UNIVERSITY OF CALIFORNIA, SAN DIEGO

**Non-Rigid Surface Detection for Gestural Interaction with Applicable Surfaces**

A thesis submitted in partial satisfaction of the
requirements for the degree
Master of Science

in

Computer Science

by

Andrew Ziegler

Committee in charge:

Professor Serge Belongie, Chair
Professor David Kriegman
Professor Jürgen Schulze

2012

The thesis of Andrew Ziegler is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

Chair

University of California, San Diego

2012

DEDICATION

To Sarah, for supporting me throughout all my endeavors.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to acknowledge Serge Belongie and David Kriegman for being excellent mentors and guiding me through my studies at UCSD.

This thesis is based on a publication which I coauthored with Serge Belongie: "Non-Rigid Surface Detection for Gestural Interaction with Applicable Surfaces" at the IEEE Workshop on Applications of Computer Vision (WACV), Breckenridge, CO, USA, January 2012.

ABSTRACT OF THE THESIS

**Non-Rigid Surface Detection for Gestural Interaction with Applicable Surfaces**

by

Andrew Ziegler

Master of Science in Computer Science

University of California, San Diego, 2012

Professor Serge Belongie, Chair

In this work we present a novel application of non-rigid surface detection to enable gestural interaction with applicable surfaces. This method can add interactivity to traditionally passive media such as books, newspapers, restaurant menus, or anything else printed on paper. We allow a user to interact with these surfaces in a natural manner and present basic gestures based on pointing and touching.

This technique was developed as part of an ongoing effort to create an assisted reading device for the visually impaired. However, it is suited to general applications and can be used as a practical mechanism for interaction with screenless wearable devices. Our key contributions are a unique application of non-rigid surface detection, a basic gesturing paradigm, and a proof of concept system.

# Section 1

# Introduction

In recent years the size of portable devices such as smart phones and personal digital assistants have remained essentially unchanged since they rely on screens to provide visual feedback to the user. In addition there must be some mechanism for the user to provide input. Recently there has been a trend towards using touch screens in these devices, making the screen the sole limiting factor.

Screen-less devices provide the ultimate in portability since there is no arbitrary limit on how small they may become. However, there is not an obvious method to interact with such devices nor a way to provide visual feedback to the user. One approach to cope with this is to replace the screen with a projected user interface and use hand gestures for interaction [14]. This approach requires a surface on which the display may be projected and will not work in brightly lit environments. Another strategy is to abandon displays altogether and rely on the user's imagination to provide "visual feedback" [7]. This is an interesting concept, but puts the burden of imagining the display on the user. Neither of these approaches are accessible to the visually impaired and in this work we develop a novel approach that is useful to the sighted and visually impaired alike.

We suggest using whatever surface the user has available to them as the source of visual feedback as well as the input medium. This could be a newspaper, restaurant menu, book, magazine or even the nutrition label on a can of soup. We make only the assumptions that the surface is isometric with the plane, which is equivalent to there being zero Gaussian curvature everywhere on the surface. Our

method allows a user to interact with these surfaces in a natural manner through pointing gestures. In this way the presence of a screen-less device becomes less pronounced to the user and from his perspective the previously passive surface becomes interactive.

Imagine a visually impaired user sitting in a restaurant touching a menu to have entrée items read aloud. A sighted user who notices an intriguing article in a magazine and uses a hand gesture to virtually clip it. A tourist in a foreign country reading a newspaper touches a news headline to have it translated audibly into her own language. Our method provides a mechanism for this type of interaction with these surfaces that are already ubiquitous in daily life.

In the next section we review related work in more detail. In section 2 we describe our method and the assumptions we make. In section 3 we present a basic gesturing paradigm and describe how metadata can be used to create this interactive experience. Finally, in section 3.4 we discuss our proof of concept implementation.

## 1.1 Related Work

There many examples of systems that allow the user's hands or fingertips to act as an input device [1, 7, 11, 12, 14, 15, 21, 25].

In [14] Mistry et al. present a wearable projector camera system that attempts to make information that is typically confined to paper and screens tangible through hand gestures. In the same vein we describe a method to allow users to interact with information printed on paper. Unlike their method, ours is designed to work with actively deforming surfaces and the information is made seamlessly tangible without visual augmentation.

In [7] Gustafson et al. the authors introduce the notion of an imaginary surface. They challenge the need for any display at all and propose that a user's imagination can provide the necessary visual feedback. In their system a user extends his thumb and forefinger to create an "L" shape. The thumb and forefinger represent basis vectors of an imaginary planar surface. With his remaining hand

the user can gesture to draw on the surface. User studies showed that this activity becomes less precise as the user's drawing hand moves farther from the origin of the surface. In our system users have the benefit of the familiar appearance of printed media to provide visual feedback and gestures are kept simple with limited reliance on the user's imagination.

In [25] Zhang presents a system that uses an ordinary piece of paper to allow users to control a computer by pointing. This work is similar to ours, but with the computer monitor as the source of visual feedback. The method relies on quadrangle detection and uses a homography to map the user's detected fingertip into a rectified fronto-parallel planar reference frame. This technique requires that at all times the paper be rigid, planar, rectangular and no edge can be fully occluded or out of frame. Our method works with arbitrarily shaped surfaces that may be actively deforming and does not require a complete view of the surface for interaction. This is essential when developing a device for the visually impaired since it will be difficult for these users to frame the surface with the camera completely.

