Enabling Hand-Crafted Visual Markers At Scale

William Preston[1], Steve Benford[1], Emily-Clare Thorn[1], Boriana Koleva[1], Stefan Rennick-Egglestone[1], Richard Mortier[2], Anthony Quinn[3], John Stell[4], Michael Worboys[5]

[1]University of Nottingham, UK; [2]Cambridge University, UK; [3]Central Saint Martins College of Art and Design, UK; [4]University of Leeds, UK; [5]University of Greenwich, UK

{pszwp, pszsdb, pszet, pszbnk, pszsre}@nottingham.ac.uk, richard.mortier@cl.cam.ac.uk, a.quinn@csm.arts.ac.uk, j.g.stell@leeds.ac.uk, m.worboys@greenwich.ac.uk

ABSTRACT

As locative media and augmented reality spread into the everyday world so it becomes important to create aesthetic visual markers at scale. We explore a designer-centred approach in which skilled designers handcraft seed designs that are automatically recombined to create many markers as subtle variants of a common theme. First, we extend the d-touch topological approach to creating visual markers that has previously been shown to support creative design with two new techniques: area order codes and visual checksums. We then show how the topological structure of such markers provides the basis for recombining designs to generate many variations. We demonstrate our approach through the creation of beautiful, personalized and interactive wallpaper. We reflect on how technologies must enable designers to balance goals of scalability, aesthetics and reliability in creating beautiful interactive decoration.

Author Keywords

Visual markers; topological markers; fiducial markers; patterns; computer vision; image recognition.

ACM Classification Keywords

H.5.2m. Information interfaces and presentation (e.g., HCI): User Interfaces – Input devices and strategies.

INTRODUCTION

As locative media and augmented reality enter the mainstream and spread into everyday settings, so the design of aesthetically pleasing and yet scalable visual markers becomes ever more important. Aesthetics are important to ensure that markers fit harmoniously within the carefully designed interiors in which we live, literally becoming part of the 'fabric' of our homes, workplaces and public spaces. Scale is important so that unique markers can be applied to individual products that then become personalised to their owners' needs. Creating aesthetic visual markers at scale would enable us to decorate our world with interactivity.

What we mean here by 'visual markers' are images that have been designed to be recognizable by computers. They range from visual codes that encode an identity (either globally or relative to a specific application or local context) to fiducial markers that convey aspects of position and orientation relative to a camera. There is currently a gap between two broad approaches to creating visual markers. The first is handcrafting in which designers create bespoke images, either beginning with a blank page and following a set of drawing rules [5] or by selecting natural images that can be recognized by a computer [3, 22]. These are aesthetically pleasing but difficult to deliver at mass scale. The second involves the algorithmic generation of visual markers that are designed to operate robustly at scale but at the cost of limited aesthetics and interaction [11, 12].

We describe how we collaborated with skilled visual designers to establish new techniques to bridge this gap. This involved extending the existing d-touch topological rules for hand-drawing visual markers. First, we introduce the extensions of area order codes and visual checksums that enable designers to embed information into designs so as to support scalability and reliability while maintaining flexibility over their aesthetic. Second, we introduce an algorithm for automatically combining small numbers of handcrafted seed markers to create many distinct variations that adhere to a common visual design.

We describe how we implemented these ideas and worked with designers to demonstrate their feasibility by creating interactive wallpaper. Each panel of our wallpaper contains multiple markers disguised within an overall pattern. Moreover, each individual roll is mass-customised to contain unique codes while clearly adhering to the common theme and matching other rolls. Feedback from designers suggests that the approach can generate acceptable designs, though raised issues of ensuring visual flow and balance.

We reflect on our experience to draw out wider lessons for how future visual marker technologies need to be open to designers so as to help them manage the complex trade-offs between scalability, aesthetics and reliability. Adopting a wider viewpoint, we also suggest that it is time to look beyond the design of individual discrete visual markers to instead contemplate a world that is liberally decorated with beautiful interactivity.

DIS 2017, June 10-14, 2017, Edinburgh, United Kingdom

ACM 978-1-4503-4922-2/17/06…$15.00

DOI: http://dx.doi.org/10.1145/3064663.3064746

RELATED WORK

Machine-readable visual marker technologies fall into two general camps, those that are algorithmically generated and those that are hand-crafted.

The most well-known algorithmically generated marker technologies are barcodes and QR codes. There is also a wider family of 2D matrix marker technologies where black and white squares are arranged to encode data. These kinds of codes are engineered to be scalable in terms of supporting a very large 'code space' (number of uniquely distinguishable codes) and also to be reliable. In the extreme, a QR code can be generated for any web address. On the other hand they have a very limited aesthetic. Various techniques have been developed to overcome this limitation such as embellishing barcodes [8, 1] or QR codes [15] to make them more attractive. Some algorithms generate markers with aesthetic qualities beyond black & white lines and squares; ReacTIVision's Amoeba marker set was created using a genetic algorithm to have a more organic look, albeit with a relatively limited palette [2]. An alternative is to completely hide markers by using light outside the visible spectrum [19].

Hand-crafting includes various approaches in which the designer either selects or draws more natural looking images that contain the correct balance of features to make them recognisable by an image processing algorithm. Vuforia Image Targets, for example uses feature detection to recognize and reconstruct the pose of markers from the presence of natural features such as corners. This leads to impressive interaction capabilities and reliability (e.g., occlusion handling) but does not work well with some kinds of designs (less 'cornery' patterns such as circles are not recommended) [22]. Blippar applies machine learning to identify classes of objects such as vending machines [3]. An alternative approach is to provide designers with a set of drawing rules for creating visual markers from scratch. D-touch, for example, employs a topological approach in which designers follow simple drawing rules (see below) to embed codes into hand-crafted images [5]. This provides designers with great flexibility for creating aesthetic designs, including those in which codes are disguised within wider patterns, but raises the problem of reliability [16]. Subsequent research explored how multiple codes might be embedded into larger images such as pieces of public 'wall art' through the use of paths (scanning a sequence of codes), groups (scanning multiple codes at once) and by switching between colour filters so as to recognize different layers of codes within a design [21]. ARToolKit falls between these approaches. It makes any image placed within a specific frame readable by using the frame to recreate the planar image, reducing this to a fixed resolution matrix and comparing this against a library [14].

