Final Project Report

3D Reconstruction of Infants' Cranial Shape using Mobile Devices

EN 601.656 Computer Integrated Surgery II

David Shi
Biomedical Engineering,
Computer Engineering
dshi9@jhu.edu

Tara Tang
Biomedical Engineering
ttang14@jhu.edu

Table of Contents

1. Introduction

1A. Clinical Motivation

Before an infant is six months old, their skulls are easily deformed, which can result in a number of cranial defects. One of the most common defects, deformational plagiocephaly/brachycephaly (DPB), has seen a rise in case numbers ever since parents were recommended to put babies on their backs to reduce the risk of sudden infant death syndrome (SIDS). Cranial deformity is becoming a pediatric epidemic, affecting up to 46% of newborns in the United States [1]. While DPB is non-synostotic, meaning it does not damage the brain and only causes facial asymmetry, other cranial defects like craniosynostosis, which causes highly morbid brain damage and requires early surgical intervention, are more serious [2, 4].

Early detection of DPB, craniosynostosis, and other skull-related deformities can prevent long-lasting trauma and potentially allow for helmet therapy or minimally invasive surgery. If deformities are not detected after six months, treatment can get much more complicated and require open surgery to prevent brain damage and death [6].

With this project, we aim to improve the way skull shapes are measured and examined by pediatricians and primary care providers. By generating a 3D model of a baby's head, we can provide physicians with a more effective, accurate, and detailed method for evaluating and diagnosing children with skull deformities.

1B. Prior Work

Currently, pediatricians only have a simple measuring tape to measure circumference and reference a head-circumference-for-age percentile chart for diagnosis of skull defects [1]. More often than not, when parents are concerned about their infant's misshapen head, the deformity is too subtle for easy detection, and the physician dismisses their concerns. Consequently, 86% of infants with DPB are not identified [6].

PediaMetrix has developed computer vision methods that operate on 2D images to provide a better approximation for circumference and other metrics, such as cranial index (CI) and cranial vault asymmetry index (CVAI). However, 2D images have limitations, as they come only from the top-down view of an infant's head. Thus, 3D parameters like volume and height cannot be measured. To address this, a 3D model of a baby's head would provide better detection of cranial defects, providing more information than what a 2D image can offer. Though conventional 3D scanners are available, they are expensive and only exist in the offices of neurosurgeons, plastic surgeons, and helmet clinics, not in the offices of general pediatricians. A mobile app solution that can generate accurate 3D models would be easily accessed by pediatricians through smartphone and tablet technology and enable point-of-care testing.

1C. Goals

The goal of this project is to develop a software pipeline to reconstruct an accurate 3D model of a baby's head, using depth information from a sensor and mobile application. This project is motivated by the need for better techniques in diagnosing cranial deformities in babies younger than six months of age, as described above.

The work done in this project will be used as the initial foundation for PediaMetrix's deformity diagnosis tool. This tool will ultimately incorporate our 3D model reconstruction with a machine learning algorithm that can analyze the model to identify potential deformed areas and make a diagnosis.

By providing a starting point for a quicker and easier method of diagnosis for pediatricians, we hope to reduce the need for helmet therapy and physical therapy for children who were not diagnosed early enough, which is currently a large financial hurdle for many families [6]. Additionally, we also hope to enable the early detection of more severe conditions like craniosynostosis to avoid open surgery or brain damage.

