
FingerInk: Turn your Glass into a Digital Board

Alaa Halawani

Dept. of Applied Physics &
Electronics,
Umeå University,
Sweden.
alaa.halawani@tfe.umu.se

Haibo Li

Dept. of Applied Physics &
Electronics,
Umeå University,
Sweden.
haibo.li@tfe.umu.se

Abstract

We present a robust vision-based technology for hand and finger detection and tracking that can be used in many CHI scenarios. The method can be used in real-life setups and does not assume any predefined conditions. Moreover, it does not require any additional expensive hardware. It fits well into user's environment without major changes and hence can be used in ambient intelligence paradigm. Another contribution is the interaction using glass which is natural, yet challenging environment to interact with. We introduce the concept of "invisible information layer" embedded into normal window glass that is used as an interaction medium thereafter.

Author Keywords

Authors' choice; of terms; separated; by semi-colons
Mandatory section to be included in your final version.

ACM Classification Keywords

H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous. See:
<http://www.acm.org/about/class/1998/> Mandatory section to be included in your final version.

General Terms

See list of the limited ACM 16 terms in the instructions and additional information:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI'13, April 27 – May 2, 2013, Paris, France.

Copyright 2012 ACM 978-1-XXXX-XXXX-X/XX/XX...\$10.00.

<http://sheridanprinting.com/sigchi/generalterms.htm>
Optional section to be included in your final version.

Introduction

Interaction between man and machine is an everlasting field of exploration. Researchers always aim at achieving more natural ways of communication between human and computer. With advancement in both hardware and software capabilities, more and more interaction devices and algorithms are being introduced to reach this goal.

One main problem is that people are usually forced to adapt to the machine environment, wasting time trying to learn how to use the new equipment that broke into their own space. This could turn to be stressful and frustrating. Hence, a crucial point is how to use advanced technology to support humans in their daily life in a way that does not introduce any stress or burden when dealing with this technology. Is it possible to let the machine fit to the human environment and not the opposite?

Models like ubiquitous computing or ambient intelligence tackle this issue. They consider embedding intelligence into everyday objects with minimum changes to the user's environment. These models are well-defined concepts with audible theories. Though, in real life things are different.

We feel that there is a gap between theory and practice of ambient intelligence. Many contributions assume special setups for their systems to work in reality. This greatly affects the flexibility of interaction. The limitations are mainly caused by two issues: Technology and hardware requirements (including interaction medium itself). Both issues are interrelated in reality.

Technology involves algorithms used to identify and track human parts (e.g. hand) needed for interaction, and then

recognize the action itself. This is a very challenging task if to be accomplished in real-life scenarios without any predefined conditions and with reasonable hardware costs. Successful systems usually require specific (usually costly) hardware like special types of screens, cameras or projection mechanisms. In brief, the challenge is to develop a robust and flexible technology with cheap hardware requirements.

The contribution we present here addresses this challenge. We consider scenarios where finger touch is used for interaction, which is natural and very common. We introduce a very robust and reliable vision-based finger detection and tracking technique that can be used in real-life scenarios without any need for pre-defined or controlled conditions. Besides robustness, the proposed finger tracking method is completely natural, cost-effective and computationally efficient, and hence can be used in many HCI systems.

On the user interface level, we present a contribution that takes advantage of the proposed finger tracking to realize ambient intelligence in an office environment. Specifically, we use office glass as an interaction medium by turning it into an invisible, interactive multi-functional information layer with minimum additional hardware and no extra space requirements. We consider this as inspiration and guidance to approaching ambient intelligence in real life situations.

Scenario

Before introducing our finger tracking algorithm, we would like to present the user interface scenario we are dealing with. We want to use window glass or office glass as an interaction medium. To our knowledge, the use of glass for interaction in real life scenarios has not been

tackled in previous works.

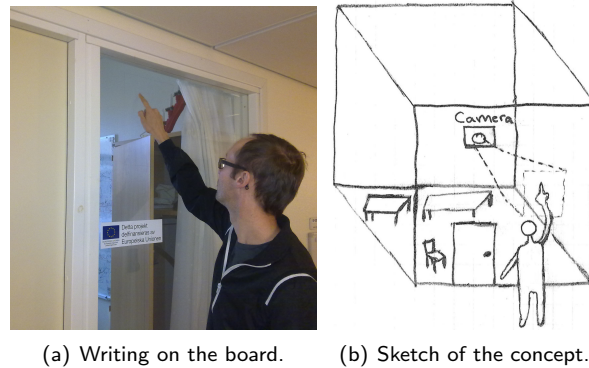


Figure 1: Concept of invisible information layer embedded into office glass.

