The Visual Computer (2019) 35:885–897
https://doi.org/10.1007/s00371-019-01681-y

ORIGINAL ARTICLE

AniCode: authoring coded artifacts for network-free personalized animations

Zeyu Wang[1] · Shiyu Qiu[1] · Qingyang Chen[1] · Natallia Trayan[1] · Alexander Ringlein[1] · Julie Dorsey[1] · Holly Rushmeier[1]

Published online: 9 May 2019

Abstract

Time-based media are used in applications ranging from demonstrating the operation of home appliances to explaining new scientific discoveries. However, creating effective time-based media is challenging. We introduce a new framework for *authoring* and *consuming* time-based media. An author encodes an animation in a printed code and affixes the code to an object. A consumer captures an image of the object through a mobile application, and the image together with the code is used to generate a video on their local device. Our system is designed to be low cost and easy to use. By not requiring an Internet connection to deliver the animation, the framework enhances privacy of the communication. By requiring the user to have a direct line-of-sight view of the object, the framework provides personalized animations that only decode in the intended context. Animation schemes in the system include 2D and 3D geometric transformations, color transformation, and annotation. We demonstrate the new framework with sample applications from a wide range of domains. We evaluate the ease of use and effectiveness of our system with a user study.

Keywords Authoring time-based media · Encoding animations · Personalized demonstrations · Network-free communication

1 Introduction

Time-based media, such as videos, synthetic animations, and virtual reality (VR) experiences, are replacing text and static images in communication. The popularity of time-based media for communication is evident from the ubiquity of "how to" videos and virtual tours presented as panoramic videos. Advances in software and hardware have made it possible to render high-quality animations and capture high-resolution digital videos. However, current methods of producing and consuming time-based media have significant shortcomings. Production of useful expository content requires time and expertise. The same content is created for all consumers and lacks personalization. Viewers require network access to receive the content. There are no alternatives for delivering videos to a viewer with little or no network access. Perhaps, more importantly, in an age where personal data are easily collected and tracked, there is a need for applications that protect privacy by being network free.

We propose a new alternative to traditional video generation and sharing. We distinguish between *authors* who have content to be communicated and *consumers* who view and use the content (Fig. 1). Our system enables authors to *create*

Zeyu Wang and Shiyu Qiu have contributed equally to this work.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s00371-019-01681-y) contains supplementary material, which is available to authorized users.

✉ Shiyu Qiu
sherry.qiu@yale.edu

Zeyu Wang
zeyu.wang@yale.edu

Qingyang Chen
qingyang.chen@yale.edu

Natallia Trayan
natallia.trayan@yale.edu

Alexander Ringlein
alexander.ringlein@yale.edu

Julie Dorsey
julie.dorsey@yale.edu

Holly Rushmeier
holly.rushmeier@yale.edu

[1] Department of Computer Science, Yale University, 51 Prospect St, New Haven, CT 06511, USA

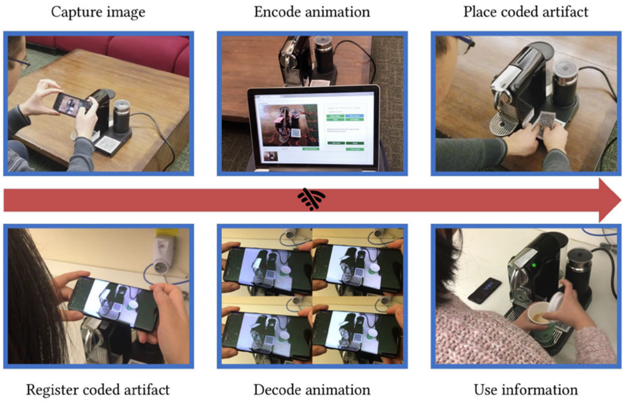


Fig. 1 Authoring and consuming network-free personalized animations. Top: the author takes a picture, specifies parameters and regions to be animated, and encodes the information in a code following our animation schemes. Bottom: with the authored code, our mobile application on the consumer side registers the view and decodes a personalized animation. The consumer obtains visual information, such as how to use a coffeemaker, without having to access the Internet

coded artifacts that allow consumers to *generate* time-based displays on their own device.

An example use case is a demonstration of operating a complicated piece of equipment. An animation is far more effective than a manual with text and still images. A consumer can generate a video of the operation as it appears at the consumer's own location. The consumer uses a smartphone camera to capture a code affixed to the equipment. The code is used to apply authored image transformations to generate the animation showing how to operate the equipment. Since the imagery is in the consumer's visual context, the motion of objects is clear to the consumer. Communication is made private and robust to network instability by generating the animation on the consumer's local device without Internet access. Since the consumer needs line-of-sight access to the equipment and affixed code, the animation can only be generated when the consumer is in the intended context. This provides protection for any sensitive information about the equipment that should not be shared broadly. The system leverages the image captured by the consumer, so that a limited amount of information needs to be encoded to produce an animation. Typically, a video 20 MB in size can be generated from 200 bytes of coded information—effectively a 100,000 compression ratio.

Our work makes the following contributions:

- We propose an alternative framework for time-based communication based on authoring a coded artifact that is imaged to create an animation on demand.
- We show that the new framework offers enhanced (but not absolute) robustness and privacy by not relying on the presence of an Internet connection and requiring consumers to be on-site and in the correct context to be able to view the animation.
- We present an authoring interface that enables non-experts to author animations and create augmented reality (AR) content. We conduct a user study to evaluate the usability of our interface.
- We present a mobile application that allows the consumer to image a coded artifact and reconstruct a personalized animation on the local device.
- We show how this framework improves the documentation of products and educational sequences, and offers a new methodology for presentation of scientific results to the general public with sample applications drawn from several fields.

2 Related work

Our work builds on a long history of generating animations in computer graphics [28], authoring multimedia [16], and expository media creation [2,3]. We rely on recent work on communication using time-based media, authoring systems, AR-assisted mobile communication, and network-free communication.

2.1 Time-based media

Time-based media can be more effective than text-based media in education [30], entertainment [20], and advertising [5]. Animations make good use of a viewer's cognitive capacity [33] and promote conceptual understanding [8]. Producers [15] and analysts of social media [19] describe video as powerful because it can incorporate other media and present rich content. Video can motivate action, enhance communication, and reach a wide audience. Digital technologies for creating time-based media are now broadly accessible and frequently applied to tell stories. In cultural heritage, for example, it is well recognized that any data need to be presented with context [25]. Storytelling from data is an important motivation for the system described in this paper, but our work is not on the humanist or psychological principles of what makes a story effective, but rather on creating a new framework for authors to apply them.

2.2 Authoring systems

Authoring for multimedia systems has progressed since the seminal COMET system [16]. One recent line of research investigates the possibilities for improving the authoring and presentation of video tutorials. There have been systems that include the ability to add bookmarks to a captured video and then add multimedia annotations to the video [10]. Other work also considers the process of effective video capture.