2. Technical Approach

2A. Schematic

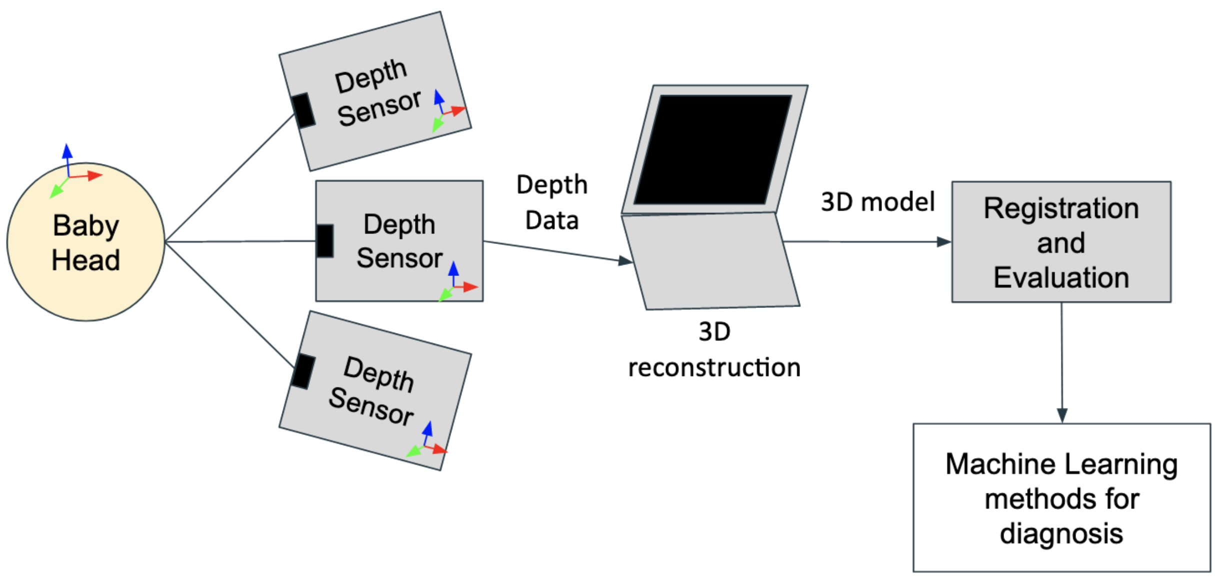

Figure 1: Schematic of software pipeline

See below for more details on each step of this workflow. In general, we begin with a baby head (doll or phantom), and we will use our depth sensor to collect data of the head from multiple angles, each in their own local coordinate systems/frames. Then, all of this data will go into our sequential ICP algorithm for 3D reconstruction and output a registered 3D model. Then, we can evaluate it against other reconstructions and ground-truth of the same object, and we can visualize our result with a mesh. The final white box in the schematic is not within the scope of this project, and it represents the direction that PediaMetrix intends to go with our project results. Ultimately PediaMetrix's goal is to input 3D models into a machine learning algorithm to perform automatic diagnosis.

2B. Data Collection and Preprocessing

2B.1. iPad and Sensor

A depth sensor (Structure Sensor, Occipital) attached to an iPad is provided by PediaMetrix. The iPad has a custom application built by Dr. Güler installed on it, which allows the user to take color images (RGB) and depth images (D) that are automatically uploaded to an Amazon AWS bucket. The data is then available for download in the Amazon bucket. A depth map is akin to a grayscale image that acts as a heat map for the spatial distance from camera to object. Each pixel is populated with this measured distance instead of a brightness value. To get the distance between adjacent pixels, we will need the intrinsic parameters (focal lengths and optical center) of our camera, which we are given. Using this information, we can recreate the depth map in 3D space [5].

Figure 2: Structure Sensor attached to an iPad, provided by PediaMetrix.

2B.2. Baby Doll and Phantom

For testing our pipeline, we are provided with a baby doll that we can take images of. This is what we used to evaluate our registration algorithm. PediaMetrix also provided us with a 3D-printed head phantom of a baby and its corresponding STL file, which contains the surface geometry information for the phantom. The STL file acts as a ground truth for our phantom. The doll and head phantom are shown in **Figure 3**.

Figure 3: Example RGB data: (Left) Baby doll. (Right) 3D-printed head phantom.

2B.3. Preprocessing

For pre-processing, we first crop the depth map to focus on the head of the baby. During data collection, our camera app automatically displays a field of view that is 25% of the camera's actual field of view. Thus, it is safe to assume that the entire object is confined to the center 25% of the image data. Therefore, we crop to the head by deleting the first 25% and last 25% of the rows and columns. We also threshold depth by eliminating points that are too far from the camera according to the distance at which the depth sensor and camera app consider to be optimal. Both these techniques help to eliminate outliers and ensure that only the head and neck are considered for ICP, reducing unwanted contributions from the background. We also apply a Gaussian blur on the depth map to ensure a smoother depth transition at an edge, where errors can occur with sharp edges. In doing so, we eliminate some noise that can happen at the edge of objects. After evaluation of various amounts of Gaussian blur, we settled on a kernel size of 7 and sigma value of 0.1.

2C. 3D Reconstruction

Our pipeline is in Python and primarily employs the use of Open3D, a Python library for 3D data and image processing. This library has numerous methods for point cloud generation, registration, and visualization [7].