# Section 2

# Non-Rigid Reference Frames

Here we describe the notion of a non-rigid reference frame. When viewing the perspective projection of an actively deforming non-rigid surface, the location of features can change dramatically. Examples of this include newspapers, books and magazines warping simply from the pressure of a person's hand holding them comfortably. These surfaces may warp further if a person adjusts his grasp or touches a point on the surface; this is demonstrated in Figure 2.3. In our system we want to give users a method of gestural interaction with such actively deforming surfaces and thus we must find a stable reference frame to detect gestures. To cope with this we seek a point to point mapping from the perspective projection of a warped surface to a rigid planar reference frame. In effect this mapping creates a stable reference frame in the presence of active deformation. We refer to this effect as a non-rigid reference frame which is illustrated in Figure 2.1.

## 2.1 Applicable Surfaces

Applicable surfaces are those that are isometric with the plane. These surfaces are commonplace in man-made environments since any inextensible surface cut from a plane is by definition isometric with the plane. This means anything made from a sheet of paper is naturally modeled as an applicable surface. Thus for our system it is a reasonable assumption that the surfaces with which users will interact are applicable surfaces. We use the isometry to find a point to point

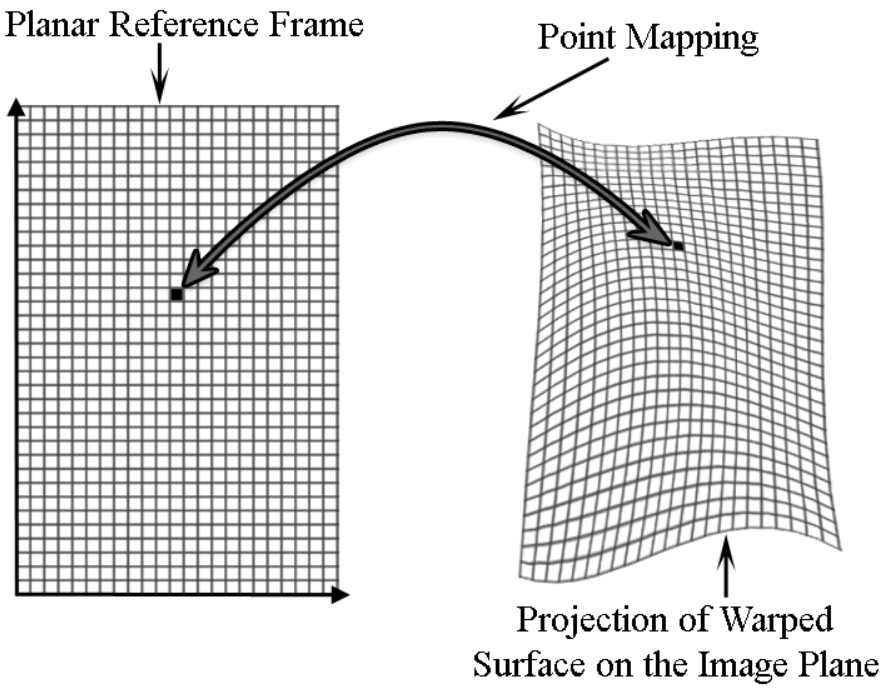


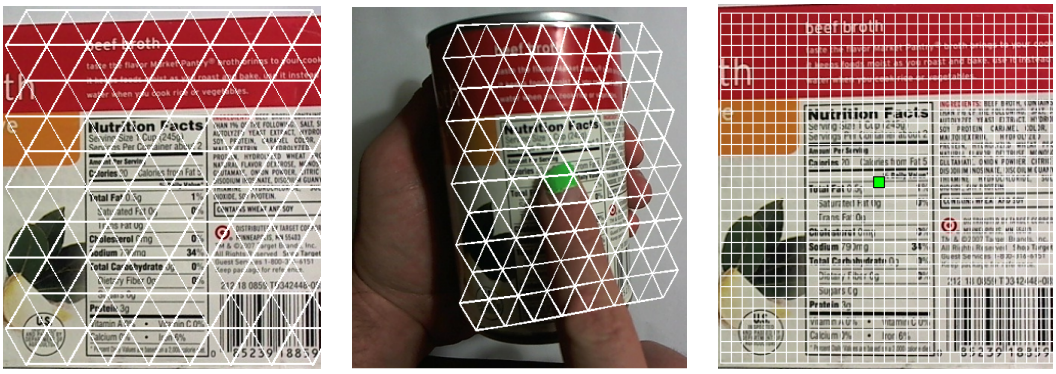

**Figure 2.1**: Illustration of a non-rigid reference frame.

mapping from the perspective projection of a warped applicable surface.

## 2.2 Non-Rigid Surface Detection

We use non-rigid surface detection to approximate the point to point mapping. Pilet et al. [17] describe a finite element method that gives a good approximation to this mapping, originally intended for graphical augmentation [18].

The method presented in [17] requires a planar reference view of the surface and feature point correspondences with a potentially deformed view of the surface. First, a triangulated 2D mesh $M$ of hexagonally connected vertices is generated to cover the planar reference frame. Using the set $C$ of feature correspondences between the two images Pilet et al. sought a transformation $T_S$ to warp the undeformed mesh $M$ onto the target image such that the sum of squared distances of the subset $G \subset C$ of inlier correspondences is minimized while keeping the deformation as smooth as possible. To accomplish this the locations of detected



(a) (b) (c)

**Figure 2.2**: A user touches the nutrition label on a can of soup. (a) The planar reference mesh. (b) The resulting warped mesh from non-rigid surface detection. (c) The mapped location of the colored marker's centroid in the planar reference frame.

