Designable Visual Markers

Enrico Costanza
e.costanza@ieee.org

Jeffrey Huang
jeffrey.huang@epfl.ch

Media and Design Lab
EPFL, Switzerland

ABSTRACT

Visual markers are graphic symbols designed to be easily recognised by machines. They are traditionally used to track goods, but there is increasing interest in their application to mobile HCI. By scanning a visual marker through a camera phone, users can retrieve localised information and access mobile services. One missed opportunity in current visual marker systems is that the markers themselves cannot be visually designed; they are not expressive to humans, and thus fail to convey information before being scanned. This paper provides an overview of d-touch, an open source system that allows users to create their own markers, and control their aesthetic qualities. The system runs in real-time on mobile phones and desktop computers. To increase computational efficiency, d-touch imposes constraints on the design of the markers in terms of the relationship of dark and light regions in the symbols. We report a user study in which pairs of novice users generated between 3 and 27 valid and expressive markers within one hour of being introduced to the system, demonstrating its flexibility and ease of use.

Author Keywords

Visual marker recognition, visual marker design, mobile devices, mobile HCI, UI toolkits, fiducial recognition, user studies.

ACM Classification Keywords

H5.2. [Information Interfaces and Presentation]: User Interfaces – Input devices and strategies.

INTRODUCTION

Visual markers are graphic symbols designed to be easily recognised by machines. They are used to relate physical objects to computer systems. Examples include the barcodes used on most commercial goods to keep track of stock in shops and warehouses, and more recent *2D-barcodes* [11, 10, 8, 21] which are easier to read with low-resolution cameras, such as those included in mobile phones. Beyond stock control, there has recently been interest in both the academic and commercial community for using this technology to enhance mobile applications and facilitate access to mobile services [26, 18, 15]. Visual markers[1] are often considered as an alternative to RFID tags. They are not exactly equivalent because visual markers are *read-only*, whereas some RFID tags can be rewritten, but they can support similar types of interaction with the advantage that they can be created with normal printers and accessed using standard photo cameras already available on most mobile devices. RFID requires special readers devoted only to this purpose, integrated only in a limited number of devices.

Similar to traditional barcodes, the shape of existing visual markers is solely based on maximising their readability by machines: they are not visually meaningful to people, and different markers of the same family are generally not easy to distinguish from one another by looking at them. In other words, one missed opportunity in current visual marker systems is that the markers themselves cannot be designed, they are not expressive to humans, and thus fail to convey information to people before being scanned. For example, information could be conveyed about the type of digital content or mobile service associated with the markers, or the project they belong to or the person or company who created the content. Further, an interesting marker design could simply attract attention so that it would be scanned with a mobile device. We argue that it is fundamental to empower both application creators and end-users to visually design their own markers – giving visual markers the same visual dignity and expressive potential currently given to icons, and even opening up functional markers to the paradigm of user-generated-content.

In this paper we introduce d-touch, a visual marker recognition system that allows users to create their own visual markers, controlling their aesthetic qualities and what they visually communicate to others. By allowing the creation of markers that support interaction both visually and functionally, d-touch can enhance most applications normally supported by visual markers, including interactive guides, mobile service access, mobile games, interactive story telling systems and augmented reality applications that have broad visual appeal and are not constrained to ugly glyphs. As demonstrated by the study presented below, markers that are both functional and visually expressive can be easily pro-

CHI 2009, April 4 - 9, 2009, Boston, Massachusetts, USA.
Copyright 2009 ACM 978-1-60558-246-7/09/04...$5.00.

[1] Visual markers are also referred to as *visual tags* or *fiducial markers*. We prefer the term *marker* over *tag* to avoid confusion with keywords and metadata *tags*.

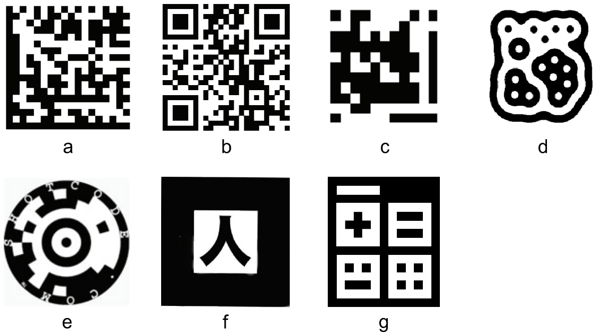


Figure 1. Example markers from the literature: a) Data Matrix[11]; b) QR-codes [10]; c) M. Rohs's visual codes[21]; d) reacTIVision markers[3]; e) TRIP[8]; f) AR Toolkit[14]; g) early d-touch marker[6].

duced by a wide spectrum of users, without specific training. We envision the system being used by professional designers as well as end users, enabling both groups to design markers as visual icons with high expressive or communicative power. Because d-touch markers are designable, end users and designers can consciously determine their look and feel, including the degree to which they are immediately recognisable as markers to be scanned. The design can range from icons that are obviously scannable (*explicit*) to ones that are hidden in the overall design and only accessible to a closed circle or upon closer look (*ambiguous*). For applications in which immediate user recognition of the markers is essential, designers may define conventions for the marker placement, e.g. markers may be placed at the bottom right corner of posters or below text in museum labels. Specific application scenarios for professional designers include the creation of highly polished, explicit visual markers that follow the design guidelines of corporate identity, or the incorporation of ambiguous markers in visual communication, such as ads, that are not recognisable at first glance. Application scenarios for end users include hand-drawn expressive visual markers left in the environment to leave location-specific information and traces. Markers could be used as hidden, secret symbols that are ambiguous and only noticed and scanned by an inner circle – echoing established urban phenomena such as graffiti and tags, as well as older hobo codes.

d-touch runs in real-time on both mobile phones and desktop computers and it is released under an open source license. The creation of markers which are both machine- and human- readable is possible because d-touch recognition is based on topological features of the markers rather than their geometry. Marker recognition is not based on shape, but on the relationship of dark and light regions. We present an evaluation of the creation of d-touch markers, designed to understand how much people can visually express within the constraints imposed by the system. During the study pairs of novice users generated between 3 and 27 valid markers within one hour of being introduced to the system, demonstrating its flexibility and ease of use.

d-touch was initially developed for tracking objects in the context of tangible user interfaces and augmented reality [5, 6, 7], and the visual appearance of early markers was optimised solely for resolution and recognition accuracy. The work presented here is based on recent technical developments that take advantage of topology-based recognition to allow a wider range of visual expression through the markers. The next section provides an overview of related work, followed by a description of the d-touch system, its performance, an example application and the requirements that it poses for the markers' design. A user study to evaluate how successfully novice users create expressive markers which satisfies the d-touch requirements is then reported, followed by discussion of the results and conclusive remarks.

