Estimating and Reconstructing a T-Shirt's Configuration into Simulation through Tracking Fiduciary Markers for the Purpose of Laundry Folding

Ruben Zhao

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2012-140
http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-140.html

May 30, 2012

Acknowledgement

This research paper would not have been possible without the contributed work from Wei Wang and Siyu E as partners in the project. In addition, we would like to acknowledge our supervisors, Professor Pieter Abbeel and John Schulman, for their guidance, support and assistance in our work. Also, I would like to acknowledge the assistance and insight from Professor Jitendra Malik, Dr. Christian Bersch and Daniel Aranki on our project. Lastly, this project would not have been able to be done without the existence of great open source libraries in Bullet Physics, OpenCV, ARToolKitPlus and the Kinect.

Estimating and Reconstructing a T-Shirt's Configuration into Simulation through Tracking Fiduciary Markers for the Purpose of Laundry Folding

By

Ruben Zhao

Partners: Wei Wang, Siyu E

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Engineering

in

Electrical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Pieter Abbeel

Professor Jitendra Malik

Spring 2012

Abstract

We are considering the problem of bringing a piece of clothing from a known initial start state to a desired spread state autonomously using the PR2 (Personal Robot 2). In analyzing this problem, we will be using a simulation engine, Bullet Physics, to simulate the behavior of our clothing article and on it develop a series of manipulations by which the clothing article can be brought to the spread state in simulation. As this simulation will become significant in deciding the best action for the PR2 to take, we worked to ensure the simulation would be as realistic as possible. Additionally, in order for the simulation to have meaning to the real configuration of the clothing article, we will develop an algorithm by which the PR2 will be able to generate an accurate guess of the configuration of the clothing article through its cameras. In our work, this generation will be done on visually marked up clothing so as to allow easier detection of important locations of the clothing article. Lastly, the simulation is used to generate the control inputs for the PR2's motors and servos so that the PR2 will be able to act according to what the simulation outputs.

Introduction

In the world of Personal Robotics today, simulators have a great value. Through the use of simulation, developers are able to test their manipulation and vision algorithms before implementing it on their expensive Personal Robots and Personal Robots themselves are able to use the simulation in practice to learn what action it should take in a certain scenario. The main criterion for these simulators is that they must be able to realistically and reliably reproduce all the important actions that the robot is trying to do. Here lies our main motivation in why creating such a simulator is valuable to the future of Robotics where our value comes with how accurate we are able to simulate reality. In our work, we will be creating a simulator using the Bullet Physics engine to model the manipulation of clothing articles so that the PR2 (Personal Robot 2) will be able to use it and learn what is the next best action in its clothes folding application. In addition, our simulator will be used to test the vision and manipulation algorithms of other researchers on the PR2 to ensure correctness and accuracy before moving it on to the actual robot.

As stated, our simulator will be using Bullet Physics as its physics engine to determine the effects of cloth manipulations. Bullet Physics is an open source physics engine that is readily accessible in C++. In addition it is used in many applications today from video games to movies where the same realism demand as our project is required. Although its counterpart, Maya, is a specialized engine for clothing and thus probably more realistic for our purposes, the fact that Bullet Physics was both readily programmable and that it was a general physics engine (which would allow for easy transitions to other non-clothing applications in the future), we decided to use Bullet Physics for our physics engine. The first step for our project was to understand how this physics engine worked and understand the various parameters that controlled the different environment of simulation. In addition, we had to understand how to create manipulations of clothing on this engine and also how to gauge how realistic these manipulations were. This second part was done by using an existing manipulation algorithm for cloth folding built on the PR2 and testing it in Bullet Physics while attempting to replicate the original process as accurately as possible.

The second question for our simulator is how to initialize the clothing article's orientation. For our simulator to have meaning to the real world, we needed to create a method by which the initial configuration of the clothing article can be perceived in the real world and then transported into simulation. This work was done through the use of BCH marked up T-Shirt, creating and saving an exact mesh model of the T-Shirt into Bullet and using ARToolKitPlus to determine the location of each marker at initialization. The returned locations from ARToolKitPlus were then used to manipulate the mesh model in simulation so that it could match the real shirt's configuration. The problem we addressed in this part of the project was how to maximize the detection of markers using ARToolKit and additional computer vision algorithms and also how to minimize the error of the simulated initial configuration from the actual shirt configuration. My contribution on this second part was focused mostly on the simulation side where I had to create an interface by which the simulation would accept a list of marker positions and robustly display the guessed initial configuration from it.

