StandARone: Infrared-Watermarked Documents as Portable Containers of AR Interaction and Personalization

Mustafa Doga Dogan
Adobe Research
San Jose, CA, USA
MIT CSAIL
Cambridge, MA, USA
doga@mit.edu

Alexa F. Siu
Adobe Research
San Jose, CA, USA
asiu@adobe.com

Jennifer Healey
Adobe Research
San Jose, CA, USA
jehealey@adobe.com

Curtis Wigington
Adobe Research
San Jose, CA, USA
wigingto@adobe.com

Chang Xiao
Adobe Research
San Jose, CA, USA
cxiao@adobe.com

Tong Sun
Adobe Research
San Jose, CA, USA
tsun@adobe.com

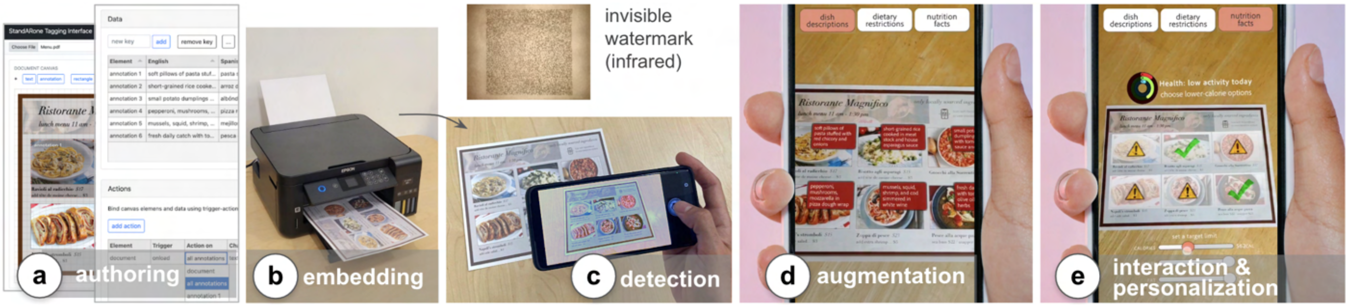


Figure 1: *StandARone* are physical documents that fully contain the AR interaction and personalization instructions in the paper medium itself. This example shows (a) how a designer can add interactive AR content (annotations, buttons, sliders) to a restaurant menu and bind it to custom data, such as descriptions of the dishes in various languages or their nutrition facts. (b) The AR content is embedded into the physical paper using an invisible watermark printed with IR inks. (c) The embedded AR content is unlocked on mobile devices to (d) augment the menu with the dish descriptions, and (e) interactively personalize it based on the user's device settings and profile (e.g., language or dietary restrictions).

ABSTRACT

Hybrid paper interfaces leverage augmented reality (AR) to combine the desired tangibility of paper documents with the affordances of interactive digital media. Typically, the instructions for how the virtual content should be generated are not an intrinsic part of the document but rather accessed through a link to remote resources. To enable hybrid documents to be portable containers of also the AR content, we introduce StandARone documents. Using our system, a document author can define AR content and embed it invisibly on the document using a standard inkjet printer and infrared-absorbing ink. A document consumer can interact with the embedded content using a smartphone with a NIR camera without requiring a network connection. We demonstrate several use cases of StandARone including personalized offline menus, interactive visualizations, and location-aware packaging.

CCS CONCEPTS

• **Human-centered computing** → *Human computer interaction (HCI).*

KEYWORDS

augmented reality; mixed reality; paper interfaces; documents; infrared imaging; watermarking; fabrication

ACM Reference Format:
Mustafa Doga Dogan, Alexa F. Siu, Jennifer Healey, Curtis Wigington, Chang Xiao, and Tong Sun. 2023. StandARone: Infrared-Watermarked Documents as Portable Containers of AR Interaction and Personalization. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems (CHI EA '23), April 23–28, 2023, Hamburg, Germany*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3544549.3585905

CHI EA '23, April 23–28, 2023, Hamburg, Germany

ACM ISBN 978-1-4503-9422-2/23/04.
https://doi.org/10.1145/3544549.3585905

1 INTRODUCTION

Documents are a pervasive medium for capturing, sharing, and communicating information [13]. Prior research has leveraged augmented reality (AR) technology to create *immersive* or *hybrid paper-digital interfaces* [14]. These systems combine the tangibility of paper documents with the affordances of interactive digital media. Some of the explored benefits of hybrid paper-digital interfaces include enriched input interactions [15, 27], enhanced indexing and search [19], and contextual and personalized content [3, 22]. A defining attribute of print media that is lost with *hybrid* paper-digital interfaces is the document's *portability* [13], i.e., all the information that the author originally intended to share, is fully self-contained and preserved in the document. The *Portable Document Format* or *PDF*s extended this defining attribute to the digital domain by making sharing and viewing files easy regardless of the operating system or software [9]. We introduce *Stand-alone AR* documents (shortened as "*StandARone*"), a portable hybrid document format that is the container of both physical and digital AR contents.

In most AR implementations, the physical and digital contents are only remotely linked, requiring manual work for end users to access experiences and thus hindering immediate immersion. To launch an individual AR experience on a viewer's phone, the user often needs to enter a URL, scan a visible QR code, or install a specific app on their phone. On the research side, much of prior immersive AR paper investigations have focused on the interaction design for different domain consumers [3, 12, 19–22]. Designing portable AR applications remains a difficult task [2] and few works have considered how the digital interaction data can be an intrinsic part of the physical document. To address these limitations, we investigate a watermarking method using infrared (IR)-based desktop printing to store the intended AR content (i.e., assets and interactions) as an intrinsic part of the document.

