Visual Monitoring of Surface Deformations on Objects Manipulated with a Robotic Hand

Fouad F. Khalil, Phillip Curtis and Pierre Payeur
School of Information Technology and Engineering
University of Ottawa
Ottawa, ON, Canada
[fkhalil, pcurtis, ppayeur]@site.uottawa.ca

***Abstract*—Nowadays dexterous manipulation of rigid objects using a robot hand can be achieved fairly well. However, grasping and manipulating deformable objects is still challenging as the force and tactile sensors which are commonly used in such applications can only provide local information about the deformation at the contact points. In this paper, a vision framework is proposed for 3D visually guided grasping and manipulation of deformable objects. This visual monitoring framework, which uses state-of-the-art computer vision methods, provides a robotic hand system with comprehensive monitoring of the deformable object that it manipulates as it tracks its deformation. Stereoscopic vision is used to detect and track in real time the deformation of non-rigid objects in three dimensions and within a complex environment. The technique is tested successfully in real robotic operation conditions using the Barrett hand. The actual object shape is rendered in the 3D virtual environment of the *GraspIt!* robotic simulator which also displays the hand configuration.**

Keywords- dexterous manipulation, deformable objects, 3D vision, deformation estimation.

I. INTRODUCTION

Most of the work in the literature on robotic grasping and handling that incorporates force and torque sensors mounted on a robotic hand has been successfully achieved and reported on rigid objects [1]. In [2], a survey is presented on real-time controlled robotic hand grasping methods for rigid objects, and on the selection of contact points for different types of grasps, whether using hand fingertips only or incorporating the hand palm as well. Among other works reported on rigid objects manipulation, Tegin and Wikander [3] introduced the use of tactile sensing to improve the robotic hand grasping control beyond the traditional capabilities based on information collected from the hand joints torque and position. *GraspIt!* [4] served as their simulation environment to test and evaluate control algorithms using different tactile sensor layouts mounted on the Barrett hand [5]. Morales *et al.* [6] developed a framework based on tactile sensing to guide the Barrett hand for picking up books off a bookshelf. Kragic *et al.* [7] developed a vision-guided robotic hand grasping system that integrates real-time vision sensor with online grasp planning. Later on, Kragic *et al*. [8] also developed a vision-guided robotic hand grasping system that integrates vision sensor with tactile feedback to grasp objects for real world tasks. Peer *et al.* [9] used both vision and force sensors to develop a dexterous tele-manipulation system with the Barrett hand.

Furthermore, three complementary sensing approaches that are vision, force and tactile, have been integrated to perform more elaborate manipulation. Allen *et al.* [10] developed a system with the Barrett hand that can grasp the lid of a canister and unscrew it from the base using a combination of tactile/force and vision sensing. Combining vision data with force and tactile measurements provides the necessary information to achieve dexterous manipulation [11], which is characterized by the capability of controlling the deformation of a manipulated object within the hand workspace. Dexterity involves hand palm and fingers to perform motions with ability and stability. However, the literature that addresses the combination of vision information with force and tactile measurements for robotic hand manipulation processes remains limited.

In contrast to the manipulation of rigid objects, which has been extensively studied in the literature, investigating the manipulation of deformable objects represents a more recent undertaking. In reality, several objects exhibit such characteristics as compliance and alterable shapes. The robotic manipulation of deformable objects represents a challenging problem, and is therefore a significant research area with potential use in several applications.

For deformable objects, the elasticity of the manipulated materials makes force and torque sensors difficult to rely on as only the behavior of contact points can be directly monitored. Therefore, incorporating live visual data is crucial in order to analyze the shape and deformation of the object over its entire surface during the manipulation. The supplementary information provided allows the robot to extract and strategically position the gripper while adjusting the effort applied on the contact points to ensure the stability and the integrity of the object [12]. Hence, the grasping process can be improved significantly using vision feedback where the relationship between the object deformation and the robot hand pose can be effectively estimated.