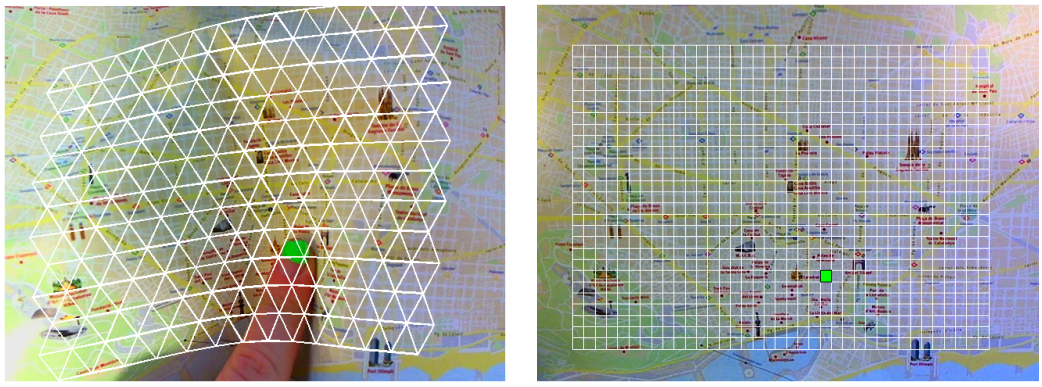
corresponding features in the reference view are represented by the barycentric coordinates of the mesh triangles containing them. They proceed to define $T_S$ as a point to point mapping parameterized by the state matrix $S = (X, Y)$ where $X$ and $Y$ are column vectors containing the pixel coordinates of the warped mesh vertices. The mapping is defined to be

$$T_S(p) = \sum_{i=1}^{3} B_i(p) \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{1}$$

where $p$ is a point on the original surface, the $B_i(p)$ are the barycentric coordinates from the reference mesh and the $[x_i, y_i]^\top$ are the pixel coordinates of the containing triangle's vertices with respect to $S$.

In our system, we seek a point to point mapping from the warped view back to the planar reference view. Given a point contained within the warped mesh we find the barycentric coordinates of the point with respect to the vertices in the warped mesh and use the known locations of the corresponding vertices in the reference mesh to find the inverse mapping. More precisely we define an inverse transformation

$$T_S^{-1}(p) = \sum_{i=1}^{3} B^i(p) \begin{bmatrix} x'_i \\ y'_i \end{bmatrix} \tag{2}$$



(a) (b)

**Figure 2.3**: A tourist points to a location on a city map. (a) The map distorted by the pressure from his finger and the appropriately warped mesh. (b) Despite the distortion, the location of the colored marker's centroid is correctly located in the planar reference frame.

where $p$ is a point contained within the warped mesh, the $B^i$ are the barycentric coordinates of the point computed with respect to the reference vertices and the $[x'_i, y'_i]^T$ are the locations of the vertices in the reference mesh. Essentially the warped surface is broken up into small local rigid reference frames and $T_S^{-1}$ is a piecewise function mapping any point in the warped reference frame onto its corresponding position in the reference view. Since a mesh is used, as the number of facets increases, $T_S^{-1}$ approaches the precise point to point mapping that we seek.

To find $T_S^{-1}$ we must first warp the reference mesh onto the target image, to do this we use the energy minimization technique developed in [17] and [18]. In [17] an energy function $\epsilon(S)$ is defined such that when minimized the squared distance between corresponding points is minimized and the mesh is warped as smoothly as possible. The function is defined as

$$\epsilon(S) = \lambda_D \epsilon_D(S) + \epsilon_C(S) \tag{3}$$

where $\epsilon_C$ is the correspondence energy, $\epsilon_D$ is the deformation energy and $\lambda_D$ is a constant.

The deformation energy ensures that the mesh is deformed smoothly, more precisely it should penalize mesh configurations that are not the result of perspec-

tive transformations. Thus $\epsilon_D$ is defined to be the sum of the squared second derivatives at each node in the mesh since this approximates curvature. As shown in [18] the deformation energy can be written using finite differences as

$$\epsilon_D(S) = \frac{1}{2} \sum_{(i,j,k)\in E} (-x_i + 2x_j - x_k)^2 + (-y_i + 2y_j - y_k)^2 \tag{4}$$

where $E$ is the set of all triplets $(i, j, k)$ such that vertices $v_i, v_j, v_k$ are collinear. As shown in [18] it is possible to write equation 4 in matrix notation allowing the use of the fast semi-implicit minimization scheme described in [9].

The correspondence energy takes into consideration the precise registration of the mesh in the warped view. In [17] it is defined as

$$\epsilon_C(S) = -\sum_{c\in C} w_c \rho(||c_1 - T_S(c_0)||, r) \tag{5}$$

where the $w_c$ are weights in the interval $[0, 1]$ and $\rho$ is a robust estimator defined as

$$\rho(\delta, r) = \begin{cases} \frac{3(r^2-\delta^2)}{4r^3} & \delta < r \\ 0 & otherwise \end{cases} \tag{6}$$

In each iteration of minimization the radius of confidence $r$ is decreased and $\rho$ classifies those correspondences outside the radius as outliers. This ensures deterministic termination which is essential for an interactive system as opposed to a method like RANSAC [5].

## 2.3 Obtaining the Planar Reference View

To perform non-rigid surface detection as described in [17] and [18] knowledge of a planar reference view is necessary. In this work we assume this planar view is available to us. This is reasonable since in our primary use case a visually impaired user will be interacting with printed media. Typically printed media is prepared digitally and thus there is a priori knowledge of the planar view.

There are a number of ways to obtain this planar view even when the planar state is not known a priori; we briefly describe three of them here. When there is no previous knowledge of the planar reference view it is still possible to synthesize one.

