Claycode: Shape-Free and Stylizable 2D Codes

Marco Maida, Marco Perronet

Abstract

1 Introduction

Similarly to QR codes, some aesthetic codes such as ARTcodes [1] use a matrix to encode the data. Other authors have employed steganography . . .

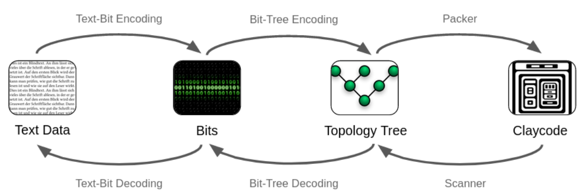
Topological markers encode information in the structure of the image rather than its pixel-level content. Examples of this approach include ReacTIVision [2], ARTag [3], D-touch [4][5][6], and Seedmarkers [7]. Previous work has explored how this technology can be used in practice to create aesthetic interactive scannable objects such as tableware [8] or speakers [7], and digital media such as wallpapers [9].

One of the major challenges for aesthetic markers is ensuring that the marker can be recognized as such by users [10]. In this paper, we address this issue both in the case where the raw data is encoded (section X), and when the data is defined by the art (section Y).

2 Pipeline Design

At a high-level, we separate the pipeline into three components: text-to-bit encoder, bit-to-tree encoder, and packer.

Figure 1: *Encoding-decoding pipeline.*



3 Bit-To-Tree Encoding

Within the context of claycodes, we are generating *rooted trees* [11]. Contrary to *plane trees* [12], in rooted trees the ordering among children does not matter. As a consequence, our encoding must be robust to different permutations of children while decoding (i.e. tree-to-bit), ensuring that isomorphic trees map to the same data.

Intuitively, the optimal solution would be a bijection between binary strings and rooted, unlabeled, unordered k-ary trees. However, it is not trivial whether such bijection would yield trees that are wide enough to be easily packed in practice. Therefore, we opted for an empirical approach and explored different different encoding strategies and evaluated them.

3.1 Encoding Strategies

We explain squares etc.

3.2 Evaluation

We compare the different strategies among each other.

3.3 Generation and data capacity

We compare the data capacity of claycodes to QR codes using the best encoding strategy to our knowledge. Here, give an idea of how the packer works.

4 Claycode Markers

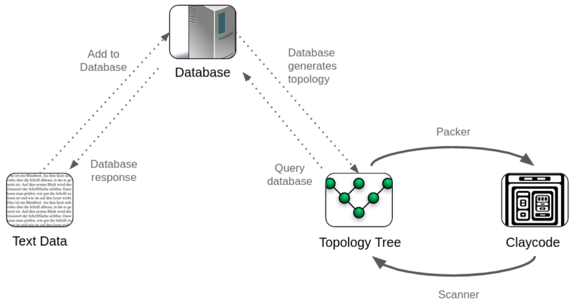
In this section we explore an alternative application of Claycodes: rather than encoding data in the code itself, we generate a topology starting from an arbitrary image and use it as the code. Under this design, the actual data is stored elsewhere (e.g. in a cloud application) while the code itself serves as a key to the data.

It has been shown in practice that the requirement for an internet connection does not hinder the adoption of a scanner application. In the case of Navilens [13] . . .

4.1 Art-Defined Markers

This approach allows the markers to be stylizable with minimal constraint on the artist.

Figure 2: *Revised encoding-decoding pipeline.*



4.2 Fragmented Markers

Using a dedicated editor, the artist can choose which areas are dedicated to the main subject, and which areas are dedicated to the floating fragments. Previous work has explored the idea of using an editor to help artists draw specific data while not violating the constraints of the marker [14]. Similarly to art-defined markers, this approach ensures that the whole image is stylizable, but it increases the burden on the artist and does not achieve error correction.

To achieve error correction based on redundancy, we can use a similar approach to instead generate multiple topologies embedded in an arbitrary image. This type of error correction is resistant to certain classes of tampering that QR codes are vulnerable to.

