

# Developing an AR Pop-up Picture Book and its Effect Editor Based on Teaching Motions

Hitomi Kataoka <sup>\*</sup>, Tadachika Ozono <sup>\*</sup>, Toramatsu Shintani <sup>\*</sup>

## Abstract

We propose an augmented reality (AR) pop-up picture book (AR book), which is a digital picture book using the AR technology. Digital picture books increase the interest of children in reading and are often used to encourage children to read. Conventional picture book writers have few technical skills and expertise in developing AR contents. We aim to develop an effect editor for AR books to aid the writers create expressive AR books. We propose a new intuitive method for obtaining editing effects in AR books using teaching motions based on demonstrations of writers using real-world objects. The results of our experimental evaluation demonstrate that the system can be effectively used to edit 3D motion effects in AR books.

*Keywords:* Augmented Reality, AR Pop-up Picture Book, Motion Teaching, Effect Editor

## 1 Introduction

Picture book writers create picture books to help improve children's interest in reading and to encourage children to read. Digital picture books increase the interest of children in reading and are often used to encourage children to read. One approach to improve children's interest in reading is to develop digital picture books using the augmented reality (AR) technology. The availability of expressions that are not observed in conventional picture books, such as the immersion and interaction with real-world objects in the AR space, may increase a user's interest in reading.

Zhang et al. [1] reported that the addition of AR content to a picture book improves the reader's interest, fun, and understanding. Therefore, we assume that the development of digital picture books using the AR technology helps readers to immerse themselves in reading and also improves their understanding of the subject matter. We developed an AR pop-up picture book (AR book), a picture book in the AR space.

We aim to develop an effect editor for AR books to help writers to create expressive AR books without help of developers. The developer is an expert who can create effects of AR objects by programming or dedicated tools for experts. While developing an AR book, writers apply motion effects, such as movement and rotation, to the AR objects in the book based on a storyline. Writers had to input various parameters, including the position and

---

<sup>\*</sup> Department of Computer Science, Graduate School of Engineering, Nagoya Institute of Technology

direction, to place the AR objects in the AR space and to achieve motion effects. The main disadvantage associated with inputting parameters is that it is difficult for writers to develop AR books. We explore various methodologies to enable the writers to achieve editing effects in an AR space without any programming requirement. The creation of characters in an AR book with interactive behaviors is essential for the development of AR books; however, it is time-consuming, for example, the creation of an interactive behavior such as moving or rotating a character with six degrees of freedom and changing the pose of the neck or the hands. Further, the representation of interactive behavior requires various natural motions that are time-consuming to enter programmatically as numerical values. Also, expressing the interactions among multiple characters is challenging because of the synchronization of multiple actions. This study tries to eliminate the process of inputting numerical values, enabling writers to easily edit the complex motion effects.

Our system captures the intended motion effect based on demonstration of a writer, which is used to edit motion effects of AR object in an AR book. In the system, writers are not required to input the position and direction parameters of AR objects in the program; further, the system captures the intended motion based on the writer's demonstration using tablet devices without using any specialized device for motion tracking. We developed an interface that enables writers to perform operations, such as tap and swipe, to edit effects and to develop an AR book without programming. We proposed a method that can only edit the horizontal movement effect with a 5% error for an AR book[2]. This paper proposes more detailed experiments that evaluate the performance of capturing 3D movement and rotation effects.

The remains of this paper are organized as follows. Section II presents the related works, and Section III explains AR books with an example. Section IV shows the implementation of a support system for achieving editing effects. Section V presents an experiment to evaluate the editing effects method based on teaching the motion, and, in Section VI, the experimental results and the editing effects method are discussed. Finally, we conclude this study in Section VII.

## 2 Related Work

Various digital picture books that improve children's interest in reading have been developed [3][4]. Unlike the printed picture books, digital picture books provide access to additional features such as animation, sound, and touch-interactive opportunities[5]. Some recent studies have also discussed digital picture books based on the AR technology. Lin et al. [6] examined the learning motivations and creative capabilities of students using text-based learning, picture-based learning, and AR digital picture books. Additionally, they observed that the learning motivation and understanding of students were improved using AR digital picture books.

