondences between them (Figure 1). Secondly, we identify rigid motions by identifying point correspondences residing on common rigid objects. This involves a single matrix representation for tracking and processing all correspondence points. Lastly, each rigid motion transforms the RGB-D images into a common volume for a final, textured volumetric representation.

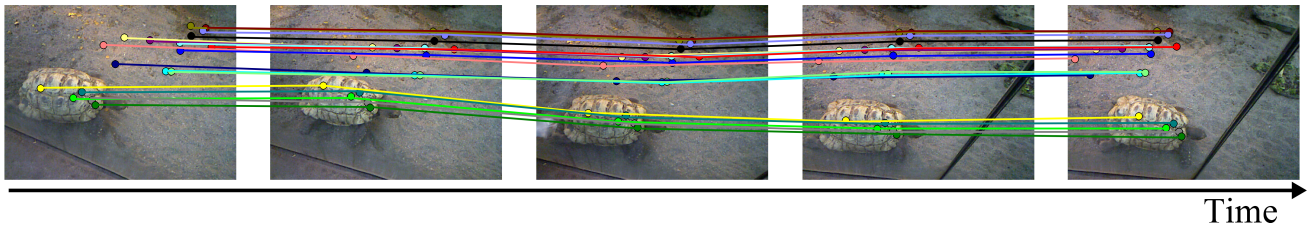

Fig. 1. Example of feature points matched across frames.

A. Initial Capture

RGB-D information is gathered using the low-cost Kinect 1.0, modified to work on portable battery, facilitating mobile capture. Each frame results in a 2D RGB image as well as a 16 bit 2D image which can be converted to depth.

To increase the quantity of points available for correspondence between frames, the RGB images are processed using seven (7) different feature-point extraction methods - SURF, MSERF, MinEigen, Harris, FAST, BRISK and SIFT, spanning the popular SIFT plus all feature types readily available in Matlab. For the provided sequences, we work with feature points visible in all frames. Feature points which disappear from view are discarded from consideration. Notice in Figure 2 that after features from Figure 1 of invalid depth are removed, the number of available features dramatically falls off to maintain inter-frame correspondence. This motivates the use of multiple types of features to improve correspondence and resulting reconstruction quality.

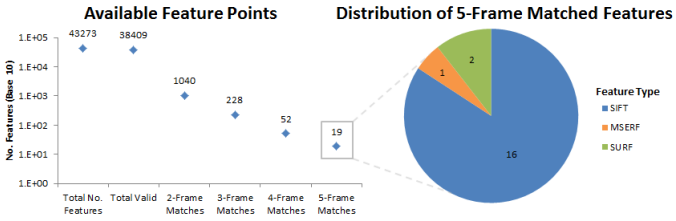

Fig. 2. Reduction of total features from Figure 1 sequence, after discarding invalid depth, and then matching across frames.

B. Feature Labelling

Labelling of features refers to the identification of feature points undergoing a common rigid transformation across frames. Feature points with the same transformation are given the common label. This is fundamental in identifying regions of the RGB-D images which reside on the same rigid object.

For the resulting n features, their 3D coordinate is computed by projecting using the depth map and camera specifications. This results in an Euclidean coordinate in metres, $(0, 0, 0)$ centered at the camera, with the z-axis projecting out from the camera, the y-axis projecting up from the camera, and x-axis perpendicular to both and right of the camera's view. The resulting 3D feature coordinates are then used to populate an $n \times n$ symmetric matrix D storing the point-pair distances. The point-pair distances in the matrix are updated each frame, as the feature points may have changed position relative to each other as would be expected in a dynamic scene. Figure 3 illustrates an example of how the distance matrix is updated between frames. Notice that the camera position may change, as the point-pair distances are relative to the pairing, and unrelated to absolute coordinates.

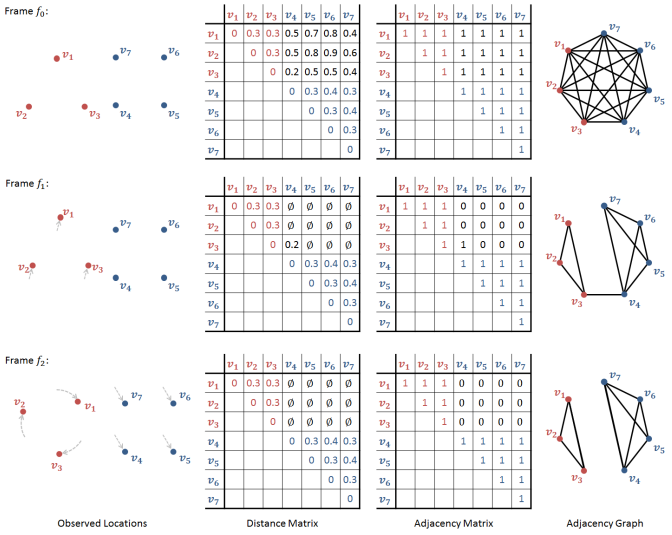

Fig. 3. Illustrative example of algorithm used for identifying rigid bodies.