For example, an improved first-person viewpoint is developed using Google glass for capture [11], which emphasizes the importance of annotation systems, linked diverse data types, and reconsidering content capture.

Animation generation has long been a feature of information and scientific visualization systems [32]. Recent work considers the authoring of 360-degree videos with branching [14,23], which emphasizes both the design of the authoring and navigation interfaces. Other work focuses on authoring and visualization tools for end users in mobile and cross-device applications [7,17], which keeps more people in the loop of content production and takes advantage of multimodal interaction to assist visual sensemaking.

2.3 Augmented reality

The popularization of VR and AR provides a new approach to communication. Content creation is a key area in AR development. Some applications focus on superimposing virtual objects on a real environment based on coplanarity in single images [13]. Similar work includes manipulating existing objects interactively with partial scene reconstructions based on cuboid proxies [38]. Authoring time-based media for AR has also become possible by mixing raw video footage using sparse structure points [12]. Recent AR devices such as Microsoft HoloLens provide the possibility of interactive scene editing in the real world based on data from the sensor [37]. Some industrial products can create AR tutorials from normal videos, offering users a personalized experience [29]. In particular, smartphones with high-resolution cameras provide extraordinary resources for acquiring and processing visual information, making mobile VR and AR possible. For example, smartphones are used to receive input and create augmented environments in sites that require interactive expository resources [4].

These systems have interesting insights into content creation for AR. However, they require the author to explicitly transfer a large amount of data to the consumer via Internet communication. In this paper, we develop methods for users to access personalized animations without an Internet connection, which makes the communication more robust to network instability and helps preserve the user's privacy. Compared with prior AR systems, our new system augments user-captured images to produce animations locally and offers a valuable tool to provide information and engage users without relying on a communication network. Our mobile application is not confined to a specific scene.

2.4 Network-free communication

Data privacy has become an important focus in systems and applications. When a smartphone is connected to a public network, data privacy is at risk. For instance, users of the fitness tracking app Strava unwittingly gave away locations of secret US army bases [18]. We explore visual communication without Internet connection. Our work can be considered a variation of "Visual MIMO" [6]. Optical communication systems are designed using active light-emitting systems and cameras. Methods for manipulating light-emitting systems and then using computer vision techniques on the receiver end have been explored extensively [36]. Additional techniques such as AirCode [21] and FontCode [34] embed information in a network-free environment. AirCode exploits a group of air pockets during fabrication and the scattering light transport under the surface. FontCode alters the glyphs of each character continuously on a font manifold and uses font parameters to embed information. AirCode and FontCode have not been used for the direct communication of visual or time-based information.

Our system uses a printed code affixed to or embedded in an object and depends on a photograph of the object itself to drive the generation of a video for communication. Rather than using coded artifacts to redirect consumers to Web sites, we use the coded information to segment and animate a consumer-captured image to present visual information. After a one-time installation of an application on their mobile device, the consumer never has to access the Internet to observe information provided by the author. This improves robustness of the communication, because the consumer does not need network access that may be difficult in remote, congested, or restricted areas, and the author does not have to ensure that a server providing videos is always available. This improves privacy (although does not guarantee it), because neither the consumer's request nor the information they receive is transferred over the Internet. It also requires people to be on-site to be able to view the animation, restricting those away from the scene from obtaining the information.

3 Framework overview

The key pieces of our new framework are (A for *author* and C for *consumer*):

A1 The author takes an image of a static scene containing objects and a reference QR code as landmarks.
A2 After image segmentation, the author selects segments and specifies animation information in the authoring interface.
A3 The author generates a new QR code based on the animation information and reference landmarks and then prints it to replace the reference QR code.
C1 The consumer opens the mobile app on a smartphone and directs the view to the object with the authored QR

code. The app takes a picture when the registration error is below a set threshold.

C2 The app processes the consumer's image. For each keyframe, it generates a mask of the region to be animated based on segment features matched with those in the author's image.

C3 The app generates the animation locally on the fly using only the consumer's image and decoded information. The consumer views a visually contextualized demonstration privately.

4 Methodology

In this section, we describe the design and implementation of our new system. First, we explore image segmentation algorithms to identify the regions to be animated. Then, we show four animation schemes to convey the author's ideas. Next, we present an interface that assists authors in converting an animation into a small set of encoded numbers. Finally, we describe a network-free mobile application that can guide consumers to the correct view, decode the authored information, and generate a personalized animation locally on the fly.

4.1 Regions to be animated

Authors need to specify a region of interest (ROI) to be animated in each keyframe. With an ROI mask, we can apply image transformations. An ROI could be defined by polygon vertices in image coordinates, but the ROI may be in a somewhat different location in the consumer's image than in the author's. To make the ROI specification more robust, we use image segmentation.

Image segmentation assigns a segment ID ($1 \leq ID \leq K$) to each pixel where K is the number of segments in the image. Among many, image segmentation algorithms are superpixel-based methods. After considering various options such as simple linear iterative clustering (SLIC) [1] and superpixels extracted via energy-driven sampling (SEEDS) [9], we selected linear spectral clustering (LSC) [22] that produces compact and uniform superpixels with low computational cost.

In the LSC algorithm (applied in Fig. 2), a pixel $\boldsymbol{p} = (L, a, b, x, y)$ in the CIELab color space (every component is linearly normalized to $[0, 1]$) is mapped to a ten-dimensional vector in the feature space. Taking as input the desired number of superpixels K, the algorithm uniformly samples K seed pixels and iteratively performs weighted K-means clustering. It achieves local compactness by limiting the search space of each cluster. After the algorithm converges, it enforces the connectivity of superpixels by merging small superpixels into neighboring larger ones.

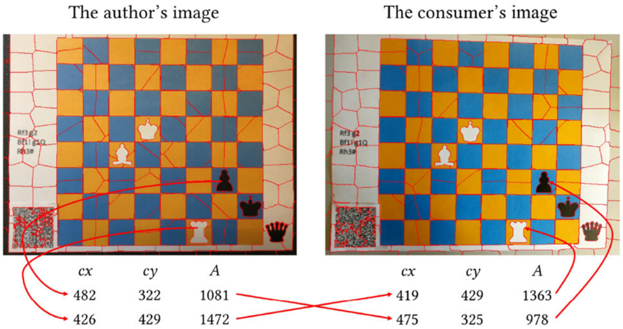


Fig. 2 Image segmentation results on both images using the LSC algorithm. The QR code stores segment features of the author's ROIs. The consumer-side app generates masks of the consumer's ROIs based on feature matching

The author chooses two parameters of the image segmentation algorithm (via sliders and interactive feedback): the average superpixel size and the superpixel compactness factor. Once the algorithm achieves a satisfactory result, it generates a matrix of segment IDs with the same size as the original image. For each keyframe, the author can select one or multiple segments as the ROI. We extract segment features with a view to coding efficiency. We use three numbers to represent a segment, its center coordinates (cx, cy), and its area A. The consumer-side app executes the LSC algorithm using the same parameters on the consumer's image. Before generating the animation, the app generates an ROI mask for each keyframe based on matching the new segmentation result and the author's encoded features in terms of weighted mean squared error. Mathematically, if the author selects n segments for the ROI in the current keyframe, together with $3n$ features cx'_i, cy'_i, A'_i where $1 \leq i \leq n$, the matched ROI in the consumer's image is the union of all the matched segments:

$$
\begin{aligned}
\mathrm{ROI} &= \cup_{i=1}^{n} \mathrm{segment}_{ID_i} \\
ID_i &= \underset{j}{\mathrm{argmin}} \quad (cx_j - cx'_i)^2 \\
&\qquad\qquad + (cy_j - cy'_i)^2 + \frac{1}{1000}(A_j - A'_i)^2.
\end{aligned}
$$

4.2 Animation schemes

To create an animation, the author selects an ROI and applies a transformation in each keyframe. Common operations include 2D and 3D geometric transformations, color transformation, and annotation (Fig. 3). All the animation schemes except annotation can use linear or quadratic interpolation during the animation to generate changes at different temporal rates.