5 Scanner

High-level explanation of scanner pipeline.

5.1 Topology extraction

We employ pre-processing on the image to remove noise, then we classify each region with a unique id, and build an undirected graph describing which regions are touching. We call this a *touch graph*. In the following, we describe an algorithm which extracts a tree given a touch graph.

5.2 Avoiding false positives

We employed different strategies including CRCs, known prefixes, and known number of descendants.

6 Packer

While drawing a given code, we need to ensure it is as small as possible while still being scannable.

7 Conclusion

References

[1] Zhe Yang et al. "ARTcode: preserve art and code in any image". In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. UbiComp '16. Heidelberg, Germany: Association for Computing Machinery, 2016, pp. 904–915. ISBN: 9781450344616. DOI: 10.1145/2971648.2971733. URL: https://doi.org/10.1145/2971648.2971733.

[2] R. Bencina, M. Kaltenbrunner, and S. Jorda. "Improved Topological Fiducial Tracking in the reacTIVision System". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*. 2005, pp. 99–99. DOI: 10.1109/CVPR.2005.475.

[3] M. Fiala. "ARTag Revision 1, A Fiducial Marker System Using Digital Techniques". In: (Jan. 2004).

[4] Enrico Costanza and Jeffrey Huang. "Designable visual markers". In: *Conference on Human Factors in Computing Systems - Proceedings* (Apr. 2009). DOI: 10.1145/1518701.1518990.

[5] Enrico Costanza, Simon Shelley, and John Robinson. "D-touch: A Consumer-Grade Tangible Interface Module and Musical Applications". In: (Jan. 2003).

[6] Enrico Costanza, Simon Shelley, and John Robinson. "Introducing Audio d-touch: A Tangible User Interface for Music Composition and Performance". In: (Jan. 2003).

[7] Christopher Getschmann and Florian Echtler. "Seedmarkers: Embeddable Markers for Physical Objects". In: Feb. 2021, pp. 1–11. DOI: 10.1145/3430524.3440645.

[8] Rupert Meese et al. "From codes to patterns: designing interactive decoration for tableware". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '13. Paris, France: Association for Computing Machinery, 2013, pp. 931–940. ISBN: 9781450318990. DOI: 10.1145/2470654.2466119. URL: https://doi.org/10.1145/2470654.2466119.

[9] William Preston et al. "Enabling Hand-Crafted Visual Markers at Scale". In: *Proceedings of the 2017 Conference on Designing Interactive Systems*. DIS '17. Edinburgh, United Kingdom: Association for Computing Machinery, 2017, pp. 1227–1237. ISBN: 9781450349222. DOI: 10.

1145/3064663.3064746. URL: https://doi.org/10.1145/3064663.3064746.

[10] Liming Xu et al. "Recognizing the Presence of Hidden Visual Markers in Digital Images". In: *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*. Thematic Workshops '17. Mountain View, California, USA: Association for Computing Machinery, 2017, pp. 210–218. ISBN: 9781450354165. DOI: 10.1145/3126686.3126761. URL: https://doi.org/10.1145/3126686.3126761.

[11] Takeaki Uno Shin-ichi Nakano. "Efficient generation of rooted trees". In: (2003).

[12] Shin-ichi Nakano. "Efficient generation of plane trees". In: *Information Processing Letters* 84.3 (2002), pp. 167–172. ISSN: 0020-0190. DOI: https://doi.org/10.1016/S0020-0190(02)00240-5. URL: https://www.sciencedirect.com/science/article/pii/S0020019002002405.

[13] Universidad de Alicante. *Method of detection and recognition of visual markers of long reach and high density*. Patent ES2616146A1, 2016.

[14] Joshua D.A. Jung, Rahul N. Iyer, and Daniel Vogel. "Automating the Intentional Encoding of Human-Designable Markers". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, pp. 1–11. ISBN: 9781450359702. DOI: 10.1145/3290605.3300417. URL: https://doi.org/10.1145/3290605.3300417.