Also, some recent researches have focused on eliminating the challenges associated with the development of AR content. Various technologies, such as Apple ARKit and Google ARCore, enable the usage of AR in smartphone and tablet applications without any requirement for specialized hardware. The progress of mobile AR technologies increases the potential of AR content creators. Despite the availability of these technologies, experienced mobile application developers still find it challenging and time-consuming to create AR applications.; consequently, it prevents the widespread of AR contents usage. The support systems for the AR contents provide better opportunities for the widespread of

the AR applications. Moreover, Mota et al. [7] proposed that the authoring tools for AR applications can be classified into two tools: the programming and content design tools. The latter completely removes the needs of programming skills, which are anticipated to be more suitable when we consider the number of future users. Thus, our proposed system is an appropriate approach for non-skilled programmers. Neveling et al. [8] proposed a prototyping tool that supported the AR interface design using paper and Play-Doh. They proved that non-digital tools and materials are flexible for designing AR interfaces and for demonstrating challenging interactions with the AR content.

Further, some researchers proposed methods to create animations using demonstrations because teaching motions based on demonstrations using real-world objects is a practical approach to develop expressive motions. It helps non-experts by removing the challenge of designing expressive motions. For example, Barnes et al. [9] proposed a system that enables users of all skill levels to swiftly create cutout-style animations by performing character motions. By demonstration, they confirmed that even first-time users, including an eleven-year-old can perform the motions by manipulating physical objects. Held et al. [10] proposed a system for producing 3D animations by physically manipulating puppets in front of a Kinect depth sensor. Their system supports layered animations to produce animations in which multiple characters move simultaneously. The users can separately capture motions with different characters and combine them into one animation. Slyper et al. [11] proposed a system for creating gestures for robotic toys. However, we propose a motion editing system based on teaching by demonstration using a single-camera-based AR system.

Based on previous studies, we can propose a system that enables writers to easily develop AR picture books. The proposed system eliminates the requirement for entering numerical values for achieving editing effects in an AR picture book and provides an intuitive interface for writers. This study aims to support the AR picture book writers for intuitively developing AR picture books without writing any code. Our system edits the motion effects of an AR object based on the motion captured from the writer's demonstration, using only an iPad and the existing real-world objects.

### **3 AR Pop-up Picture Book**

#### **3.1 Features of an AR Book**

An AR book is a digital picture book that is displayed virtually on a real-world flat surface using the AR technology. Our system exhibits the following three features:

- the system contains three-dimensional (3D) AR objects, such as characters, houses, and trees, on a plane in the AR space;
- the system animates the AR objects placed in the AR space;
- the system displays the text of the picture book one sentence at a time so that the users can read at their own speed.

The AR book represents the depth and distance of AR objects, which cannot be represented in conventional printed picture books, using 3D AR objects. We intend to help the readers to understand a story by animating the AR objects; displaying one sentence at a time appeals to readers who are not used to reading long stories. The AR book promotes reading by



Figure 1: The AR pop-up picture book.

improving the reader’s interest and understanding to support the transition to reading picture books containing a large amount of text and to books containing only text in the future. When the AR book begins, the AR objects of the picture book can be observed on a real-world flat surface, which makes the readers feel that the picture book is spread on the floor. Further, the readers can immerse themselves into reading because they can see the AR picture books from various directions by moving around.

### 3.2 Implementation

Placing the AR objects on a plane in the AR space requires the AR book to recognize the real-world flat surface and to subsequently place the AR object on it. Our system uses Apple’s ARKit to detect the planes. As the AR picture book displays a camera view, ARKit detects horizontal surfaces, such as floors, based on camera images.

We implemented various motion effects, such as jumps, sways, and horizontal movements, to visually express the movements of characters and emotional effects such as heart, sweat, and grumpy. Further, we applied animation effects to the AR objects in accordance with the flow of the story. For example, we applied a motion effect to a character who is speaking and an emotional effect to depict the emotion of that character.

Fig. 1 denotes an example of the AR book. The system displays the AR book on the floor of the detected real-world plane. When 3D AR objects, such as picture book characters and buildings, are displayed in the AR space, an emotion effect can be observed above the head of a character. The system displays a text of the picture book at the bottom of the screen, one sentence at a time. The story presented in the picture book progresses to the next sentence when the user taps the screen.

While developing the AR book, the writer sets the parameters of the AR object, including the position, direction, speed, pose, and emotional effect, for editing the effects of the character’s motion. The writer should enter the duration of the movement and the distance to move the AR object in the X, Y, and Z directions to move the character in the AR book. Entering such complex information in the form of numerical values is time-consuming and challenging for writers.