This paper will discuss our approaches to these two problems. The first section will discuss the relevant parameters of Bullet Physics and how they relate to the simulated reactions to manipulations in the clothing articles. The second section of this paper will the methodology by which the cloth's guessed initial configuration in simulation was created and lastly we will discuss the final obtained results and accuracy from the actual shirt configuration to the simulated configuration.

Literature Review

Simulation has long been crucial to determining the next best action for an agent. The use of it to determine actions in laundry folding have been addressed in work by Cusumano-Towner et al.[1], Christian Bersch et al.[2] and more recently, John Schulman et al.[3] In the first two works, the author tackles on the problem of using computer vision accompanied with simulation to determine what action to take in folding laundry. The latter piece focuses more on reproducing the perceived article using computer vision in simulation as accurately as possible and to track the article in simulation as it is manipulated in the real world.

In the work done by Cusumano-Towner, they utilize a Hidden Markov Model (HMM) to estimate the current state of the clothing article. In this process, they use the height/contour of the observed article at each timestep to determine which state the clothing article is in through matching this data with their simulated clothing article. Once they can reliably predict that the clothing article is in a particular state, the robot proceeds to bring the clothing article to a desired state (in the case of a towel, this is the spread state with grippers on adjacent corners). The remaining actions to fold the article are automated from the state they are currently in. In this work, the accuracy of the simulation is not as important as only simple measurements are needed.

The cost is that the number of measurements needed to determine the state is higher and thus more effort is wasted in determining the initial configuration of the clothing article. Despite this, the work done can still very accurately spread out a towel in a short amount of time and then very reliably fold the article.

Christian Bersch et al. add on an extra layer to the above method to include more measurement data through the use of fiduciary markers. These marker positions are the new measurement data and through it, they can determine an approximate configuration for the clothing article through one 360 degree scan of the article at the start. As a result, the number of iterations required to determine the clothing article's configuration is lowered substantially and the rest of the work relies on bringing the clothing article to its desired state. Due to this improved measurement data, the number of grasps for the entire process drops down to an average of 5 per run[2]. Also the accuracy of the runs were also improved as due to the extra measurements, the robot would always know whether it was in the correct state. The cost of their method is that the 360 degree scan of the clothing article brings a lot of time complexity bringing their average run time to almost 20 minutes[2] whereas Cusumano-Towner's group were able to complete their folding in less than 10 minutes[1].

Recently, John Schulman has introduced another method for folding laundry with simulation by using the Kinect and Bullet Physics. His work concentrates on using a real physical simulation to place the article in simulation while updating it through measurements with the Kinect and point cloud data. As a result, he is able to very reliably track the configuration of the clothing article from a set initial configuration. The aim here is that in a later work, with the real physical simulation, the robot will be able to do a look-ahead to determine which action will allow it to reach its goal fastest. As the simulation will be able to run at a much faster rate than the robot can physically act, this will allow for much better planning and overall less actions required to fold the clothing article. The cost here is that for his algorithm to work, the initial state of the clothing article must be set to a known state for the Kinect and the point cloud data to be properly initialized.

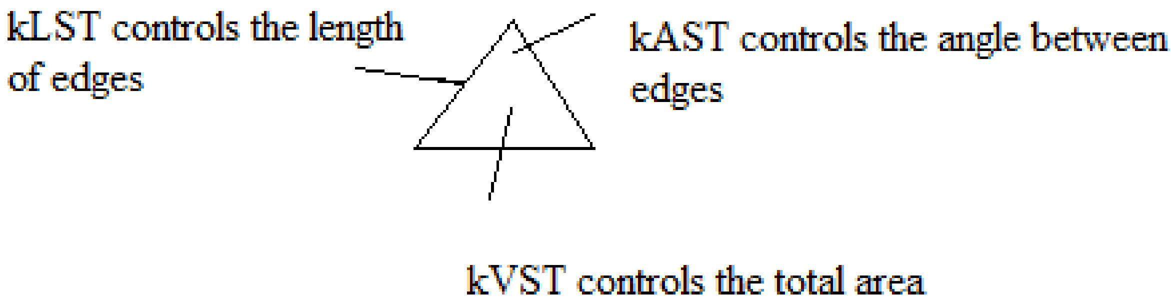
Methodology

In order to create our simulation interface, we had to construct a simulation environment that replicated that of the robot's laundry folding task and also a vision system on our marked clothing articles so that the simulation would be able to discern the initial clothing configuration. As stated previously, our underlying physics engine for the simulation was Bullet Physics and our aim was to create a semi-realistic simulation for our clothing article and allow for the vision tracking to bring the simulation to match reality.