Our data should consist of multiple images of a baby's head from all angles. This can be taken with the baby still and the camera rotating around the baby, or moving the baby while keeping the camera still. By using the provided camera intrinsic parameters, we can use Open3D to generate 3D point clouds from these depth images. Our next step is to register these point clouds together such that they are in a common coordinate system.

The approach we have decided to take for this registration is a sequential Iterative Closest Point (ICP) pipeline. ICP is a commonly used algorithm for iteratively comparing two point clouds in order to find a rigid frame transformation (consisting of a rotation and translation) to convert from one to another. In our case, we are defining the world frame as the frame of the first point cloud we collect. Please refer to the **Appendix** at the end of this report for a more detailed description of the basic ICP methodology.

To accomplish a sequential ICP pipeline, we iterate over every image in order. This assumes that images were taken consecutively and do not jump back and forth between significantly different angles. For each image, we perform ICP to register it to the previous image and store the resulting transformation in a list. When we are done, we compose all of the transformations in reverse order to sequentially register frame n to frame n-1, and then to frame n-2, etc. until we reach frame 1. The reason we do not do a pairwise registration that registers every frame to every other frame is because of the general smoothness of a baby's head, which lacks rich and definitive features. We have observed that this registration scheme will result in all point clouds overlapping as much as possible, instead of overlapping slightly and gradually growing the reconstruction to cover all angles.

As mentioned above, Open3D has several functions relevant to ICP, which we are heavily relying on. For our limited trajectory, no special frameworks or data structures are necessary to store the final point clouds.

Once we have registered all of our point clouds to a single coordinate frame (frame 1), we can simply combine them into one point cloud. Then, to avoid redundancy, we can down-sample to a desired voxel size and generate a mesh using Open3D. This mesh will be the final visualization of the 3D reconstruction that we generate, and is heavily impacted by the method of reconstruction and the voxel size we downsample to.

2C.1. Robustness Testing

In order to make our pipeline robust to baby motion, we also intend to collect data where the baby is in slightly different positions between images. This movement will simulate certain amounts of “fidgeting” from the baby while we collect data. Motion is expected to compromise the fidelity of the depth data as we increase the amount of motion.

In general, we expect motion to present as increased noise in our data. We can potentially address this noise with common noise-reduction and smoothing algorithms, adapted to depth maps. If motion is minimal, our ICP approach should still hold, as we are cropping our data to only include the head, which is a rigid body. We are also not incorporating use of RGB data, which would otherwise cause difficulties with motion because the location of RGB features relative to each other would change.

2D. Evaluation

During evaluation, our “test points” will be the point cloud making up the ground-truth or Occipital reconstructed model, and we will register those to the same frame as our own output point cloud. ICP with identity initialization is less reliable here, because our result and the ground-truth frames could be wildly different. Thus, we are simply running our evaluation program with an initialization transformation that is manually found through simple composition of rotations and translations. As long as that initialization is close enough, ICP can be used to bridge the gap and generate the final registration for evaluation purposes.

Once our test and ground-truth models have been registered to each other, we can get comparison metrics from them. The most meaningful metric we are calculating is the average surface distance, which simply calculates the distance between every test point and its nearest point in the ground-truth point cloud. The average of this distance represents how close our pipeline output is to the actual baby head phantom.

3. Results

3A. RGB images

Figure 4: Different images captured by the RGB camera. Baby was fixed while the camera was moved around the baby.

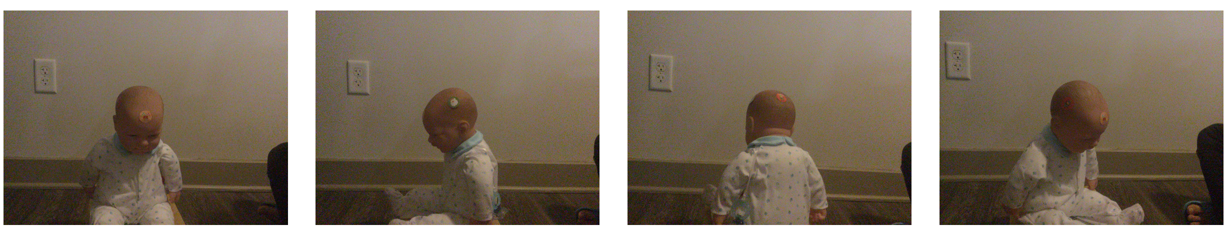

Figure 5: Different images captured by the RGB camera. Baby was rotated around while the camera was fixed.