There are a few approaches that attempt to mix the scalability of automatically generated markers with a degree of hand-crafting. Chu et al. [4] presented a method to integrate halftone images into QR Codes, the result is a cross between the two with the QR Codes landmark features still visible. Yang et al. [24] presented a similar system for varying the color in an image to encode data resulting in noisy interactive images. Vuforia's VuMark enables the creation of a large number of designs that share a common visual theme [23]. It does this by having the designer specify a number of elements with 2 states that can be used to represent binary data. However, it introduces constrains that may limit artistic style; e.g. the outer shape must be a non-symmetric polygon made up of a limited number of straight edges and both states of elements must be block coloured shapes.

There is a great deal of work involved in generating large numbers of hand-crafted images. Either the designer must instruct the computer to recognize a large number of existing images, being careful to ensure that they are sufficiently distinguishable from one another, or they must hand-draw many variants of a basic design from scratch, or design with interchangeable states in mind. Not only will generating large numbers of designs take a great deal of time, but it can also become increasingly challenging as the code space increases, as designers must ensure that the computer can distinguish ever finer levels of detail. Indeed feature based image detection, such as methods based on Scale Invariant Feature Transform [25], have good occlusion tolerance hindering their ability to differentiate between very similar images such as those in Figure 1. Alternatively the d-touch approach can scale to large numbers of codes in theory but there will be a practical limit to the fineness of detail that designers can cope with without some more automated support. Certainly, the examples of the d-touch topological approach have to date been limited to small code spaces [5, 16]. The challenge of scale becomes even more difficult if a large number of designs need to share a common visual theme. Now the designer must create large numbers of images that are sufficiently different that the computer can distinguish between them while being sufficiently similar that a human sees them as being part of a common family.

This distinction between the scalability and reliability of algorithmically generated markers on the one hand, and the aesthetic of hand-crafted ones on the other, defines the central challenge of our paper. How can we obtain the best of both worlds – generate large numbers of markers that share a beautiful and common design aesthetic and that can also be scanned reliably? Put simply, how can we enable the hand crafting of beautiful markers at scale?

APPROACH

There are two key aspects to our approach to this challenge: the technical choice to extend the d-touch system and the methodological approach of following a design-led process.

Extending D-touch

Technically, we chose to extend the d-touch topological approach to hand-crafting visual markers first proposed in [5]. We chose this as previous research has shown it to be especially suited to use by graphic designers. Specifically, trained designers appreciated the openness of the d-touch rules, quickly learning how to draw valid markers before then figuring out how to creatively exploit the rules to embellish and otherwise disguise these within a variety of patterns [16]. It has also been shown to be usable by novice users with basic support [6]. As researchers, we were drawn to the inherent openness of the d-touch rules that makes them both readily implementable and extensible. d-touch has been extended in the past: extending its detection functionality with orientation information [2] and extending its aesthetic opportunities by introducing color filters [21].

Given the centrality of d-touch to what follows, we now briefly review the rules as set in [5] before introducing various extensions below. A valid d-touch marker comprises a three level hierarchy. First, a *marker* is a continuous dark shape. Second, this shape must contain a number of light *regions*. Finally, these regions are populated with varying numbers of solid *blobs* (these cannot themselves be hollow – i.e., be further sub-regions). The number of regions gives the number of digits in the code. The number of blobs in each region gives the value of each digit. These are then written in ascending order. The shapes of the regions and blobs and ordering on the page are not considered, giving designers great flexibility to create varying (and indeed multiple) visual designs for each code or indeed to create visually similar variants of different codes. Figure 1 shows similar looking variants to two different d-touch codes, in this case each with five regions, but varying in the numbers of blobs.

Figure 1. Two visually similar d-touch markers that embed different codes and their topology hierarchies.

From the outset, we identified our key goal as scalability with sub-goals of maintaining aesthetic freedom and reliability.

Scalability – we wanted to enable designers to embed more information into their drawings so as to allow for more codes to be generated. Although the d-touch code space is unlimited in theory, practical limitations on camera resolution limit the numbers of regions and blobs that can be included in a marker. We therefore explored whether d-touch might be extended to recognize other features to allow additional information to be introduced. We also wanted to empower designers to be able to create large numbers of markers *in practice*.

Aesthetics – we wished to ensure that the topological approach remained open to the creativity of the designers. It was important that any new extensions did not close down the scope of possible designs by requiring designers to create a particular style of image, for example one dominated by a certain kind of recognizable visual feature.

Reliability – it was desired to maintain the reliability of scanning, and ideally to also bring aspects of this under the control of designers so that they could manage the various trade-offs involved between scale, aesthetics and reliability.

Design-led process

We were keen to involve designers throughout our research process, understanding their needs and drawing on their expertise to help find an appropriate balance between scalability, reliability and creative aesthetic. This involved an iterative process, cycling between technical development, working with designers through workshops and creating prototype artefacts, before reflecting on these. This process unfolded as follows:

- We considered various possibilities for how we might extend d-touch to address our goals.
- We implemented a selection of these in a mobile app so that designers could experiment with them.
- We ran a day-long workshop with four professional designers to test out these possibilities. Two worked as illustrators in the advertising sector, the third described themselves as working in a range of areas from textiles to business logos. The fourth worked in ceramics. We invited the participants to complete three tasks: creating a visual design for a code of our choosing; creating a second one for a code of their choosing; and creating a harmonious family of designs.
- We refined our implementation and then commissioned two of the designers to create an artefact – the interactive wallpaper that we describe below – to further test and demonstrate our extensions.
- We developed a separate design tool to automatically generate large numbers of new designs from a few hand-crafted seed designs.
- We used this to generate variants of the wallpaper and ran a feedback session with its designers.

EXTENDING TOPOLOGICAL MARKERS

We present the results of this process in two parts. In this section we document the rationale for, implementation of and initial experience with two complementary extensions to the topological approach: *area order codes* and *visual checksums*. In the subsequent section we introduce the approach to automatically combining designs.