StandARone supports both an authoring and consumption workflow. *Authors* can use *StandARone*'s authoring UI to create AR assets (e.g., text labels, buttons, simple vector graphics) and define interaction behavior for these assets. The AR contents are encoded in a watermark and printed invisibly using near-infrared (NIR) ink. Authoring an AR experience with *StandARone* does not impact the authors' aesthetic intent as the added tags are completely invisible.

Consumers can decode the watermark using an IR camera in their smartphone. The watermark includes all data required for the consumer to launch and interact with the AR scene. *StandARone*'s approach preserves the document as the portable container of all information, including AR interactions and behavior. Figure 1 shows an overview of *StandARone*'s workflow used to create a hybrid restaurant menu with interactive AR content.

To demonstrate the capabilities of *StandARone*, we developed several applications and conducted a technical evaluation of the watermarking capacity. We take inspiration from prior work that has established user value in augmenting paper [3, 19, 21, 22]. Our demonstrations focus on showcasing the breadth of applications that *StandARone* can support and the additional benefits that using this method provides to the user (e.g., network-free, unobtrusive, portability). Our evaluation shows that *StandARone* documents could unobtrusively store up to 3,080 characters of data for virtual content generation. *StandARone*'s authoring interface leverages this capacity to embed the intended AR assets and interaction data into the document and thus does not rely on any remote database. This capacity allows the AR applications to be entirely portable within the physical document and all interactions to remain private between the physical document and the user's phone.

2 RELATED WORK

Combining the digital and the physical worlds has been one of the main uses cases of AR. Below we explain how paper is used for this purpose and how tags are used to specify paper-based interactions.

2.1 Hybrid Paper-Digital Interfaces

The benefits of hybrid paper-digital interfaces have been extensively explored in prior work [1, 19, 22, 24]. Han et al. conducted a systematic literature review and defined hybrid paper-digital interfaces as *"any interface embedding digital or electronic functionality in physical paper to enable its use as an input or output device."* Leveraging AR to add a digital information layer is a common technique used. *PapARVis* [3] and *Paper Trail* [22] explore authoring workflows that support educational and data visualization use cases. These include dynamic media such as animations and interactive UI components. *Dually Noted* [21] and *HoloDoc* [19] demonstrate the utility of augmenting paper for improving productivity tasks. These systems maintain the AR content in a separate database (e.g., cloud client), and are configured for a specific use case, which hinders portability. On the other hand, *AniCode* [30] uses QR codes to store entire animation sequences contextual to an object. A remaining limitation is that the pre-encoded content is not interactive and the visible QR codes become obtrusive in the user's view as longer sequences are embedded.

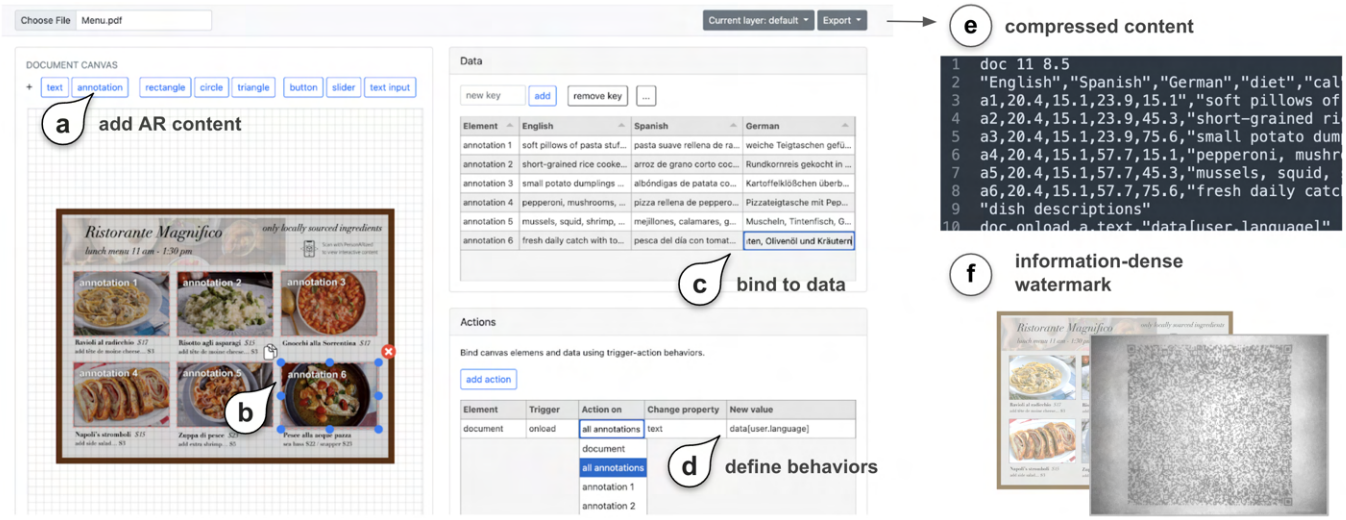
In this work, we investigate unobtrusive IR watermarking as the fabrication method to enable hybrid documents to be portable containers of all information. Our aim is to preserve documents' defining attribute of portability by making the digital content an intrinsic part of the document, rather than separately accessed through external resources.

2.2 Tags for Linking Digital Information to Physical Documents

Want et al. [31] envisioned how tags can be used to unobtrusively connect physical documents with virtual representations and computational functionality. They demonstrated the utility of "invisibly, seamlessly, and portably" linking physical objects to digital functionality and actions that are "naturally associated with their form."