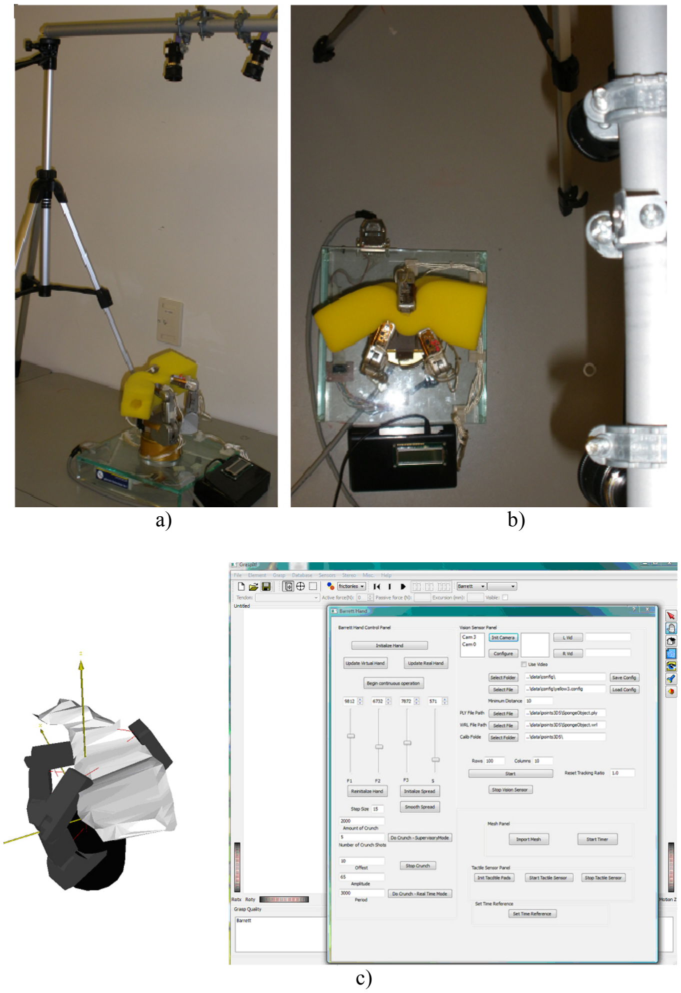
Nastar and Ayache [13] presented a deformable model that can be used to track and analyze deformation of dynamic structures in 2D or 3D medical images. Sullivan *et al.* [14] focused on solving the traffic problem using a parameterized deformable model able to represent different vehicle classes. In robotics, some successful attempts have been reported towards tracking and recovering grasp points on a non-rigid object surface in 1D and 2D using visual sensors, as well as in achieving some feedback control from those points. In an early effort, Sullivan and Papanikolopoulos [15] proposed a model-based approach for tracking of deformable objects based on visual servoing. Active contours of the object boundary are defined by a set of control points that are updated at a certain frame rate during the tracking process such that an energy function is minimized. The idea of using active contours, or "snakes", to track objects in the context of grasping was introduced earlier by Kass *et al.* [16]. Later on, Hirai *et al.* [17] proposed a real-time visual tracking method for the manipulation of deformable planar parts. They combined tactile and visual feedback to control the motion of a deformable object in 2D with respect to the positions of selected markers considered as reference points. They used this information to track the deformation. Foresti and Pellegrino [18] focused on automating the handling of deformable objects using vision techniques only. Their vision system worked along with a hierarchical self-organizing neural network to select proper grasping points in 2D. Other visual tracking methods of deformations in 1D, such as linear object deformation, and in 2D, such as planar object deformation, are reviewed by Henrich and Worn [19]. Similarly, Saadat and Nan [20] present an insight review of some of these visual tracking methods related to specific industrial applications where characteristics differ based on the type of industry and shape of the deformable object.

The technique developed in this paper aims at providing stereoscopic 3D shape information to a robotic system which is involved in dexterous manipulation of deformable objects. Such visual integration can be readily used in a later stage with the force and tactile measurements to achieve better controlled grasp and more elaborate manipulation. A real-time image processing approach is developed that extracts the tridimensional surface deformation of an object positioned in the Barrett hand and displays it in the *GraspIt!* virtual environment. The dynamic reconstruction of the shape of the object in three dimensions is useful to drive the robotic interaction and hence to control the object's deformation.