Area Order Codes

Our first extension introduces an ordering to the codes. Costanza previously considered adding an ordering to topological markers by making the region's centre points collinear and reading them in order of distance from a pivot or pre-defining a fixed shape for markers [7]. We opted for an alternative strategy of utilising knowledge of the relative sizes of the regions in the marker to order the code as in our early discussions designers felt this is something they could control. Consider the example shown in Figure 2. Under the original d-touch approach of [5] this corresponds to the code 1.1.3.3.4 (five white regions all joined together that contain 1, 1, 3, 3 and 4 solid blobs respectively). Under the area order extension, the code is given an ordering according to the size of its constituent regions. Thus, the example of Figure 2 now becomes the code 4.3.1.3.1, which can be recognized as being distinct from other orderings that would arise when the regions are given different relative sizes. The rationale for proposing this extension was as follows:

- It increases the available code space: for a design with *R* regions where each region can contain up to B blobs there would be B^R total codes.
- The approach is largely shape invariant so that designers would still enjoy a great deal of flexibility over the relationship between visual aesthetic and embedded code – i.e., could easily create different looking patterns for a common code or vice versa.
- It is backward compatible with the original d-touch algorithm. Extended codes can still be read as d-touch codes. Indeed, this may introduce new creative potential in terms of being able to treat extended codes as either being distinct instances or part of a family group.

Due to perspective, the apparent region areas will change depending on the camera angle. This must be considered when designing the interaction and markers, taking steps to either prevent or embrace this. This can be prevented by using a reasonable step change in scale between each region or using a checksum (described later). Or embraced by allowing different codes to be intentionally read at different angles and triggering different actions. Figure 3 shows a map of what codes can be read at what angles for a simple marker. A map like this can be used to help plan the interaction. This is a special case, 3 regions arranged in a triangle, where all orderings can be read from a single marker. As you increase the number of regions, vary their size and shape, many orderings become impossible to read. This allows you to create markers with the same structure and numbers that reads differently by changing the position of the numbers in the image.

At the workshop, all four designers managed to successfully complete all three tasks for this extension. Figure 4 shows a representative selection of initial sketches that emerged from the workshop, illustrating area order codes and visual checksums.

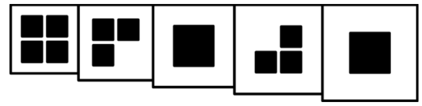

Figure 2. A simple topological marker.

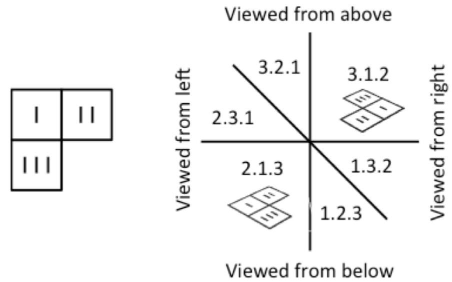

Figure 3. A d-touch marker and a map of the orderings that can be read at different angles under the area order extension.

The workshop revealed that designers were able to understand and successfully employ this approach. All four professed their enjoyment, commenting that the extension offered an appropriate level of challenge: *"It was a bit of a challenge but not in a frustrating way"*. They also noted that it was possible to anticipate area ordering when initially planning a design: *"Before I'd even drawn it I knew that I'd do a bigger bit here and a smaller bit here… I thought about that before I'd even put pen to paper"*. Although one did also reveal a more improvised approach: *"I just draw what I wanted to draw and then put the number into the regions… according to its size"*.

While our designers could generally reason about area ordering, this was not always easy. One encountered shapes with deceptive surface areas (Figure 4a). Their design contained two regions with almost equal areas that caused them to be read in different orders depending on the camera angle, while appearing to the eye to be quite different sizes.

We then introduced an additional task for our designers – to deliberately create a visual image that would read differently from different angles. We asked them to draw the code 1.2.3.4.5, but its order and what angles it would be read from were left open to the designers. Two of our designers used near symmetrical images with large regions on each side to achieve this (e.g. Figure 4b, right). When changing the camera angle from the left to right the order of the two large regions swaps over yielding a different code while the other regions maintain the same order.

The other two designers used non-symmetrical images to create three or more orderings from different angles (e.g. Figure 4b, left). One of these uses that same strategy as the symmetric images, a few large regions that change order

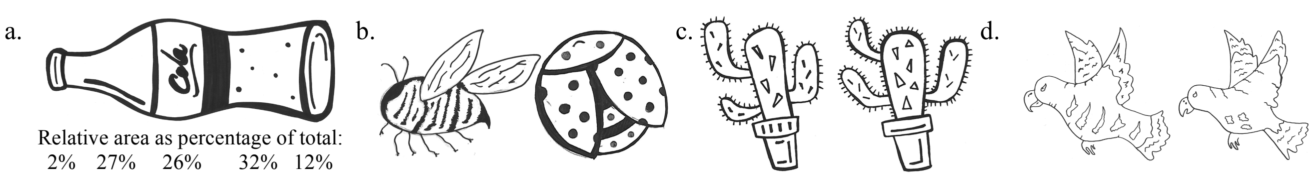

Figure 4. Example of topological markers drawn in the workshop: a. deceptive area sizes found while using area order, b. images that can be read differently at different angles using area order, c. & d. similar images with different codes using visual checksum.

depending on angle, the other placed smaller regions at either end of a long region relying on an apparent change in the smaller region's size.

Visual Checksums

With increasingly complex visual markers comes the associated challenge of maintaining reliability. Indeed, algorithmic approaches such as barcodes include redundant information in the form of checksums, an idea that has also been proposed for increasing the reliability of topological markers [16]. We therefore introduced a visual checksum mechanism into the drawing rules, but in a way that was intended to be comprehensible and open to designers. We extended the drawing rules to allow for an optional additional region that encodes a checksum value as a number of hollow-blobs (i.e. *non-solid* shapes). The checksum is calculated in a similar way to ISBN-10:

$$1+\left(-1+\sum_{position=1} region\ position \times value\right) \bmod 7$$

For example, Figure 5 shows this applied to our previous example from Figure 2. The checksum is calculated as 1 + (-1 + 1×1 + 1×2 + 3×3 + 3×4 + 4×5) mod 7 = 2, so two hollow blobs are now added into an additional region.