To connect paper documents to digital content, machine-readable printed codes, such as barcodes, have been actively used as a more compact and cheaper alternative to RFID tags [16]. 2D barcodes, such as QR codes, store information in the form of contrasting bits, but affects the original design as it occupies space and impacts the visual look of the final artifact [11], particularly those that store large amounts of data [32]. Thus, researchers explored making invisible or more unobtrusive camera-detectable tags to create a more seamless, discreet user experience [6, 7].

Other works investigated subtle pixel manipulations to encode data *invisibly* [10, 25, 34, 37]. For example, *Chartem* [10] encodes



Element	English	Spanish	German
annotation 1	soft pillows of pasta stuf...	pasta suave rellena de ra...	weiche Teigtaschen gefü...
annotation 2	short-grained rice cooke...	arroz de grano corto coc...	Rundkornreis gekocht in ...
annotation 3	small potato dumplings ...	albóndigas de patata co...	Kartoffelklößchen überb...
annotation 4	pepperoni, mushrooms, ...	pizza rellena de peppero...	Pizzateigtasche mit Pep...
annotation 5	mussels, squid, shrimp, ...	mejillones, calamares, g...	Muscheln, Tintenfisch, G...
annotation 6	fresh daily catch with to...	pesca del día con tomat...	ten, Olivenöl und Kräutern

Bind canvas elemens and data using trigger-action behaviors.

Element	Trigger	Action on	Change property	New value
document	onload	all annotations	text	data[user.language]
		document all annotations annotation 1 annotation 2		

```
doc 11 8.5
"English","Spanish","German","diet","cal
a1,20.4,15.1,23.9,15.1","soft pillows of
a2,20.4,15.1,23.9,45.3,"short-grained ri
a3,20.4,15.1,23.9,75.6,"small potato dum
a4,20.4,15.1,57.7,15.1,"pepperoni, mushr
a5,20.4,15.1,57.7,45.3,"mussels, squid,
a6,20.4,15.1,57.7,75.6,"fresh daily catc
"dish descriptions"
doc.onload.a.text."data[user.language]"
```

Figure 2: Authoring and embedding. (a, b) The user loads a document and adds virtual elements to be overlaid in AR. The user binds the elements to (c) tabular data. On export, all components (e) are compiled into a string and (f) embedded it into a watermark.

subtle pixels into chart background, however, it only supports digital versions of charts, i.e., the data cannot be robustly decoded once they are printed. Methods that support printed documents can only leverage colored pixels and are thus limited in the amount of stored information. For instance, *StegaStamp* [25] encodes 7 characters into an image, and *FontCode* [34] encodes 1.77 bits per character (e.g., a 1,000-character paper abstract would store 221 encoded characters). These methods are unable to store information in non-printed white space, which is a typical part of documents. We investigate a watermarking approach to leverage the whole document area, including text, graphics, and white space, to increase capacity and support the embedding of a wide range of AR assets.

2.3 Infrared-Based Tags

To store invisible tags, infrared (IR) based approaches have emerged as a robust alternative. In addition to patterns projected by external sources [18], researchers have used materials that pass, reflect, or absorb IR light to embed patterns. For instance, *MiniStudio* [17] tracks objects by adding IR-reflective stickers with fiducial markers using screen printing. *InfraredTags* [7] covers fiducial markers with an opaque material that passes IR light to make them imperceptible.

As for IR-absorbing materials, *Spyn* [23] adds such inks into knitted artifacts to embed hidden information. *HideOut* [33] creates hidden tracking markers from IR-absorbing ink to project digital imagery on physical paper, however, a spray gun is needed to evenly coat the paper surface. Wang et al. [29] showed how different IR-absorption characteristics of CMY and black ink can be used to subtly embed fiducial markers, however, it only works for dark areas. Similar to this work, we leverage the convenience and accessibility of commodity inkjet printers, which allow us to print hybrid interfaces rapidly using the encoded data, but we also make use of the white space in the document.

To summarize, *StandARone* has two main benefits that allow it to rapidly create portable and privacy-preserving hybrid document experiences: (1) We utilize the whole document area, including text, graphics, and white space to increase the amount of data that can be unobtrusively stored in the document. (2) We can rapidly create these encoded documents without having to use complicated or manual processes such as spraying or screen printing, thanks to commercially available inkjet printers and inks.

3 STANDARONE WORKFLOW AND USE CASES

In the *StandARone* workflow, authors define the AR interactions and print the document with the AR content (Section 3.1), and consumers decode the content from the document using a mobile phone (Section 3.2). We describe the workflow using an example application, where a designer creates an interactive menu experience for a restaurant's guests.

3.1 Authoring: Embedding AR Content in an Information-Dense Watermark

Adding AR elements: As shown in Figure 2, the designer first imports the desired document, i.e., the paper menu she designed, into *StandARone*'s tagging user interface (UI). The author defines the AR content by placing input and output elements on the canvas (Figure 2a-b). Elements that can be added include plain text, annotations (i.e., callout), shapes such as rectangles and circles, as well as input elements such as buttons and sliders.