2D Transformation We use a 2D rigid transformation to describe object motion in the image plane. The author can

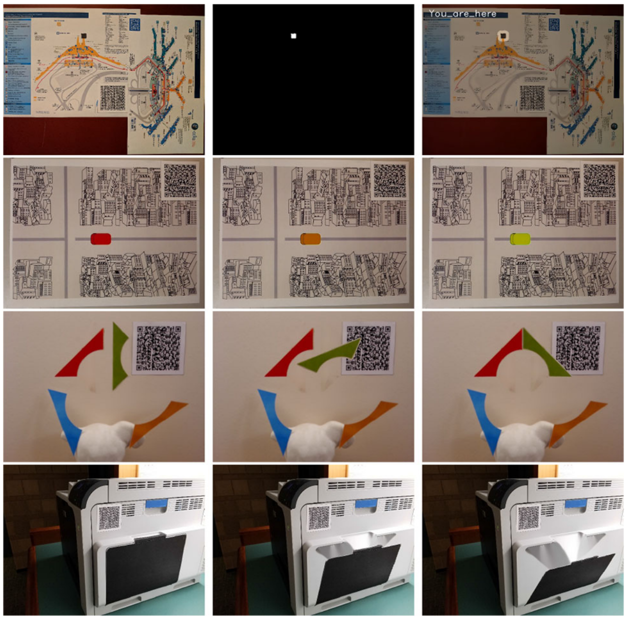


Fig. 3 Animation schemes. From left to right in the first row: original image, ROI mask, and annotation frame. Second row: color transformation frames. Third row: 2D transformation frames. Fourth row: 3D transformation frames

specify translation t_x, t_y as well as rotation θ after selecting an ROI. We assume the rotation is around the ROI center (c_x, c_y) since it is more intuitive to authors. The author needs to specify the duration T of the current keyframe in seconds and interpolation method. Assuming 30 frames per second (fps) in the video, we apply the following geometric transformation to the selected ROI in the ith frame:

$$M = \begin{bmatrix} 1 & 0 & rt_x \\ 0 & 1 & rt_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & c_x \\ 0 & 1 & c_y \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos r\theta & \sin r\theta & 0 \\ -\sin r\theta & \cos r\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -c_x \\ 0 & 1 & -c_y \\ 0 & 0 & 1 \end{bmatrix},$$

where r is the interpolation ratio. $r = \frac{i}{30T}$ for linear interpolation and $r = \frac{i}{30T}^2$ for quadratic interpolation. After the current ROI is moved to a different location, there may be a hole at the original location. We use the Telea algorithm [31] for image inpainting before generating the animated frames. We also dilate the ROI mask to further improve the visual effect.

3D Transformation When authors want to describe object motion in 3D space, a 2D rigid transformation is not sufficient. In our current system, we assume no depth image information is available. Therefore, we use a subspace of 3D transformation, which is 2D perspective transformation on the image plane, because it can describe motion in 3D space. Instead of translation and rotation, the author needs to specify four pairs of corresponding points $\boldsymbol{p}_j = (x_j, y_j)$, $\boldsymbol{p}'_j = (x'_j, y'_j)$ before and after the transformation to obtain the homography. Given the duration T in seconds, 30 fps, in the ith frame, we can solve the transformation matrix M using a linear system solver as follows:

$$w_j \begin{bmatrix} x_j + r(x'_j - x_j) \\ y_j + r(y'_j - y_j) \\ 1 \end{bmatrix} = \begin{bmatrix} m_{xx} & m_{xy} & m_{xw} \\ m_{yx} & m_{yy} & m_{yw} \\ m_{wx} & m_{wy} & 1 \end{bmatrix} \begin{bmatrix} x_j \\ y_j \\ 1 \end{bmatrix},$$

where $j = 1, 2, 3, 4$, r is the interpolation ratio, and w_j is a normalizer for homogenous coordinates.

Color Transformation Color transformation can help emphasize an image region or can produce artistic effects. To preserve textures, we convert the image from RGB space to HSV space and change the hue channel given Δh from the author. In the ith frame, we change the RGB values of pixels within the ROI mask as follows, where we use the same interpolation ratio r defined in previous animation schemes:

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = HSV2RGB \left(RGB2HSV \left(\begin{bmatrix} R \\ G \\ B \end{bmatrix} \right) + \begin{bmatrix} r\Delta h \\ 0 \\ 0 \end{bmatrix} \right).$$

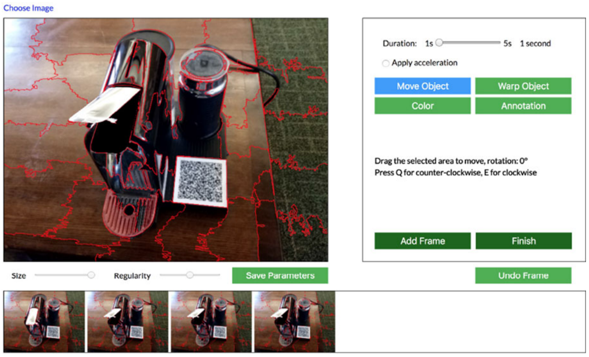
Annotation In expository videos, it is useful to provide textual annotation to explain functionality. A QR code is capable of encoding some short text for an ROI. Given this information, we dilate the ROI mask more and perform bilateral filtering on the background to focus the attention of the consumer [35]:

$$\tilde{I}(\boldsymbol{p}) = 0.7 \frac{\sum_{\boldsymbol{q} \in \Omega} I(\boldsymbol{q}) \cdot f(||I(\boldsymbol{q}) - I(\boldsymbol{p})||) \cdot g(||\boldsymbol{q} - \boldsymbol{p}||)}{\sum_{\boldsymbol{q} \in \Omega} f(||I(\boldsymbol{q}) - I(\boldsymbol{p})||) \cdot g(||\boldsymbol{q} - \boldsymbol{p}||)},$$

where $\boldsymbol{p}$ is a non-ROI pixel in the image, Ω is a window centered in $\boldsymbol{p}$, f is the range kernel for smoothing differences in intensities, and g is the spatial kernel for smoothing differences in coordinates, both with a sigma of 80. Note that we also reduce the intensity of the non-ROI region by 30% so that the ROI stands out. The text is presented at the top left corner of the filtered image. Annotation also works well with color transformation if the author wants to emphasize an ROI.

