

Cut-Out Animation Using Magnet Motion

Srinivas Anumasa

Computer Science

Shri Ramswaroop Memorial University

Lucknow, 225003, India

srinu.0922@gmail.com

Avinash Singh

Computer Science

Shri Ramswaroop Memorial University

Lucknow, 225003, India

singhavinash66@gmail.com

Rishi Yadav

Computer Science

Shri Ramswaroop Memorial University

Lucknow, 225003, India

rishiyadavjec@gmail.com

Abstract

In this paper a new video-based interface for creating cutout-style animation using magnets is presented. This idea makes room for users of all skill levels to animate. We created an interface which is generally a closed box with a camera placed at the bottom. Roof of the box acts like a multi touch interface. A cast of physical characters are designed by the animator using paper, markers and scissor. If animator wants to animate using this physical characters, he pastes them to magnets (we call them as PMagnets) and place them under the roof (puppets facing the camera). These characters are controlled by other magnets (CMagnets) on the top of the roof. Here we require, a simple foreground extraction algorithm to extract the characters and to render them onto a new background. Our system "Cut-Out Animation using Magnet Motion" runs in real time (i.e 30 Frames/Sec). Therefore animator and the audience can instantly see the animation.

Keywords: Cut-Out Style Animation, Background Replacement, Magnets.

1. INTRODUCTION

Animation is an illusion of movement created by rapidly displaying a sequence of images. Creating an animation sequence is a time consuming process. Hand drawn or stop motion animation is a laborious process in which every frame is manually drawn or crafted. For computer generated animation several tools like Maya, Flash, etc. are available in the market. Even though they provide a good interface for creating easy animation sequence one must be an expertise and requires training to gain control over the interface provided.

Puppetry animation is a real time story telling animation which is famous from past few centuries. Puppeteers use threads to control the motion of puppets and bring them to life. Puppeteers and their controls are often visible to the audience. Although the quality of Puppetry animation is not good as hand drawn animation, still it attracts kids because of its own way of storytelling.

Cut-out animation is one way of storytelling in which the characters are flat 2d characters designed using papers, scissors and colours. Traditional Cut-out animation is not a real time like Puppet shows, each frame is manipulated by adjusting the character cut-outs. It is a time consuming process.

In this paper we present a novel technique for cutout style animation which is mostly inspired by the work of Connally Barnes, David E. Jacobs Video Puppetry: A Performative Interface for

Cutout Animation. Authors in paper [1] proposed a simple cut-out style animation in which the characters are controlled by hand of a puppeteer. These Puppets are tracked and rendered onto a new background in the computer. In our approach each character is attached to magnets (PMagnet(s)) and placed on one side of the screen which are controlled by magnets (CMagnet(s)) on the other side of the screen. Thus giving life to the Puppets by concealing the Puppeteer. Also, if the animator wants to render on to the new background no tracking and object detection algorithms are required, just a simple background subtraction algorithm works well.

The main contribution of our work is the introduction of a new interface for creating cut-out style animated stories by using magnets and simple background subtraction algorithms. Designing this interface is easy, cheap, neither required complicated algorithms nor specialized skills, also no formal training was required for animation. However the animation produced by this method cannot match the visual quality of a studio production, but they are useful in many scenarios, including kids' productions or the class of animations such as "South Park" or "JibJab" that tries to simulate a "cut-out" style animation.

2. RELATED WORK

2.1. Video Puppetry.

Our work is inspired by the work of Connelly Barnes, David E. Jacobs [1]. The system in [1] is divided into two modules Puppet builder and Puppet theatre. For the first module authors created an interface which allows users to add new puppets into the database. A user first draws a character on paper with markers, crayons, or other high-contrast media. Next the user cuts out the puppet and places it under the video camera. The puppet builder captures an image of the workspace and processes the image in several stages before adding the puppet to a puppet database. Scale - invariant feature transform SIFT [2] features of each character are detected and stored in a database which is used in the second module.