II. INTRATED VISION, SIMULATOR AND ROBOT HAND

The computational platform used in this framework is *GraspIt!* which is an interactive simulation, planning, analysis, and visualization virtual environment for robotic hand grasping and manipulation, introduced by Miller and Allen [4]. *GraspIt!* is considered as one of the major software platforms for testing and evaluating robotic grasps. Its features satisfy the guidelines presented by Loffler *et al.* [21], which are required for robot manipulator control software development in terms of flexibility, real-time support and modularity. This software package covers several commercial well-known robotic hands, including the Barrett hand which can be simulated while manipulating an object within a virtual 3D workspace. The contact selection mechanism and collision detection engine in the software can prevent the robot members and grasped object from passing through each other and hence determine the proper contact locations. The grasp analysis class can evaluate grasps using a variety of different quality measures. The dynamics engine can compute contact and friction forces in real-time to allow the evaluation of user-written robot control algorithms, and therefore provides the ability for a combination of grasp planner/evaluator [22].

As an extension to the simulator, our integrated framework allows a physical interconnection between the simulator and the real Barrett hand. With this solution, the robot hand can mimic the simulated one, and vice versa, both in the supervisory and real-time modes of operation of the hand. For our experimentation, the manipulation and deformation of objects is achieved using a real Barrett hand, connected to the *GraspIt!* simulator. The experimental system configuration is shown in Figure 1.



Cam 3
Cam 0
..\data\config\
..\data\config\yellow3.config
Minimum Distance 10
PLY File Path ..\data\points3DS\SpongeObject.ply
WRL File Path ..\data\points3DS\SpongeObject.wrl
Calib Folde ..\data\points3DS\
Rows 100 Columns 10
Reset Tracking Ratio 1.0
9812 6732 7872 571
Step Size 15
2000 Amount of Crunch
5 Number of Crunch Shots
10 Offest
65 Amplitude
3000 Period
Barrett

Figure 1. The system setup with steroscopic vision sensor and Barrett hand manipulating a deformable object: a) lateral view, b) vertical view, and c) the *GraspIt!* simulator interface.

The integrated robotic platform is also extended with a stereoscopic vision system that utilizes two Point Grey Research Flea 2 industrial cameras [23]. They provide non-compressed video streams via a FireWire bus (IEEE-1394b), with a resolution up to 640x480 pixels at 30 fps. However, due to the time required by the algorithm to execute, a 320x240 pixels resolution is used to achieve real-time tracking of surface deformation. Considering the size of the objects, the cameras are positioned at about 20 cm from each other, with an angle of about 30° between their respective optical axes to ensure that the deformed object covers a large part of the image planes.

III. Image Processing and surface Shape Estimation of a Deformable Object

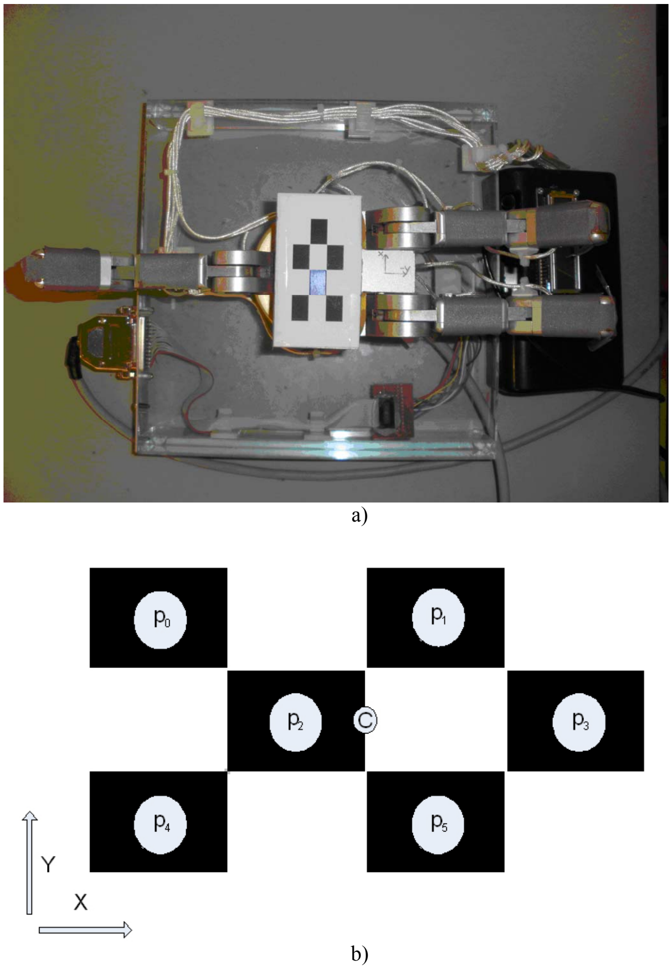
There are important differences between the way rigid and deformable objects can be handled [17]. While the manipulation of a rigid object requires only the control of its location, the manipulation of a deformable object requires controlling both the location of the object and its deformation. In the handling process of rigid objects, grasping and manipulation can be performed independently. Hence, grasping of a rigid object requires the control of grasping forces only, while manipulation of a freely moving rigid object results in the change of its position and orientation only. On the other hand, grasping and manipulation interfere with each other in the handling of deformable objects. Grasping forces yield the deformation of a non-rigid object, which may change the shape and location of the object. Consequently the contact between fingers and the object may be lost and grasping may be compromised due to the deformation at the fingertips. Therefore, in the handling of deformable objects, grasping and manipulation must be performed simultaneously. This reality imposes severe constraints on live visual tracking of both operations.