In our work, we consider door environment in work offices (office front), but the concept is directly applicable to many other environments. We assume having a glass window in the door environment, and that is common in offices. Office front is a rich interaction environment. Examples are opening the door or writing a notice on the notice board. Typically, a special device is needed to accomplish each task (e.g., notice board to write a notice and electric code lock to open the door).

We aim at embedding intelligence into the environment to help the user in accomplishing such (and other) tasks. We accomplish this by introducing the principle of invisible information layer that we embed in the frontal glass, thus turning it into a “Digital Board”. To “write” on this digital board, some type of “ink” should be used.

Like with other interaction devices, we consider using touch with the finger as a natural way of interaction. We

mainly require in our design the use of the index finger, as can be seen in Figure 1(a). The user uses the index finger to write on the glass. As the “writing” using the finger is invisible, it resembles the well-known invisible ink, and hence we call it FingerInk.

As it represents a multifunctional information layer, the board is assumed to serve for several different functions. For example, it can be used as a notice board to leave a message for the office user in case of his absence, or it can be used for personal code identification, instead of the electric code lock, to grant access for authorized personnel to enter the office. With some suitable projection technique, it can be used as a drawing board or a discussion board. These are just examples of the functions. Others can be thought of.

From hardware point of view, we only need an off-the-shelf camera that is mounted behind the glass inside the office. This is clarified in the sketch of Figure 1(b). The user uses his finger to write on the glass. This is captured by the camera and analyzed by the computer to accomplish the required task (finger tracking, code recognition, etc.).

Using glass for interaction is attractive for many reasons. Firstly, we can find glass windows nearly everywhere, which means our interaction medium is available all the time. Secondly, it fits perfectly and naturally in the ambient intelligence paradigm; an interaction board with no changes to the user environment. Lastly, it can be a very simple and cheap setup. In the case we are introducing, we only need to install an off-the-shelf camera behind the glass. No additional hardware or extra space is required.

Yet, many algorithmic challenges come along with the simplicity of this setup (and that is why, in our opinion, it

was not considered in previous works): First of all, reflections on the glass may cause serious failures to many algorithms and sensors. Moreover, dynamic and cluttered backgrounds, in addition to changes in lighting conditions are serious problems to be solved.

The finger detection and tracking technology we are introducing overcomes these challenges, and achieves very robust and reliable results without assuming any predefined conditions.

Finger Detection and Tracking

We want first to stress that the proposed finger tracking technology is not limited to the above-described scenario of glass interface. In reality, it can be used in many HCI scenarios where finger touch is the given way of interaction (e.g. [2]). We proposed glass for interaction, and we consider this as a second contribution for the reasons mentioned above.

We need some simple, reliable, and efficient method to accomplish the crucial task of finger tracking, given that all what we are using are an off-the-shelf camera and a PC. Previous work has tackled finger tracking in many different ways.

Some relied on extra hardware like infrared camera to detect the skin [4]. We aim at avoiding extra hardware in our work. Moreover, the work described in [4] describes a special-purpose setup with certain hardware configuration.

In [2], the authors use a special-purpose device, consisting of a special camera with infrared sensors and LEDs, to accomplish hand tracking. Additionally, the authors mention that the setup requires controlled lighting conditions to work.

Other contributions used patches placed on the fingers [3]. We feel this is unnatural and limits the flexibility in our case.

Other methods use color segmentation or background subtraction (see [6] for review). This might work for systems with simple backgrounds and constant illumination conditions, both of which are not assumed in our work.

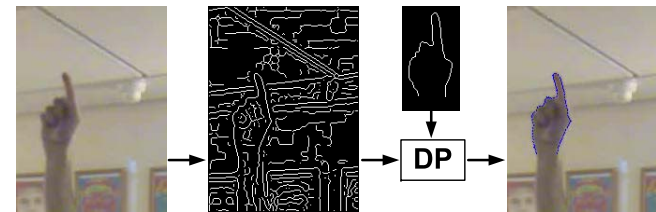


Figure 2: Using template matching with DP to detect and track the finger.