RELATED WORK

Visual marker recognition can be considered as a special case of *object recognition*, where the objects, i.e. the markers, are designed in conjunction with the recognition algorithm in order to optimise performance, achieve high recognition rate, low false positives and high processing speed. Barcodes are probably the earliest example of visual markers, dating back to the 1950s [12]. They started to be used commercially at the end of the 1960s and are still used in most shops. Barcodes are designed to be read through high resolution scanners, and they are generally not easy to decode through standard consumer grade webcams and camera phones, at least without special macro lenses. A more recent generation of visual markers, which can be more easily read by low cost imaging cameras has been presented over the last twenty years. Proposed applications include mixed reality systems [19, 14], video post-production [24, 13] and, more recently, human-computer interfaces for camera-phones [16, 17, 21]. Because most systems rely on geometrical feature detection both to localise the markers within input images and for encoding unique identifiers in each marker, the markers' visual appearance is strongly constrained. In the vast majority of cases, the shape (i.e. the geometry) of the markers is generated algorithmically, following techniques derived from communication coding, without allowing human input on the visual design.

For example, ARToolkit markers [14] (Figure 1 f) are characterised by a thick black square frame containing arbitrary grayscale patterns. The system uses straight line detection and quadrilateral fitting to locate the markers and if a marker is found, its pattern is extracted and cross-correlated with all known patterns. As a consequence of this the system speed decreases the more patterns used and the more markers that are contained in the input image. The patterns used for ARToolkit markers can be customised arbitrarily, however, later reserch [9] suggested to apply digital communication coding techniques to improve the system's performance, at the cost of customisation. The TRIP system [8] is based on edge detection followed by fast ellipse fitting to locate and track markers, known as *Spot Codes*, which are composed of concentric circles and arcs (Figure 1 e). Because ellipses are projection invariant, the system is robust to perspective distortion.

A number of systems have common characteristics and they can be referred to as *2D barcodes*. In all of them bits of information are encoded in a matrix of black and white dots, generally with the protection of some coding. The markers also include lines to facilitate orientation recovery. Examples include Cybercode [20], QR-codes [10] (Figure 1 b), Data Matrix [11] (Figure 1 a), and Visual Codes [21] (Figure 1 c). Algorithms to detect this type of markers are also available for mobile phones [10, 11, 21], and several players in the mobile communication industry are promoting standards and a common infrastructure for them [2]. The markers are generated automatically through coding algorithms and they do not allow any aesthetic personalisation or tuning from human input.

The d-touch recognition system is based on the topology of the markers, rather than their geometry [5]. Initially the system was developed for tracking objects in tangible user interfaces and augmented reality applications [6, 7], as the topology-based approach allows fast performance even when multiple markers are recognised in the same image. An example marker used in tangible interfaes is show in Figure 1 g. The shape is designed for compactness and to encode additional information in the position of the parts of the symbol. An initial attempt to design d-touch markers for a mobile application taking into account aesthetic aspects was reported in a position paper [4], however, in that case markers were still mostly based on the rectangular grid of Figure 1 g, while the work presented here extends the expressive range considerably. Two systems other than d-touch, VPS system [13] and reacTIVision [3] (which was derived from d-touch – Figure 1 d), are known to use topology rather than geometry for marker detection, however, in both of them markers are generated through algorithmic techniques, with little or no input from the user regarding aesthetic aspects of the markers.

Fast watermark detection [17] can be an alternative approach to visual marker recognition. This technique encodes information such as an identifier over any 2D image, invisibly. However, the method still relies on geometry, and it requires images to either have a dark border on a white background or to be framed with a thin black rectangle. Moreover, the method is not as fast as other marker recognition techniques, as it is reported to take slightly more than one second to process an image of size 288×352 on a mobile phone.

Given that for most other marker recognition systems symbols are designed algorithmically, no precedents were found in the literature for user studies similar to the one proposed here.

D-TOUCH TECHNICAL OVERVIEW

d-touch is based on image topology and in particular on adjacency information of connected components, represented in a “region adjacency tree”. Images are first transformed into 2-levels (pure black and white), then they are segmented into connected components: sets of adjacent pixels of the same colour, such that for any pair of pixels in the set it is possible to find a path (of pixels) joining them within the same set (this is the same concept used by the “paint bucket” operation in paint programs). Note that because the images at this point are binary, the connected components are either black or white. The region adjacency tree is then constructed in the following way: to each connected component corresponds a node, two nodes are connected if the two related regions share a border – in the case of binary images if two regions share a border it means that one is contained in the other, and the region adjacency graph is guaranteed to be a tree [22], which is easier to deal with computationally. The region adjacency tree of an example binary image is illustrated in Figure 2. The white background region a contains 3 black regions: b, e and f – note that all the black pixels in the scissors are connected (region b). Region b, in turn, contains two white regions: c and d.

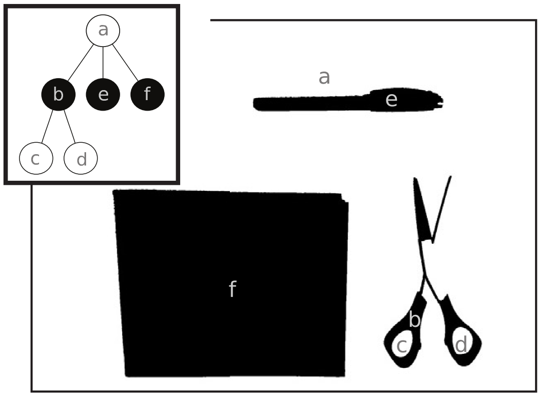


Figure 2. An example of binary image and its region adjacency tree.