This section details the image processing approach that is used to extract the information necessary for 3D reconstruction of a deformable object surface in real-time. The same procedure is applied on the respective video streams originating from two cameras in a stereoscopic setup.

A. *Calibration*

In order to extract the necessary parameters that define the stereo camera system in the robotic work cell, calibration must be performed on the cameras and between the stereoscopic vision setup and the robot hand. The intrinsic calibration is initially performed separately on each camera following Zhang's approach [24]. The method provides the radial lens distortion, the image center, and the focal length of each camera. Next, the extrinsic calibration builds upon the methods of Chen *et al.* [25] and Ihrke *et al.* [26] to estimate the rotation matrix and the translation vector defining the relationship between the cameras. The procedure is integrated in a computational framework developed by Bériault *et al.* [27]. As the system is dedicated to operate in complex environments where the image dynamically contains not only the deformable objects but also the robotic hand and fingers as well as other miscellaneous objects, and where the illumination conditions can significantly influence the texture properties of the object which consequently affect the estimation of the surface shape, consideration is also given to proper exposure parameters for the sensors.

Once the stereo pair has been calibrated, the calibration between the workspace of the Barrett hand and the stereo pair is determined. To achieve this, a 4x3 checkerboard calibration pattern is positioned in the palm of the Barrett hand, as shown in Figure 2a. Due to the relatively small area of the calibration pattern compared to the size of the image, the user manually selects the region in which the pattern is located. The centers of the black squares in the checkerboard pattern are used as features, and are sorted according to their known geometrical distribution, as shown in Figure 2b. One of the squares is color-coded to prevent confusion in the distribution of the pattern. The centers are found using the mean image value as a threshold, and blob analysis.


a)

b)

Figure 2. a) Calibration checkerboard sitting in the palm of the Barrett hand, and b) sorted and labeled checkerboard pattern.

Once points from the checkerboard pattern are sorted for each camera in the stereo pair, an implicit correspondence is assumed (i.e. point p_i in the left view is assumed to match p_i in the right view), and the respective 3D point coordinates, $\vec{p}_i$, for each corresponding set of blobs are then calculated via mid-point triangulation. This provides the necessary information to extract the rotation and translation vectors of the transformation from the stereo pair to the calibration pattern. To obtain these estimates, the X direction vector is first determined as shown in Eq. (1). Next, the Y direction vector is determined, while ensuring that it remains orthogonal to the X vector (see Eq. (2)). Finally the Z direction vector is computed as the cross product between the X and Y direction vectors as shown in Eq. (3). The rotation matrix which links the calibration pattern reference frame to the stereo pair reference frame is given by the concatenation of the direction vectors into a matrix, as shown in Eq. (4).