Observe that for rigid bodies, distances of points on the same rigid body remain constant. As the distances are updated for each frame, if a distance significantly differs from a prior frame, the values are flagged as invalid. This corresponds to two points not residing on the same rigid body.

Whereas Perera and Barnes [9] identify trial-and-error sequence-specific thresholds using standard deviation and averages, an error threshold based on error metrics intrinsic to the Kinect camera are used to identify signficant differences. From Khoshelham and Elberink [5], the error function for a point z metres from the camera on the z axis, is expressed as:

$$E_z(z) = (1.87z^2 - 1.84z + 2.21) \times 10^{-3}. \quad (1)$$

Notice this error changes with distance and position from the camera. While Khoshelham and Elberink [5] describe how to calculate the errors E_x, E_y for x and y respectively, experimentation demonstrates that their contribution is negligible and we use E_z to represent the error for a 3D coordinate. As feature points may be at different z distances, and therefore

different errors, we use standard error propagation from the point-pair distance calculation to produce:

$$E(z_1, z_2) = E_z(z_1) + E_z(z_2) \tag{2}$$

for z_1, z_2 the z coordinates of the respective 3D feature points. E is then used as our error threshold for identifying significant differences between frames, and must be computed for each frame and each point-pair. Figure 3 illustrates how point-pair (v_2, v_4) becomes invalid in frame f_1. Figure 4a-e illustrates distance matrix updating of the Figure 1 sequence.

An adjacency matrix graph representation is computed by:

$$A_{i,j} = \begin{cases} 1, & \text{if } D_{i,j} > E(z_i, z_j) \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

from the $n \times n$ distance matrix D. Figure 4f illustrates an example of such a resulting adjacency matrix. Edges represent point-pairs likely belonging to the same rigid object, or not yet observed to be on different objects. Whereas Perera and Barnes [9] immediately apply the maximal cliques algorithm to their adjacency graph, we speed up the process by:

- Identifying connected components, efficiently computing rigid transformations [13], and computing residuals to identify if all points are on a common rigid object
- For points not identified above, performing transformation-residual checks on objects identified in a prior frame
- For points not identified above, computing max cliques from the adjacency graph, and using the transformation-residual to confirm rigid transformations.

In performing transformation checks, the possibly exponential time for finding max cliques within the graph is reduced.

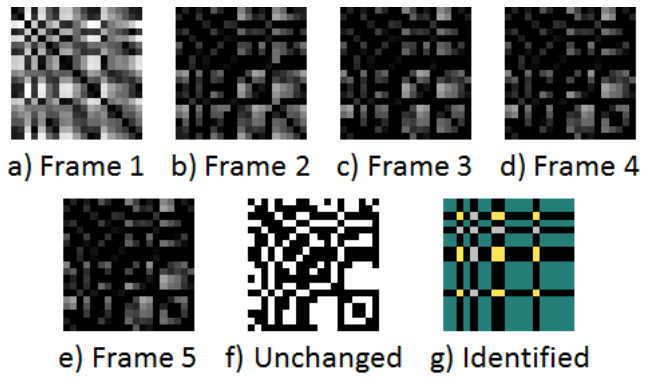

Fig. 4. a-e. Frames 1-5 visualizing distance matrix. Black indicates point-pair distance exceeding error threshold. f. Visualization of resulting adjacency matrix. g. Visualization of adjacency matrix after clique finding, rigid validation is applied. Using the presentation format from SiftFu.

At this point, feature points from the original RGB-D sequence have been automatically labelled for each calculated object. Figure 5 illustrates the initial adjacency graph (a) which is pruned using the distance matrix to the resulting labelled graph (b), and with points labelled for each of the objects. It is worth noting that this process does not require a priori information on the number of rigid objects present.