4.3 Authoring interface

After obtaining the image to animate and its segmentation, the author can add transformations to the scene using the authoring interface. The interface (Fig. 4), which runs on a laptop or desktop machine, requires the author's image and the segmentation result as a matrix of segment IDs and a table of segment features. When an image is loaded, the system



Drag the selected area to move, rotation: 0°
Press Q for counter-clockwise, E for clockwise

Fig. 4 Authoring interface. The author can upload an image and select segments to form an ROI for each keyframe. The author also selects an animation scheme in each keyframe and interactively specifies information to be encoded. The canvas is updated in real time. Our system generates an animation preview and a QR code when authoring is finished

detects the reference QR code and stores the landmark coordinates. The author can tune the two segmentation parameters by adjusting sliders and visualizing the segmentation result.

The author can apply an animation scheme to a selected ROI in the image. Selected segments are highlighted in red. Highlighted segments form an ROI for the current keyframe, for which the duration can be indicated using a slider in the interface. After the author selects an animation scheme from the right panel, corresponding instructions appear. We use more user-friendly terms of “Move Object” and “Warp Object” rather than the more technical term of “2D/3D transformation”. For a 2D transformation, the author can drag and rotate the ROI. For a 3D transformation, the author can drag four green pins to the corners of the ROI and drag four purple pins to new positions in order to warp the ROI. When the purple pins are dragged, the interface computes a homography and warps the ROI in real time. For color transformation, the author can use a slider to adjust hue. The initial value of the slider is the average hue of the current ROI, and color change is displayed in real time. For annotation, the author can simply type in the desired text. After the addition of each keyframe, the interface updates the canvas and matrix of segment IDs. A thumbnail of the current canvas is rendered at the bottom so the author can keep track of all the keyframes. The author can also undo a keyframe if not satisfied with the previous frame.

When authoring is finished, the system generates a preview animation and a new QR code. Due to the limited storage capacity of the QR code, the author is alerted if the amount of information to be encoded has exceeded the maximum allowed. As a trade-off between capacity and size, the version 13 QR code that we use can encode 483 alphanumeric characters (about seven keyframes) with the data correction level M. The generated QR code uses the following encoding:

- 8 numbers, xy-coordinates of four landmarks of the reference QR code.
- 4 numbers, parameters of the LSC algorithm, the first two of which can be tuned in the interface. We reserve all four in case they should be used.
- An integer m, the number of keyframes to be animated. For each of the following m lines
    - An integer n, indicating how many segments there are in the current ROI.
    - For each segment $3n$ integers, the xy-coordinates of the segment center and the segment area.
    - An integer τ to represent the animation type of the current keyframe.
        - If $\tau = 0$ or $\tau = 4$, this type is 2D transformation. Following are 4 numbers, i.e., horizontal and vertical translation, rotation, and duration. Temporal interpolation is linear when $\tau = 0$ or quadratic when $\tau = 4$.
        - If $\tau = 1$ or $\tau = 5$, this type is 3D transformation. Following are 16 numbers, i.e., xy-coordinates of four pairs of corresponding points before and after the perspective transformation. Another number follows for duration. Temporal interpolation is linear when $\tau = 1$ or quadratic when $\tau = 5$.
        - If $\tau = 2$ or $\tau = 6$, this type is color transformation. Following are 2 numbers, i.e., change of the hue channel and duration in seconds. Temporal interpolation is linear when $\tau = 2$ or quadratic when $\tau = 6$.
        - If $\tau = 3$, this type is annotation. Following are a string and a number, i.e., text content of the annotation and duration in seconds.

4.4 Consumer-side application

The consumer-side application (Fig. 5) works on a handheld mobile device. When the consumer holds a smartphone toward a scene with authored objects, the app detects and decodes the QR code using the ZXing QR code framework [27]. The app recognizes the landmark coordinates of the reference QR code in the author’s image. Four static red circles are rendered at these locations on the screen. The app also draws four green dots on the screen in real time at the landmark coordinates of the current QR code based on the detection result. The consumer registers the view to the author’s specification by aligning the dots with the circles.

To generate the animation, the app performs the LSC image segmentation using the same parameters on the consumer’s image downscaled to 640 × 480 to achieve good runtime performance without compromising much visual quality. The app generates an ROI mask for each keyframe by

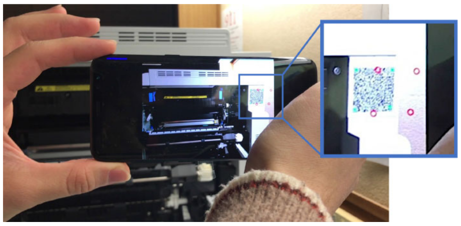


Fig. 5 Consumer-side application. Red circles are where the landmarks should be, and green dots are current detected landmarks. The animation begins once the view is registered

matching segment features with the features stored in the QR code. A transformation stack is maintained for each segment to keep track of its location, since segments can be moved through 2D/3D transformations. After segment matching, the accumulated transformation is applied to each segment to obtain the intended ROI mask. Then, other parameters in this keyframe are used to generate the image sequence using our animation schemes. Our current implementation is an Android app using OpenCV in Java [26].

5 Sample applications

We demonstrate sample applications in the following areas: expository media, cultural heritage and education, creative art and design. We use static figures to summarize the examples; the animations are available in our supplementary video. Our portfolio includes more examples and their detailed descriptions.

5.1 How do I use this?

A common problem is explaining how to use a product. Products are designed, to the extent possible, with well-known visual affordances (e.g., the handle on a cup). Simplified diagrams are sometimes printed on objects or booklets. Online videos illustrate usage. All of these approaches have potential problems; an obvious visual affordance may not be possible to incorporate into the design, people may not be able to understand simplified diagrams, and online videos require network access and may show the product in a different view or lighting which make it difficult for users to relate to their particular situation.

Our system allows the consumer to view a video on usage of a product based on the consumer's current visual context. A specific example is shown in Fig. 6. The consumer needs to understand that they should lift the lid, put in a capsule, and put a cup under the nozzle to make coffee. They also need to know how to add water, clear the trash bin, and make milk foam. The author places a generated QR code on the coffeemaker; then, the consumer views the animation generated on the fly in the mobile app. The author describes the lid being lifted using 3D transformation and the trash bin being pulled out using 2D transformation. Annotations are shown to indicate where to put the cup and make milk foam.