$$\vec{X} = \frac{(\vec{p}_1-\vec{p}_0)+(\vec{p}_3-\vec{p}_2)+(\vec{p}_5-\vec{p}_4)}{\left\|(\vec{p}_1-\vec{p}_0)+(\vec{p}_3-\vec{p}_2)+(\vec{p}_5-\vec{p}_4)\right\|} \tag{1}$$

$$\begin{aligned} \vec{Y}' &= (\vec{p}_0-\vec{p}_4)+(\vec{p}_1-\vec{p}_5) \\ \vec{Y} &= \frac{\vec{Y}' - (\vec{Y}' \cdot \vec{X})\vec{X}}{\left\|\vec{Y}' - (\vec{Y}' \cdot \vec{X})\vec{X}\right\|} \end{aligned} \tag{2}$$

$$\vec{Z} = \vec{X} \times \vec{Y} \tag{3}$$

$$R = \begin{bmatrix} \vec{X} & \vec{Y} & \vec{Z} \end{bmatrix} \tag{4}$$

Finally the translation between the stereo camera pair to the calibration pattern is calculated by determining the center of the calibration pattern, *C*, which corresponds to the center of the Barrett hand's palm. Computing the mean value of the points p_0 to p_5 works well to estimate the *y* and the *z* components. The *x* component needs to be handled differently, because a simple average among the six points produces a value offset in the negative *x* direction from *C*. To compensate for this, the center of points p_0, p_1, p_4, and p_5 is calculated, and is averaged with the center of points p_2 and p_3, yielding the coefficients shown in Eq. (5).

$$T = \begin{bmatrix} \frac{1}{8}(p_{0x}+p_{1x}+p_{4x}+p_{5x})+\frac{1}{4}(p_{2x}+p_{3x}) \\ \frac{1}{6}\sum_{i=0}^{5} p_{iy} \\ \frac{1}{6}\sum_{i=0}^{5} p_{iz} \end{bmatrix} \tag{5}$$

$$Q_{CB/SP} = \begin{bmatrix} R & & & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

The rotation matrix, *R*, and the translation vector, *T*, can then be combined into a homogenous transformation matrix, $Q_{CB/SP}$, (Eq. (6)), representing the transformation from the stereo pair to the calibration pattern. The homogenous transformation matrix which goes from the Barrett hand to the calibration pattern (Eq. (7)) is obtained based upon the information on the Barrett hand reference frame, as defined in *GraspIt!*, as well as the thickness of the calibration pattern, that is 4mm. The final transformation from the Barrett hand to the stereo pair is calculated as shown in Eq. (8).

$$Q_{CB/BH} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

$$Q_{SP/BH} = Q_{CB/BH} \cdot Q_{CB/SP}^{-1} \tag{8}$$

B. 3D Surface Shape Estimation

Once the calibration is performed, the acquisition of 3D data on the surface of the deformable object using the stereo pair can commence. The first step which is performed is the rectification of the stereo pair images [28]. Following rectification, segmentation of the camera images is performed by directing the user to select the object of interest in the first frame of each camera's video feed, which is tracked via colour histogram in future frames. From this selected region, the colour histogram representing the object which we wish to monitor is calculated based upon the normalized RGB colour space. The latter is selected since it is fairly immune to changes in lighting, while allowing for a better segmentation of object when compared to the HSV colour space. To perform the segmentation of the object of interest from the background, histogram back-projection is used, followed by blob analysis, to eliminate small objects and objects far away from the selected region.

With the object now segmented from the image in both video feeds from the stereo pair cameras, Konolige's block matching stereo algorithm [29], is used to calculate the disparity map. The rejection of areas which are not of interest is performed using the segmented objects in each video feed as masks to the disparity map. To improve computational performance for the creation and the rendering of the 3D surface mesh, as well as minimize the effects of noise from the stereo algorithm, the disparity map is sampled at regular intervals along rows and columns, taking into account the starting and ending point of the objects contour on the disparity map along each row, to produce vertices for the 3D representation, as shown in Figure 3.

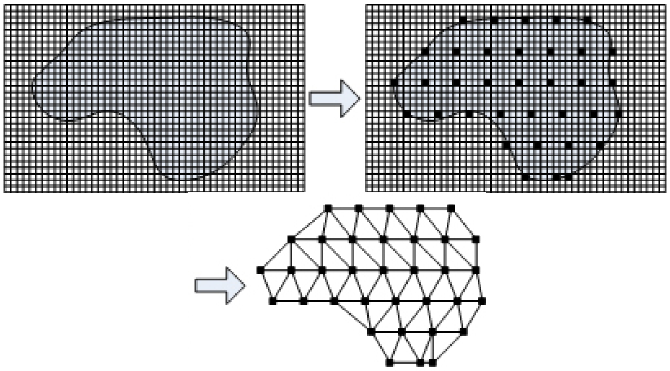


Figure 3. Example of a disparity map (left), its uniform sampling (center), and the resulting mesh (right).