Markers are also defined in terms of their region adjacency tree, so the recognition of markers in a scene image is reduced to searching for the markers’ adjacency trees as subtrees of the scene image adjacency tree. This search is known as the *subtree isomorphism* problem, and it can be solved with an $O(n \times m^{1.5}/log(m))$-time algorithm, where m is the number of nodes in the target tree and n the number of nodes in the scene tree [23]. The adjacency structure of the markers is limited by 3 constraints to simplify this search from a computational point of view, and to reduce the likelihood of false recognition. The first constraint is that markers can have only 3 levels of nesting, which we name *root*, *branches* and *leaves* – the *root* is the outermost level of the marker, the region enclosing it, any region of the marker connected to the root is named *branch*, and regions connected to *branches* different than the root are *leaves*. Regions of the marker that are connected to the root and do not contain any leaves are called *empty branches*. The second constraint is that markers must have at least 3 branches and the third that at least half of the branches must contain leaves. In other words, a valid marker can be composed of a black region containing 3 or more white regions, and at least half of these white regions must contain one or more black regions – however the colours can be inverted. Because the first step of the recognition process is to convert the images into pure black and white, the markers can actually be drawn in any colour, as long as they present reasonable contrast. For simplicity, the rest of the paper will consider only black and white markers. Four examples of the “minimal” valid marker structure are illustrated in Figure 3, other examples are shown in Figure 4.

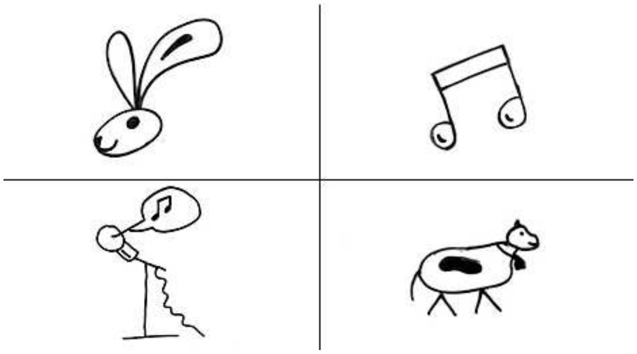


Figure 3. Four example "minimal" valid d-touch markers: each is a black region containing 3 white regions (the minimum number of regions allowed); 2 of the white regions contain 1 black region, one contains none. These examples were generated by subjects in the user study.

The marker definition is based purely on topology and it leaves considerable freedom for the geometry, i.e. the shape, of the symbols. Therefore, it is possible to design symbols that carry visual meaning to people, but can also be efficiently recognised by machines. Based on the rules definition it is difficult to predict how successfully people will be able to express themselves visually while still following the rules. To investigate this issue, a user study was designed and performed, as described in the following sections.

Each marker is associated with a numerical ID, derived directly from its adjacency tree: this is a sequence of non-negative integer numbers representing the number of leaves contained in each of the branches, followed by the colour of the root. IDs for some example markers are shown in Figure 4. Because the system takes into account only the topology, it is possible to have symbols with different shapes map to the same topological structure, and therefore to be considered identical, such as the four in Figure 3.

In principle, it would be possible to calculate the maximum number of different d-touch markers that can be defined for a fixed camera resolution, assuming a specific type of geometry, such as a rectangular grid. However, that would not be very informative given that the main strength of d-touch compared to other visual marker recognition systems lies in human-designed markers. For this reason, considerations relative to the number of different markers supported by the system are based on the results of the user study and discussed below.

By definition d-touch markers are scale and rotation invariant. In fact they are even resilient to other types of distortion such as stretching, as long as their topology is not affected. However, their recognition is dependent on the camera being able to capture their structure. In practice this means that if some of the elements of the markers are too small compared to the camera resolution, or if they are distorted by blur, the recognition would fail. To design *robust* markers – markers that can be recognised at lower resolution and are resilient to blur – it is important to take into account the size of details compared to the overall size of the symbol.

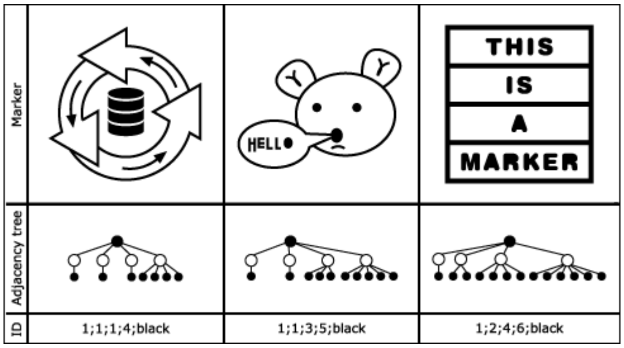


Figure 4. Examples of d-touch markers fulfilling the constraints imposed by the system. Each of the black and white connected components corresponds to a node in the adjacency tree on the second row. The ID, on the bottom row, is constructed as the the number or leaves in each of the branch nodes, followed by the colour of the root node.

Implementation and performance

d-touch runs in real-time on desktop PCs (Linux, MS Windows and Mac OS) and on Symbian OS mobile phones. The topology-based approach does not involve floating point operations, making the system particularly suitable for embedded devices. The source code is platform independent, it is written in C++, includes Python bindings and is publicly available under the terms of the GNU Public License (GPL) [1]. The desktop version uses a standard socket based client/ server architecture and makes it possible to develop applications in other programming languages. Some of the data structures require reallocation if their size grows beyond initial reserved space, significantly slowing down the processing. However, it was empirically noticed that for an image of size $w \times h$ pixels, the tree size is generally smaller than $l = w \times h/100$. Larger adjacency structures generally correspond to noisy images, generated for example by lag in the camera's automatic gain control when there is a change in the scene illumination. For this reason, if the graph construction grows beyond l the current frame is dropped.