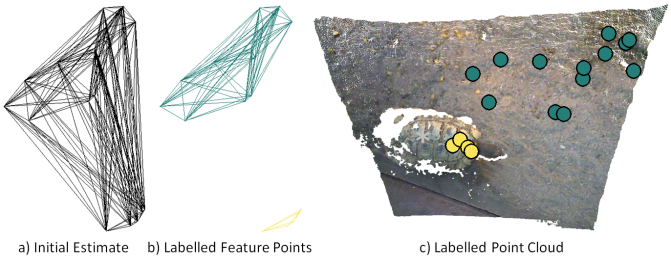

Fig. 5. a. Initial adjacency graph; b. Updated adjacency graph with labels; c. Labelled feature points in 3D space.

C. Conversion to Volumetric Representation

To construct a fully 3D representation of the resulting objects an adaptation of the truncated-signed distance (TSDF) SiftFu approach of Xiao et al. [15] is used. This results in a textured volumetric representation of each object.

Note that SiftFu relies on SIFT features and RANSAC for identifying a rigid transformation with which to transform the depth information into the TSDF-calculated volume. Instead, for each of the identified rigid objects a volume is created, and the object's corresponding per-frame rigid transformation is used to transform the depth information in the volume. Readers are directed to the SiftFu work of Xiao et al. [15] for further details on converting RGB-D to TSDF-calculated volumes.

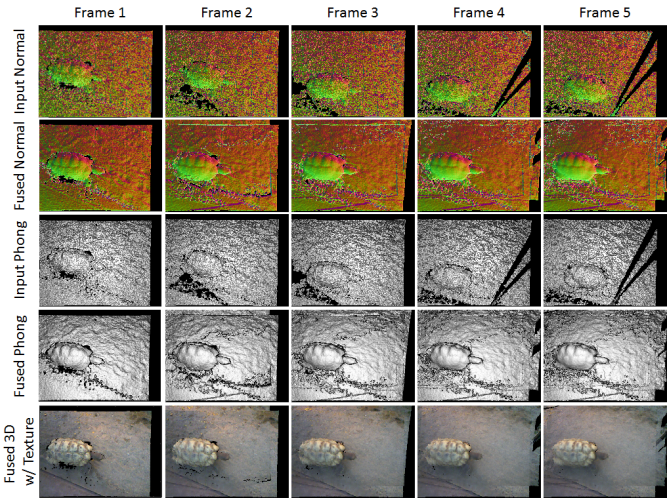

Fig. 6. Visualization of TSDF output across frames. Using presentation format from SiftFu [15].

D. Validation

A leopard tortoise at the Calgary Zoo was filmed using a Kinect 1.0 device. To maximize frame count and overlapping feature points, five consecutive frames were manually extracted to illustrate the reconstruction process. Computation was performed on Matlab using a 64-bit Intel Core i7-3770 with 16 GB RAM. Matlab implementation for the following were used: SIFT [7], SiftFu [15] - adapted for this work, and Point Cloud Library [11] - for RGB-D to 3D projection.

IV. Results

From Figure 6, observe how the holes on the surface representation decrease as more frames of input are aligned. Also notice how the posterior of the specimen initially is not visible. As more surface is made visible to the camera, the reconstruction is able to incorporate and amalgamate the additional surface information. This results in a whole model volume representation of the observed surface of the tortoise.

Figure 7 visualizes the final result of the five frame leopard tortoise sequence. Note: The washed out texture is an artifact of the 3D visualization used, as the bottom left corner of Figure 6 illustrates strongly merged texture.

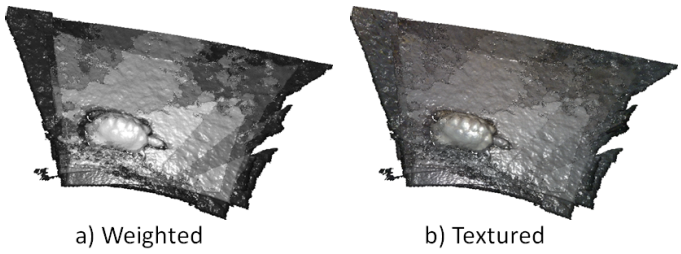

Fig. 7. Volumetric Representation. a. Visualization of weighting. Lighter areas represent more overlapping frame volumes, darker areas represent fewer overlapping frame volumes; b. Textured result.