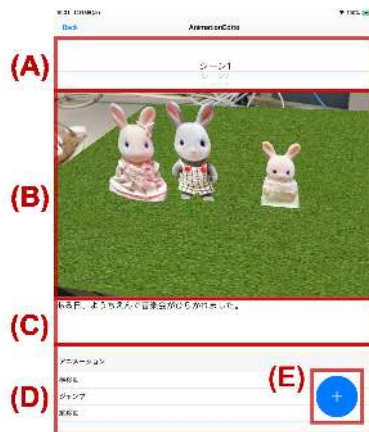


Figure 2: UI for editing effect.

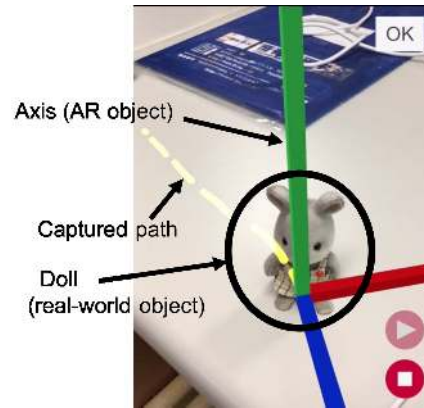


Figure 3: Example of editing effects obtained by capturing the motion.

## 4 Editing Effects by Teaching Motion

### 4.1 Editing Effects Using Real-world Objects

Fig. 2 denotes the user interface (UI) for editing effects. Users can edit the effects on the screen after registering a text and position of characters in the picture book. Fig. 2(A) provides the scene selection, and Fig. 2(B) displays the preview of an AR picture book for the currently selected scene in the camera view. Fig. 2(C) displays the text of the currently selected scene, whereas Fig. 2(D) provides a list of the effects registered in the currently selected scene. Further, users can tap the Fig. 2(E) button to add effects. When the button is tapped, users enter the name of the effect to be added, select the target character, and begin a demonstration motion teaching.

Fig. 3 denotes an example of editing effects based on teaching motions. In the figure, we used a doll as a real-world object. The camera view displays an axis and an AR object based on the position and direction of the detected doll in the AR space. There are three axes: the axis perpendicular to the desk, the axis parallel to the desk, and the axis facing the doll. We move the doll on the desk from left to right at regular intervals, and the axis keeps track of the changes. The camera view displays the yellow line, the AR object, and the path obtained by capturing the motion. We illustrate an example of the editing motion effects for multiple characters in one scene in Fig. 4. The camera view displays the movement path of the registered character (the blue line on the left) and of the character that the user is teaching a motion (the yellow line on the right) on the desk. Our system enables users to teach motions by demonstration while reviewing the movement path of other characters to edit the motion effect for the interaction between multiple characters.

Fig. 5 depicts the UI for motion effect creation by teaching. The left of Fig. 5 is the screen before beginning with the teaching. (A) denotes the “Effect Creation Finish” button, (B) denotes the “Effect Preview” button, and (C) denotes the “Teaching Start” button. Tapping the “Teaching Start” button changes the screen to a teaching screen, as denoted in the center of Fig. 5, whereas tapping the “Teaching Finish” button ends the teaching and saves the captured path. When the “Effect Preview” button is selected, the screen changes to the motion effect preview screen, as presented on the right of Fig. 5. On this screen, users can confirm the motion effect created by teaching.

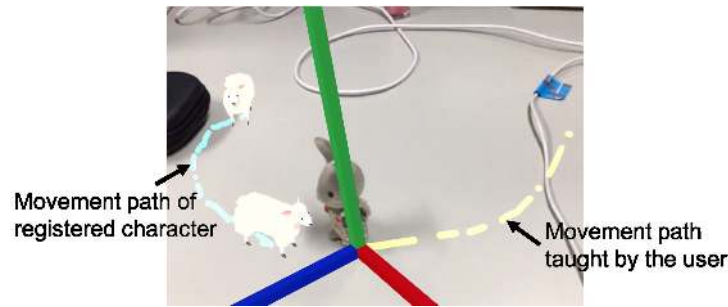


Figure 4: Example of editing effects for multiple characters in one scene.