Associating elements with data: The added elements are populated in the tabular data panel, where the designer can add associated information. The tabular data has columns linked to designer-defined *keys* (Figure 2c). The keys are used to filter the data and retrieve user- and device-relevant pieces of information. This approach

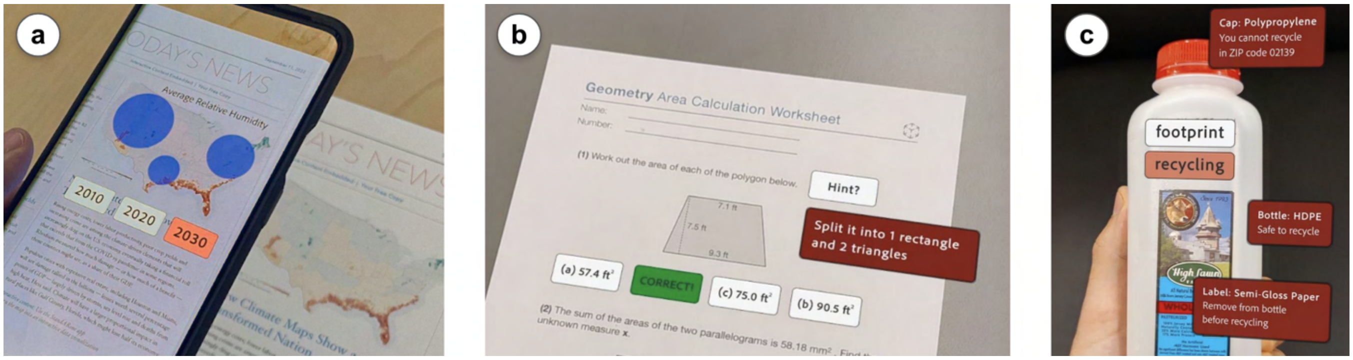


Figure 3: Use cases of StandARone: (a) Interactive data visualization, (b) learner-aware, interactive educational materials, (c) product packaging that adapts to the consumer's location.

avoids the need to connect to an external server and maintains all the data in the document.

In this example, the designer adds detailed descriptions of the dishes in different languages to accommodate local customers and tourists. Based on the language chosen for the operating system of the user's device, the relevant language content will be retrieved in the AR view. Other keys local to the device that the author can choose to add include the user's age, demographics, and other personal or app-specific information.

Defining interaction behaviors: The user can now leverage the added data by defining trigger-action behaviors (i.e., "if this then that" [28]) for the virtual elements. In Figure 2d, the text of each annotation is updated (action) when the document is loaded (trigger). Other triggers that the user could add include when a button is pressed or when a slider's value is changed. The new text will now be dynamically chosen based on the user's preferred language.

Exporting and printing the watermarked document: The user clicks the export button, which automatically compresses the AR content into a watermark and downloads the file (Figure 2e,f). Next, the designer sends the file to the inkjet printer. One of the tanks of the printer already includes the NIR ink, which is used for invisible watermarking.

3.2 Consumption: Decoding the AR Tags and Viewing the Document

Once the document is printed with the watermark, consumers can use the IR camera in their mobile devices to unlock and interact with the embedded AR content. In this example, the user sees directly on the menu that it has AR content embedded (i.e., the label: "Scan with *StandARone* to view interactive content"). Thus the user launches the mobile application, which automatically captures the human-visible document and the invisible AR watermark simultaneously. The app then decodes the watermark and shows the embedded AR content to the user.

The dish descriptions are shown in English since the restaurant guest's phone is set in this language (Figure 1d). The guest can also display the other embedded data using the menu at the top of the screen. As shown in Figure 1e, the "nutrition facts" view further shows a slider, which allows users to filter the meals based on a calorie target.

Overall, this example showcases a simple scenario when users may not have network access or not want to use it, i.e, an international tourist can still view accurate translations even if they do not have a network connection. The fitness and diet information is personal and sensitive, so *StandARone* allows the document to leverage these data without sharing it with third parties. Figure 3 and the supplemental video figure demonstrate other use cases explored with *StandARone*. These include interactive data visualizations, learner-aware interactive educational materials, expandable references in printed publications, and product packaging that adapts to the consumer's location.

4 TECHNICAL IMPLEMENTATION AND WATERMARKING EVALUATION

We describe the technical implementation of *StandARone*, including its authoring interface, NIR desktop printing, and watermark detection pipeline. We finally evaluate the data embedding capacity of the watermarking method.

4.1 User Interface for Adding AR Elements

The Web-based interface is implemented using the *Fabric.js*[1] canvas library and the *Tabulator*[2] interactive table library. Once the author has added the AR content, the UI automatically compiles the position, scale, and orientation of all the virtual elements, the key-based database, and the trigger-action list into a single tabular data source, which is compressed as a string (Figure 2e). The string is then converted into a machine-readable watermark (Figure 2f).

4.2 NIR Watermarked Document Printing

To print *StandARone* documents, we use an off-the-shelf inkjet printer (*Epson ST-200*). The cyan, magenta, yellow (CMY) channels of the printer were filled with the appropriate color inks, and the K (black) channel was filled with an 850nm NIR ink. This ink passes light at visible wavelengths, and thus appears mostly transparent to