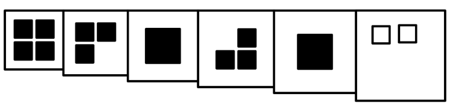

Figure 5. A topological marker using the visual checksum extension.

This additional region increases reliability by protecting against certain classes of recognition error such as when interference (dirt, reflections or shadows) causes blobs to appear and/or disappear. Table 1 shows that this checksum prevents a large number of errors. It might seem like using an extra region would reduce the number of codes verses using that region for data but against other proposed validation methods (such as those in [16]) it actually produces more codes as it does not require gaps in the code space. Although not backwards compatible with existing d-touch applications it would be easy to modify them to accept these new markers alongside existing ones.

Number of blobs effected by interference	1	2	3	4
Percentage of errors prevented	100	84	86	83

Table 1. Potential interference errors prevented by the visual checksum (given the number of regions is 5 with 1-6 blobs).

The area order and visual checksum can be used together by including both in the visual image and either calculating the checksum value over the code in area order or the conventional ascending value order. These two options have different (and useful) effects. Calculating the checksum using ascending value order allows for various permutations of the code to be read at different angles. Calculating the checksum based on area order guards against the code being read from an unintended angle. In either case the area size of the visual checksum region is not used in any calculation.

All four designers were able to successfully draw markers with visual checksums once given the required checksum value. However, they did require support from a software tool to calculate this value in the first place. They generally agreed that introducing hollow blobs could make their illustrations more aesthetically interesting by introducing a new variant on what they referred to as 'mark making': *"I just think it's more interesting, just mark making"*. However, there were concerns about working with codes with larger checksum values: *"I was hoping for a lower checksum number, but it came up with seven"*. Some felt it was best if the number of hollow blobs was similar to the number of 'normal' blobs so as to avoid jarring visual contrasts: *"If you had low numbers like 1.2.2 you'd hope the checksum to be a low 2 or 1"*. Extending the checksum calculation tool to also provide a list of all possible codes and their associated checksums would allow the designers to browse and choose the codes they felt most comfortable with. Finally, they commented on a tendency to use the largest region as the checksum region as this gave more space for hollow blobs: *"[It] makes sense to me to put them in bigger spaces because there is less chance of them not working. If I had to put 6 hollow dots in [a small region] it would be quite difficult whereas dots I could just put in 6 small dots. I do have to have them in the bigger area"*.

Implementation

We implemented the extended d-touch rules in a mobile app for Android and iOS devices. Our image processing workflow consists of colour filtering [21], threshold, contour detection [20] and topological code detection [5] (Figure 6). This is applied to frames from a video feed and runs in real time at up to 30 frames per second (on an iPhone 6). To improve recognition in challenging lighting conditions we use Otsu's threshold method [18], tiled and varying the number of tiles between frames. Like other topological marker systems we do not support partially occluded markers. The reliability of the markers depends on their design; Meese et al. suggested minimum line thickness

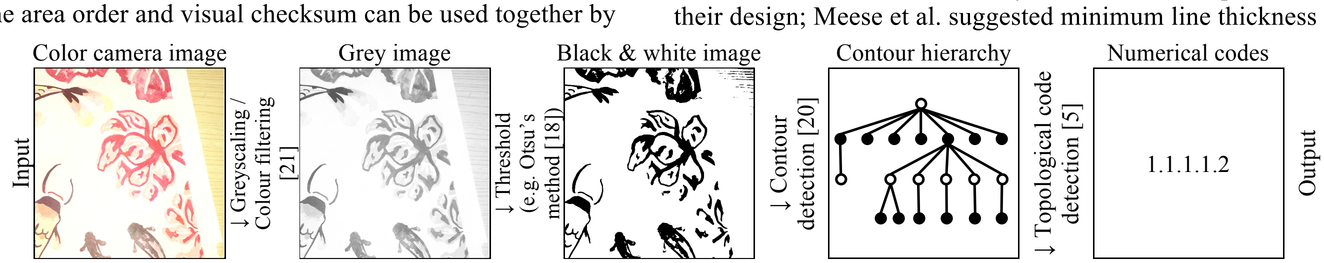

Figure 6. Image processing workflow for detecting topological markers. The marker is the flower.

and spacing to ensure reliability [16]. The software contains a configurable mapping of codes to websites. When a user scans a marker they are taken to the webpage mapped to the code encoded in the marker.

SEMI-AUTOMATIC GENERATION OF MARKERS

While designers are able to use our techniques to manually create individual visual markers, it is infeasible for them to handcraft and test many thousands of markers. Scaling up requires a degree of automation somewhere in the process. We therefore developed an approach to generating large numbers of new markers by automatically combining elements from a handful of handcrafted seed images.

Our approach exploits the regional structure of the topology by automatically extracting region contents (blobs) and then swapping them around to create new combinations. Figure 7 illustrates the basic principle. The three seed images (1.1.1, 2.2.2 and 3.3.3) have been manually designed to share a common region structure (the white regions are the same size and in the same place in each seed image) but with varied region contents (numbers of solid blobs). Fixing the region structure in this way allows for region contents to be copied and pasted between different markers to generate new variations such as the marker 1.1.3 as shown in Figure 7. In this case it draws its content from two of the initial seed images.

The number of new markers that can be generated in this way depends on the range of values provided in the seed images. However, the combinatorial approach means that even a small set of seeds can soon generate many new markers. Figure 8a shows the seven new conventional d-touch markers that can be generated by combining the three seed images.

The visual checksum extension is handled in the same way as the rest of the marker: the designer needs to include an extra region in their structure for the checksum and supply seed designs to cover all eventualities under our modulo-7 checksum scheme. The software works out the required checksum for each new code and inserts the value from the appropriate seed image into the checksum region.

This combinatorial approach also works with the area order extension generating additional area order markers by ordering the regions by area (Figure 8b).

Implementation

Our implementation works by:

1. Extracting the common region structure (size & position) from the seed images, counting the blobs and creating a set for each region's variations (e.g. for Figure 7: {1A, 2A, 3A}, {1B, 2B, 3C}, {1C, 2C, 3C}).
2. Creating a list of codes using the Cartesian product of these sets (e.g. {1A, 1B, 1C}, {1A, 1B, 2C}, {1A, 1B, 3C}, {1A, 2B, 1C}, …), and filtering by validation (e.g. the visual checksum).
3. Going through this list and creating individual markers by using one seed image as a base and masking the required regions from the other seed images (Figure 9).