Simulation Engine:

The purpose of the simulation engine is to allow for the tracking algorithm to manipulate the clothing article in simulation into a realistic configuration. To this end, our simulation needed to be semi-realistic and include features such as bending constraint and self-collision while still runnable in real time as the robot needs to be able to keep track of its environment in simulation without it lagging behind. In order to accomplish this, our simulation aspect required three steps – set-up of a basic clothing article, set-up of self-collision, and finally the creation of a scale model of our clothing article into simulation.

During the first stage, we set up a OpenGl scene which utilized Bullet Physics for its callback function to determine how the simulation behaved at each time-step. Our clothing article (initially a towel) was created as a triangle mesh model and Bullet modeled its behavior as a mass-spring system where the clothing article was allowed to stretch/shrink and whenever this happened, it would create a force to counteract this stretch/shrink which was atypical of how normal cloth behaves. In order to manipulate the behavior of the clothing article to match our circumstance, we have three main material properties to manipulate. These were kLST (linear strain), kAST (angular strain) and kVST (volume strain). As shown below, these variables control the strain of their respective names with a value of 1 denoting a 100% stiff spring (no stretching allowed) and a value of 0 implying a flexible spring. A value of 0 however did not imply that the spring could stretch infinitely but rather Bullet Physics defined some amount for maximum stretching.



For the purposes of the cloth simulator, our cloth article should allow minor growth in volume and be flexible in linear and angular stiffness. As cloth should not be completely flexible but still maintain some fluid dynamics, kLST should be set to around 0.4 to 0.5 while kAST is set to 0 and kVST is set to around 0.7. For the cloth, this seemed to produce a fairly realistic simulation. For the later mesh models however, kLST also set a constraint on the rigidity of the overall cloth shape and would cause, for example, the general shape of a T-Shirt mesh to maintain a semi-rigid shape regardless of any manipulations acted upon it. This was because in Bullet, irrelevant links were made across triangles to merge the Thus for the later mesh models of T shirt and other articles, kLST was dropped down to 0 while kVST was increased to 0.9 to try ensure that the deformation was still realistic.

The other aspects of Bullet parameters control the level of detail that the clothing article is simulated with. The parameters that act in this way are the number of triangles in the mesh, and the number of position and velocity (piteration and viteration respectively) iterations conducted at each time step. The number of triangles control how many triangles are used to model the clothing article while the number of iterations indicates how many times at each time-step the position and velocity of each node of the clothing article would be refined by the effects of its nearby nodes. As expected, increasing these parameters create a more realistic simulation but at the cost of time. In our case, as we needed a semi-realistic simulation that runs in real time, we used a low piteration and viteration count (2 and 1 respectively) and decided it was more important to have a larger triangle count (in each case setting this so that the simulation runs at exactly real-time) to ensure that the cloth motion was as fluid as possible. This was because a low number of iterations can mostly be resolved through the tracking of the points while even with perfect tracking; too few triangles would limit the ability of the clothing article to bend.

Vision System:

In order for the simulation to know the initial configuration of the clothing article, we introduced a vision system on a marked up T-Shirt. The T-Shirt was marked with fiduciary BCH markers that could be detected using the ARToolKit library. However, the problem here lies that with a deformable object, ARToolkit's detection algorithm was insufficient and was unable to accurately discern the number of markers we needed. As a result, we introduced a method by which the deformed marker image could be detected through unwarping the image and determining the marker identity through the detected edges. A second problem was that to detect the markers, ARToolkit needed a working distance of approximately 40 cm. The Kinect depth camera however required at least 50 cm to work. As a result, we also have to establish a matching algorithm that allows us to determine the marker ids first at a close distance and match those markers to a further image to determine the depth value of each marker.

In order to unwarp a marker image, a transformation was first made with the corner locations of the marker given by ARToolkit. This transformation was able to bring an arbitrary marker configuration into a square configuration. After this transformation however, the marker was still difficult to identify as due to the warping, the length of each segment was still inconsistent. The second phase of the algorithm was to run an edge detection algorithm to determine where each segment boundary lied. Once found, theses edges were used to segment the marker and determine its bitmap. Due to some occlusion and noise, the found bitmap often did not exactly match the marker's actual bitmap and the last step was to find the closest bitmap from our set of markers using its Hamming distance. As a result, we were able to discern the identity of the marker despite its deformation.