---

[1] http://fabricjs.com/
[2] http://tabulator.info/

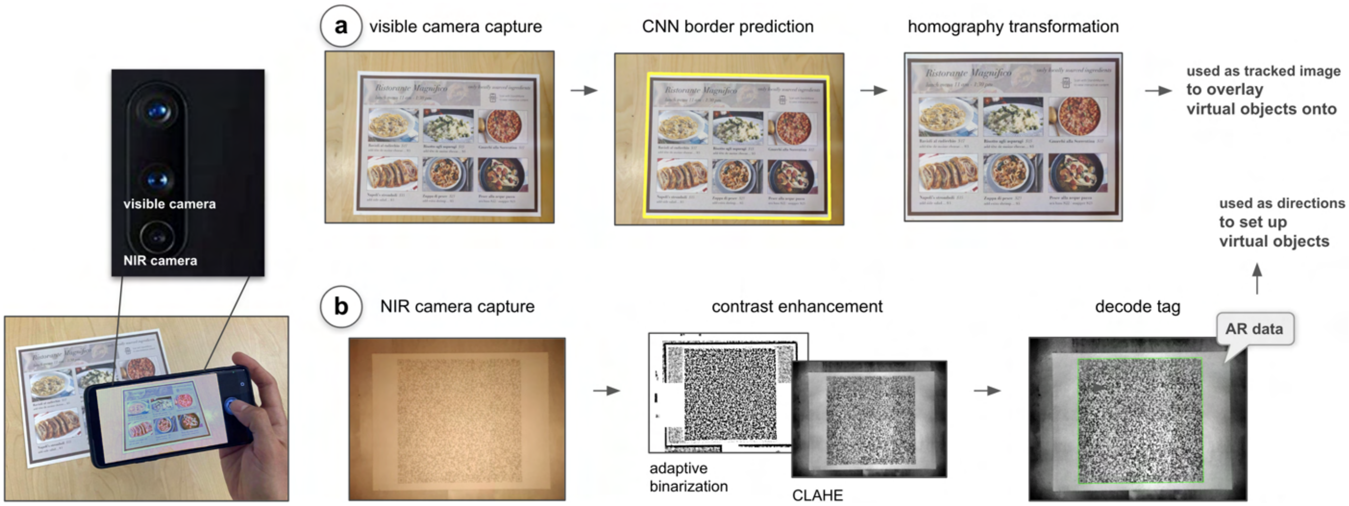


Figure 4: Decoding process. We use a mobile phone with an embedded NIR camera. The phone captures both the (a) visible and (b) invisible content and processes them for tracking the document and setting up the AR scene using the embedded data.

the naked eye, but has high absorbance intensity in the NIR range, with the peak absorbance wavelength at 850nm. Thus, it is captured as black by NIR cameras.

Since the ink loaded into the "black" channel is non-black (for human sight), we instead generate the black visible content by mixing the C, M, Y inks, as their combination results in black color. We do this by changing the color space of the visual part of the document to CMYK and adjusting the four channels such that the digital K channel is blank. The visible color content of the document is printed initially, then the sheet is fed back to overlay the invisible content (watermark) with the NIR ink in the printer's K channel.

4.3 Tag Detection with NIR Imaging

Figure 4 shows the capturing and decoding process. The visible camera capture is mainly used for tracking the document in AR. For this, we detect the document borders using a machine learning model based on *PageNet* [26], which identifies the main document region to segment content from both textual and non-textual border noise using a convolutional neural network (CNN). The identified borders are used to rectify the document via a homography transformation. The rectified image is used by *Unity*'s *AR Foundation* to track the physical document and overlay digital content on it.

The NIR camera capture is enhanced using optical and digital methods. We placed a small IR light LED and IR filter on the back of the phone to reduce noise [7]. The captured image is digitally enhanced in two ways: (1) We apply a CLAHE [35] filter to the image, which generates a grayscale output with increased contrast and sharpness. (2) We separately apply an adaptive threshold to the original image, and then dilate and erode it based on the image gradient, which generates a binary output. Both the grayscale and binary outputs are then used as inputs to the code reader library (*Dynamsoft SDK*[3]). We plan to use machine learning methods to increase the robustness of this detection in the future [5, 8]. The detection result is used to recover the AR string. It is then parsed by *Unity* to automatically add the author-specified digital content to the AR view based on the user's profile. The generated content is filtered and displayed based on the user's interactions.

4.4 Evaluation of the Watermarking Method

In this section, we conduct a preliminary evaluation to assess the breadth of data encoding capabilities based on NIR watermarking. Watermark detection is impacted by the document's distance from the camera and light source (i.e., NIR LED). If far away, the illuminance decreases, reducing the contrast and thus the detectability of the NIR-printed codes.

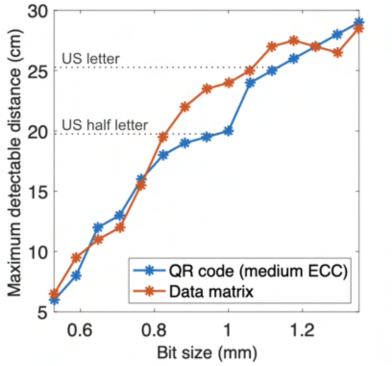
To evaluate how the document distance impacts the smallest detectable bit (i.e., code module), we printed the same code in 20 different sizes, corresponding to a bit size range of 0.4-1.9 mm. The graph in Figure 5 shows the maximum distance the document can be at for the bits of different sizes to be detected. The horizontal lines mark the distances needed for the camera to fully capture a US letter-size colored sheet (8.5"x11") and a US half letter-size colored sheet (5.5"x8.5"), i.e., 25.5cm and 19.5cm.