3B. Depth point clouds (cropped)

Figure 6: Corresponding depth images captured by the depth camera from **Figure 5**. Depth maps are cropped for more accurate reconstruction without artifacts.

3C. Ground Truth, Reconstruction

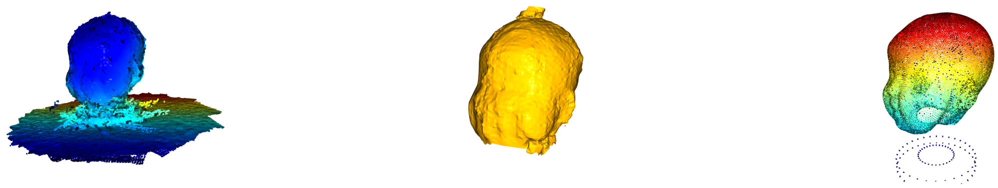

Figure 7: Results for the 3D printed phantom shown in **Figure 3**. (Left) Registered point clouds in a single coordinate frame. (Middle) Mesh generated from the registered point clouds (voxel size 5 mm). (Right) Ground truth point cloud.

The data for this final result was taken on the 3D printed phantom shown in **Figure 3**. Data was collected with small amounts of motion between images to check for robustness. Overall, we found that the result was on par with our previous motionless datasets, suggesting that with small amounts of motion, consecutive ICP is still able to match frames together. This makes sense, since we are only considering cropped depth information and the data is unaffected by the baby's position in relation to the surroundings.

For this final result, we compared our pipeline output point cloud to the ground-truth mesh, which we converted to a point cloud using Open3D. This procedure follows the evaluation described in **Section 2D**. Our average surface distance was 4.5097 millimeters. This indicates a relatively accurate reconstruction, below our threshold for the expected deliverable.

However, despite the moderately high accuracy, our model had clear room for improvement. The most significant issue with our final reconstruction was an appearance of two faces, which we believe is likely due to our pipeline's lack of loop closure detection (discussed more in **Section 4A**). Thus, although the small average surface distance suggests a well-aligned skull area, the obvious misalignment of the faces shows that our results can definitely be improved. In the future, a properly aligned reconstruction can also be used to calculate more clinically relevant metrics like cranial vault asymmetry (CVAI), which will show how effective our model might be at assisting with an actual diagnosis.

3D. Documentation

In addition to our code, we provided thorough documentation of our pipeline, program structure and algorithmic approach. This document can be found on our CIS II Wiki page.

4. Discussion

4A. ICP Discussion

Through our experiments, ICP is found to be not as robust to noise. During our initial attempts, we did not crop to the baby's head, which led to strange behavior as the ICP algorithm tried to minimize distances that were not part of the baby itself. This led to bad reconstructions that aligned background features rather than baby features.

This is what inspired our thresholding and cropping method. For the thresholding portion, we removed points that were farther than a threshold (70 cm) instead of replacing them with a max distance. This is to ensure that we only preserve points of importance and not build a background patch far in the distance. For the cropping portion, as described above, the iPad camera had crosshairs and a focus distance range that could essentially center the baby in the depth map. When we applied this cropping, we found that the ICP worked much better as it was able to align the isolated areas of importance much better.

In general, it is important to note that a consecutive ICP approach is prone to drift as the error from every registration builds up with the composed transformations. This issue is commonly dealt with using camera odometry and loop closures – in other words, when the camera returns to a viewpoint it had already seen previously, it can account for that built-up error. While we did not have time to implement loop closure optimization for our algorithm, it is definitely a promising method to approach in future work. Without loop closure, we frequently observed a reconstruction that appeared to have two faces, because the first and last frame no longer line up due to built-up error.

4B. A Vision-Based Approach

We also tried to implement a vision-based method that used the RGB images that were also taken by the iPad. This approach is based on a paper titled "An Evaluation of the RGB-D SLAM System" by Endres et al. [3]. The crux of this approach is using feature detectors on the pair of RGB images in order to detect visually strong features, instead of relying exclusively on depth information. We tried implementing this with both the Oriented FAST and Rotated BRIEF (ORB) and Scale-Invariant Feature Transform (SIFT) feature detectors.

With the features detected, we can generate a set of point correspondences in the 2D images, and then project those points to a 3D space using the depth information and simple computer vision algorithms. Then, we can perform RANSAC registration using the correspondences only, as provided by an Open3D function. RANSAC is the acronym for Random Sample Consensus, a common way to deal with noisy data and eliminate outliers.