The matching algorithm to determine the depth and identification of each marker was done by first taking a close snapshot (< 40 cm) of the clothing article to determine the identity of each found marker. Then a second snapshot is taken at a further distance which now had the depth of each found marker. In order to match each marker to a found depth and identity, a linear assignment algorithm was applied where the cost of each match was set as the distance moved (after normalizing the image so both had the same width/height and center). By minimizing the sum of all costs with the constraint that each marker can only be matched once, we were able to create a robust method by which we solved the Kinect's minimum depth range and ARToolkit's maximum detect range allowing our tracking system to determine both the depth and identification of each marker found.

Discussion/Results

Simulation Reconstruction:

From our final developed application, we were able to accurately reproduce the T-Shirts configuration for the found markers on the shirt. Through an initial ARToolkit detection, unwarping unidentified markers and then matching marker positions across frames, we were able to identify and manipulate approximately 80% of the markers that were visible to the camera. These manipulated markers were placed in Bullet Simulation and gave an accurate reconstruction of the original configuration for those locations. The problem in reconstruction lied with the unfound markers as these locations can only be estimated by the locations of the found markers which in the case of deformable objects, does not provide much restriction to its location other than that it was obscured from view.

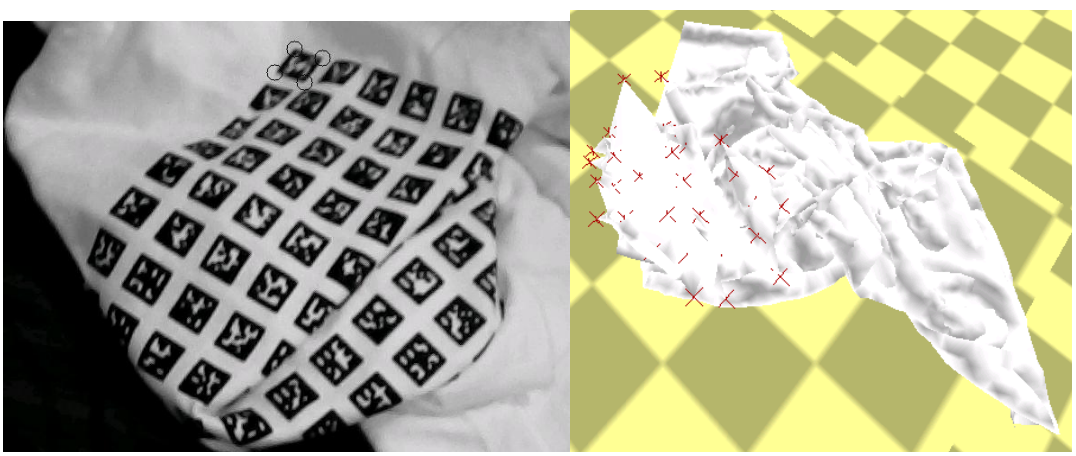


Fig 2. Reproduction of shirt (left) in Bullet Simulation (right) without depth

From figure 2, we can see that in the reconstruction process, we had a fairly accurate reconstruction of the shirt in simulation with two slightly wrong positions when reconstructed without depth information of the markers. In particular, the fold pattern found on the left image can be found also on the right image. The main problem with having no depth comes with identifying the exact x and y world locations from the x and y pixel locations and not the depth itself as for the most part, the depth in the image is unable to be strongly differentiated by the Kinect.

Fig 3. Reproduction of shirt (left) in Bullet Simulation (right) with depth

Figure 3 shows a reconstruction phase with the depth information. With the depth information, we performed a perspective projection from the found pixel locations and the focal length of the Kinect. From this, we returned a much more accurate reconstruction of the shirt as now we have the exact x,y,z world coordinates of the markers rather than an assigned x,y,z coordinate based upon the x,y pixel data. One problem encountered when using the Kinect was that the Kinect's resolution proved to be too poor and at the depth distance, detection of the markers almost completely failed with our original T-Shirt. As both the matching and unwarping algorithms require at least the detection of the markers, we were required to print new larger markers on the T-Shirt so that the Kinect could still detect the marker at the depth distance. As the Kinect was still unable to identify robustly all markers at this distance, the matching algorithm was still required.