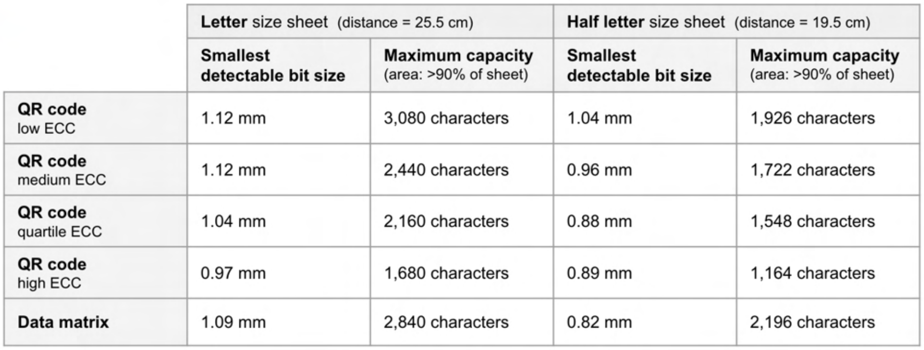
For these two sheet sizes, we estimated the maximum amount of data that can be stored based on the smallest detectable bit size when the full sheet is imaged under high IR illumination. This bit size depends on the code type, e.g., QR codes vs. Data Matrix codes, and QR codes with different levels of error correction codes (ECC). Thus, we repeated this evaluation for different code types.

Figure 5 shows the smallest detectable bit size and the estimated bound for the data capacity for each code type. The results indicate that higher levels of ECC can help detect smaller bits in QR codes. However, since the ECC also occupies space, it ends up reducing the amount of data encodable by the user. The maximum capacity values were calculated based on at least 90% coverage of the whole sheet using IR-printed codes. For letter size, we used 20 codes (5x4, each 5.1cm x 5.1cm), resulting in 90.7% coverage. For half letter, we used 6 codes (3x2, each 6.6cm x 6.6cm), resulting in 93.2% coverage.

---

[3] https://www.dynamsoft.com/





	Letter size sheet (distance = 25.5 cm)		**Half letter** size sheet (distance = 19.5 cm)	
	Smallest detectable bit size	**Maximum capacity** (area: >90% of sheet)	**Smallest detectable bit size**	**Maximum capacity** (area: >90% of sheet)
QR code low ECC	1.12 mm	3,080 characters	1.04 mm	1,926 characters
QR code medium ECC	1.12 mm	2,440 characters	0.96 mm	1,722 characters
QR code quartile ECC	1.04 mm	2,160 characters	0.88 mm	1,548 characters
QR code high ECC	0.97 mm	1,680 characters	0.89 mm	1,164 characters
Data matrix	1.09 mm	2,840 characters	0.82 mm	2,196 characters

Figure 5: Evaluation of detection distance. Left: The further the document, the larger the bits must be to be detected. Right: Based on the smallest identifiable bit for a given distance, we can embed various amounts of data.

Using these layouts, we can embed up to 3,080 characters on a letter sheet (20 *Version 7* QR codes with low ECC) and 2,196 characters on a half letter sheet (6 72x72 Data Matrix codes). This capacity would be sufficient to embed a wide range of interactions into a single sheet, for instance, for the menu application in Section 3, ~1,800 characters were needed. As future work, methods based on paper folding [36, 38] could be used to stack multiple layers of data together. We note that the capacity estimates represent a conservative bound as we use a binary encoding, which allows users to embed an extensive range of characters (*ISO/IEC 8859-1*). The capacity could be increased for applications where only numeric or alphanumeric data needs to be embedded [32].

5 DISCUSSION

In this section, we discuss the limitations of using *StandARone* from the perspectives of authoring and consumption, and how these can be addressed in the future.

Authoring multimodal AR content: *StandARone*'s authoring interface is currently limited to 2D visual assets, e.g., 2D shapes and input elements. In the future, we plan to enable authors to add multimodal content as well, such as audio (text-to-speech, which could also be used for accessibility such as embedding alt text for images), animations, and 3D virtual shapes around the document. Supporting multimedia assets will be important to expand to more use cases explored in prior work [1, 22]. Additionally, including 3D assets can allow users to leverage the document as a spatial anchor in 3D space. For instance, a product manual sheet could be placed next to a physical object to display instructions around it with 3D arrows.

NIR imaging and AR form factor: In our prototype, we used NIR-based watermarking and detection tools. NIR cameras are increasingly available in common handheld devices (e.g., *iPhones* use it for facial recognition), but not all platforms provide 3rd-party developers access to the raw NIR stream (e.g., processed depth maps can be accessed on *iOS*). We argue that the interest in NIR applications will increase as their use cases are demonstrated by similar projects.

In our current implementation, *StandARone* documents are marked with a small icon or label on the document (e.g., Section 3.2) to allow user discoverability of the embedded AR content. We envision that next-generation AR hardware form factors can more fully leverage *StandARone*'s utility. Compared to using a handheld device, *always-on* AR glasses could be constantly scanning the environment for hidden AR content. We next plan to implement this in current AR headsets such as *Microsoft HoloLens* which comes with an integrated but low-resolution NIR camera.

6 CONCLUSION

We presented *StandARone*, which are real-life documents that fully contain the AR content in the paper medium itself. *StandARone* stores the content by using an information-dense watermarking method based on desktop inkjet printing and IR inks. We demonstrated the *StandARone* workflow, which consists of authoring the AR content using a web-based tool that generates the watermark, and consuming the printed hybrid document on mobile devices, as well as several use cases. We described the implementation of the watermark generation, the inkjet printing process of IR inks, and its detection based on mobile IR imaging. We evaluated the printed watermark's storage capacity. We discussed how *StandARone* documents could include multimodal content in the future and viewed using immersive, headset-based form factors. We envision *StandARone* as a step towards a future where objects inherently communicate how they should be viewed and interacted with in AR [4].

REFERENCES

[1] Andrea Alessandrini, Alessandro Cappelletti, and Massimo Zancanaro. 2014. Audio-augmented paper for therapy and educational intervention for children with autistic spectrum disorder. *International Journal of Human-Computer Studies* 72, 4 (2014), 422–430. Publisher: Elsevier.