Region masks are created using a bitwise 'or' over the area covered by the region in the base image and seed image as well as any differences between the images that overlap the region. This was implemented in Python using OpenCV.

If using the area order extension and the ability to read different codes at different angles, the software maps what angles produce what orderings by emulating the reading of the marker from many viewing angles. This allows for producing markers that read unique codes at every angle.

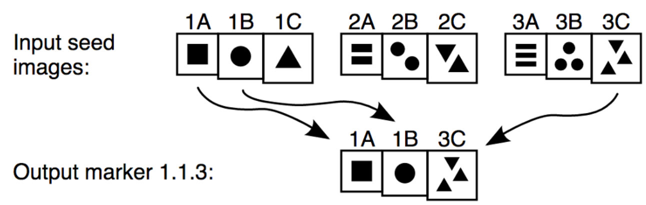

Figure 7. A generated marker created from combining elements of the seed images. The resulting marker takes 2 regions from the first seed (1A & 1B) image and 1 from the third (3C).

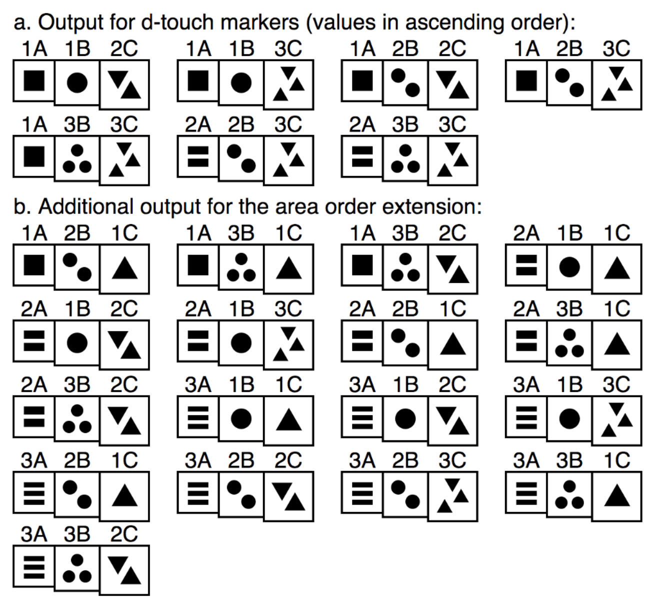

Figure 8. A small-scale example of the output of semi-automatic marker generation.

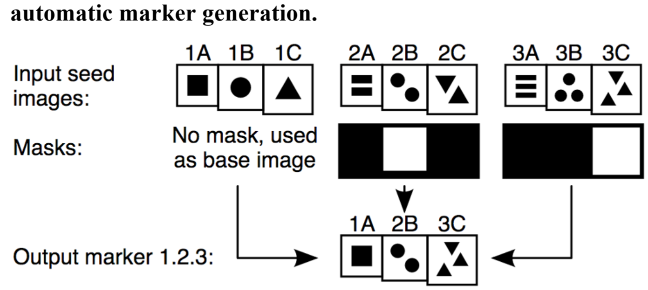

Figure 9. The process of generating a marker created from combining elements of the seed images with image masks.

DESIGNING INTERACTIVE WALLPAPER

We alighted on the challenge of designing interactive wallpaper as a way of exploring our new techniques in practice. We were also motivated by the idea that each individual roll of paper should be personalised, i.e., uniquely coded so that it could be tailored to deliver individual interactions. And yet we also wanted all of the rolls to share a common visual identity, being recognizable as the same design and also being suitable laying side-by-side when decorating a room. Ideally, we wanted the variations to be indistinguishable to humans, at least without close inspection. In short, each decorated room should contain dozens of visual markers, all of which should be distinct from those in thousands of very similarly decorated rooms so that owners could associate their own stories and information with their own rooms.

The initial design

We commissioned two of our designers to create some seed designs. The brief was that the wallpaper should be aesthetically desirable and also richly interactive, meaning that each panel should contain multiple distinct visual markers so as to support potentially complex narratives. In order to work within a sufficiently large code space we asked them to create codes of 7 regions, with each region containing up to 6 blobs. In theory this could generate 279,936 unique codes (6^7). However, we also asked our designers to use visual checksums to increase reliability and to experiment with area order markers that triggered different codes from different angles. Both of these techniques can constrain the code space and so in practice the number of usable markers would be somewhat smaller than this depending on the specific choices they made.

Our designers decided on a wildlife theme, drawing inspiration from the book and TV series *The Animals of Farthing Wood* [9]. Part of their final design can be seen in Figure 10. Markers are hidden in the 4 larger motifs with 4 smaller non-interactive motifs (snail, frog, flower/dragonfly and insect) in-between, all overlaid on a leaf background. The design reveals the great skill of the designers in being able to disguise the markers within a wider pattern. Compared to the earlier examples, it is difficult to spot them. As an aside, this inevitably raises the question of how people know how to interact with such hidden codes. This is not our focus in this paper, but we note that [16] has previously suggested various options including providing cues on the mobile app that could be switched on and off according to the nature of the experience (is it meant to be obvious or a playful 'treasure hunt') and the familiarity of the user with the artefact (are they encountering it for the first time or have they lived with it for many years).

Auto-generating variants

We now step through the auto-generation of one of the wallpaper panels, the hedgehog. Our designers provided 7 seed images that shared a common region structure (Figure 11). These included several variations of each region in terms of numbers of blobs: 2 of the regions had four variations; 1 region had 5; and 4 regions had 6. Their designs included all 7 possible variants of the visual checksum region. They used area ordering to ensure that their designs could be read differently from various angles. Given these 7 seeds and the constraints of area order codes, our software was able to generate 20,664 distinct markers that read different codes at between 2 and 8 different angles (yielding 116,352 readable codes in total) and a further 144 markers that read the same code from every angle.