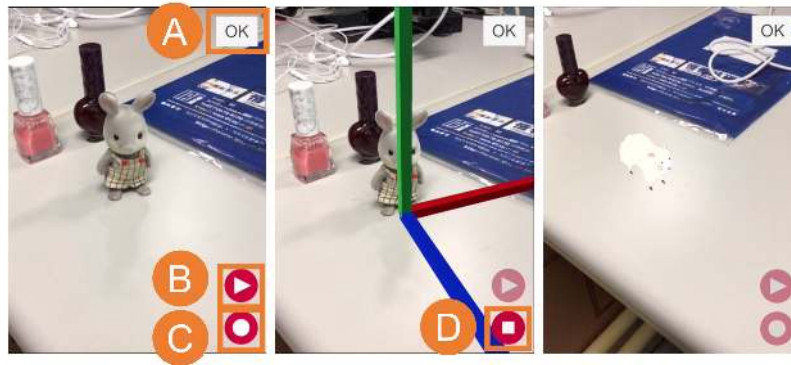


Figure 5: The UI for motion effect creation by teaching. The left denotes the screen before teaching starts, the center denotes the teaching screen, and the right denotes the preview of the screen in effect.

When users make effects of interaction among multiple characters, they first make the effect of the individual character and then overlap the individual effects. Therefore, the precision of the motion tracking should be high enough.

## 4.2 System Architecture and Implementation

This section describes the architecture and the implementation of the mechanism that supports the editing effects. We illustrate the system architecture in Fig. 6. Our system comprises (1) AR Scanner, (2) AR Tracker, (3) Editing Effect Subsystem, (4) AR Picture Book Display Subsystem, and (5) Scenario DB. AR Scanner records the spatial features of a real-world object from a camera image and creates an AR reference object to detect the real-world object. AR Tracker detects the real-world object based on a camera image and tracks the motion of the AR reference object registered in Scenario DB. Further, Editing Effect Subsystem registers the positions and directions captured by AR tracking as an effect in Scenario DB. AR Picture Book Display Subsystem loads the scenario and the AR objects from Scenario DB and displays an AR picture book on a real-world flat surface. Further, we implemented the system using ARKit for displaying the AR objects, scanning, and tracking.

We used AR Scanner in Fig. 6 to scan a real-world object and exported the AR reference object file using the functions provided by ARKit in detecting the real-world object. Then, to scan a real-world object, we resized a bounding box in the camera view using swipe operation and defined a region that contains a real-world object. Further, we viewed the real-world objects from different angles and scanned all the faces, including the top and

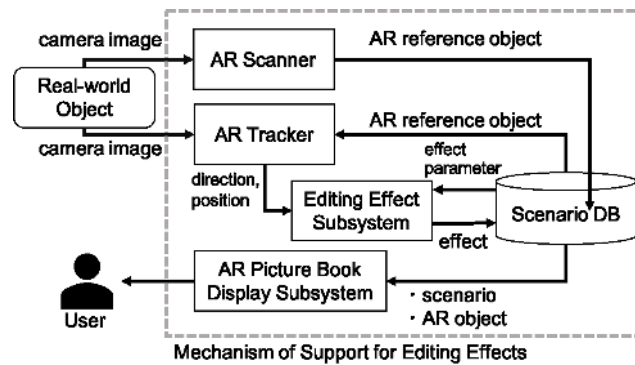


Figure 6: The system architecture of the support mechanism for editing effects.



Figure 7: An example of the experiment.

sides, of the bounding box. We used AR Tracker to examine the position, direction, and scale of the detected real-world object and saved the position and direction of the detected real-world object when an update event occurs. At the end of the demonstration, AR Tracker deleted the AR object from the camera view and saved the positions and directions of the learned motion captured as CSV files.

## 5 Evaluation

In this section, we describe the assessment of the 2D/3D editing effects for a single character using our proposed system. We employed a recall of obtaining the editing effects using the teaching motion as the performance evaluation value. Whereas, it is essential to reproduce each intended effect while considering the overlapping effects for multiple characters.

### 5.1 Procedure

We measured the accuracy of our editing method through the editing effects users apply around an obstacle on a desk that is a real-world object. We marked the target path in advance, and a user moved a real-world object along with the marks. Furthermore, we replayed obtained effects to observe the AR object character's movement over the marks. Fig. 7 shows an example of the experiment. We used an upside-down white cup as an obstacle (see the center in Fig. 7) and a doll as a real-world object (see right in Fig. 7). The motion of the doll was tracked on the solid-color desk. The environment for motion