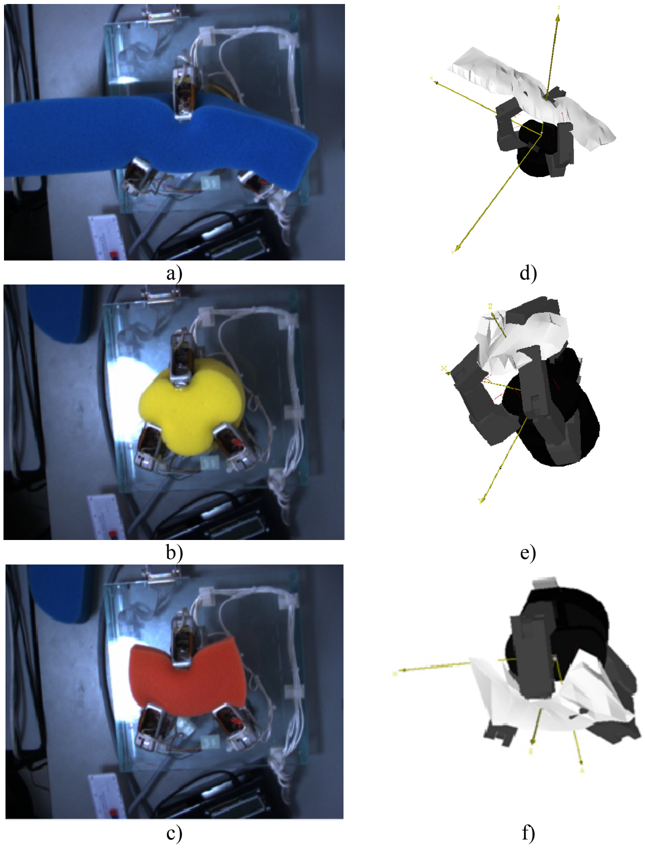
IV. Experimental Results and Analysis

Several tests were conducted with the integrated experimental setup described in section II. At each test cycle the video information is processed and the corresponding mesh that represents the object's surface deformation is generated and updated to reflect the object during the manipulation. It is then displayed live on the *GraspIt!* simulator platform. Under the operating conditions described, the video processing runs at approximately 7 frames per second when running on a 3.4 GHz Pentium 4 CPU. This frame rate is sufficient for most dexterous robotic manipulation where interaction occurs at reduced pace to ensure stability of the grasp.

The deformable objects considered here are of four different pieces of foam with plain color, as depicted in Figures 4a, 4b, 4c and 5a, respectively. They are all exhibiting similar elastic characteristics. To test the object elasticity in this 3D visual monitoring algorithm, the hand starts contracting and relaxing its grip on the object repetitively over a period of time lasting several minutes. The 3D elasticity monitoring algorithm was verified with these objects. Figures 4d, 4e and 4f show the corresponding 3D reconstructions in the *GraspIt!* graphical environment. The display of the objects optimizes the visual information about the reconstructed objects' surface using the axes rotation and zooming features of *GraspIt!*.

Figure 5 shows the evolution of the manipulation involving the fourth test object when an increasing force is applied by the hand on the object at three instances in time covering the object from low deformation (Figure 5a), to moderate deformation in which the object starts to bend (Figure 5b), and finally when the object is completely deformed (Figure 5c). In the extreme situation (Figure 5c), the mesh points surrounding the contact points with the fingers of the Barrett hand cannot be extracted in sufficient number to allow 3D reconstruction of the full surface shape because of the occlusions that result from the fingers structure. However, knowledge about the location of the surface in these areas can readily be computed from the robotic hand kinematics.