Using a portable structured light system or structure from motion it is possible to obtain a point cloud representation of the deformed surface. This point cloud representation can be used to synthesize a planar reference view with methods such as those described in [4, 8, 19]. In the absence of range data, methods such as the ones described in [6, 20, 24] could be used to obtain a planar reference view from monocular images. Finally, in the special case of documents with dense text it is possible to use the texture flow to synthesize the planar view as described in [13].

# Section 3

# Gestures

Our system offers a new method of interaction with common place objects such as newspapers, restaurant menus, magazines, and books. For visually impaired users this system can provide access to information unavailable to them without assistance from other people.

## 3.1 Fingertip Detection and Tracking

Our method relies on being able to accurately detect fingertips. There are many systems that have been successfully developed to reliably detect hands and fingertips [1, 3, 10, 11, 12, 15, 22, 25]. In our implementation we use the simple approach of colored markers adhered to the user's fingernail. We then find the centroids of the detected markers in the video frames and use these as the estimated location of the fingertip.

## 3.2 Metadata Registration

After obtaining the planar reference view there is a chance to process the planar view for a particular application. For instance in a common use case users will be interacting with text and at this step the planar view would be run through an OCR engine. The results of OCR are typically hierarchical bounding boxes of characters, words, sentences, and paragraphs with the detected text attached as

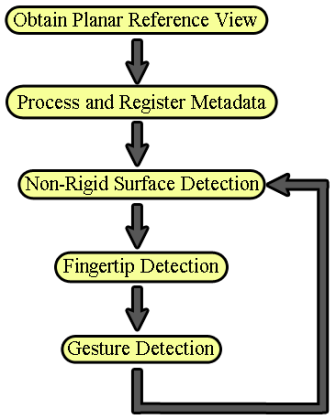



**Figure 3.1**: The high-level flow of our method.

metadata. Since this processing is done on the planar reference view, the location of these bounding boxes can easily be represented in that reference frame. As illustrated in Figure 3.2 the mapped location of the user's fingertip in the planar reference frame can be used to detect which of these bounding boxes the user is touching. In the case of a visually impaired user or a tourist at a restaurant the bounding boxes would contain entrée items and the metadata would be the description or the description's translation. The menu item pointed to by the user can be detected and its associated metadata could be read aloud to the user.

There is no restriction to the bounding shapes that can be registered in the planar reference frame nor the action that should be performed when activated. This is analogous to the traditional GUI paradigm of buttons and events.

## 3.3 Basic Gestures

In our system we have several pieces of information available to develop gestures. At any given time, we have knowledge of how many fingertips are present and their locations in the planar reference frame. In addition, since we capture a video stream we can store data from previous frames to develop complex gestures.

In [1] Argyros and Lourakis present sensible design criteria for gestures which we take into account in our own system. Specifically they suggest that gestures should be intuitive and unambiguous. This is beneficial for both the user

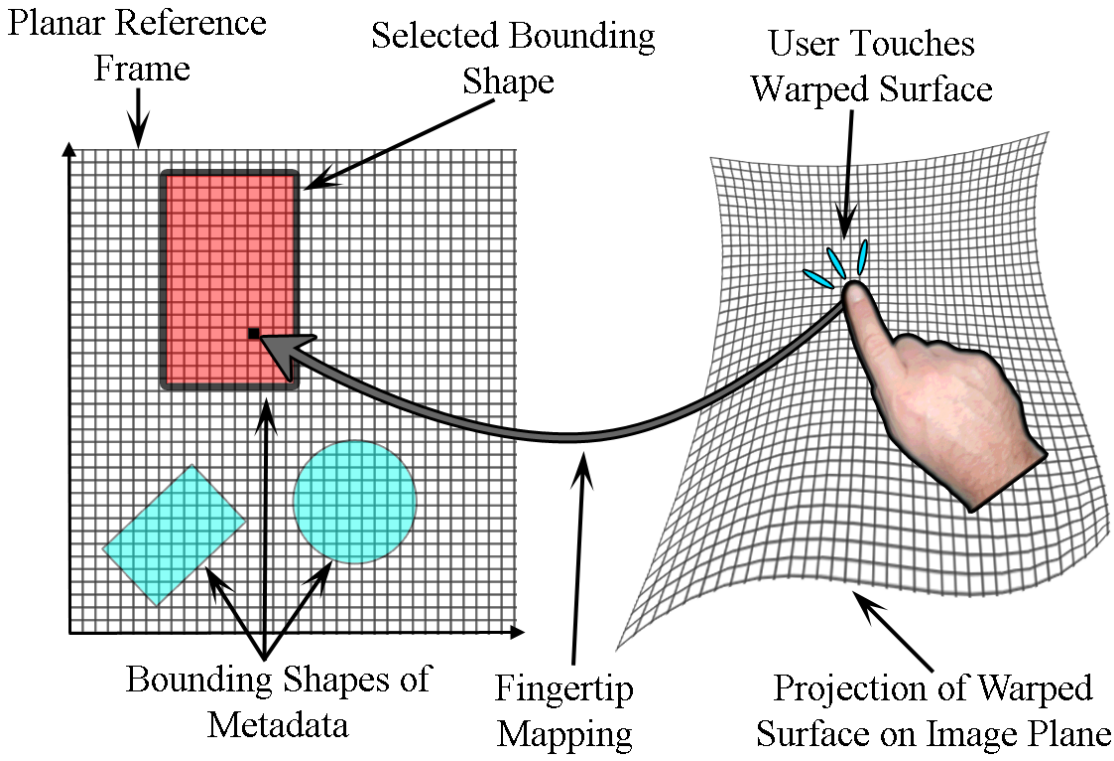



**Figure 3.2**: The pointing gesture: The user holds his finger still for a short time and the fingertip location is mapped into the planar reference frame. The registered bounding shapes are tested to see if any contain the fingertip location. A predefined action is performed for any bounding shapes the user's fingertip is touching.

and system designer since gestures are kept simple and are easy to detect. In our typical use case the user is expected to be holding the surface with which they are interacting. Thus we designed our basic gestures to require the use of only one hand, but the user is free to use either hand for gesturing at any time.