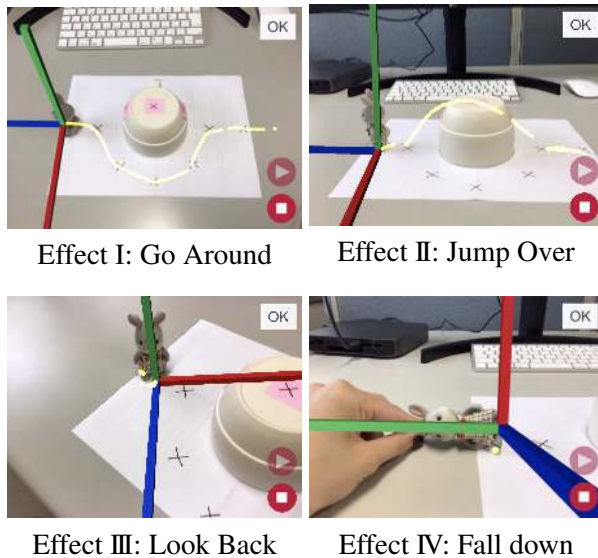


Figure 8: Examples of editing effects for the experiment.

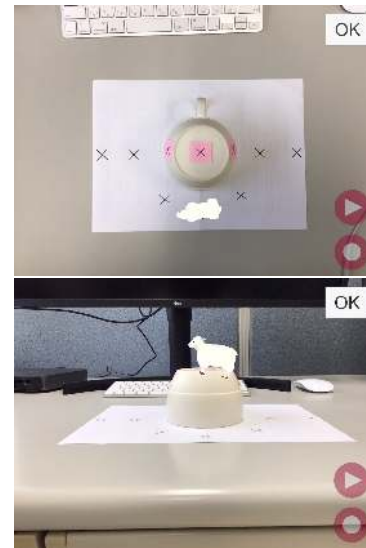


Figure 9: The examples of the experiment. The upper and the lower image denote Effect I and II, respectively. These are similar paths, but the system accurately distinguishes them as planar or stereoscopic.

teaching had a total of 10 marks on white paper and the white cup. We used an iPad running iOS 12.3.1 for conducting the experiment.

We evaluated the motion teaching for the following four effects: Effect I (Go Around), Effect II (Jump Over), Effect III (Look Back), and Effect IV (Fall Down). The user repeated motion teaching ten times each. Fig. 8 displays examples of the effects. The upper left of Fig. 8 represents Effect I viewed from above. The system detected the doll on the left of the screen and displayed a yellow line as the captured path. The user moved the doll along the seven marks from right to left to detect its position. The doll moved horizontally in Effect I. The upper right of Fig. 8 represents Effect II viewed from the side. The system displayed the path obtained by teaching as a yellow line as with Effect I. The lower left of Fig. 8 depicts Effect III creation by teaching. The user rotated the doll  $180^\circ$  on the mark. The user rotated it at  $45^\circ$  intervals on the vertical green axis to detect its direction. The lower right of Fig. 8 displays Effect IV creation by teaching. The user rotated the doll  $90^\circ$  on the blue axis from back to front on the mark to detect its direction and returned. We measured recall that is the percentage of successes. The number of successes is the number of correct replays. Specifically, the numbers of successes of Effect I and II are the numbers of marks where the system displays the AR object at the correct mark, and the number of successes of Effect II I and IV are the numbers of correctly detected rotations.

## 5.2 Results

We played the obtained effects to observe the motions of the AR object. We found them both from the above (the upper of Fig. 9) and from the side (the lower of Fig. 9). Fig. 10 displays the results. The horizontal and vertical axes of Fig. 10 represent Effect I-IV and



their recall, respectively. The experimental results denote that our proposed system can edit 3D movement and rotation effects with practical and significant performance.

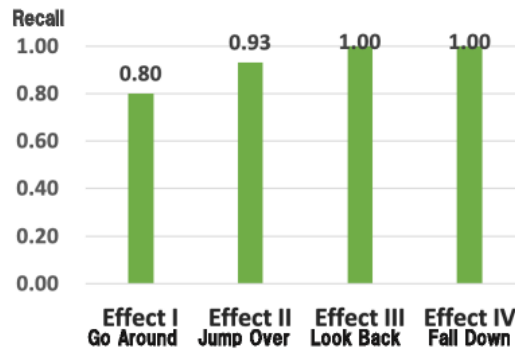


Figure 10: Recalls in Effect I - IV.

## 6 Discussion

The experimental results demonstrate that our system can capture the intended 3D effects. Therefore, we conclude that the system provides enough performance to overlap the effects for multiple characters. In most results of the Effect I, the AR object correctly passed over the marks; however the system achieved only two marks twice out of ten times. Consequently, the shape of the obtained path looked the same as the target path and deviated on the whole. Then, moving the device quickly may cause the shift. In such a situation, users manually modify the captured paths, for example, moving points of the path.