We propose the concept of *dynamic programming-based template matching* to achieve robust finger detection and tracking. The idea is based on storing a **single** contour template of a hand (or finger only) and then using it to match the contours present in the edge image of the contents of the interaction space. We search in edge information extracted from the acquired image using the very standard Canny edge detector.

To accomplish the search we use dynamic programming (DP) in order to insure a globally optimal solution. The procedure is abstracted in Figure 2.

The scheme possesses several advantages that are able to overcome the previously-mentioned challenges.

First of all, it is robust to illumination changes, since it

relies only on edge information, which is insensitive to lighting changes.

Secondly, it is also robust to the presence of clutter in the background. Even when the edge image is noisy, the detection results are highly reliable, thanks to the powerful DP search we are using. It is worth noting that we do not put any constraints on the background used. The algorithm is assumed to work for any given background content.

Moreover, the template deformation, that the algorithm allows, enables the detection to tolerate scale and orientation changes up to certain limit (can reach $\pm 50\%$). This flexibility makes it possible to use a **single** template for different users. I.e., there is no need to store a separate template for each user. Additionally, Once the hand (or finger) is detected, the fingertip is immediately determined without any further computations, as we already know its position in the template itself.

The technical details are briefly described in the following section.

Template Matching via DP

DP is an algorithm that is used to ensure a globally optimal solution of a problem, as it always returns the highest score match. It has proven to be successful in many contributions [7, 1, 8, 5].

Hence, we decided to adapt the DP algorithm to the task of hand detection and finger tracking. Namely, we use DP to search for the best fit of a hand template in a given image.

We are aware that the DP-based search is a classical algorithm. Our contribution is in the art of using the DP

algorithm, in combination with the concept of deformable templates, to achieve simple, elegant and robust detection and tracking.

Following is a brief clarification of the concept.

As can be seen in Figure 2, the inputs to the DP module are the binary template and the binary edge image. A **single** template is acquired offline from one of the users and stored to be used for online hand detection for **all** users.

Two main issues are to be considered here: Template deformation and template matching.

Deformable Templates

In order for the template to be usable, it should adapt to different hand shapes and tolerate certain range of transformations. This cannot be achieved if it is treated as a rigid entity. Instead, it is divided into shorter segments that are allowed to move within a certain range during the matching process. Deformation is introduced as a result of the controlled segment movements.

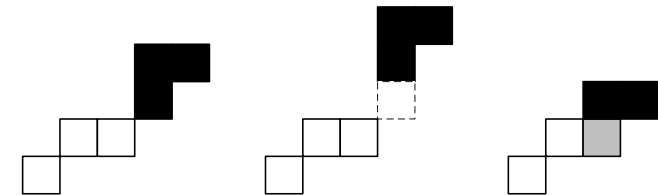


Figure 3: (*left*) Original arrangement of two segments (3 pixels each) in the template. (*middle*) Deformation caused by introducing a 1-pixel gap between the segments. (*right*) Deformation caused by introducing a 1-pixel overlap of between the segments.

The example in Figure 3 illustrates the concept. It is

assumed that the template is divided into several 3-pixel segments. The left column of the figure shows the original spatial arrangements of two segments in the template. By allowing the segments to move one pixel, a set of deformations can be achieved by introducing a gap (middle column) or an overlap (right column) between the segments.

Segments can be shifted to any position in the image to be searched, provided that the relative displacement between two consecutive segments is not greater than one pixel. I.e., if two consecutive segments are shifted by $o = (o_x, o_y)$ and $p = (p_x, p_y)$ respectively, then relative displacement is governed by:

$$|o - p| = \max(|o_x - p_x|, |o_y - p_y|) \leq 1. \quad (1)$$

The degree of template flexibility is then governed by the segment length. If each segment contains 3 pixels, then an overall shrinkage or enlargement of around $1/3 = 33\%$ can be introduced.