There still remains a problem with how to reconstruct the unfound marker positions. This area of the T-Shirt was very loose and Bullet had no way of knowing where it should be in the many positions that it could be. From this we learn that it is very important that:

1. The markers need to be more spread rather than clustered. With a more spread marker pattern, Bullet will be able to better guess the positions of unfound markers. This can be shown in part from the improvement from figure 2 to figure 3. Although it was partly the result of adding depth information, part of the improvement was from the fact that the markers now (although fewer by almost a factor of 2) covered the entire shirt.

2. One image only gives the program limited information. Even for a normal person, seeing the image on the left in figure 2, it will be difficult to accurately guess the exact configuration of the shirt. Thus, we propose that for a more accurate reconstruction, a few manipulations and re-measurements should be conducted to gain more information of the shirt's configuration.

One method to correct the position of the unfound markers is to place it in a location that obscures it from view in Bullet. The problem that we ran into here is that without 100% detection rate of the visible markers, it is impossible to determine what location should be obscured and what location should be visible. In the case presented above, the fold should be obscured but could just as well be visible and unable to be identified by the unwarping algorithm. As a result, our final simulation does not handle undetected markers. This can be seen in figure 3 where the unfound markers were allowed to drop as a result of gravity.

Unwarping:

Through the unwarping algorithm described before, we were able to unwarp and accurately detect marker identities that ARToolKitPlus is unable to detect. ARToolKitPlus works well to detect flat objects. However when the object is bent to a certain degree, it is no longer able to accurately match the marker's features (through SIFT) to the warped marker. Figure 3 shows an example of the unwarping algorithm from start to finish on marker #0.

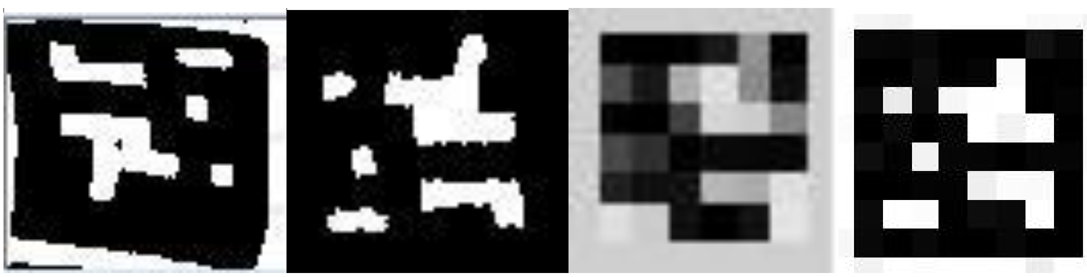


Figure 3 (From left to right):
original image from camera (unidentified by ARToolKitPlus)
Homography Transformation to Square
Edge detection and segmentation based on found edges
Threshold Image and Hamming Distance match to actual marker

Through the above unwarping algorithm, it was also found that for it to work, the markers had to have exactly 7 defined edges in both x and y directions. Overall, most BCH markers fit into this category and thus when choosing markers to place on the T-Shirt, the markers need to be examined removing any markers without 7 edges in both x and y directions. This was not problematic in the final solution as out of the 4000 BCH markers presented, our T-Shirt only required 180 on each side which could be easily picked out from the set.

Matching:

For the matching algorithm, one of the main complications that needed to be resolved was how to deal with the entrance of a new marker into the next frame. As the new marker does not exist in the previous frame, it should not be matched to any previous marker or else the matching of the other markers in the image would be violated. Two methods were attempted to resolve this. The first was to assign a max search range for every unidentified marker in the new frame. This max search range would be set at half the distance between the two nearest markers in the previous frame. The result of this will be that the camera must be moved at a steady pace such that the position of the marker in one frame is relatively close to the position of the marker in the next frame. Additionally as this method is relatively fast (as compared to linear assignment) it will allow for a much higher fps (frames per second) and hence allow for a bit quicker movement of the camera. We found that with this method, the whole process of starting a close ID measurement to a far depth measurement could be done reliably within 5 seconds.

The second method tested was through linear assignment. In this method, each unmatched assignment would be given a base occlusion cost while the costs of the other matches are set as the distance from one marker to the next. As compared to the above method, linear assignment took a bit longer to process with little improvement in accuracy that the max search range was able to give. Thus as a result, we chose to keep the nearest neighbor method with the max search range to deal with the entrance of new markers.