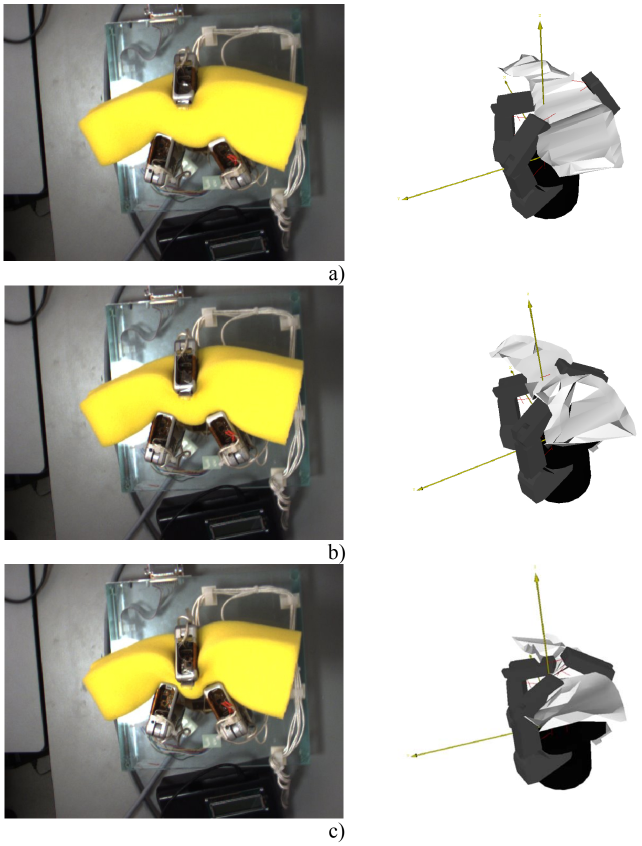
The video processing algorithm adopted in this framework can extract points based upon the colour and texture of the object itself using the colour histogram and block matching stereo algorithm as explained in section III. The image processing and 3D reconstruction parts of the algorithm generate a set of 3D points that dynamically follow the evolution of the shape of the surface of the object. In order to observe the deformation, a surface mesh is computed directly from the set of 3D points. The 3D mesh of the object is produced based on the known geometrical relationships between the sample points, as illustrated in Figure 3. The mesh is displayed in real-time by the *GraspIt!* virtual environment to validate the proposed solution.


a) d)

b) e)

c) f)

Figure 4. a-c) Three tested objects manipulated by the hand by applying forces, and d-f) the corresponding live graphical display of estimated 3D surface shapes.


a)

b)

c)

Figure 5. The fourth tested object manipulated by the hand with different forces (left), and the corresponding live graphical display of estimated 3D surface shape at different compression stages (right).

This 3D visual deformation monitoring algorithm will further be integrated with the tactile and force feedback taken from sensors mounted on the hand fingers and palm to perform the overall dexterous manipulation. The force/tactile sensors have a crucial role in the later stage to refine the pose/orientation of the hand and to ensure stability while performing the manipulation process. It is expected that fusing vision/force and touch information in the feedback loop can provide the necessary control law to allow on-line grasp planning and dexterous manipulation in a more efficient and effective way.

V. Conclusion

This paper presents a 3D vision algorithm that copes with object deformation and fully integrates visual 3D deformation monitoring with the *GraspIt!* simulator and the Barrett hand. This integrated solution provides in real time the necessary information required to track the surface deformation of a non-rigid object manipulated by the Barrett hand in a real robotic work cell. The proposed visual deformation monitoring system demonstrated sufficient robustness for our application in its initial testing stage. Further development will aim at increasing the density of reconstructed points to provide more accurate surface shape estimation, as well as deal with a wider range of working conditions, in terms of the deformable object attributes and elasticity.

ACKNOWLEDGEMENTS

The authors acknowledge the support from the Natural Sciences and Engineering Research Council of Canada to this research.

References

[1] A. Bicchi and V. Kumar, "Robotic grasping and contact: a review," *Proc. IEEE Conf. on Robotics and Automation*, pp. 348-353, San Francisco, CA, 2000.

[2] Z. Yu and J. Gu, "A survey on real-time controlled multi-fingered robotic hand," *Proc. Canadian Conf. on Electrical & Computer Eng.*, pp. 975-980, 2008.

[3] J. Tegin and J. Wikander, "Simulating tactile sensors in robotic grasping," *Proc. Third Swedish Workshop on Autonomous Robotics*, 2005.

[4] A. T. Miller and P.K. Allen, "*GraspIt!* a versatile simulator for robotic grasping," *Robotics and Automation Magazine*, vol. 11, pp. 110-112, 2004.

[5] ---, Barrett BH8 Robotic Hand, *http://www.barrett.com/robot*

[6] A. Morales, M. Prats, P. Sanz and A. Pobil, "An experiment in the use of manipulation primitives and tactile perception for reactive grasping," *Proc. of the Robotics: Science & Systems Manipulation Workshop - Sensing and Adapting to the Real World*, 2007.