[2] Narges Ashtari, Andrea Bunt, Joanna McGrenere, Michael Nebeling, and Parmit K. Chilana. 2020. Creating Augmented and Virtual Reality Applications: Current Practices, Challenges, and Opportunities. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376722

[3] Zhutian Chen, Wai Tong, Qianwen Wang, Benjamin Bach, and Huamin Qu. 2020. Augmenting Static Visualizations with PapARVis Designer. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–12. https://doi.org/10.1145/3313831.3376436

[4] Mustafa Doga Dogan, Patrick Baudisch, Hrvoje Benko, Michael Nebeling, Huaishu Peng, Valkyrie Savage, and Stefanie Mueller. 2022. Fabricate It or Render It? Digital Fabrication vs. Virtual Reality for Creating Objects Instantly. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 5. https://doi.org/10.1145/3491101.3516510

[5] Mustafa Doga Dogan, Steven Vidal Acevedo Colon, Varnika Sinha, Kaan Akşit, and Stefanie Mueller. 2021. SensiCut: Material-Aware Laser Cutting Using Speckle Sensing and Deep Learning. In *Proceedings of the 34th Annual ACM Symposium on User Interface Software and Technology*. ACM, Virtual Event USA, 15. https://doi.org/10.1145/3472749.3474733

[6] Mustafa Doga Dogan, Faraz Faruqi, Andrew Day Churchill, Kenneth Friedman, Leon Cheng, Sriram Subramanian, and Stefanie Mueller. 2020. G-ID: Identifying 3D Prints Using Slicing Parameters. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376202

[7] Mustafa Doga Dogan, Ahmad Taka, Michael Lu, Yunyi Zhu, Akshat Kumar, Aakar Gupta, and Stefanie Mueller. 2022. InfraredTags: Embedding Invisible AR Markers and Barcodes Using Low-Cost, Infrared-Based 3D Printing and Imaging Tools. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New Orleans LA USA, 9. https://doi.org/10.1145/3491102.3501951

[8] Mustafa Doga Dogan, Veerapatr Yotamornsunthorn, Ahmad Taka, Yunyi Zhu, Aakar Gupta, and Stefanie Mueller. 2022. Demonstrating InfraredTags: Decoding Invisible 3D Printed Tags with Convolutional Neural Networks. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 7. https://doi.org/10.1145/3491101.3519905

[9] Betsy Fanning. 2017. *Preservation with PDF/A*. Technical Report. Digital Preservation Coalition. https://doi.org/10.7207/twr17-01 Edition: Second.

[10] Jiayun Fu, Bin Zhu, Weiwei Cui, Song Ge, Yun Wang, Haidong Zhang, He Huang, Yuanyuan Tang, Dongmei Zhang, and Xiaojing Ma. 2021. Chartem: Reviving Chart Images with Data Embedding. *IEEE Transactions on Visualization and Computer Graphics* 27, 02 (Feb. 2021), 337–346. https://doi.org/10.1109/TVCG.2020.3030351 Publisher: IEEE Computer Society.

[11] Christopher Getschmann and Florian Echtler. 2021. Seedmarkers: Embeddable Markers for Physical Objects. In *Proceedings of the Fifteenth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '21)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3430524.3440645

[12] Philippe Giraudeau, Alexis Olry, Joan Sol Roo, Stephanie Fleck, David Bertolo, Robin Vivian, and Martin Hachet. 2019. CARDS: a mixed-reality system for collaborative learning at school. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*. 55–64.

[13] Lisa Gitelman. 2014. *Paper knowledge: Toward a media history of documents*. Duke University Press.

[14] Feng Han, Yifei Cheng, Megan Strachan, and Xiaojuan Ma. 2021. Hybrid Paper-Digital Interfaces: A Systematic Literature Review. In *Designing Interactive Systems Conference 2021 (DIS '21)*. Association for Computing Machinery, New York, NY, USA, 1087–1100. https://doi.org/10.1145/3461778.3462059

[15] David Holman, Roel Vertegaal, Mark Altosaar, Nikolaus Troje, and Derek Johns. 2005. Paper windows: interaction techniques for digital paper. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 591–599.

[16] Lars Erik Holmquist. 2006. Tagging the world. *Interactions* 13, 4 (July 2006), 51. https://doi.org/10.1145/1142169.1142201

[17] Han-Jong Kim, Ju-Whan Kim, and Tek-Jin Nam. 2016. miniStudio: Designers' Tool for Prototyping Ubicomp Space with Interactive Miniature. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, San Jose California USA, 213–224. https://doi.org/10.1145/2858036.2858180

[18] Johnny Lee, Scott Hudson, and Pau Dietz. 2007. Hybrid infrared and visible light projection for location tracking. In *Proceedings of the 20th annual ACM symposium on User interface software and technology (UIST '07)*. Association for Computing Machinery, New York, NY, USA, 57–60. https://doi.org/10.1145/1294211.1294222

[19] Zhen Li, Michelle Annett, Ken Hinckley, Karan Singh, and Daniel Wigdor. 2019. HoloDoc: Enabling Mixed Reality Workspaces that Harness Physical and Digital Content. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, 1–14. https://doi.org/10.1145/3290605.3300917

[20] Wendy E Mackay, Guillaume Pothier, Catherine Letondal, Kaare Bøegh, and Hans Erik Sørensen. 2002. The missing link: augmenting biology laboratory notebooks. In *Proceedings of the 15th annual ACM symposium on User interface software and technology*. 41–50.