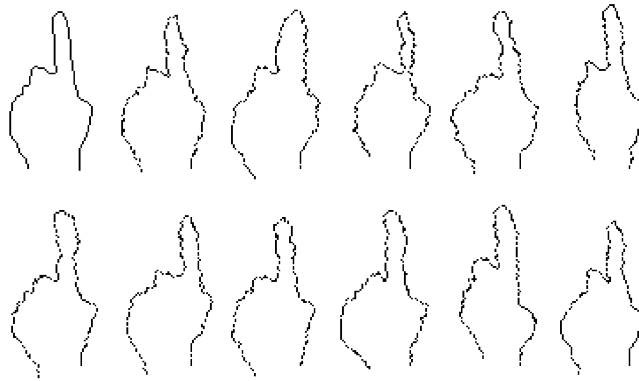


Figure 4: A random selection of possible deformations of the original template (upper left).

A mixture of gaps and overlaps will result in a set of possible deformations. Figure 4 shows a random selection of possible deformations of a template (upper left) with segment length of 2 pixels (up to 50% flexibility).

Template Matching

The template is divided into n segments. The query image to search in is of size $w \times h$ pixels. The Viterbi DP algorithm is used for the matching process. If the process is viewed as a trellis, each column in the trellis corresponds to a segment, and nodes in that column correspond to possible pixel shift positions of that segment in the image. Arcs connecting nodes in two consecutive columns are governed by Equation 1. This is abstracted in Figure 5.

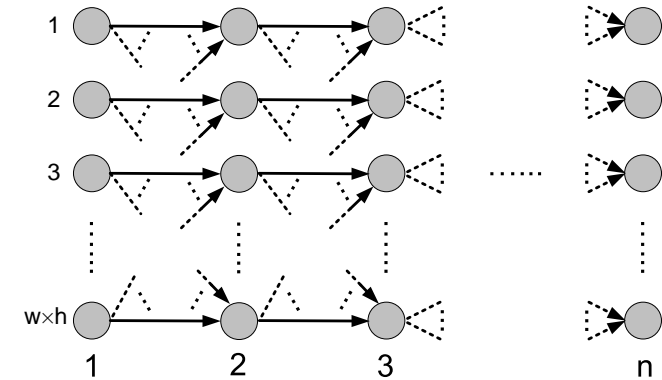


Figure 5: Illustration of the trellis structure governing the search process.

We search for the path through the trellis that maximizes the accumulated score, R . For segment i shifted to position p , R is given by:

$$R(p, i) = \max_{p' \in \Omega(p, i)} \{R(p', i - 1)\} + V(p, i), \quad (2)$$

$$\Omega(p, i) = \{p' \in \Omega : |p' - p| \leq 1\}.$$

Ω is the set of all possible shifts. $V(p, i)$ is the local reward given to node (p, i) and it is equal the number of edge pixels segment i will cover if placed at position p .

When R has been calculated for the last segment, the algorithm backtracks the optimal path starting from the node with the highest R value in the last column.

Results

Setup

We ran the experiments in one office at our department. The glass window at the front door of the office served as our digital board (cf. Figure 1(a)). To run the tests, videos of users in action were recorded using the camera of a Nokia X6 mobile phone.

The videos taken by this camera are of very normal quality, and any other webcam could be used.

Analysis of the videos was carried out using a desktop computer with a core 2 Duo 2.93 GHz Processor. 8 different volunteers participated in our experiments.

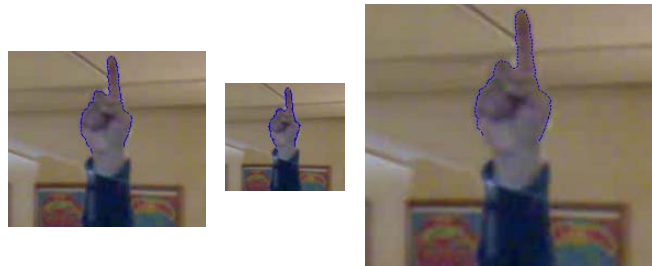


Figure 6: Scaling up to $\pm 50\%$ can be tolerated.

Hand Detection & Finger Tracking

A single hand template of one of the volunteers was acquired offline and used for tracking hands of all eight

users. The standard canny edge detector was used to obtain edge images from the video frames.

For matching with DP, the segment length was set to 2 pixels. This achieves a flexibility of $\pm 50\%$ of the original template size. This is a very acceptable range for our case.

This is demonstrated in Figure 6, where 3 versions of an image are shown. To the left, one can see the original image with the template size matching the hand size to great extent. The middle shows the image resized to half, and the right shows the image resized to 1.5. In all three cases, the template fits the hand very well.