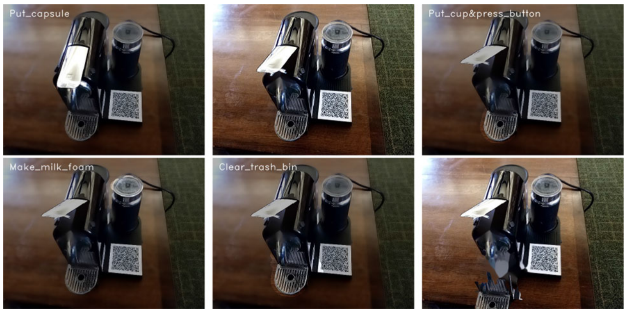


Fig. 6 Coffeemaker. This communicates the operations of lifting the lid, placing the cup, making milk foam, and clearing the trash bin. Animation in supplementary video

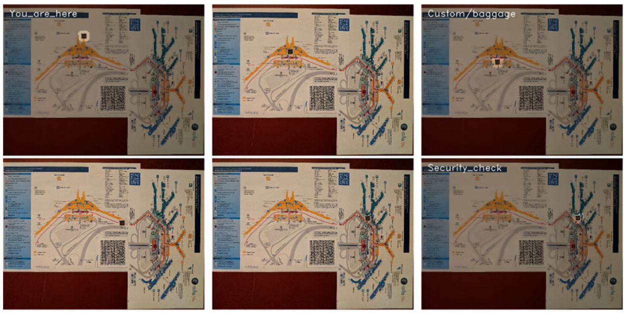


Fig. 7 Navigation in an airport. International arrivals need to pass through customs, board the airport shuttle to change terminals, and go through security again for domestic flights. Animation in supplementary video

One of the strengths of our system, offline data, can be extremely useful in navigation. For example, when travelers arrive at an airport to make a connecting flight, they are often unfamiliar with the complex layout of the airport. They may not have network access or may not want to communicate their current location to others. In the example in Fig. 7, a passenger from an international flight has arrived at Chicago O'Hare International Airport and needs to pass through customs, board the airport shuttle to change terminals, and go through security for domestic flights. Using our system, an animation shows the path to customs, shuttle stops, and security checkpoints. The animation highlights the route that could be difficult to follow on a static map. The passenger obtains this visual information without disclosing their location or other sensitive information.

5.2 Cultural heritage and education

Most cultural heritage artifacts on display at museums and historical sites cannot be touched by visitors; their purpose is usually explained through text. However, text may not effec-

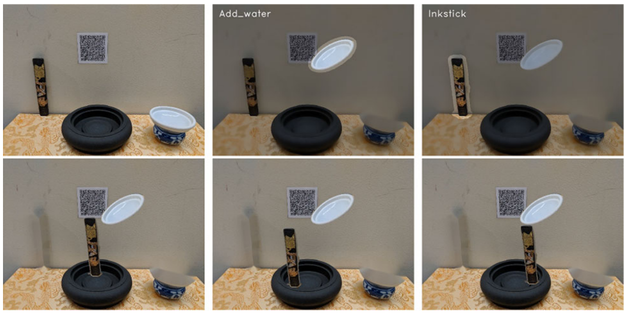


Fig. 8 Ink grinding. First, water is added to the inkstone. Then, the inkstick is highlighted with annotation and moved over the inkstone to show the process of grinding ink. Animation in supplementary video

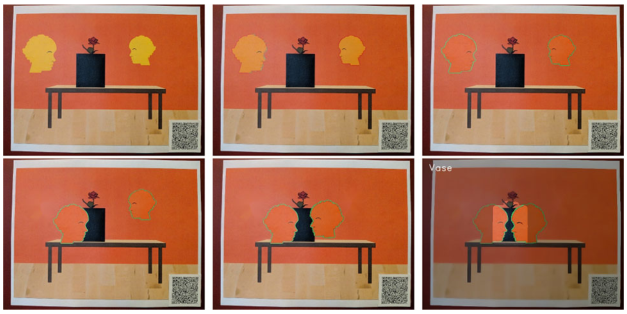


Fig. 9 The Rubin vase. Two yellow heads gradually change color to match the red background, and a vase is formed by arranging the heads and the box in a specific manner. Animation in supplementary video

tively convey foreign concepts to visitors. This is especially true when there are multiple artifacts used in a complex manner. Providing information through kiosks, Wi-Fi, and cloud-based storage can be financially infeasible for small museums and remote sites. In such cases, visual communication is more effective, and viewing the animation on a smartphone without accessing the Internet gives the added bonus of a private and personalized experience.

An example in the heritage domain is the process of grinding ink in Fig. 8, which was part of the standard method of writing in ancient Asia. First, the water container is lifted and moved over the inkstone to show the addition of water. Then, an annotation is displayed identifying the inkstick. Finally, the inkstick is moved to the location of the inkstone and ground over its surface to demonstrate ink production.

5.3 Creative art and design

Artists can design animated posters using our system. For example, the poster in Fig. 9 initially shows two people appreciating a flower in a black box. The artist changes the color of the heads to merge them into the background and warps them in such a way that a vase is carved out from the black box using color and perspective transformations. With an authored QR code, the consumer is able to understand ideas that previously could not be conveyed through a static poster. Educators can also use this type of animation to show the psychological concept of bistable images intuitively.

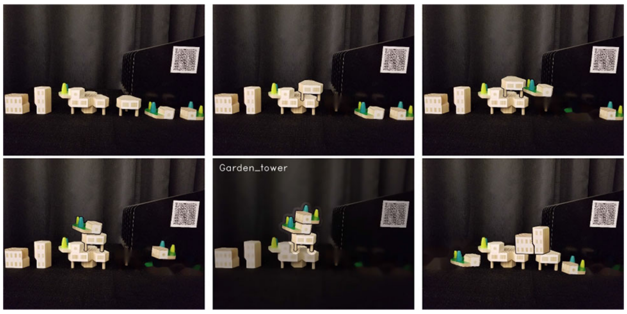


Fig. 10 A Blockitecture® set. The first five pictures show the process of building a garden tower, while the last picture shows the possibility of a different configuration. Animation in supplementary video

Building blocks are another example of design. Many designers and architects use blocks to develop ideas in 3D, such as through the Blockitecture® set in Fig. 10. In Blockitecture, simply shaped blocks can be rearranged in many different configurations. Using our system, the author can use a single scene to explore various configurations without physically rearranging blocks. The author can take one image of the scene and create multiple different animated designs through different transformations on the blocks. This Blockitecture can be expanded to larger scenarios where designers can take an image of a construction in progress and then use our system to view various possibilities for the construction.

6 Evaluation

We evaluated our system with a user study. The goal of the study was to test whether people find our novel system easy to use and whose aspects of the authoring interface are the most helpful in creating an animation. We also sought to explore how well people understand the animation decoded through the mobile app and how robust our system is to changes in environment lighting and camera position. People may be familiar with time-based media, authoring systems, and network-free communication as separate entities, but to our knowledge there is no existing system that embodies all three. Therefore, obtaining user feedback on authoring and consuming personalized time-based media in a network-free environment is important to determine whether such a system is effective for communication.