The processing speed depends on the number of connected components in the input image (this is directly reflected in the size of the tree and on the number of operations required for its construction). However, it is independent of the number of markers contained in the image. On a *desktop PC* powered by an Intel Pentium D processor at 2.8 GHz with 2MBytes of cache and 1GByte of RAM, running Ubuntu Linux 6.10, the average time required to process a 640×480 frame was 16.45 ms (std. dev 0.83, N=65000 frames), corresponding to about 60 frames per second. On a *Nokia N73* mobile phone the time to process a frame is 86.0 ms (std. dev 18.2, N=144), i.e. about 12 frames per second in viewfinder mode with a resolution of 240×180 pixels, and it goes down to an average of 347.3 ms (std. dev 85.6, N=14) when processing 640×480 images. This level of performance has a strong impact on usability as it allows the creation of an effect similar to *mouse hover* when a camera phone is pointed to a d-touch marker. The marker can be highlighted in the viewfinder to show users that it can be selected.

Example Mobile Application: uWiki

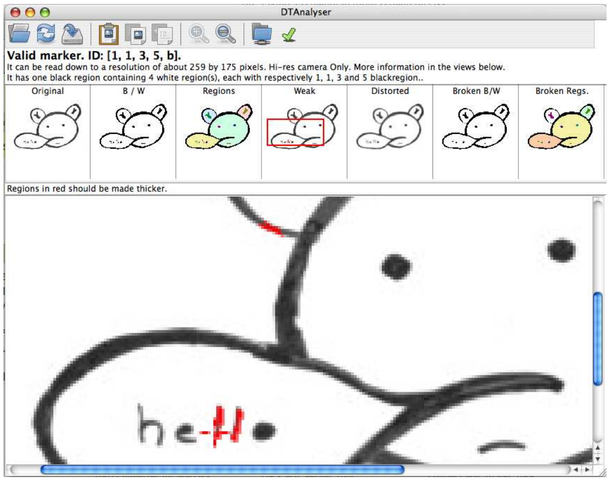
To deploy the d-touch system in mobile applications we have developed *uWiki*: a wiki-based system that makes it possible to associate multimedia content with d-touch markers. uWiki includes a mobile client running on Symbian S60 devices, written in Python and C++, and a server, based on the open source TikiWiki engine [25]. By scanning a d-touch marker through the mobile client, users can retrieve multimedia content associated to it, and edit or add new material. The system also provides a web-based interface to the same information, to allow richer access through personal computers when available.

DTAnalyser

The DTAnalyser is a desktop application developed to help users design valid markers. It is GUI-based and runs on Windows, OS X and Linux. Users can import candidate markers as bitmap files or through copy and paste from other applications. A screen shot is shown in Figure 5. The DTAnalyser does not provide any drawing functionality, it is designed to be used in conjunction with existing graphic applications, given their availability under both commercial and open source licenses. To avoid ties to a specific platform, it was preferred to develop the application as stand-alone rather than as a plug-in. For any imported graphics, the DTAnalyser shows how the image is transformed into black and white and how it is segmented into connected components – this information is rendered through a coloured map. The application checks whether the imported graphics comply with the structure of d-touch markers; if they do not, it attempts to detect how the proposed symbol violates the d-touch rules and presents this information to the user.

In the case of valid markers, the application displays the marker ID, and performs a robustness analysis. Low resolution scanning of the marker is simulated by resizing the image with a low-fidelity method (pixel replication). The image is iteratively scaled down until its topology becomes different from the original. The distorted image is compared with the original to detect which elements are most likely to be corrupted, normally corresponding to smaller details of the symbol. The elements that make the marker *weak* to low resolution scanning are displayed to the user, as illustrated in Figure 5, together with the the black and white and segmented representations of the distorted image. The minimum (simulated) resolution at which the marker can be successfully read is also displayed. Given that most mobile phones on which the d-touch recognition runs have a viewfinder resolution of 240×180 pixels, markers which are readable at a resolution under 100×100 pixels are suggested to be "good for a mobile phone", those readable below 200×200 pixels "not easy for a mobile phone" and the others as "high-resolution cameras only".

As shown in Figure 5, all the available views of the candidate markers are scaled down and displayed as thumbnails at the top of the application window. Each view can be selected by clicking on it with the mouse and in this way displayed on the main panel of the application, where it is possible to zoom in on details. It is worth underlining that the DTAnalyser does not attempt to automatically fix the imported graphics, it rather points out problems and let the users modify their design (back in the original drawing application) to solve them.



Valid marker. ID: [1, 1, 3, 5, b].

It can be read down to a resolution of about 259 by 175 pixels. Hi-res camera Only. More information in the views below.

It has one black region containing 4 white region(s), each with respectively 1, 1, 3 and 5 blackregion..

Regions in red should be made thicker.

Figure 5. A screenshot of the DTAnalyser application, highlighting in red the elements of the marker that make it unreadable when scanned at low resolution.

EVALUATION: DRAWING FUNCTIONAL MARKERS

The rules that define valid d-touch markers, described above, allow the creation of symbols that can both be read at the topological level by the recognition algorithm and have an iconic meaning for people. While the rules are in principle very flexible in this sense, humans usually refer to graphics in terms of shapes and composition, rather than nesting of connected components. Thus it was not obvious how easily people would be able to create symbols that carry expressive meaning while complying with the d-touch rules. A user study was designed and run to address this question. At a more general level, the study also aimed to explore the design space of d-touch markers, in terms of topological complexity of the symbols drawn, number of unique IDs generated and "collisions" of marker IDs. Two experiments were designed: the first one examined the ability of users to draw valid markers, evaluating also the effect of information provided by the DTAnalyser application, while the second experiment focussed on the creation of markers which are not only valid, but can also be scanned at low resolution. More formally, our study aimed to test the following hypotheses:

1. people can, with minimal training, draw markers both recognisable by the system and expressing a concept or message which can be understood by others;

2. the information about which features of a symbol violate the d-touch constraints, provided by the DTAnalyser application, aids in the creation of functional markers;

3. taking into account limitations due to scanning resolution *robustness* of the markers does not reduce significantly the expressive ability.