As compared with previous work, the strength of this work is its ability to reproduce the T-Shirt’s configuration in a real physics simulation. With the Customo-Towner paper, they were able to fairly accurately estimate the state of the clothing article through the use of their HMM with a semi-realistic simulation as its state estimator. Our work will be able to more accurately estimate the clothing article’s state through a single image to determine its pose in simulation. From there, the same HMM model can be applied to determine the exact clothing configuration with a more accurate simulator.

As compared with Christian Bersh's paper using the same fiduciary markers, they were able to pinpoint the T-Shirt's configuration also through the marker locations. The benefit this work brings is that because the configuration is reproduced within a real physics simulation, the robot will be able to use the simulator to determine its next best action after a series of simulated manipulations. Here, it will be weighing the cost of the simulation versus the possibility of an inaccurate regrasp (as in Christian Bersh's case) of the clothing article. As processing power increases, the former method will become more and more utilized.

John Schulman's work involving bringing the PR2's environment into simulation will link strongly to our work as currently; his work is unable to determine the original configuration of the clothing article. Once determined through our work, his point cloud tracking algorithm will be able to accurately take over the simulator and bring the clothing article from its initial state to a folded state. The problem here lies that our work will need to return the clothing article's configuration with utmost certainty and will probably require a few manipulations and re-measurements to determine its configuration. As a result, a strong follow-up to this work would be to implement this sequence of manipulations and re-measurements in both the simulation and the real work simultaneously so that the simulator can better learn the T-Shirt's configuration. A second more relaxed method is to determine a cost efficient realistic method to deal with the hidden markers such as either obscuring them from view or guessing from the contours created by the other markers.

Conclusion

In this paper, we have discussed an algorithm to determine the initial configuration of a T-Shirt and to reproduce it in simulation. From our work, the main recurring result was that the more regions of the shirt that were marked up, the more accurately the simulation replicated the shirt.

This does not translate directly to the number of markers as we found that most of the marked locations had little use in conveying the shirt's configuration as its neighbors had already mostly given that information. In addition, it seemed that the information gained by markers at different locations around the shirt was mostly independent on where the marker was physically on the shirt implying that the spread of marker locations was much more important than the locality of the markers.

From this work, general shirt configuration detection methods can be analyzed as in the general method, rather than markers, important features of the shirt are used. From our work, it is conclude that these features be sparse and distinguishable (either by an assignment algorithm or physical look) from one another or there will be a lot of missing data for the simulator. Current state-of-the-art detection methods are only able to detect sleeves and pockets of untextured shirts which would be enough to construct a crude shirt configuration but in order for an accurate reconstruction, the clothing article will need to run through a series of manipulations and measurement updates similar to that of a particle filter as the information is just not there for a one run analysis.

The strength of this work will be that such innovation can be tested against this algorithm for correctness as the markers provide a ground truth for where each location of the clothing article lies. Thus, for the reader, we recommend that you treat this work as a base test for your vision and manipulation algorithms for the process of laundry folding.

Acknowledgements

This research paper would not have been possible without the contributed work from Wei Wang and Siyu E as partners in the project. In addition, we would like to acknowledge our supervisors, Professor Pieter Abbeel and John Schulman, for their guidance, support and assistance in our work. Also, I would like to acknowledge the assistance and insight from Professor Jitendra Malik, Dr. Christian Bersch and Daniel Aranki on our project. Lastly, this project would not have been able to be done without the existence of great open source libraries in Bullet Physics, OpenCV, ARToolKitPlus and the Kinect.

References

[1] Bringing Clothing into Desired Configurations with Limited Perception,

Marco Cusumano-Towner, Arjun Singh, Stephen Miller, James O'Brien, Pieter Abbeel.

In the proceedings of the *International Conference on Robotics and Automation (ICRA)*, 2011

[2] Bimanual robotic cloth manipulation for laundry folding,
Christian Bersch, Benjamin Pitzer, S¨oren Kammel.
In the proceedings of the Intelligent Robots and Systems (IROS), 2011

[3]Tracking Deformable Objects with Point Clouds

John Schulman, Pieter Abbeel

The thesis of Ruben Zhao, titled Estimating and Reconstructing a T-Shirt's Configuration into Simulation through Tracking Fiduciary Markers for the Purpose of Laundry Folding, is approved:

Chair ____________________________________ Date __________________

____________________________________ Date __________________

University of California, Berkeley