We spent quite a bit of time trying to implement and debug this approach, since it made sense to us that using RGB features in the image would help the registration algorithm. To enhance the RGB features of the image, we put stickers (provided by PediaMetrix) on the baby doll's head. However, we found that even with these stickers, both ORB and SIFT were still not finding very many features to match. Furthermore, we found that the registration based on correspondences would often apply drastic transformations to the point clouds in order to line up the corresponding points, disregarding the alignment of the rest of the points. We believe that if there were some way to penalize large amounts of movement in a way that the algorithm would try to stay close to the identity matrix, this problem could be remedied. Regardless, we decided to discard this vision-based approach for now and focus on using depth only.

4C. Baby Heads

Baby heads were difficult to work with because they are not feature-rich, as their heads resemble more of a smooth sphere. This meant that the depth point clouds we extracted were quite smooth, and ICP would often have difficulty distinguishing two aligned heads from two misaligned heads. This made the final reconstructions prone to artifacts in alignment as the registrations were good approximations, but were not perfect representations of the baby head. Furthermore, this restriction prevented us from reasonably implementing a pairwise registration method where every frame was registered to every other frame. Typically, we would store all of the pairwise transformations in a pose graph which could then be optimized using a loss function to prune the less accurate edges. However, by trying to register viewpoints of a baby head that were far away from each other, our pairwise results ended up lining all of the point clouds directly on top of each other such that we had several layers of point clouds on one side of the reconstruction and empty space on the other side. This is most likely fundamentally due to the smoothness of the baby heads, such that registration did not find two completely overlapped viewpoints to be problematic.

5. Progress Evaluation

5A. Dependencies

All of our dependences were easily resolved except for the shaker we had intended on using for motion robustness testing. While our mentors were ready to order the shaker for us, we ended up spending too much time implementing our reconstruction algorithm, such that extensive motion robustness testing with the shaker set-up was no longer possible.

5B. Deliverables

Since we did not have time to work with the shaker, we were only able to achieve a small amount of motion robustness by manually introducing motion into our data. We did this by capturing slightly non-contiguous frames where the baby position and camera angle were slightly off. This robustness was part of our minimum deliverables. Aside from that, we were able to meet our expected deliverables, which consisted of: 1) collected and processed depth data, 2) working software pipeline and documentation, and 3) accuracy evaluation results with a surface distance of less than 5 millimeters.

5C. Timeline

For our timeline, we got held up troubleshooting the reconstruction and trying different methods to improve it. We lost time trying to fine tune parameters like depth thresholding, fitness thresholding, and the specific hyperparameters of ICP. Additionally, implementing the vision-based method and debugging it took much longer than we thought and did not produce better results for us than just using the depth maps.

	February				March				April				May			
Week	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Research and Planning																
Literature Review																
Data Processing (Depth Data)																
Reconstruction Algorithm																
3D Model Generation								[M]								
Registration Algorithm																
Documentation																
Evaluation and Improvement																
Ground-Truth Comparisons										+						
Baby Motion Robustness									-							
Final Evaluation												[M]				

Figure 8: Timeline proposed during Checkpoint Presentation (3/23/2021). Blue column: week of presentation. The "+" in "Ground-Truth Comparisons" was an added week, while the "-" in "Baby Motion Robustness" was a removed week. Even with these adjustments, we were unable to dedicate much time to Baby Motion Robustness due to our difficulties with debugging and optimizing our Registration Algorithm.

5D. Reflection

Our group did not have much prior experience with 3D reconstruction, so we got off to a slow start trying to research and understand the task at hand and what potential approaches existed. Motion robustness was a little ambitious as a minimum deliverable given how much time we spent on implementing and debugging the pipeline for a working reconstruction. We also lost time on implementing alternate pipelines, like the vision-based solution, which should have been better in theory but was limited in its scope within our Open3D framework.

David and Tara met with Can, Dr. Seifabadi, and Dr. Güler every week, which was very helpful for progress check-ins, getting any questions answered about our approach, debugging, and new ideas to try. Our management plan was very effective, with the PediaMetrix GitHub repository providing an efficient way to share code with the whole group and Google Drive allowing David and Tara to work on the weekly slide decks for the mentor meetings, as well as various presentations and reports for class. In general, David and Tara worked very well together, meeting several times a week over Zoom to work on code together via screen share. We also occasionally worked separately, especially when we were simultaneously developing the depth and vision-based approaches, but communication over Zoom during this work time kept both of us in the loop on each other's work.