While teaching motion based on the writer's demonstration using real-world objects, the system also captures the speed and pose of characters along with their position and direction. The system continuously saves the position information about the real-world object as it is updated; therefore, it does not consider the motion teaching speed. As future work, we aim to realize the motion effects anticipated by the writers by applying the motion teaching speed to the AR objects. Also, we aim to enable writers to edit the character pose easily by applying the pose of the real-world objects to the AR objects.

Teaching motion with small movements is time-consuming because the user waits for the detection of the real-world object more often; hence, small movements may not be accurately tracked, and the coordinate system may shift. Therefore, our system accurately teaches the motion of the real-world object at intervals. Additionally, interpolating and smoothing functions for captured paths are useful.

## 7 Conclusion

In this paper, we proposed an AR pop-up picture book, a picture book in the AR space. Moreover, we proposed an effect editor using teaching motions for creating expressive AR books to support writers who are not developers. Furthermore, we proposed a method for obtaining editing effects by teaching motion based on the writers' demonstration using real-world objects. By illustration, our proposed system supports teaching motions while reviewing the movement path of other characters to edit the effect for multiple characters in one scene. The experimental results denote that our system can edit 3D movement with

significant performance. Finally, we developed an intuitive interface to edit effects easily without programming. The interface provides support for basic operations, such as tap and swipe, and enables writers to develop an AR book.

## Acknowledgments

This work was supported in part by JSPS KAKENHI Grant Number 19K12097, 19K12266. The authors would like to thank Enago ([www.enago.jp](http://www.enago.jp)) for the English language review.

## References

- [1] J. Zhang, Y. Sasaki, T. Tsuruno, Y. Matsuura, N. Utsunomiya, and H. Yamamoto, "Investigation of Kansei Added Value in Book Publishing Filed by Using AR Contents" *International Journal of Affective Engineering*, vol. 14, no. 3, 2015, pp.151-155.
- [2] H. Kataoka, T. Ozono, and T. Shintani, "Realizing an Effect Editor for AR Pop-up Picture Books by Teaching Motions" *IDDC2019, IEEE*, 2019, pp. 989-994.
- [3] H. Stichnothe, "Engineering stories? A narratological approach to children's book apps," *Nordic Journal of ChildLit Aesthetics*, vol. 5, 2014, pp. 1-9.
- [4] B. Sargeant, "What is an ebook? What is a Book app? And why should we care? An analysis of contemporary digital picture books," *Children's Literature in Education*, vol. 46, no. 4, 2015, pp. 454-466.
- [5] F. Serafini, D. Kachorsky, and E. Aguilera, "Picture Books in the Digital Age," *The Reading Teacher*, vol. 69, no. 5, 2016, pp. 509-512.
- [6] P. Lin, Y. Huang, and C. Chen, "Exploring imaginative capability and learning motivation difference through picture e-book," *IEEE Access*, vol. 6, 2018, pp. 63416-63425.
- [7] R. C. Mota, R. A. Roberto, and V. Teichrieb, "Authoring tools in augmented reality: an analysis and classification of content design tools" *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality*, 2015, pp. 164-167.
- [8] M. Nebeling, J. Nebeling, A. Yu, and R. Rumble, "ProtoAR: Rapid Physical-Digital Prototyping of Mobile Augmented Reality Applications" *Proc. of CHI'18*, no. 353, 2018, 12p.
- [9] C. Barnes, D. Jacobs, J. Sanders, D. Goldman, S. Rusinkiewicz, A. Finkelstein, and M. Agrawala, "Video Puppetry: A Performative Interface for Cutout Animation" *ACM Transactions on Graphics (TOG)*, vol. 27, 5, no. 124, 2008, pp. 1-9.
- [10] R. T. Held, A. Gupta, V. Curless, and M. Agrawala, "3D Puppetry: A Kinect-based Interface for 3D Animation" *Proceedings of the 25th annual ACM symposium on User interface software and technology*, 2012, pp. 423-434.
- [11] R. Slyper, G. Hoffman, and A. Shamir, "Mirror Puppeteering: Animating Toy Robots in Front of a Webcam" *Proceedings of the 9th International Conference on Tangible, Embedded and Embodied Interaction*, 2015, pp. 241-248.