The second module is puppet theatre, generally it is an object tracking environment. The user moves the puppets for animation, an over head camera which is used to track the animated characters by removing the user's hands and renders the characters on a new background. The SIFT [2] features are used to identify all puppets every 7-10 frames. Between SIFT updates, optical flow on Kanade- Lucas- Tomasi KLT [3] features are used to track the movement of puppets in real time.

But, this system has some limitations in handling every puppet. For example every puppet must occupy some significant portion in every video frame. So that every character can be tracked robustly. This system can handle 5 to 6 characters at a time, because as the number of characters gets increased the system gets slow. It is difficult to handle occlusions which may interfere with the tracking part. Because in animation there are many situations where characters move over each other. Puppets cannot be moved quickly because the KLT algorithm assumes that the displacement of tracking objects in between frames is small.

2.2. Sketch-n-stretch.

Authors in [4] designed an interface which is specially designed to simulate cut-out style animation. This is a two (virtual) layered interface. The user first selects the background image on which the foreground needs to render, then he selects cut-out pen from the menu and draws the border of the cut-out and selects the pencil tool to draw the character or he selects an image of a character stored in computer memory and then he selects an empty cut-out. By rotating, scaling and translation the animator can animate the character. This is not a multi touch interface it is not possible to animate two or more characters at the same time, so a time line option is provided. Each character is individually animated and combined using the time line.

3. PRINCIPLE OF OUR APPROACH

Traditional cut-out animation is a time consuming process, because for each frame the animator has to adjust the position of each Puppet or a part of the Puppet. Authors in [1] came with an which are time expensive. These algorithms are required to remove hands of animator which are

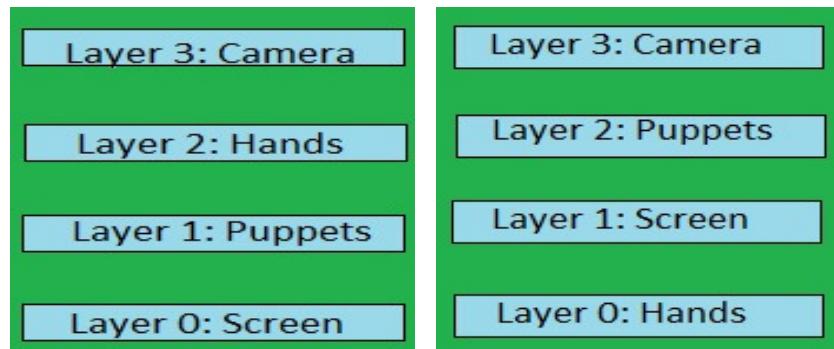


FIGURE 1: (a) (b)

in between the camera and Puppets. If we think this process as four layered animation, as shown in Figure 1(a) Layer 3 is camera, Layer 2 is hands (controller), Layer 1 Puppets, Layer 0 as screen. We tried to rearrange this layered system for simple cut-out style animation Figure 1(b) shows our layered system Layer 4 is camera, Layer 2 Puppets(With PMagnets), Layer 1 screen, Layer 0 controller (controlling the motion of Puppets (pasted with PMagnets)) with CMagnets.

4. PROPOSED WORK

Our system follows two steps. Puppet design and deciding number of P(C)Magnets required for his character is the first step in our system. Generating animation sequence is the second step. In this step animator control's the motion of characters using CMagnets. A background subtraction algorithm is required in this step.

4.1` Puppet Design

As shown in the Figure 2(a) animator can design the physical animation character for his animation sequence.

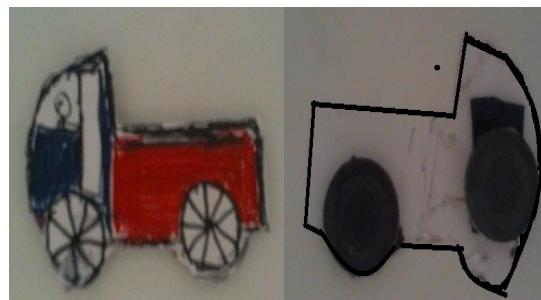


FIGURE 2: (a) (b)

When animator gets completed with their designing part, they have to decide what kind of motion does their character or part of the character will have. This motion decides number of P(C)Magnets required. For example, the character may have motion such that it does not include any rotation so it requires one P(C)Magnet. In Figure 2(a) the character will have some rotational motion. So, it requires two P (C) Magnets Figure 2(b) is the back side of the physical puppet attached with 2 PMagnets which are used to control the motion using other CMagnets. Between

PMagnets and CMagnets there will be a cardboard (roof of our interface) of small thickness, so that CMagnets can control PMagnets using magnetic field.