6. Conclusion

6A. Significance

From our semester, we have successfully developed a working pipeline for a baby head reconstruction using a mobile device (iPad). This pipeline demonstrates that a mobile solution is possible for capturing volumetric information of a baby's head. It paves the way for Pediametrix to continue developing a viable clinical solution that pediatricians can use to assess cranial deformities.

6B. Moving Forward

1. We can refine the sensor interface to choose a region of interest, and do a cropping of the depth map during the data acquisition instead of during the processing. This reduces the size of the data that we have to import and download, and cuts down on processing time.

2. Vision-based should yield more accurate registrations if we can ensure stronger RGB features and penalize large transformations, as discussed in **Section 4B**.

3. Taking videos instead of a series of photos might be better, as the correspondence between frames would yield more accurate transformations as the frames are very close to one another. The downside to this might be longer processing time for the reconstruction as there is a lot of redundant information.

4. Higher resolution sensors can pick up more subtle shapes in the depth map to capture more intricate features to help in our reconstruction's accuracy and smoothness. This might lead to an increase with processing time as there would be more points collected in the point cloud.

5. Implementing loop closure detection can greatly reduce the error in the final reconstruction by accommodating for the error that builds up during consecutive ICP registrations.

7. Appendix – ICP Methodology

Once the 3D reconstructed model is created, it will be registered to a reference world frame. The preferred method for this registration process is an Iterative Closest Point (ICP) algorithm. ICP is a commonly used algorithm for iteratively comparing two point clouds in order to find a rigid frame transformation (consisting of a rotation and translation) to convert from one to another.

In our case, we are defining the world frame as the first point cloud we collect for our 3D reconstruction. The reconstruction will be generated in reference to that first point cloud, such that our resulting model will already be in the world frame. Then, during evaluation, our "test points" will be the points making up the ground-truth model, and we will use ICP again to register those to the same frame as our own model.

We begin the ICP procedure with an initial guess for the frame transformation, **F**, from the test points to the model points. Once we apply **F** to the test points, we search for the closest points in the model. This gives us a set of point pairs consisting of a test point (transformed to the world frame) and its corresponding closest model point.

We compare the distances between these point pairs with a predefined distance threshold and consider those with distances under the threshold to be "inliers." We use these inlier pairs to generate a new guess for **F**. Then, we begin a new iteration, applying this new **F** to all the test points and finding the next transformation using inliers. As we continually iterate, the number of inliers should increase as our **F** brings the test points closer to the model points. Eventually, the overall error between the transformed test points and the model points will be small enough that we can terminate the algorithm and consider the current **F** to be our registration frame transformation.

8. References

[1] Boston Children's Hospital. Deformational Plagiocephaly. Retrieved from https://www.childrenshospital.org/-/media/Centers-and-Services/Departments-and-Divisions/Plastic-and-Oral-Surgery/PlagiocephalyBook_interactive.ashx?la=en&hash=153679AB2E613576AD9A07E4C32B0F66AF0130B7

[2] Center for Disease Control and Prevention (2020). Facts about Craniosynostosis. Retrieved from Facts about Craniosynostosis – CDC.gov

[3] Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremers, D., & Burgard, W. (2012, May). An evaluation of the RGB-D SLAM system. In 2012 IEEE International Conference on Robotics and Automation (pp. 1691-1696). IEEE.

[4] Fenton, R., & Gaetani, S. A. (2019). A pediatric epidemic: Deformational plagiocephaly/brachycephaly and congenital muscular torticollis. Contemporary PEDS

Journal, 36(2). A pediatric epidemic: Deformational plagiocephaly/brachycephaly and congenital muscular torticollis

[5] Hartley, Zisserman. (2003). Multiple View Geometry in Computer Vision (2nd ed.). New York, NY: Cambridge University Press.

[6] Losee, J. E., & Mason, A. C. (2005). Deformational plagiocephaly: diagnosis, prevention, and treatment. Clinics in plastic surgery, 32(1), 53–viii. https://doi.org/10.1016/j.cps.2004.08.003

[7] Qian-Yi Zhou, Jaesik Park, Vladlen Koltun. (2018). Open3D: A Modern Library for 3D Data Processing. arXiv:1801.09847.

9. Acknowledgements

Industry Partner: PediaMetrix Inc.

Thank you to our mentors, Can Kocabalkanli, Dr. Reza Seifabadi & Dr. Özgür Güler, for all of your assistance, support, and encouragement.