To avoid ambiguity our gestures are activated by holding fingers still for a certain amount of time. This is simple to detect since we merely need to check that the location of the fingertips stays within a small radius of their positions in prior frames. If their locations do not drift out of this radius for a specific number of consecutive video frames we detect an action.

Our most basic and primary gesture is pointing. To perform this gesture the user simply points to a location on the surface; Figures 2.2 and 2.3 demonstrate this gesture. When a pointing action is performed bounding shapes registered in

the planar reference frame are tested to see if they contain the location of the fingertip. If a bounding shape does contain the fingertip location its predefined action is performed; this is analogous to a graphical button and a mouse click in a traditional GUI. If there are many registered bounding shapes, which could be the case with dense text, it would be inefficient to test each one. Since we detect gestures in a planar reference frame it is possible to use binary space partitioning to achieve sub-linear time for this operation.

Selection is our other basic gesture and demonstrates the concept of gesture modes. Since pointing is an extremely natural gesture it is desired that all our gestures will be derived from this. However, this introduces ambiguity to our system. To overcome this we have a concept of gesture modes. To enter another gesture mode the user presents a certain number of fingers and holds them still for a short amount of time. As depicted in Figure 3.3 a user extends two fingers to enter selection mode and then gestures a bounding box to select a region of the surface. To gesture the bounding box the user indicates diagonal corners of the box with pointing gestures. Once the region is selected it is registered in the planar reference frame. The user can then point to his selection to execute a predefined action such as saving the selection as an image or activating each registered bounding shape in the selected region. In the case of text this could mean reading aloud all text that is selected in the region or copying the text to the system clipboard of a portable device. If the user desires to clear his selection without executing any action on it he can present two fingers again before another pointing gesture.

## 3.4 Current Implementation

Our current system is implemented in Matlab as a proof of concept and processing is done with pre-captured videos off-line. The videos are captured using a web camera mounted on goggles worn on the user's head. This way the camera sees whatever the user's head is pointing towards. We chose this arrangement with the visually impaired in mind since pointing the camera at the surface relies solely on proprioception. The user would simply point his head towards the finger being

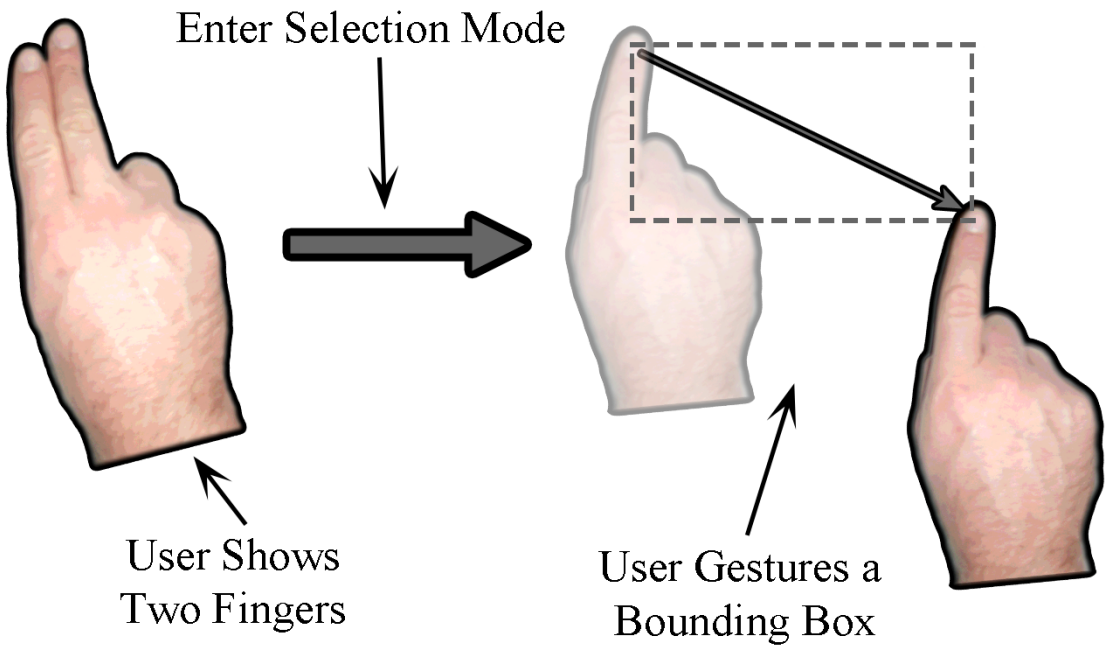



**Figure 3.3**: The selection gesture: The user shows two fingers to enter selection mode. Using two pointing gestures the user draws a bounding box to select a region of the surface.

used for gesturing. This is also a natural position for the camera to be mounted for sighted users.