Algorithm run time was not calculated, and the presented work is not real-time. However, due to the efficient error thresholding and resulting accuracy of the distance matrix, clique finding is not required in this example and less than 2% of the time is spent labelling feature points. The dominant portion of time is spent in the SiftFu implementation which can be demonstrated to work in real-time speeds by using comparable algorithms [2]. The remaining portion of the system - RGB-D capture, feature extraction, feature point labelling - has been designed to be readily parallelized.

V. Conclusion

The process presented successfully demonstrates an automatic reconstruction approach from RGB-D frames to final textured volumetric representation. The merging and adaptation of existing techniques is instrumental in accomplishing this reconstruction, including the use of camera-specific error calculations for thresholds, and refactoring volumetric reconstructions to work with per-object rigid transformations. The resulting reconstruction demonstrates its ability to be applied on real-world rigid specimen.

Future work includes parallelization of the system to improve speed, and expanding the process to automatically perform reconstruction on live data. Additionally, back-calculating temporarily unavailable feature points can be applied to improve the quality of rigid motion estimations, and increase the number of avaiable feature point data used in a given frame. Finally, expanding this work to identify and support hierarchical rigid motions and applying the work to alternative specimen will also be explored.

This work provides a solid step in facilitating automated reconstruction of dynamic objects, and demonstrates its application to live specimen.

VI. Acknowledgements

Research supported by Alberta Innovates Technology Futures and the Computer Science department at the University of Calgary. With permission from the Calgary Zoo. We wish to thank Mark Sherlock for assisting with data collection and Shannon Halbert for suggestions on figures.

References

[1] Ylenia Chiari, Bing Wang, Holly Rushmeier, and Adalgisa Caccone. Using Digital Images to Reconstruct Three-Dimensional Biological Forms: A New Tool For Morphological Studies. *Biological Journal of the Linnean Society*, 95(2):425–436, 2008.

[2] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, Pushmeet Kohli, Vladimir Tankovich, and Shahram Izadi. Fusion4d: Real-time performance capture of challenging scenes. *ACM Trans. Graph.*, 35(4):114:1–114:13, July 2016.

[3] Peter L. Falkingham. Low Cost 3D Scanning Using off the Shelf Video Gaming Peripherals. *Journal of Paleontological Techniques*, (11):1–9, June 2013.

[4] Oscar Fernandez, Elsbeth van Dam, Lucas Noldus, and Remco Veltkamp. Robust Point Cloud Segmentation of Rodents using Close Range Depth Cameras in Controlled Environments. In *Visual Observation and Analysis of Animal and Insect Behavior (VAIB'14), ICPR Workshop*, August 2014.

[5] Kourosh Khoshelham and Sander Oude Elberink. Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. *Sensors*, 12(2):1437–1454, 2012.

[6] Scott Krig. Interest Point Detector and Feature Descriptor Survey. In *Computer Vision Metrics: Survey, Taxonomy, and Analysis*, pages 217–282, Berkeley, CA, 2014. Apress.

[7] D.G. Lowe. Object Recognition from Local Scale-Invariant Features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157, 1999.

[8] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. KinectFusion: Real-time Dense Surface Mapping and Tracking. In *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality*, ISMAR '11, pages 127–136, Washington, DC, USA, 2011. IEEE Computer Society.

[9] Samunda Perera and Nick Barnes. Maximal Cliques Based Rigid Body Motion Segmentation with a RGB-D Camera. In *Proceedings of the 11th Asian Conference on Computer Vision - Volume Part II*, ACCV'12, pages 120–133, Berlin, Heidelberg, 2013. Springer-Verlag.

[10] David A. Ross, Daniel Tarlow, and Richard S. Zemel. Learning Articulated Structure and Motion. *Int. J. Comput. Vision*, 88(2):214–237, June 2010.

[11] Radu Bogdan Rusu and Steve Cousins. 3D is Here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[12] Joaquim Salvi, Carles Matabosch, David Fofi, and Josep Forest. A Review of Recent Range Image Registration Methods with Accuracy Evaluation. *Image and Vision Computing*, 25(5):578 – 596, 2007.

[13] Olga Sorkine. Least-Squares Rigid Motion Using SVD, Feb 2009. Technical Notes.

[14] Charlotte Winter. 3D Laser Scanning of Taxidermy Owls, Apr 2012. Year End Exhibition - Shrewsburys College of Art & Technology.

[15] Jianxiong Xiao, A. Owens, and A. Torralba. SUN3D: A Database of Big Spaces Reconstructed Using SfM and Object Labels. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1625–1632, Dec 2013.