EXPERIMENT 1: VALID MARKERS

Experimental Design

The first experiment was designed as between-groups with two conditions: a "Feedback" condition in which participants were given detailed information if their drawing violated the d-touch rules, and a "No Feedback" condition in which they were just told whether or not their drawing was a valid marker, without explanations. The experiment was carried out in pairs and all sessions were video recorded to allow the analysis of conversation between the participants. Participants were given a total of 1 hour to both study written instructions and to draw as many valid markers as possible.

The written instructions introduced the d-touch system and its rules to define valid markers, illustrated through a number of examples, and described the drawing task. Only markers with black root were considered, so the rules could be simplified as follows: "a valid marker can be composed of a black region containing 3 or more white regions, and the majority (i.e. more than half) of these white regions must contain one or more black regions. This makes exactly 3 levels of nesting – it must be no more and no less. However, there is no limit in the number and shape of the regions." No mention was made of scanning resolution issues, as this aspect was not covered by the first experiment. The instructions subsequently briefed the participants to draw as many markers as possible that could be placed in a public space to attract attention to any of the following topics: "Music", "Animals", "Pollution/Energy Consumption" and "Children". It was made clear that markers could be distributed in any way subjects liked, from having all of them belong to one topic to an even distribution. Participants were instructed that the level of detail and accuracy should be just enough for someone else to guess which of the four topics each marker is related to.

The experiment was designed for subjects without specific drawing skills nor familiarity with graphic software applications. For this reason, the drawing took place on a white dry-erase board with a black pen: an informal medium that affords playful drawing and easy modifications. While the ability to hand-draw markers can be advantageous in some situations, we imagine marker design to take place mostly through graphic design software tools – we choose to use the whiteboard in the experiment to avoid effects related to familiarity and learning curves of specific software applications or digital drawing devices (e.g. tablet computers), we expected participants from the university to be already familiar with this medium.

To check the validity of the drawings as markers an early version of the "DTAnalyser" was used, running on a standard desktop computer running MS Windows XP. This version differs from the one described above in that it does not provide information about the segmentation in connected components, it does not include zoom functionality and the interface is based on tabs, rather than thumbnails. The application was connected to a webcam pointed at the board, firmly attached to a desk (so it could not be misplaced). When participants wanted to check their drawings they had to explicitly import them by clicking on a button. The imported images had a resolution of 320×240 pixels. A separate window showed the continuous video feed of images from the webcam, displayed in normal or thresholded (black and white) mode according to participant preference. All the information displayed to the users, all the images they checked as well as their actions within the DTAnalyser were logged with accurate timestamps. The experimental setup is illustrated in Figure 6.

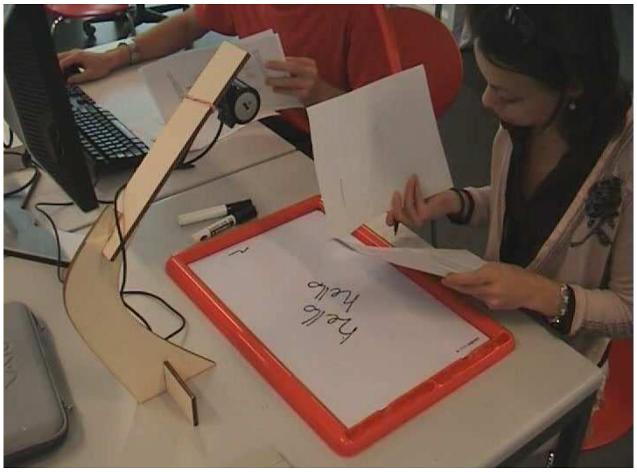


Figure 6. The experimental setup.

In both experimental conditions the DTAnalyser would inform users whether or not the proposed drawing is a valid d-touch marker. In the case of a drawing that is not a valid d-touch marker the application behaviour varied depending on the experimental condition. In the "Feedback" condition the DTAnalyser provided information about detected violations of the d-touch rules, namely the presence of nesting beyond three levels, and less than half of the white areas containing black regions. In both conditions the application did not provide any information about scanning resolution or robustness of the markers.

Participants were asked to save their drawings as soon as they were valid and the participants were satisfied with them. A "save" function was included in the application. This required participants to name their drawings according to the relative theme, and it was made available only for valid markers. When attempting to save the system would also inform the users if a marker with the same ID had already been registered within the same experimental session – in such a case participants were asked to modify the current marker to avoid the ID collision. Subjects were asked to swap chairs and roles after drawing each marker, so that the person drawing always sat in front of the whiteboard and the other person in front of the keyboard and mouse. The computer monitor was visible to both participants.

Experiment Description

Thirteen pairs of volunteers were recruited from our university population, through posters and mailing lists. A total of 6 females and 20 males, age range between 19 and 38 (mean:

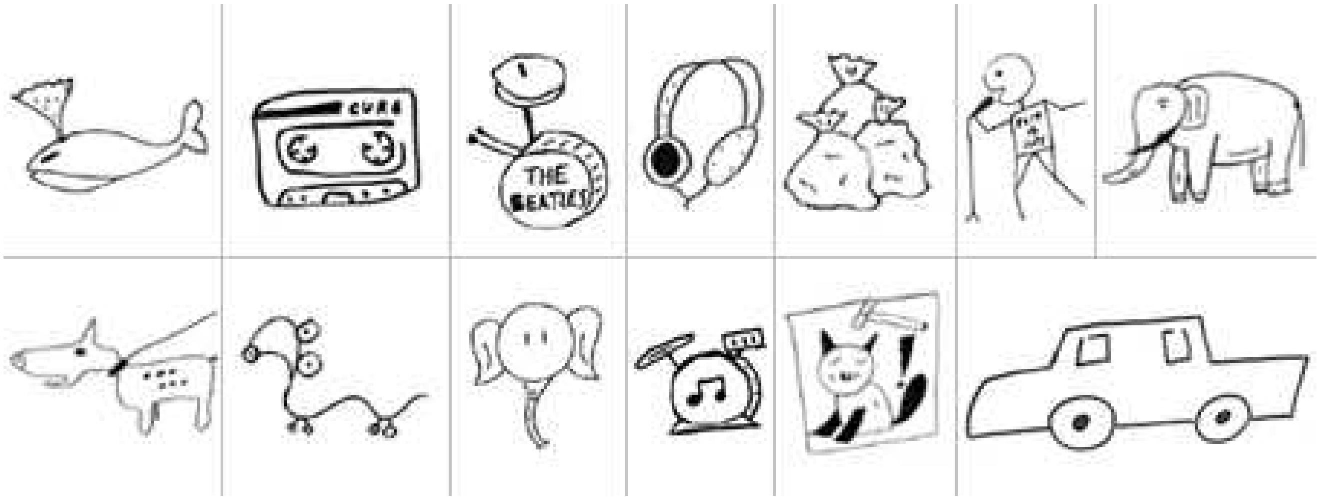


Figure 7. A selection of valid markers produced in the first experiment.