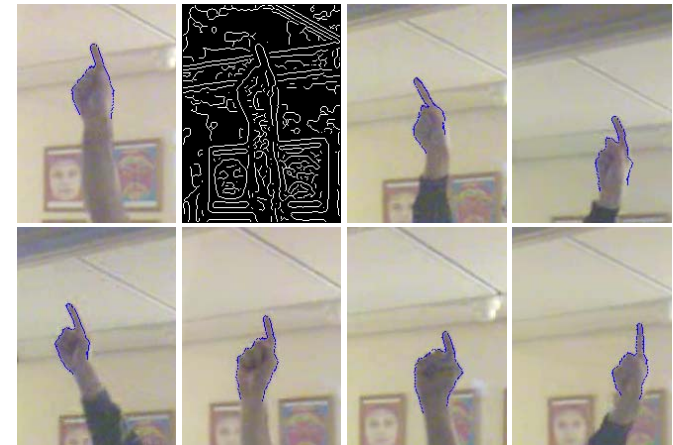


Figure 7: Sample results for hand/finger detection.

Figure 7 shows some detection results for different users of the system. The upper left image was shown together with its edge map to demonstrate the ability of the algorithm to tolerate heavy noise in edge images. In the shown case, the hand was reliably detected despite the dense presence of surrounding edge information.

Other results in the figure show how good the algorithm works for different users with different hand appearances. Notice that rotation, within an acceptable range, is also tolerated by the algorithm (e.g. lower left result). For more than 6100 acquired frames, an overall detection rate of 98.7% was achieved.

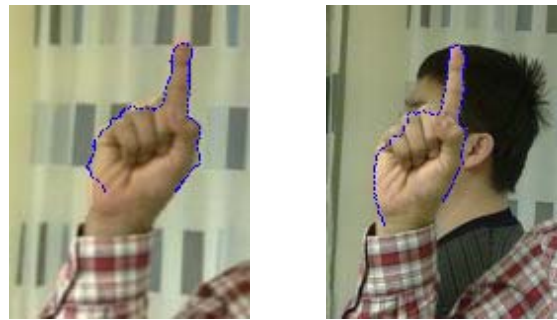


Figure 8: Testing in other circumstances. A different background (left) and a person passing by while testing (right).

Another result is shown in Figure 8. Here, a different background is used. The right part of the figure shows a dramatic change in the content of the background when a person is passing by while testing the system. The detection process is not affected by the change of the background details.



Figure 9: Sample results of finger tracking for a user writing on the digital board.

Once the hand is detected, finger tracking is easy, as the

position of the fingertip can be easily determined from the template itself. Figure 9 shows a result of finger tracking on the proposed digital board.

Personal Code Identification

As proof of concept for the FingerInk, we consider a personal code identification application that grants or denies access to some facility based on a personal passcode drawn by the user on the glass. The problem at hand is to identify this passcode.

Please note that what follows is a preliminary work concerned with proving the concept. Further enhancements are needed for this application to be used for real security-related issues.

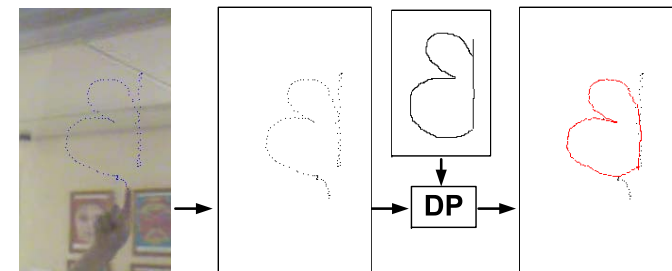


Figure 10: DP used for template matching as the first step in code identification.

The pattern drawn by the user on the board can be easily represented in a very simple binary image. This made us think of using the algorithm in hand (DP) for this application.

If we acquire offline a binary template of the user's code, we can then use the same template matching with DP algorithm described earlier to accomplish the task (see Figure 10. Binary images were inverted for better view).



Figure 11: Eight different templates for users' codes.

For this purpose, each of the 8 volunteers was asked to pick up some pattern that he has to draw on the board. Each person was asked to provide 5 samples by repeating his pattern 5 different times on the board, resulting in a database of 40 samples. One template per pattern was registered by the user offline to be used for matching.