## 3.5 Paths to Real-time

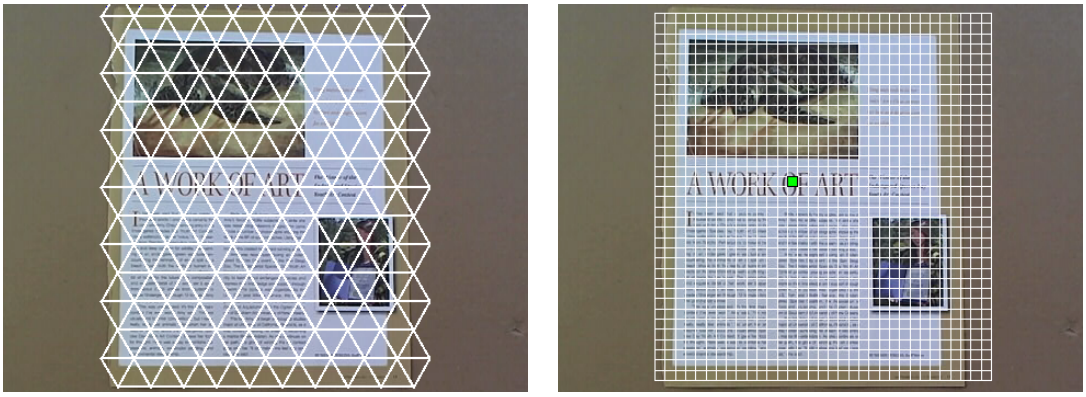
Our system is intended to be interactive and we have plans to implement a real-time version. It is expected that this system will run in real-time if optimized and compiled to machine code. Our justification is the fact that we rely solely on the method described in [17] which was shown to run at 10 frames per second on a 2.8 GHz Intel Pentium 4 machine. Currently we use SURF features implemented in Matlab [2]. Feature extraction and descriptor computation take the bulk of the processing time. Ferns features were used in [17] since they are fast detect, but this is made at a trade-off for off-line processing [16]. We prefer SURF features because there are GPU implementations that can run at 30 frames per second and require no off-line processing [23].

# Section 4

# Experiments

Our method is intended to let users interact with actively deforming surfaces and thus should be accurate despite occlusions and deformation to the surface. To evaluate the accuracy of our method first we obtain the planar reference view of a surface. We then affix a small colored marker to the surface while in a planar state and map the marker's detected centroid into the planar reference frame. Next we proceed to warp the surface and occlude large portions of it with the user's hands. The marker's centroid is detected in each frame of the captured video and mapped into the planar reference frame. We then record the difference between the mapped marker centroid and the initial mapped location which would ideally be zero. In practice we have found that in the presence of even drastic deformation this difference remains within a few pixels for a 640x480 image. In the presence of large occlusions we found similar results as long as the area directly around the marker was not completely occluded.

Our proposed gestures are based on pointing, thus this is precisely the situation that would arise in practice. Since the terminal end of a user's extended finger would be the location of the marker there would not be much occlusion in that area. Thus even if the mesh is registered poorly in the area occluded by the user's hand the portion of the mesh at the user's fingertip will be precisely registered. We designed our gestures with this in mind and are able to cope with imprecise registration unlike the system in [18] which needs global precision to perform graphical augmentation; this is demonstrated in Figures 4.1 and 4.2.



(a) (b)

**Figure 4.1**: Reference view from the quantitative experiment. (a) The planar reference mesh. (b) The mapped centroid of the marker in the planar reference frame falls in the same grid cell for Figures 4.2(a-d).

To quantitatively evaluate the accuracy of fingertip mapping we hand selected feature points and then sorted them by their distances from the marker's detected centroid. We ran non-rigid surface detection repeatedly removing the farthest feature from the centroid before the next iteration. We recorded the difference between the current mapping of the centroid and the reference mapping shown in Figure 4.3. These results suggest that it is only important to have a few good feature correspondences near the marker. In fact the accuracy of the mapping can improve when the majority of features are located near the marker; this is evident from the dips in the plots of Figure 4.3(a) and 4.3(c). This experiment was also run for varying numbers of triangular mesh elements. When there are many feature points, having more mesh elements improves the accuracy of the mapping, as expected. However, as the number of feature points becomes small the accuracy of the mapping degrades more quickly when using more mesh elements.