24.7, std. dev. 4.6). All subjects expressed interest in participating via email, showing familiarity with computers, and received 20 CHF (approximately equivalent to 18 USD) for their time. Anyone who expressed interest and was above 18 years of age was included in the study, no specific drawing skills were required to participate.

Results

The participants in one of the sessions in the "Feedback" condition drew only 3 markers, but these were considerably more complex than the ones produced in the other sessions, therefore data for this session was excluded from the analysis.

An average of 13.4 valid markers (std. dev. 5.7) were saved in each of the 1-hour long sessions, ranging from a minimum of 6 to a maximum of 27, median 13, and an overall total of 161. A selection of markers produced in the first experiment is reported in Figure 7. In the "Feedback" condition the average per session was 14.7 (std. dev 3.8) while in the "No feedback" condition 12.2 (std. dev. 6.9) and a one-way ANOVA test shows no significant difference. However, removing one outlier (distance from the average is twice std. dev) from the "No feedback" condition, the average for "No feedback" is 9.2 and std. dev 2.28, and one-way ANOVA test indicates that this difference is significant ($p < 0.05$).

In 6 of the sessions participants drew markers with the same ID as one of their previous entries. This happened between 1 and 3 times per session; in all cases subjects modified their marker and solved the ID conflict within seconds. Out of a total of 161 valid markers submitted, 116 had different IDs, corresponding to an ID collision rate of 27.9%. It must be underlined that subjects were blind to the IDs generated by other participants.

The number of inner regions, or *branches* and *leaves* in the terminology defined above, was taken as an indicator of the complexity of the markers; markers had on average 5.2 branches (std. dev. 3.1) with a minimum of 3 (the minimum accepted by the system) to a maximum of 19, median 4. Figure 8 shows the distribution of these values. The number of leaves was on average 8.8 (std. dev. 7.0), minimum 2, maximum 49, median 7. One-way ANOVA showed no significant differences either in the average number of branches or in the average number of leaves per marker between the two conditions.

Application Logs and Video Recordings

The log files collected by the DTAnalyser application show that participants tested their drawings 482 times over the entire experiment. 195 times (40.5%) the candidates were valid markers, while 287 times (59.5%) they were not. The invalid candidates were manually categorised according to the reason for being invalid; this analysis revealed that in 153 of the 287 times (53.3%) an input drawing was not valid, it was because of "artifacts" specific to the whiteboard, such as incomplete pen strokes and gaps in large filled areas, which modified the topology of the drawings. Analysis of the video recordings showed that these artifacts were often a cause of frustration, and that participants often attempted, and succeeded, to detect problems in the drawings of their partners before checking through the DTAnalyser.

Marker Coding

Two volunteers, also recruited from the university population, served as independent coders of the symbols drawn in the study. The coding took place through a simple web application where the markers were shown one at the time, and for each one the coder was asked to answer:

> Which theme do you think this drawing is related to?
> 1. Music 2. Animals 3. Pollution 4. Children
> 5. None of the above

and to also enter a freeform short description of the marker.

In 133 of 161 cases (82.6%) the coders agreed with each other. In 123 of the cases (76.4% of the total) both coders' choice was in agreement with that declared by the creator of

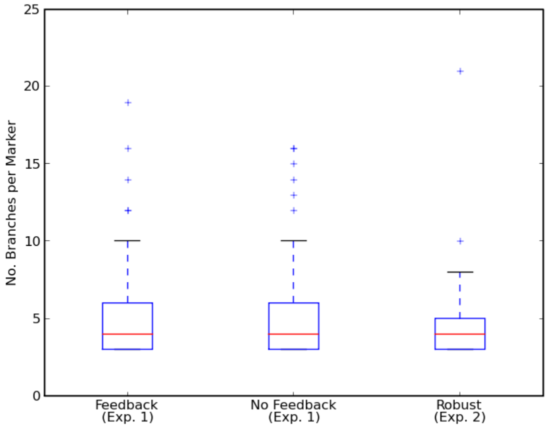


Figure 8. Boxplot of the number of branches (white regions, in this case) per marker, for the entire study. The red horizontal line indicates the median value, the box displays the inter-quartile range containing the central 50% of the data, while the whiskers indicate the extension of the data without outliers (these are shown by the crosses).

the marker. For 148 of the markers (91.9%) at least one of the two coders guessed the intention of the the creator of the marker.

Discussion

The fact that in all sessions participants managed to draw at least six valid markers, together with the high recognition rate of the marker meaning by coders, confirms the first hypothesis: people can, with minimal training, draw markers recognisable by the system and that at the same time express a concept or message which can be understood by others. The difference in the number of markers produced per sessions suggests that the DTAnalyser application can help in creating valid markers. The distribution of branches indicates that participants were able to create complex markers, but generally drew simple ones – indeed in the experiment there was no requirement nor incentive to favour complexity.

Even though participants were warned in the instructions, artifacts specific to the whiteboard were often cause of frustration. The version of the DTAnalyser application used in this first experiment was unable to help with this kind of artifacts, so their presence somehow “disturbed” the comparison of the two conditions. The visualisation of segmented components was added to the DTAnalyser to remedy to this problem. Automatic closure of open contours was considered as a possible extension of the software, however, it was noticed that sometimes users deliberately leave small gaps in the drawings as a way to reduce the levels of nesting – for example this is the case for the windows of the car on the bottom right of Figure 7 – making it very difficult to distinguish artifacts from deliberate choices.