6.1 User study

In a pilot study, we asked four users (two Ph.D. students in computer science, an illustrator, and a student from a non-computing background) to test the authoring interface by creating a sample animation. They offered feedback on improvements for the interface, such as removing redundant components, adding the undo functionality, and hints about what to do at each step of the authoring process.

For the formal user study, we chose five scenes from the previous section (Blockitecture, coffeemaker, ink grinding, navigation on a map, and the Rubin vase) and an additional simple test scene. These scenes cover various application areas, which would allow for more comprehensive results. We created an authored QR code for each scene, so that all subjects can view the same animations using the mobile app.

We recruited 20 subjects of varying age and background. Each subject was shown a short instructional video demonstrating the authoring and consuming process. Participants were able to familiarize themselves with authoring and consuming animations using the test scene. They first used the mobile app on an Android smartphone (Google Pixel 2 XL, released in October 2017) to decode an animation from the test scene by registering QR code landmarks. They then used the authoring interface on a laptop (MacBook Pro 13-inch, Late 2013) to recreate the animation based on what they understood from the animation decoded on the mobile device. We answered questions from the subjects as they worked through the test scene, to make sure they understood how to correctly use the mobile app and authoring interface.

After understanding the authoring and consuming process through the test scene, participants repeated the same procedure for the five scenes set up for this user study. The procedure, again, involves first viewing an animation using the mobile app and then using the authoring interface to recreate the animation. All participants viewed the same animation for each scene. During these five scenes, participants were not allowed to ask us questions and did not receive input from us. The order of scenes for each participant was different, to account for participants getting better at using the system over time. For each scene, we recorded the time taken for subjects to register the view on the smartphone, as well as the time taken to recreate the animation using the authoring interface. After finishing all the scenes, subjects were asked to answer questions regarding how difficult the registration process was and how well they understood what was being conveyed in the animation (as a *consumer*), and how difficult they found it to create the animations and how satisfied they were with different components in the interface (as an *author*). We also collected background information about each subject's level of expertise in video editing, computer science, and computer graphics.

6.2 Observations

The following key observations are gained through the user study:

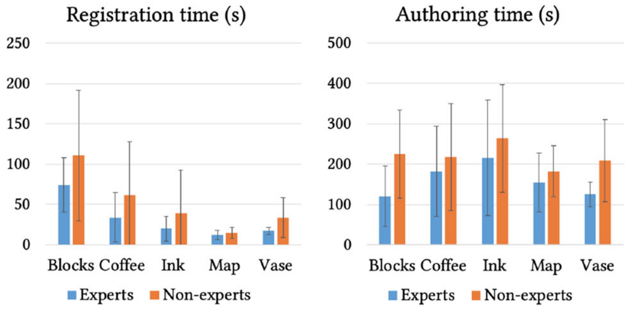


Fig. 11 Left: average time participants took to register a view. Right: average time participants took to create an animation

- Experts and non-experts alike find the authoring interface easy to use, taking on average around 3 min per animation.
- People are most satisfied with the interface components 2D transformation, color transformation, and annotation.
- Registering QR landmarks using the smartphone is an unfamiliar process to many people at first, but they quickly master it and registration time decreases significantly.
- Although registering QR landmarks can require a bit of effort, once correctly registered the decoded animation is easily understood by most viewers.
- Decoded animations are understood by viewers throughout various times of the day with varying lighting conditions and slight variation in camera position.

First, our high-level analysis focuses on scores provided by the subjects. All the scores in the survey are on a Likert scale of 1 to 5. For rating the difficulty of registering QR codes and using the authoring interface, a rating of 5 was most difficult. For understanding decoded animations, 5 was easiest to understand. For evaluating interface components, 5 indicated the highest level of satisfaction. We explicitly explained rating semantics to avoid confusion.

The average time that people took to register a view was 41.79 s. People found it moderately difficult to register QR code landmarks (average rated difficulty 3.45). Most people found it easy to understand what is being conveyed in the decoded animations (average rated level of understanding 4.00). The average time taken by people to create an animation was 189.82 s. Most people found it easy to create an animation using the authoring interface (average rated difficulty 2.19). In general, people were satisfied with the way they interacted with different interface components (segmentation tuning 3.35, 2D transformation 4.55, 3D transformation 3.55, color transformation 4.45, annotation 4.65, animation preview 4.35, undo 4.00).

We also wanted to explore difference in behavior between expert and non-expert users (Fig. 11). Based on the level of expertise in video editing, computer science, and computer graphics on a Likert scale of 1 to 5 with 5 being most experienced, we consider a participant an expert user if the three numbers add up to 8 or more, or a non-expert user otherwise. We had 10 expert users and 10 non-expert users. Expert users took 20.51 s less on average to register a view, because they might have had a better idea on how to move and tilt the smartphone in order to better register the QR code landmarks. The non-expert group had greater standard deviation, which suggests that their understanding of how to register a view varied among individuals. Expert users also took 59.97 s less on average to create an animation. However, if we compute the average animation creation time per individual, one-way analysis of variance (ANOVA) yields a p value of 0.0748, greater than the common significance level of 0.05. This means no statistical difference was observed between experts and non-experts in terms of the authoring time. Given the fact that the average authoring time was about 3 min and all participants completed each animation in under 10 min, we conclude that non-experts are able to use the authoring interface almost as conveniently as expert users.

We also asked participants for general comments on the framework regarding ease of use. The registration process was difficult for many, mainly because most of them had never done anything similar. Registering a 3D scene took significantly longer than registering a 2D poster. However, we also observed that individuals improved rapidly. Once a subject knew how moving and tilting the smartphone affected the registration of QR code landmarks, a view could be registered in 10 s. For the authoring process, most participants liked the undo functionality and all the animation schemes except 3D transformation. Some of them had a hard time tuning the segmentation parameters or warping an object. This could be due to the fact that non-experts are not familiar with ideas of segmentation and homography. Participants also gave feedback on potential improvements for the interface, such as adding a scaling functionality, preserving segment selections between keyframes, and more flexibility in the list of thumbnails. Finally, participants mentioned interesting applications of the framework, including indoor navigation, interactive advertisement, museum displays, home decoration, AR treasure hunt, employee training, educational illustrations, etc. Overall, the participants enjoyed using the system both as a consumer and as an author. They found the system to be innovative and useful.

From the sample applications and user study, we demonstrate that our system is able to handle various scenes from posters on a wall to objects displayed in the physical world. It gives reasonable results for scenes that consist of only pure color parts or specular and textured regions. To the best of our knowledge, this is the first framework to communicate videos in a network-free environment, making the communication both private and personalized. Our new framework is innovative in four respects—authors do not need to employ computer specialists to create expository media, the media incorporate new modes of presentation and interaction, Internet communication is not required, and consumers are able to personalize their use of the media. The examples in the paper and portfolio show that our system is also versatile and applicable to a variety of areas such as education, consumer product documentation, cultural heritage preservation, creative art, and others that require explanations of objects and processes.