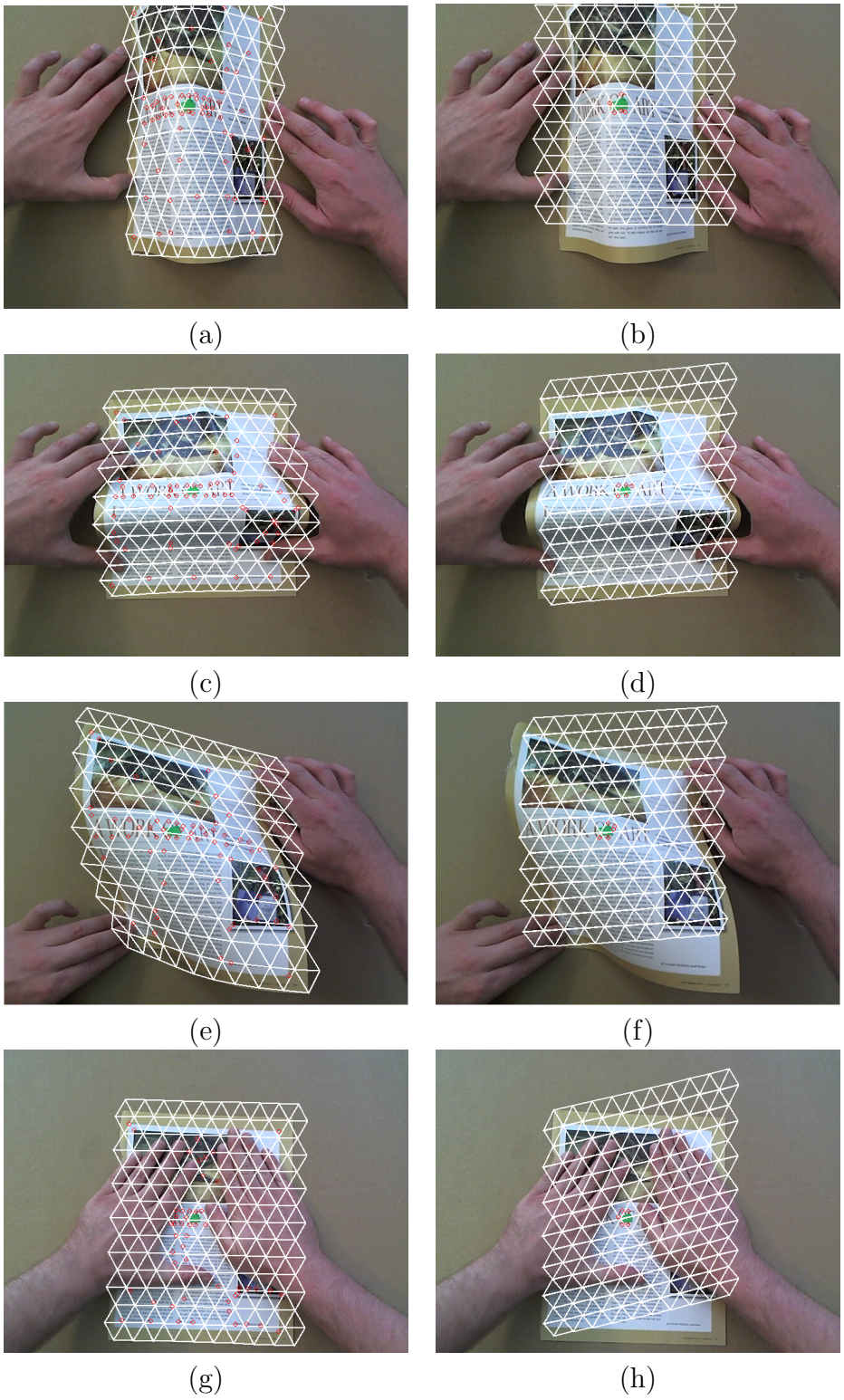


**Figure 4.2**: Images from the quantitative experiment. (a, c, e, g) meshes warped using all the hand clicked feature points. (b, d, f, h) meshes warped using the five closest hand clicked feature points to the marker's centroid in the warped view.

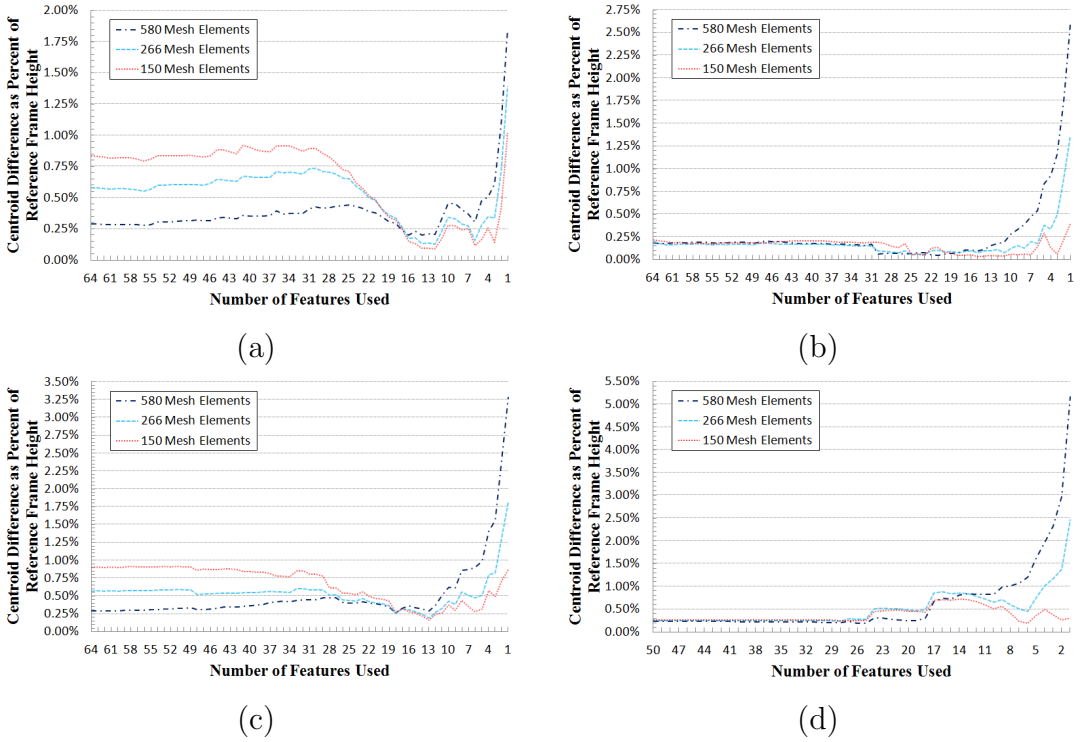



**Figure 4.3**: The hand clicked points depicted in Figure 4.2(a-h) were sorted by increasing distances from the marker centroid in their respective warped views. Non-rigid surface detection was run repeatedly removing the farthest feature at the end of each iteration. The detected marker centroid at each iteration was mapped into the planar reference frame and the difference was recorded. This experiment was then run again for different numbers of mesh elements. (a) The plot corresponding to Figure 4.2(a-b). (b) The plot corresponding to Figure 4.2(c-d). (c) The plot corresponding to Figure 4.2(e-f). (d) The plot corresponding to Figure 4.2(g-h).