EXPERIMENT 2: ROBUST MARKERS

Experimental Design

Based on the results from the first experiment, the second experiment aimed at assessing the creation of markers which

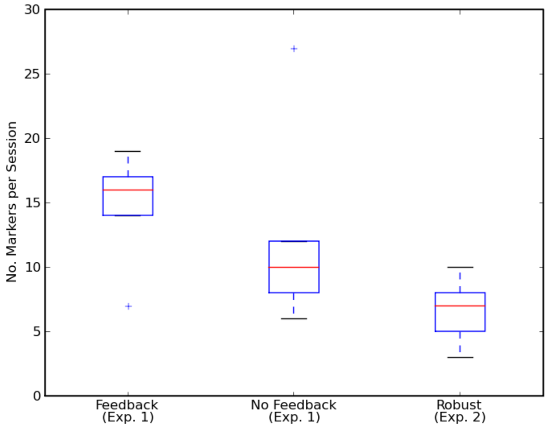


Figure 9. Boxplot of the number of valid markers created per session.

are not only valid, but also robust to blur and low resolution scanning. The experimental setup for the new experiment was nearly identical to the first one, except that this time the instructions covered the issues of marker robustness, and subjects were asked to draw markers which could be read at a (simulated) resolution of no more than 200×200 pixels or preferably no more than 100×100. The DTAnalyser application was used in its current version, as described in the “d-touch Technical Overview” section.

Experiment Description

Eight new pairs of volunteers took part in the second experiment, 9 females and 9 males, age range between 20 and 32 (mean: 24.6, std. dev. 2.8). Subjects were recruited and compensated as in the first experiment.

Results

Of the markers submitted by the participants only one did not comply with the resolution requested in the instructions and this was excluded from the following analysis. A total of 58 valid markers were submitted over the 9 1-hour long sessions, on average 6.4 valid markers in each of them (std. dev. 2.0), ranging from a minimum of 3 to a maximum of 10, median 7. A selection of the markers produced in the second experiment is reported in Figure 10. Out of the 58, 36 markers (61.1%) satisfied the more stringent requirement of 100×100 pixels. The number of distinct IDs produced was 44 (75.9%).

The average number of *branches* was 4.6 (std. dev. 2.6), ranging from 3 to 21, median 4. The number of leaves per marker was on average 6.26 (std. dev. 4.51), minimum 2 and maximum 25, median 5. Comparing these values with the results from the “Feedback” condition in Experiment 1 through one-way ANOVA reveals that the difference in number of markers produced per session is significant ($p<0.01$), as well as the difference in number of leaves per marker ($p<0.01$), while no significant differences were found for the number of branches per marker.

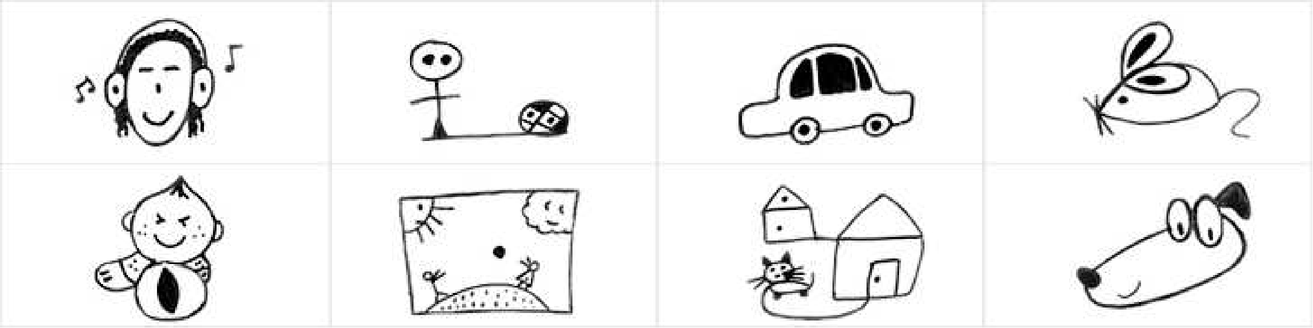


Figure 10. A selection of valid markers produced in the second experiment.

Marker Coding

The same volunteers who coded the first experiment coded the second one as well, following the very same procedure. In 47 of 58 cases (81.0%) the coders agreed with each other. In 44 of the cases (75.9% of the total) the choice of both coders was in agreement with what declared by the creator of the marker. For 54 of the markers (93.1%) at least one of the two coders guessed the intention of the the creator of the marker.

Discussion

The second experiment demonstrates that novice users can create robust valid markers with minimal training. Compared to the first experiment, the increased complexity of the task resulted in fewer markers created in each session, and in lower complexity, at least taking the number of leaves as an indicator.

Distinct IDs

Across both experiments a total of 219 valid markers were created, 149 of them (64.8%) corresponding to distinct IDs, while 34 IDs were repeated between 2 and 9 times. It must be emphasised that each pair of users was only aware of the IDs that they produced during the experiment and that they were "blind" to the ones produced by other groups. While further investigation is needed to assess how many distinct IDs can be supported by the system, we expect that the number of IDs will be large enough to cover many mobile applications. We envision markers and IDs to be application-specific, rather than having one central repository: with each application maintaining its own database of d-touch markers. Moreover, multiple markers can be combined together to increase the ID space; markers can be placed next to each other and scanned simultaneously by the client. As an example, using 2 markers from a set of 44 (as those produced in the second experiment) yields 990 combinations.

Limitations

While the whiteboard proved to be easy to use by our subjects, the artifacts suggest that the use of a digital medium, such as a tablet computer should be reconsidered. The performance recorded in the experiments is relative to a set initially empty: the difficulty of the task can be expected to increase if subjects had to add markers with unique IDs to a set already populated with hundreds of markers. More generally, the relatively small size of the ID address space (especially when compared to 2D barcodes) implies that markers cannot encode URLs directly, so the IDs must be used as keys to a database (on-line or residing on the phone itself). Similarly, it would be difficult to encode checksum information directly in the topology-based IDs, so redundancy is more likely to be included through geometrical features.