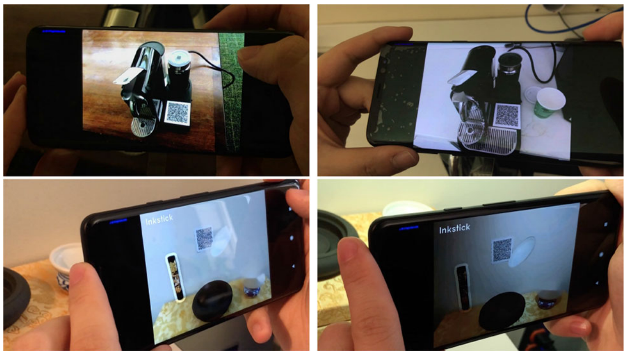


Fig. 12 Top: robustness to changes in the vantage point and environment. Bottom: robustness to different lighting conditions

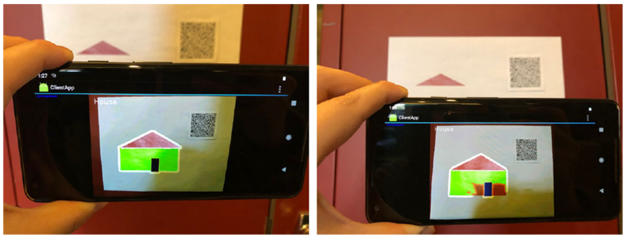


Fig. 13 Left: correct decoding when consumer's view and author's view are similar. Right: imperfect segmentation due to change in consumer's view

7 Limitations and future work

The main limitation of our system is that the segmentation and inpainting results are not always perfect when there are extreme changes in the environment, due to our choice of the LSC and Telea algorithms as a trade-off between performance and runtime on a mobile device. The consumer's view may not be exactly the same as the author's, and the object may be moved to another environment with different background and lighting. Our system is robust to reasonable changes in the vantage point and lighting conditions. Figure 12 shows animations decoded correctly in environments with specular highlights and cast shadows. The registration threshold we set in the mobile app is a trade-off between registration difficulty and the quality of decoded animations. This means if the consumer takes a picture from a slightly different view, the algorithm usually succeeds in matching the ROIs and recovering a personalized animation. However, drastic changes in the view or lighting conditions can cause failure in ROI matching, which can make decoded animations confusing. Figure 13 shows the result of imperfect segmentation when the camera is placed at a different angle than the author's view. Interestingly, we observed in the user study that participants were able to create the intended animation in the authoring interface even if some ROIs were mismatched. This shows that the author's ideas can still be effectively communicated using our system even when the visual effects are not perfect. The framework can be further improved by gaining design-related insights from more experiments such as determining the range of consumer camera angles allowed to decode a coherent animation and exploring easier ways for consumers to register a scene.

By design, our system only handles scenes whose parts are in a rigid configuration. If a consumer's scene has multiple objects that are in a configuration different from that of the author's, the correct animation cannot be decoded. In the future, this may be solved using multiple codes to support localization of objects in the scene. Furthermore, no additional objects can be introduced to the scene due to the limited capacity of QR codes and dependence on the consumer's picture to provide personalization. An improvement would be adding another animation scheme that enables introducing geometric primitives. Other ideas to enrich the animation schemes include rendering a contour around the highlighted ROI for annotation, adding image layer representations, and linking other pictures or videos that consumers can take. In particular, the framework can be generalized to use 3D information with recent smartphones that support capturing depth images. The current choice of QR code occupies a noticeable area in the scene. More alternatives such as aesthetically pleasing codes [24], watermarking techniques [34], and 3D fabricated unobtrusive tags [21] can be applied in the future. Other possible future work includes adopting image processing algorithms based on deep learning and developing new applications.

The full system is available with source code at https://github.com/zachzeyuwang/AniCode.

Compliance with ethical standards

Conflict of Interest The authors declare that they have no conflict of interest.

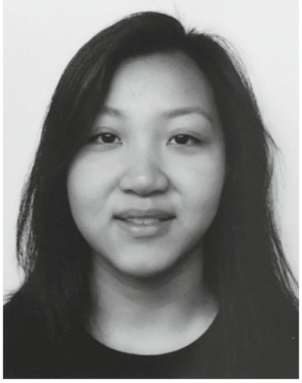
References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Anal. Mach. Intell. **34**(11), 2274–2282 (2012)
2. Agrawala, M., Li, W., Berthouzoz, F.: Design principles for visual communication. Commun. ACM **54**(4), 60–69 (2011)
3. Agrawala, M., Phan, D., Heiser, J., Haymaker, J., Klingner, J., Hanrahan, P., Tversky, B.: Designing effective step-by-step assembly instructions. ACM Trans. Gr. **22**(3), 828–837 (2003)
4. Andolina, S., Pirrone, D., Russo, G., Sorce, S., Gentile, A.: Exploitation of Mobile Access to Context-Based Information in Cultural Heritage Fruition. In: Proceedings of the International Conference on Broadband, Wireless Computing, Communication and Applications, pp. 322–328. IEEE (2012)
5. Appiah, O.: Rich media, poor media: The impact of audio/video vs. text/picture testimonial ads on browsers' evaluations of commercial web sites and online products. J. Curr. Issues Res. Advert. **28**(1), 73–86 (2006)
6. Ashok, A.: Design, modeling, and analysis of visual mimo communication. Ph.D. thesis, Rutgers The State University of New Jersey-New Brunswick (2014)
7. Badam, S.K., Elmqvist, N.: Visfer: camera-based visual data transfer for cross-device visualization. Inf. Vis. **18**(1), 68–93 (2019)
8. Barak, M., Ashkar, T., Dori, Y.J.: Teaching science via animated movies: its effect on students' thinking and motivation. Comput. Edu. **56**(3), 839–846 (2011)
9. Van den Bergh, M., Boix, X., Roig, G., de Capitani, B., Van Gool, L.: Seeds: Superpixels extracted via energy-driven sampling. In: Proceedings of the European Conference on Computer Vision, pp. 13–26. Springer (2012)
10. Carter, S., Cooper, M., Adcock, J., Branham, S.: Tools to support expository video capture and access. Educ. Inf. Technol. **19**(3), 637–654 (2014)
11. Carter, S., Qvarfordt, P., Cooper, M., Mäkelä, V.: Creating tutorials with web-based authoring and heads-up capture. IEEE Pervasive Comput. **14**(3), 44–52 (2015)
12. Chang, C.S., Chu, H.K., Mitra, N.J.: Interactive videos: plausible video editing using sparse structure points. Comput. Gr. Forum **35**(2), 489–500 (2016)
13. Cho, N.H., Wu, Q., Xu, J., Zhang, J.: Content authoring using single image in urban environments for augmented reality. In: Proceedings of the International Conference on Digital Image Computing: Techniques and Applications, pp. 1–7. IEEE (2016)
14. Chu, J., Bryan, C., Shih, M., Ferrer, L., Ma, K.L.: Navigable videos for presenting scientific data on affordable head-mounted displays. In: Proceedings of the ACM Conference on Multimedia Systems, pp. 250–260. ACM (2017)
15. Clarine, B.: 11 reasons why video is better than any other medium. http://www.advancedwebranking.com/blog/11-reasons-why-video-is-better (2016)
16. Feiner, S.K., McKeown, K.R.: Automating the generation of coordinated multimedia explanations. Computer **24**(10), 33–41 (1991)
17. Fidas, C., Sintoris, C., Yiannoutsou, N., Avouris, N.: A survey on tools for end user authoring of mobile applications for cultural heritage. In: Proceedings of the International Conference on Information, Intelligence, Systems and Applications, pp. 1–5 (2015)
18. Hern, A.: Fitness tracking app strava gives away location of secret us army bases. The Guardian (2018). https://www.theguardian.com/world/2018/jan/28/fitness-tracking-app-gives-away-location-of-secret-us-army-bases
19. Kaplan, A.M., Haenlein, M.: Users of the world, unite! the challenges and opportunities of social media. Bus. Horizons **53**(1), 59–68 (2010)