# Section 5

# Conclusions and Future Work

In the future we plan to implement a real-time system with automatic fingertip detection. Rather than using colored markers on the user's fingertips we plan to use a technique such as the one described in [11]. In a companion project we are investigating methods to acquire the planar reference view automatically when a priori knowledge of the surface is not available. This would cover the case of materials with no available digital copy such as legacy books or restaurant menus.

The method we have presented can enable a new way of interacting with the information printed on paper. In addition it has the potential to give the visually impaired independent access to this information.

# Bibliography


[1] Antonis A. Argyros and Manolis I. A. Lourakis. Vision-based interpretation of hand gestures for remote control of a computer mouse. In *In Computer Vision in Human-Computer Interaction*, pages 40–51. Springer-Verlag, 2006.

[2] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.*, 110:346–359, June 2008.

[3] Nguyen Dang Binh, Enokida Shuichi, and Toshiaki Ejima. Real-time hand tracking and gesture recognition system. In *Proceedings of International Conference on Graphics, Vision and Image Processing (GVIP-05*, pages 362–368, 2005.

[4] M.S. Brown and W.B. Seales. Document restoration using 3D shape: a general deskewing algorithm for arbitrarily warped documents. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 367 –374 vol.2, 2001.

[5] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24:381–395, June 1981.

[6] Nail Gumerov, Ali Z, Ramani Duraiswami, and Larry S. Davis. Structure of applicable surfaces from single views. In *ECCV04. 1*, pages 482–496, 2004.

[7] Sean Gustafson, Daniel Bierwirth, and Patrick Baudisch. Imaginary interfaces: spatial interaction with empty hands and without visual feedback. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, pages 3–12, New York, NY, USA, 2010. ACM.

[8] Akihiko Iketani, Tomokazu Sato, Sei Ikeda, Masayuki Kanbara, Noboru Nakajima, and Naokazu Yokoya. Video mosaicing based on structure from motion for distortion-free document digitization. In *Proceedings of the 8th Asian conference on Computer vision - Volume Part II*, ACCV'07, pages 73–84, Berlin, Heidelberg, 2007. Springer-Verlag.

[9] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[10] T. Kurata, T. Okuma, M. Kourogi, and K. Sakaue. The hand mouse: Gmm hand-color classification and mean shift tracking. In *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 2001. Proceedings. IEEE ICCV Workshop on*, pages 119 –124, 2001.

[11] Taehee Lee and Tobias Hollerer. Handy AR: Markerless inspection of augmented reality objects using fingertip tracking. In *Proceedings of the 2007 11th IEEE International Symposium on Wearable Computers*, pages 1–8, Washington, DC, USA, 2007. IEEE Computer Society.

[12] Julien Letessier and François Bérard. Visual tracking of bare fingers for interactive surfaces. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, UIST '04, pages 119–122, New York, NY, USA, 2004. ACM.

[13] Jian Liang, D. DeMenthon, and D. Doermann. Flattening curved documents in images. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 338 – 345 vol. 2, June 2005.

[14] Pranav Mistry, Pattie Maes, and Liyan Chang. WUW - wear Ur world: a wearable gestural interface. In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, CHI EA '09, pages 4111–4116, New York, NY, USA, 2009. ACM.

[15] K. Oka, Y. Sato, and H. Koike. Real-time fingertip tracking and gesture recognition. *Computer Graphics and Applications, IEEE*, 22(6):64 – 71, Nov/Dec 2002.

[16] Mustafa Özuysal, Michael Calonder, Vincent Lepetit, and Pascal Fua. Fast keypoint recognition using random ferns. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2009.

[17] J. Pilet, V. Lepetit, and P. Fua. Real-time nonrigid surface detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 822 – 828 vol. 1, June 2005.

[18] Julien Pilet, Vincent Lepetit, and Pascal Fua. Fast non-rigid surface detection, registration and realistic augmentation. *Int. J. Comput. Vision*, 76:109–122, February 2008.

[19] Maurizio Pilu. Undoing page curl distortion using applicable surfaces. In *Computer Vision and Pattern Recognition Conference*, pages 67–72, 2001.

[20] Mathieu Salzmann, Francesc Moreno-Noguer, Vincent Lepetit, and Pascal Fua. Closed-form solution to non-rigid 3D surface registration. In *ECCV (4)'08*, pages 581–594, 2008.

[21] D.R. Schlegel, A.Y.C. Chen, Caiming Xiong, J.A. Delmerico, and J.J. Corso. Airtouch: Interacting with computer systems at a distance. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 1 –8, Jan. 2011.

[22] Naoya Takao, Jianbo Shi, and Simon Baker. Tele-graffiti: A camera-projector based remote sketching system with hand-based user interface and automatic session summarization. *International Journal of Computer Vision*, 53:115–133, 2003.

[23] Timothy B. Terriberry, Lindley M. French, and John Helmsen. GPU accelerating speeded-up robust features. In *Proceedings of the 4th International Symposium on 3D Data Processing, Visualization and Transmission*, 3DPVT '08, pages 355–362, Atlanta, GA, USA, 2008.

[24] Aydin Varol, Mathieu Salzmann, Engin Tola, and Pascal Fua. Template-free monocular reconstruction of deformable surfaces. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1811 –1818, 29 2009-oct. 2 2009.

[25] Zhengyou Zhang. Visual panel: Virtual mouse keyboard and 3D controller with an ordinary piece of paper. In *In Workshop on Perceptive User Interfaces*, pages 1–8. ACM Press, 2001.