Figure 10. A section of the interactive wallpaper. The codes are hidden in the 4 large motifs: top left hedgehog 1.1.2.3.4.4.4(6), middle right birds 1.1.3.1.1.1.3(1), middle left bees 1.1.1.2.1.1.2(1) and bottom right butterflies 1.3.2.3.1.3.1(1). Image courtesy of Lilli Cowley-Wood and Liz Jeal.

Figure 12 shows how one of the generated markers was composed, drawing the contents of its regions (including the checksum region) from 6 of the 7 seeds. Figure 13 shows an example of how reading a marker from two different angles yields different codes. The images are screenshots from our app in its debug mode that outlines recognised codes and shows their values as an overlay.

Feedback from designers

We held a final debrief meeting to capture our designers' views on both the outputs and process of auto-generation.

For the former, we invited them to inspect a selection of designs that included the two auto-generated ones with the maximum and minimum numbers of blobs (most and least busy), a further random selection of auto-generated designs, and their own seed designs. Overall they felt that the auto-generated content retained the original aesthetic: *"I think that the fact that it's initially worked out by hand means that it looks very natural."* This judgement involved two key criteria. First was the importance of visual flow, that lines should flow in a natural way without sudden jarring shifts of continuity or angle: *"I think the directional line is important as well. Just having all the curves going the right way. I think that hides the code and that hides the changes quite well."* This generally proved to be the case with just a few exceptions: *"The line here is at a slightly odd angle, that's the only one I would pick out as a little odd. The angle just needs to be flipped over*". Second was balance, which largely related to density of blobs across the design. This was also generally deemed to be acceptable, at least within limits: *"Looking at them they don't look that unbalanced but I suppose if you had someone more of a perfectionist and wanted more of a balance between them so you didn't have one that looking like it had less dots and one that looked like it had loads I wonder if you could have it so that it spreads them out a little bit more."* They felt that the fluid nature of their particular design was suited to the approach and that this might not be the case with other visual styles that were more geometric or structured: *"I think that our artwork is very illustrative and fluid whereas if it was something that was more structured then it would be more obvious."*

In terms of their wider views of the process, they were excited by the opportunity to work at scale: *"You could work with bigger chains that have to produce things on a larger scale"* and also that the cutting and pasting approach largely left the artist in control: *"The computer is just cutting and pasting what you've already done I suppose but to a level that wouldn't be practical."* They stated that they would be happy to associate their names to the final designs provided that it was *"done subtly"* and *"fitted with the flow and feel of the design"*. In terms of trusting the output they stated that they would be satisfied with only visually checking a small selection of the output: *"I could sleep easy after checking 10"*.

DISCUSSION

The previous sections provide initial evidence as to the feasibility and potential utility of our three extensions to topological markers. It would appear that, given appropriate support, skilled graphic designers are able to comprehend them and apply them to create beautiful interactive designs at a greater scale than was previously possible. We now conclude our paper by reflecting on the wider implications of our work in terms of moving beyond creating individual markers to designing interactive decoration; implications for future techniques; and challenges for further work.

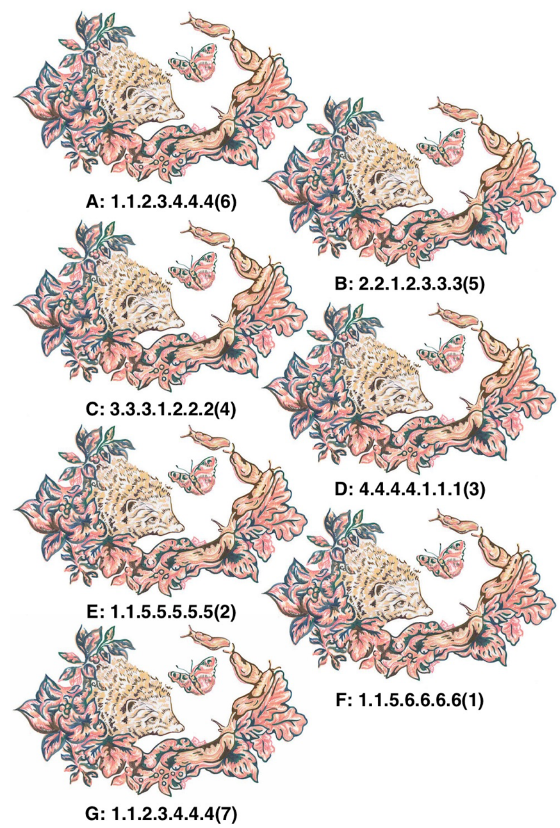

Figure 11. The seed images and their values (in area order) for the hedgehog panel of the wallpaper (checksum in brackets).

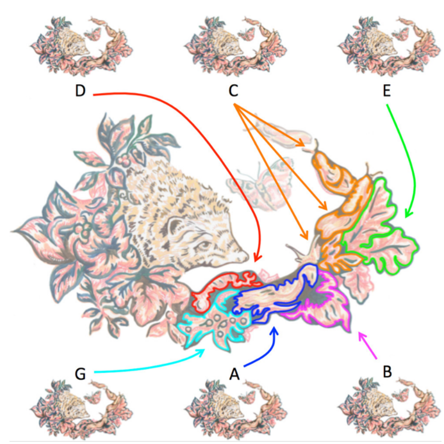

Figure 12. A generated marker (middle) and the six seed images it is composed of (top & bottom).

Figure 13. Scanning an auto-generated marker from different angles to yield different codes (1.3.5.2.6.3.1 on the left and 2.3.1.5.6.3.1 on the right). Uses debug mode to reveal the code.

Decorating the world with interactivity

The interactive wallpaper presented above is quite a long way removed from the kinds of visual markers that traditionally feature in augmented reality and locative media. Whereas the latter tend to be small, discrete and often clearly recognizable visual elements that can be attached to key locations and artefacts in the everyday world, our wallpaper is large in extent, disguises multiple markers within a wider pattern, and is intended to be a permanent background fixture of the everyday world. Put another way, our work suggests a shift in focus away from designing individual markers towards designing 'interactive decoration' in general. From fabrics and wallpapers to chinaware and clothes, our world is liberally decorated with patterns that enhance its beauty, add value to our belongings and allow us to express our tastes. We propose that it is time to move away from thinking of designing discrete markers to think more broadly of designing *interactive decoration* that can be wildly applied to the everyday world so as to cover it with interactivity. In turn, this requires engaging the design community in the development of new techniques.