20. Karat, C.M., Pinhanez, C., Karat, J., Arora, R., Vergo, J.: Less clicking, more watching: Results of the iterative design and evaluation of entertaining web experiences. In: Proceedings of the IFIP TC13 International Conference on Human-Computer Interaction, pp. 455–463 (2001)
21. Li, D., Nair, A.S., Nayar, S.K., Zheng, C.: AirCode: Unobtrusive physical tags for digital fabrication. In: Proceedings of the ACM Symposium on User Interface Software and Technology, pp. 449–460. ACM (2017)
22. Li, Z., Chen, J.: Superpixel segmentation using linear spectral clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1356–1363 (2015)
23. Liao, I., Hsu, W.H., Ma, K.L.: Storytelling via navigation: A novel approach to animation for scientific visualization. In: M. Christie, T.Y. Li (eds.) Proceedings of the International Symposium on Smart Graphics, pp. 1–14. Springer International Publishing, Cham (2014)
24. Lin, S.S., Hu, M.C., Lee, C.H., Lee, T.Y.: Efficient qr code beautification with high quality visual content. IEEE Trans. Multimed. **17**(9), 1515–1524 (2015)
25. McKercher, B., Du Cros, H.: Cultural Tourism: The Partnership Between Tourism and Cultural Heritage Management. Routledge, Abingdon (2002)
26. OpenCV team: OpenCV for Android SDK. https://opencv.org/platforms/android (2017)
27. Owen, S., Switkin, D., team, Z.: ZXing barcode scanning library. https://github.com/zxing/zxing (2017)
28. Parent, R.: Computer Animation: Algorithms and Techniques. Newnes, Oxford (2012)
29. Revell, T.: App creates augmented-reality tutorials from normal videos. New Scientist (2017). https://www.newscientist.com/article/2146850-app-creates-augmented-reality-tutorials-from-normal-videos
30. Schnotz, W.: Commentary: towards an integrated view of learning from text and visual displays. Educ. Psychol. Rev. **14**(1), 101–120 (2002)
31. Telea, A.: An image inpainting technique based on the fast marching method. J. Gr. Tools **9**(1), 23–34 (2004)
32. Upson, C., Faulhaber Jr., T., Kamins, D., Laidlaw, D., Schlegel, D., Vroom, J., Gurwitz, R., Van Dam, A.: The application visualization system: a computational environment for scientific visualization. IEEE Comput. Gr. Appl. **9**(4), 30–42 (1989)
33. Wouters, P., Paas, F., van Merriënboer, J.J.: How to optimize learning from animated models: a review of guidelines based on cognitive load. Rev. Educ. Res. **78**(3), 645–675 (2008)
34. Xiao, C., Zhang, C., Zheng, C.: FontCode: Embedding information in text documents using glyph perturbation. ACM Trans. Gr. **37**(2), 15 (2018)
35. Yeshurun, Y., Carrasco, M.: Attention improves or impairs visual performance by enhancing spatial resolution. Nature **396**(6706), 72–75 (1998)
36. Yuan, W., Dana, K., Varga, M., Ashok, A., Gruteser, M., Mandayam, N.: Computer vision methods for visual mimo optical system. In: CVPR Workshops, pp. 37–43 (2011)
37. Yue, Y.T., Yang, Y.L., Ren, G., Wang, W.: SceneCtrl: Mixed reality enhancement via efficient scene editing. In: Proceedings of the ACM Symposium on User Interface Software and Technology, pp. 427–436. ACM (2017)
38. Zheng, Y., Chen, X., Cheng, M.M., Zhou, K., Hu, S.M., Mitra, N.J.: Interactive images: cuboid proxies for smart image manipulation. ACM Trans. Gr. **31**(4), 99 (2012)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Zeyu Wang received the B.Sc. degree (Summa Cum Laude) in machine intelligence from Peking University in 2016. He is currently a Ph.D. student in computer graphics at Yale University. His research interests include virtual and augmented reality, cultural heritage, visual perception, and intelligent interfaces for creative applications.



Shiyu Qiu received the B.S. degree in computer engineering from Columbia University in 2017. She is currently a Ph.D. student in computer graphics at Yale University. Her research interests include virtual and augmented reality, intelligent interfaces for creative applications, and 3D modeling.

Qingyang Chen received the B.S. degree in computer science from Yale University in 2017. He is currently a software engineer at Google.

Natallia Trayan received the bachelor's degree in fashion design from the Academy of Arts, Architecture, and Design in Prague in 2016, as well as from Otago Polytechnic in 2015. She is currently a graphic designer at Luvly in the Kapiti Coast, New Zealand.

Alexander Ringlein received the B.S. degree in computer science and mathematics and philosophy from Yale University in 2018. He is currently a software engineer on Oculus at Facebook.

Julie Dorsey is a professor of computer science at Yale University and the founder and chief scientist of Mental Canvas, Inc. She came to Yale in 2002 from MIT, where she held tenured appointments in both the Department of Electrical Engineering and Computer Science (EECS) and the School of Architecture. She received undergraduate degrees in architecture and graduate degrees in computer science from Cornell University. Her research interests include material and texture models, sketch-based modeling, and creative applications of AI. She has received several professional awards, including MIT's Edgerton Faculty Achievement Award, a National Science Foundation Career Award, an Alfred P. Sloan Foundation Research Fellowship, along with fellowships from the Whitney Humanities Center at Yale and the Radcliffe Institute at Harvard.

Holly Rushmeier is a professor of computer science at Yale University. She received the B.S., M.S., and Ph.D. degrees in mechanical engineering from Cornell University in 1977, 1986, and 1988 respectively. Between receiving the Ph.D. and arriving at Yale, she held positions at Georgia Tech, NIST, and IBM Watson research. Her current research interests include acquiring and modeling material appearance, applications of human perception to realistic rendering, and applications of computer graphics in cultural heritage. She is a fellow of the Eurographics Association, an ACM Distinguished Engineer and the recipient of the 2013 ACM SIGGRAPH Computer Graphics Achievement Award.