[7] D. Kragic, A.T. Miller, and P. K. Allen, "Real time tracking meets online grasp planning," *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 3, pp. 2460-2465, 2001.

[8] D. Kragic, S. Crinier, D. Brunn, and H. I. Christensen, "Vision and tactile sensing for real world tasks," *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1545–1550, Seoul, Korea, 2003.

[9] A. Peer, B. Stanczyk, U. Unterhinninghofen, and M. Buss, "Tele-assembly in wide remote environments," *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Beijing, China, 2006.

[10] P. K. Allen, A. Miller, P. Y. Oh, and B. S. Leibowi, "Integration of vision, force and tactile sensing for grasping," *Int. J. of Intelligent Machines*, vol. 4, pp. 129-149, 1999.

[11] A. Bicchi, "Hands for dexterous manipulation and robust grasping: a difficult road towards simplicity," *IEEE Trans. on Robotics and Automation*, vol. 16, pp. 652-662, 2000.

[12] F. F. Khalil and P. Payeur, "Robotic interaction with deformable objects under vision and tactile guidance - a review," *Proc. IEEE Int. Workshop on Robotic and Sensors Environments*, pp. 1-6, Ottawa, ON, 2007.

[13] C. Nastar and N. Ayache, "Fast segmentation, tracking, and analysis of deformable objects," *Proc. 4th Int. Conf. on Computer Vision*, Berlin, Germany, pp. 275–279, 1993.

[14] G. D. Sullivan, A. D. Worrall, and J. M. Ferryman, "Visual object recognition using deformable models of vehicles," *Proc. Workshop Context-Based Vision*, Cambridge, MA , pp. 75–86, 1995.

[15] M.J. Sullivan and N. P. Papanikolopoulos, "Using active deformable models to track deformable objects in robotic visual servoing experiments," *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 4, pp. 2929-2934, 1996.

[16] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *Int. J. of Computer Vision*, vol. 1, pp. 321-331, 1987.

[17] S. Hirai, T. Tsuboi, and T. Wada, "Robust grasping manipulation of deformable objects," *Proc. IEEE Symp. on Assembly and Task Planning*, pp. 411-416, 2001.

[18] G.L. Foresti and F.A. Pellegrino, "Automatic visual recognition of deformable objects for grasping and manipulation," *IEEE Trans. on Systems, Man, and Cybernetics: Applications and Reviews*, vol. 34, pp. 325-333, 2004.

[19] D. Henrich and H. Worn, Eds., *Robot manipulation of deformable object*, Springer-Verlag, London, 2000.

[20] M. Saadat and P. Nan, "Industrial applications of automatic manipulation of flexible materials," *Int. J. of Industrial Robots*, vol. 29, pp. 434-442, 2002.

[21] M. S. Loffler, V. Chitrakaran, and D. M. Dawson, "Design and implementation of the robotic platform," *J. of Intelligent and Robotic Systems*, vol. 39, pp. 105-129, 2004.

[22] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 2, pp.1824-1829, Taipei, Taiwan, 2003.

[23] ---, Point Grey Research Flea2 Cameras, *http://www.ptgrey.com*

[24] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1330-1334, 2000.

[25] X. Chen, J. Davis, and P. Slusallek, "Wide area camera calibration using virtual calibration objects," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 520-527, Hilton Head, SC, 2000.

[26] I. Ihrke, L. Ahrenberg, and M. Magnor, "External camera calibration for synchronized multi-video systems," *Proc. Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision*, pp. 537-544, 2004.

[27] S. Bériault, P. Payeur, and G. Comeau, "Flexible multi-camera network calibration for human gesture monitoring," *Proc. IEEE Int. Workshop on Robotic and Sensors Environments*, pp. 125-130, Ottawa, ON, 2007.

[28] A. Fusiello, E. Trucco, and E. Verri, "A compact algorithm for rectification of stereo pairs," *Machine Vision and Applications*, Springer-Verlag, vol. 12, pp. 16-22, 2000.

[29] K. Konolige, "Small vision systems: hardware and implementation," *Proc. Int. Symp. on Robotics Research*, Hyayma, Japan, pp. 111-116, 1997.