Implications of interactive decoration

Reflection on our goals of scalability, aesthetics and reliability reveal key requirements for future techniques that aim to support this notion of interactive decoration.

Achieving **scalability** transpires to be a complex and multifaceted challenge. A baseline requirement concerns the *theoretical size* of the code space. How many codes could be generated in theory? In theory the d-touch approach is infinitely scalable by increasing the numbers and regions and/or blobs.

Second is the *practical availability* of the code space. How many codes can a designer squeeze into a given design that needs to be realized at a specified size and resolution and be recognised using cameras with a given resolution? The introduction of area order codes allows designers to squeeze more information into a given space by recognizing additional features of an image. While these factors may be relatively predictable for traditional markers such as QR codes, they will become far more contextual as we move to decorating a variety of artefacts at varying physical scales.

Next is the *practical delivery* of the code space. How can designers actually generate large numbers of designs? This is a major challenge for all of the handcrafted approaches that require designers to either select or draw, and potentially test and then apply, designs for each code that may be required. Our contribution here has been to introduce a technique to algorithmically generate many individual variants from a small number of seed designs.

Last, is the *structure of the code space*. Are 'addresses' flat or is it possible to distinguish classes from instances of things, for example a particular design of wallpaper from a particular roll of that design. This requirement is potentially important to support the mass customization [13] of future products where customers buy branded things but can then personalize it to their specific needs. D-touch already supports a structured code space in terms of the number of regions that are used in a code (one can separate five from six region codes for example). Our area order extension adds a further layer of structure – one can now read the same marker with or without area ordering.

Achieving **aesthetic** designs is also a multi-faceted challenge. New techniques would do well to avoid constraining the choice of *visual style.* The 'beauty' of d-touch is that topology is largely independent of visual style, enabling designers to draw the same code in many different ways. The same is broadly true of our area order and visual checksum extensions. However, techniques that rely on detecting particular kinds of features might constrain designers' choice (e.g. detecting corners [22] or straight contour edges [23] removes the possibility of generally circular designs). This openness of d-touch also allows designers to choose the extent to which they *disguise* detectable markers within wider patterns or not, an important aspect of aesthetic design that have been previously noted in [16]. Our experience with automatically generating interactive patterns reveals that *visual flow and balance* are important to the perceived aesthetic of a design and future algorithms would do well to accommodate these. Finally, we note the importance of adopting a *design-led methodology*, involving designers from the outset so that new techniques are responsive to their ideas and needs.

There are various well-known techniques for achieving **reliability** such as the use of checksums. Our introduction of *visual checksums* reflects the belief that it is also useful to expose these directly to designers, allowing them to incorporate them in their designs, so long as they have support for calculating what they should be or for choosing an option (number of blobs in our case) that best meets their aesthetic needs. Previous research suggests that reliability will also involve considering materials to be decorated and the context of deployment [16].

Our goals sit in tension and designers will need to carefully trade them off, making informed choices for a given design brief. Indeed, this trade-off underlies the two camps of marker technologies that we reviewed earlier, with algorithmic approaches tending to favour scale and reliability at the cost of aesthetics while hand-crafted approaches adopt the opposing stance. Our argument is that as we begin to widely decorate the world with interactivity, these choices need to be exposed to designers so that they can manage the trade-offs rather than hardwiring them into the underlying technologies.

It is unlikely that any new technique (e.g. an extension to d-touch rules) can equally address all goals. However, we argue that, ideally each new technique will target some of them without overly compromising others. Thus, our area order extension addresses scalability without compromising aesthetics and reliability while visual checksums address reliability without compromising aesthetics or scale. Overall, the aim should be to provide designers with a rich palette of techniques that allow them to carefully manage the trade-offs involved.

Future challenges for interactive decoration

While our experience suggests that our techniques are broadly feasible and have potential for creating interactive decoration, we note major challenges for future work.

The approach of algorithmically combining hand-crafted designs seems promising, but as our designers observed, it needs to be proved against a wider range of design styles that may test the aesthetics of visual flow and balance. What are the wider aesthetic constraints of mixing and matching designs in this way? Will all combinations look good next to one another and can we predict what will and won't work? More practical exploration is required.

While our designers could generally understand and work with our techniques, they require support for calculating checksums, dealing with viewing angles, previewing and testing generated outcomes and so forth. In short, there is considerable work to do in embedding new techniques into design tools, including as extensions to the tools that they naturally use (e.g., Photoshop and Illustrator).

Though not the focus of this paper, further work needs to explore the challenge of interacting with interactive decoration, especially where markers are disguised within large patterns.

APPLICATIONS

To generate ideas we have deployed the wallpaper at MozFest (an event run by Mozilla) and run a workshop with a national media broadcaster.

One application of the wallpaper, from the media broadcaster, is as a children's story telling device. Using cameras built into a tablet or toy, scanning the wallpaper could open new parts of the story or allow the reader to select characters at different points in the story. The uniqueness of the wallpaper could allow children to take their devices to a friend's home triggering interactions based on the new location. If placed in the bedroom it would allow a media broadcaster to release content accessed through the wallpaper at a child's bedtime encouraging them to get ready for bed.

Another idea is to use the wallpaper in a guest room as an enhanced interactive guest book. The wallpaper would allow visitors to 'check-in' when they arrive and attach photos or leave messages about their visit. The homeowner is then able to reflect on and curate the media. A normal guest book only documents facts but being able to virtually attach photos and videos adds another dimension while the located artifact makes the interaction feel different to sharing media online.

Using the generation software an organization could create branded visual markers for use in their ecosystem. For example a museum could create a topological marker icon, generate a number of variants and place them on the information placards that when scanned with the museum's app link to digital media about the exhibit. Visitors would be able to associate the icon with the app and know to scan it. There could also be a handful of visually different markers that indicate the type of digital media available. Scanning the markers through the museum could allow a visitor to build a digital scrapbook representing their visit that could be shared with others.

Placing aesthetic visual markers in the environment has also been proposed as method of indoor location, e.g. [10, 17]. Our generation software can produce many unique markers that could be used to give a reader's location to within a room or a few meters depending on the design. However, as topological markers do not support pose recreation, greater location accuracy would require further development.