CONCLUSION AND FUTURE WORK

This paper provided an overview of d-touch, a visual marker recognition system based on image topology. d-touch markers are defined in terms of nesting dark and light regions. We demonstrated through a user study how such system enables people to design their own functional visual markers, determining their aesthetic qualities and what they visually communicate to others. The constraints imposed by d-touch are flexible enough to allow novice users to create markers which are visually expressive and at the same time machine readable. The system is cross-platform and released under an open source license. We hope that the material presented here will open up new opportunities for the design of mobile interfaces that use visual markers to relate digital information to the physical world.

In the future we plan to conduct longitudinal studies related to the design of d-touch markers, as well as mobile applications of the system, starting by uWiki. One specific aspect to be covered by future studies on marker design is the avoidance of ID collisions. At the technical level, further investigation will explore the opportunity to use geometrical features to differentiate markers with the same topological structure.

ACKNOWLEDGEMENTS

We are thankful for the constructive comments and advice from the CHI anonymous reviewers and program committee, from Pierre Dillenbourg and Patrick Jermann at EPFL CRAFT, from Ian Oakley at Lab:USE, CMU Portugal and from our colleagues in the EPFL Media and Design Lab.

REFERENCES

1. d-touch source code.
http://sourceforge.net/projects/libdtouch.

2. Mobile codes consortium. http://www.mobilecodes.org/, February 2007.

3. Ross Bencina, Martin Kaltenbrunner, and Sergi Jordà. Improved topological fiducial tracking in the reactivision system. In *Proc. IEEE International Workshop on Projector-Camera Systems (Procams 2005)*, San Diego, USA, 2005.

4. Enrico Costanza and Leinss Mirja. Telling a story on a tag: The importance of markers' visual design for real world applications (paper presentation). In *Proceedings of the Mobile Interaction with the Real World Workshop at Mobile HCI2006*, Espoo, Finland, sep 2006.

5. Enrico Costanza and John Robinson. A region adjacency tree approach to the detection and design of fiducials. In *VVG*, pages 63–69, 2003.

6. Enrico Costanza, Simon B Shelley, and John Robinson. D-touch: A consumer-grade tangible interface module and musical applications. In *Proceedings of Conference on Human-Computer Interaction (HCI03)*, 2003.

7. Enrico Costanza, Simon B Shelley, and John Robinson. Introducing audio d-touch: A tangible user interface for music composition and performance. In *Proceedings of the 2003 International Conference on Digital Audio EffectsDAFx03*, 2003.

8. Diego Lopez de Ipina, Paulo R. S. Mendonca, and Andy Hopper. Trip: A low-cost vision-based location system for ubiquitous computing. *Personal Ubiquitous Comput.*, 6(3):206–219, 2002.

9. Mark Fiala. Artag, a fiducial marker system using digital techniques. In *CVPR (2)*, pages 590–596, 2005.

10. ISO. International standard, iso/iec18004. ISO International Standard, Jun 2000.

11. ISO. International standard, iso/iec16022international symbology specification, data matrix. ISO standard, 2006.

12. Woodland Norman J. and Bernard Silver. Classifying apparatus and method, October 1952.

13. David J. Johnston and Adrian F. Clark. A vision-based location system using fiducials. In *VVG*, pages 159–166, 2003.

14. H. Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings of the 2nd International Workshop on Augmented Reality (IWAR 99)*, San Francisco, USA, October 1999.

15. Kaj Mäkelä, Sara Belt, Dan Greenblatt, and Jonna Häkkilä. Mobile interaction with visual and rfid tags: a field study on user perceptions. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 991–994, New York, NY, USA, 2007. ACM.

16. Mathias Mohring, Christian Lessig, and Oliver Bimber. Video see-through ar on consumer cell-phones. In *ISMAR '04: Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'04)*, pages 252–253, Washington, DC, USA, 2004. IEEE Computer Society.

17. Takao Nakamura, Atsushi Katayama, Masashi Yamamuro, and Noboru Sonehara. Fast watermark detection scheme from camera-captured images on mobile phones. *IJPRAI*, 20(4):543–564, 2006.

18. Eamonn O'Neill, Peter Thompson, Stavros Garzonis, and Andrew Warr. Reach out and touch: Using nfc and 2d barcodes for service discovery and interaction with mobile devices. In Anthony LaMarca, Marc Langheinrich, and Khai N. Truong, editors, *Pervasive*, volume 4480 of *Lecture Notes in Computer Science*, pages 19–36. Springer, 2007.

19. Jun Rekimoto. Navicam: A magnifying glass approach to augmented reality systems. *Teleoperators and Virtual Environ.*, 6(4):399–412, 1997.

20. Jun Rekimoto and Yuji Ayatsuka. Cybercode: designing augmented reality environments with visual tags. In *DARE '00: Proceedings of DARE 2000 on Designing augmented reality environments*, pages 1–10, New York, NY, USA, 2000. ACM Press.

21. Michael Rohs and Philipp Zweifel. A conceptual framework for camera phone-based interaction techniques. In *Pervasive Computing: Third International Conference, PERVASIVE 2005*, number 3468 in Lecture Notes in Computer Science (LNCS), Munich, Germany, May 2005. Springer-Verlag.

22. Azriel Rosenfeld. Adjacency in digital pictures. *Information and Control*, 26(1):24–33, September 1974.

23. Ron Shamir and Dekel Tsur. Faster subtree isomorphism. *J. Algorithms*, 33(2):267–280, 1999.

24. G. Thomas, J. Jin, T. Niblett, and C. Urquhart. A versatile camera position measurement system for virtual reality tv production. In *Proc. Internat. Broadcasting Convention (IBC'97)*, pages 284–289, 1997.

25. Tikiwiki. http://tikiwiki.org/.

26. Eleanor Toye, Richard Sharp, Anil Madhavapeddy, David Scott, Eben Upton, and Alan Blackwell. Interacting with mobile services: an evaluation of camera-phones and visual tags. *Personal Ubiquitous Comput.*, 11(2):97–106, 2007.