The 8 different templates of the user codes are shown in Figure 11.

To accomplish the identification task, all of the 8 templates are matched to each of 40 samples in the database. The template with the best match to the pattern wins the identification process.

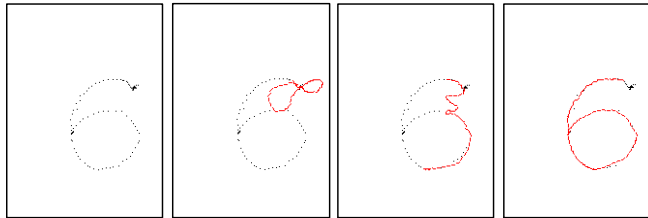


Figure 12: A test pattern (leftmost) and match results for 3 different templates. $\delta = 0.11, 0.15$, and 0.22 respectively. The rightmost result has the highest δ value and is selected as the best fit.

To determine the best match, a very simple measure is used in our case. It is supposed that the best fit for the pattern will cover as many pixels as possible of that

pattern. This is illustrated in Figure 12, that shows a test pattern (leftmost column) and the match results for three templates.

It is obvious that the rightmost result is the correct one, covering most of the pattern's points. If N_p denotes the number of pattern points and N_t the number template points (i.e., size of the template), then the fit measure is given by: $\delta = N_p/N_t$.

If two templates have the same or similar δ values, the larger template is selected as the best match. Values below 0.15 are rejected. Based on this, we ran the tests on the database, and achieved a recognition rate of 97%.

We are aware that this is not the best measure to use, and that many other algorithms (e.g., [9]) can be successfully used to recognize the drawn patterns, but our concern was to prove the concept of using glass as an interaction medium. Other issues related to other applications or uses of the system will not be discussed and are considered to be beyond the scope of the current work.

Conclusion

We showed how to achieve simple and robust finger tracking using template matching via dynamic programming. The algorithm is flexible and only requires the use of an off-the-shelf camera. It successfully overcomes the problems associated with real-life scenarios and does not assume any predefined setup. Thus it can be employed in many CHI systems. We took advantage of the propose tracking algorithm to present an example of introducing ambient intelligence in normal window or office glass. The idea is based on integrating an invisible information layer into the glass that can then be used as a multifunctional interaction medium. We showed a simple demonstration of using the system for personal code

identification as proof of concept for finger-based interacting with the glass. We believe that the CHI audience can benefit from our tracking algorithm and will be inspired by new ideas regarding the usage of the setup we introduced.

References

- [1] P. F. Felzenszwalb and R. Zabih. Dynamic Programming and Graph Algorithms in Computer Vision. *PAMI*, 33(4):721–740, Apr. 2011.
- [2] S. Gustafson, D. Bierwirth, and P. Baudisch. Imaginary Interfaces: Spatial Interaction with Empty Hands and without Visual Feedback. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology*, pages 3–12. ACM, 2010.
- [3] M. Kaltenbrunner and R. Bencina. reacTIVision: a Computer-vision Framework for Table-based Tangible Interaction. In *Proc. of the 1st intl. conf. on Tangible and embedded interaction*, pages 69–74, 2007.
- [4] H. Koike, Y. Sato, and Y. Kobayashi. Integrating Paper and Digital Information on EnhancedDesk: a Method for Realtime Finger Tracking on an Augmented Desk System. *ACM Trans. Comp.-Hum. Interact.*, 8(4):307–322, Dec. 2001.
- [5] R. Nohre. Deformed Template Matching by the Viterbi Algorithm. *PRL*, 17(14):1423–1428, December 1996.
- [6] V. I. Pavlovic, R. Sharma, and T. S. Huang. Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. *PAMI*, 19(7):677–695, July 1997.
- [7] E. G. M. Petrakis, A. Diplaros, and E. Milios. Matching and Retrieval of Distorted and Occluded Shapes Using Dynamic Programming. *PAMI*, 24(11):1501–1516, Nov. 2002.
- [8] A. Shashua and S. Ullman. Structural Saliency: The Detection of Globally Salient Structures Using a Locally Connected Network. In *ICCV*, 1988.
- [9] J. O. Wobbrock, A. D. Wilson, and Y. Li. Gestures without Libraries, Toolkits or Training: A \$1 Recognizer for User Interface Prototypes. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 159–168, 2007.