CONCLUSION

Our exploration of how to hand-craft beautiful visual markers at scale has led us to introduce three extensions to the existing topological approach – area order codes, visual checksums and an algorithmic approach to combining a few hand-drawn seed designs to create potentially many variations. Collectively, these enable designers to manage the trade-offs between the competing goals of scalability, aesthetics and reliability. Our collaboration with designers to prototype mass-customised interactive wallpaper has demonstrated the feasibility of these techniques and suggested that we need to turn our attention away from discrete markers towards a more generalised notion of interactive decoration.

To conclude on a broader note, our approach to bridging between the hand-crafted and the algorithmically generated has been to start with the former and reach out to the latter. We have retained a focus on designers hand-making individual images but considered how algorithms might help scale these up. Future work might explore the alternative strategy – algorithmically generating large numbers of visual markers with an enhanced aesthetic. For example, could we extend our current simple recombination technique with genetic algorithms as used in ReacTIVision system to generate the organic looking Amoeba marker set [2]? Whatever the approach, we suggest that designers will need to be placed at the centre of the research process so that their natural understanding of aesthetics can inform new algorithms in a deep way and so that these algorithms are in turn open to their creative practice.

ACKNOWLEDGMENTS

This research has been supported by the Horizon Digital Economy Research Institute (EPSRC Grant No. EP/G065802/1 and EP/M02315X/1) and the 'Living With Interactive Decorative Patterns' project (EPSRC Grant No. EP/L023717/1).

REFERENCES

1. Barcode Revolution website. Retrieved January 2016 from http://www.barcoderevolution.com/
2. Ross Bencina, Martin Kaltenbrunner, and Sergi Jorda. 2005. Improved Topological Fiducial Tracking in the reacTIVision System. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR'05), 99-. http://dx.doi.org/10.1109/CVPR.2005.475
3. Blippar website. Retrieved February 2016 from https://blippar.com/en/faq/
4. Hung-Kuo Chu, Chia-Sheng Chang, Ruen-Rone Lee, and Niloy J. Mitra. 2013. Halftone QR codes.*ACM Trans. Graph.* 32, 6: Article 217. http://dx.doi.org/10.1145/2508363.2508408
5. Enrico Costanza and John Robinson. 2003. A Region Adjacency Tree Approach to the Detection and Design of Fiducials. In *Proceedings of Vision, Video, and Graphics* (VVG '03), 63-69.
6. Enrico Costanza and Jeffrey Huang. 2009. Designable visual markers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '09). ACM, New York, NY, USA, 1879-1888.
7. Enrico Costanza. 2010. *Designable Visual Markers for Mobile Human-Computer Interaction*. Doctoral Thesis. Ecole Polytechnique Federale de Lausanne, The School of Computer & Communication Sciences, pp. 24-26, 30 & 114.
8. D-barcode website. Retrieved January 2016 from http://www.d-barcode.com/
9. Colin Dann. 1979. *The Animals of Farthing Wood*. Heinemann, London.
10. Zita V. Farkas, Péter Korondi, Dániel Illy, and Lóránt Fodor. 2012. Aesthetic marker design for home robot localization. In *IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society*, 5510-5515.
11. GS1. 2016. *GS1 General Specifications, release 16* [standard document].
12. ISO. 2015. *ISO/IEC 18004:2015: Information technology -- Automatic identification and data capture techniques -- QR Code bar code symbology specification*.
13. Andreas M. Kaplan and Michael Haenlein. 2006. Toward a parsimonious definition of traditional and electronic mass customization. *Product Innovation Management*, 23, 2: 168-182.
14. Hirokazu Kato and Mark Billinghurst. 1999. Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality* (IWAR '99), 85-.
15. Yue Liu, Ju Yang, and Mingjun Liu. 2008. Recognition of QR Code with mobile phones. In *Proceedings of Chinese Control and Decision Conference (CCDC '08)*, 203-206.
16. Rupert Meese, Shakir Ali, Emily-Clare Thorne, Steve D. Benford, Anthony Quinn, Richard Mortier, Boriana N. Koleva, Tony Pridmore, and Sharon L. Baurley. 2013. From codes to patterns: designing interactive decoration for tableware. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13), 931-940. http://dx.doi.org/10.1145/2470654.2466119
17. Shinya Nishizaka, Atsushi Hiyama, Tomohiro Tanikawa, and Michitaka Hirose. 2009. Robustness enhancement of a localization system using interior decoration with coded pattern. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology* (VRST '09), 147-150.
18. Nobuyuki Otsu. 1979. A threshold selection method from gray-level histograms. *IEEE Trans. Sys., Man., Cyber.* 9, 1: 62–66.
19. Hanhoon Park and Jong-Il Park. Invisible Marker Tracking for AR. 2004. In *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR '04*), 272-273.
20. Satoshi Suzuki and Keiichi Abe. 1985 Topological Structural Analysis of Digitized Binary Images by Border Following. *CVGIP* 30, 1: 32-46.
21. Emily-Clare Thorn, Stefan Rennick-Egglestone, Boriana Koleva, William Preston, Steve Benford, Anthony Quinn, and Richard Mortier. 2016. Exploring Large-Scale Interactive Public Illustrations. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems* (DIS '16), 17-27. http://dx.doi.org/10.1145/2901790.2901826
22. Vuforia website. Natural Features and Image Ratings | Vuforia Library Prod. Retrieved January 2016 from https://developer.vuforia.com/library//articles/Solution/Natural-Features-and-Ratings/
23. Vuforia website. VuMark. Retrieved September 2016 from https://library.vuforia.com/articles/Training/VuMark
24. Zhe Yang, Yuting Bao, Chuhao Luo, Xingya Zhao, Siyu Zhu, Chunyi Peng, Yunxin Liu, and Xinbing Wang. 2016. ARTcode: preserve art and code in any image. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (UbiComp '16), 904-915.
25. David G. Lowe. 1999. Object Recognition from Local Scale-Invariant Features. In *Proceedings of the International Conference on Computer Vision* (ICCV '99), Vol. 2. IEEE Computer Society, Washington, DC, USA, 1150-1157.