[21] Jing Qian, Qi Sun, Curtis Wigington, Han L. Han, Tong Sun, Jennifer Healey, James Tompkin, and Jeff Huang. 2022. Dually Noted: Layout-Aware Annotations with Smartphone Augmented Reality. In *CHI Conference on Human Factors in Computing Systems*. ACM, New Orleans LA USA, 1–15. https://doi.org/10.1145/3491102.3502026

[22] Shwetha Rajaram and Michael Nebeling. 2022. Paper Trail: An Immersive Authoring System for Augmented Reality Instructional Experiences. In *CHI Conference on Human Factors in Computing Systems*. ACM, New Orleans LA USA, 1–16. https://doi.org/10.1145/3491102.3517486

[23] Daniela K. Rosner and Kimiko Ryokai. 2010. Spyn: augmenting the creative and communicative potential of craft. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. Association for Computing Machinery, New York, NY, USA, 2407–2416. https://doi.org/10.1145/1753326.1753691

[24] Hyunyoung Song, Tovi Grossman, George Fitzmaurice, François Guimbretiere, Azam Khan, Ramtin Attar, and Gordon Kurtenbach. 2009. PenLight: combining a mobile projector and a digital pen for dynamic visual overlay. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 143–152.

[25] Matthew Tancik, Ben Mildenhall, and Ren Ng. 2020. StegaStamp: Invisible Hyperlinks in Physical Photographs. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Seattle, WA, USA, 2114–2123. https://doi.org/10.1109/CVPR42600.2020.00219

[26] Chris Tensmeyer, Brian Davis, Curtis Wigington, Iain Lee, and Bill Barrett. 2017. *PageNet: Page Boundary Extraction in Historical Handwritten Documents*. Technical Report arXiv:1709.01618. arXiv. https://doi.org/10.48550/arXiv.1709.01618 arXiv:1709.01618 [cs] type: article.

[27] Wai Tong, Zhutian Chen, Meng Xia, Leo Yu-Ho Lo, Linping Yuan, Benjamin Bach, and Huamin Qu. 2023. Exploring Interactions with Printed Data Visualizations in Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics* 29, 1 (Jan. 2023), 418–428. https://doi.org/10.1109/TVCG.2022.3209386 Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[28] Blase Ur, Melwyn Pak Yong Ho, Stephen Brawner, Jiyun Lee, Sarah Mennicken, Noah Picard, Diane Schulze, and Michael L. Littman. 2016. Trigger-Action Programming in the Wild: An Analysis of 200,000 IFTTT Recipes. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 3227–3231. https://doi.org/10.1145/2858036.2858556

[29] Hsi-Chun Wang, Wen-Hsin Liu, Chia-Long Chang, and Yi-Hui Chen. 2008. Design of Halftone-Based AR Markers under Infrared Detection. In *2008 International Conference on Computer Science and Software Engineering*, Vol. 6. 97–100. https://doi.org/10.1109/CSSE.2008.1391

[30] Zeyu Wang, Shiyu Qiu, Qingyang Chen, Natallia Trayan, Alexander Ringlein, Julie Dorsey, and Holly Rushmeier. 2019. AniCode: authoring coded artifacts for network-free personalized animations. *The Visual Computer* 35, 6-8 (June 2019), 885–897. https://doi.org/10.1007/s00371-019-01681-y

[31] Roy Want, Kenneth P. Fishkin, Anuj Gujar, and Beverly L. Harrison. 1999. Bridging physical and virtual worlds with electronic tags. In *Proceedings of the SIGCHI conference on Human factors in computing systems the CHI is the limit - CHI '99*. ACM Press, Pittsburgh, Pennsylvania, United States, 370–377. https://doi.org/10.1145/302979.303111

[32] Denso Wave. 2022. Information capacity and versions of QR Code. https://www.qrcode.com/en/about/version.html

[33] Karl D. D. Willis, Takaaki Shiratori, and Moshe Mahler. 2013. HideOut: mobile projector interaction with tangible objects and surfaces. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction (TEI '13)*. Association for Computing Machinery, New York, NY, USA, 331–338. https://doi.org/10.1145/2460625.2460682

[34] Chang Xiao, Cheng Zhang, and Changxi Zheng. 2018. FontCode: Embedding Information in Text Documents Using Glyph Perturbation. *ACM Transactions on Graphics* 37, 2 (April 2018), 1–16. https://doi.org/10.1145/3152823

[35] Garima Yadav, Saurabh Maheshwari, and Anjali Agarwal. 2014. Contrast limited adaptive histogram equalization based enhancement for real time video system. In *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. 2392–2397. https://doi.org/10.1109/ICACCI.2014.6968381

[36] Junichi Yamaoka, Mustafa Doga Dogan, Katarina Bulovic, Kazuya Saito, Yoshihiro Kawahara, Yasuaki Kakehi, and Stefanie Mueller. 2019. FoldTronics: Creating 3D Objects with Integrated Electronics Using Foldable Honeycomb Structures. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300858

[37] Peiying Zhang, Chenhui Li, and Changbo Wang. 2021. VisCode: Embedding Information in Visualization Images using Encoder-Decoder Network. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (Feb. 2021), 326–336. https://doi.org/10.1109/TVCG.2020.3030343 Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[38] Clement Zheng, Peter Gyory, and Ellen Yi-Luen Do. 2020. Tangible Interfaces with Printed Paper Markers. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*. ACM, Eindhoven Netherlands, 909–923. https://doi.org/10.1145/3357236.3395578