4.2 Animating

In this step animator generates the animation sequence using the Puppets. For this step we have

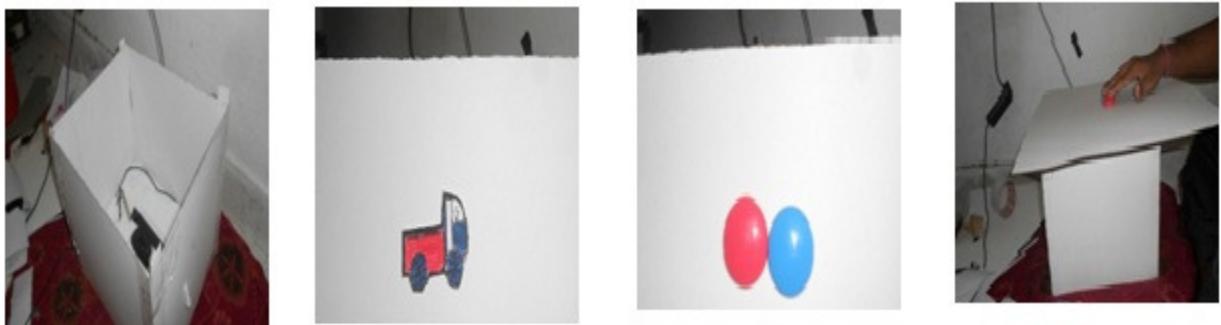


FIGURE 3: (a)

(b)

(c)

(d)

designed a small box using a white cardboard sheet which is used as a puppet theatre. As shown in Figure 3 (a) this box creates a closed environment, and has its own light source providing uniform illumination so that the lighting cannot be affected by the outside environment. A camera is placed at bottom of the box, the roof which is a white cardboard sheet acts as an interface. The camera is projected on inside of the roof where it records the motion of physical characters. The other side of the roof acts as a medium for an animator for controlling PMagnets with CMagnets. Figure 3(b) a 4 wheeler puppet is placed on inside of the roof, Figure 3(c) is the other side of the roof with CMagnets (white board magnets). Figure 3(d), user controlling the motion of the puppet.

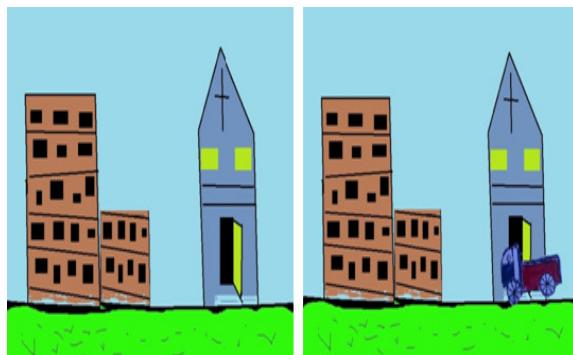


FIGURE 4: (a)

(b)

If the animator wants to render onto a new background than the foreground must be extracted. There are methods proposed in [5],[6] for extracting foreground from a scene where background information is known and it is same for all frames. In our system the background is constant, so a simple background subtraction algorithm is sufficient. Figure 4(a) is the new background, Figure 4(b) is the snapshot of an animation sequence after rendering Figure. 2(a) onto Figure 4(a). Instead of rendering onto computer generated background, the animator can use his hand drawn background, so no background replacement is needed.

5. RESULTS

Figure 5 are the snapshots of an animation sequence using Figure 2(a) as puppet. We haven't done much animation, but we believe this interface provides a simple and cost effective way of producing simple animation sequences.

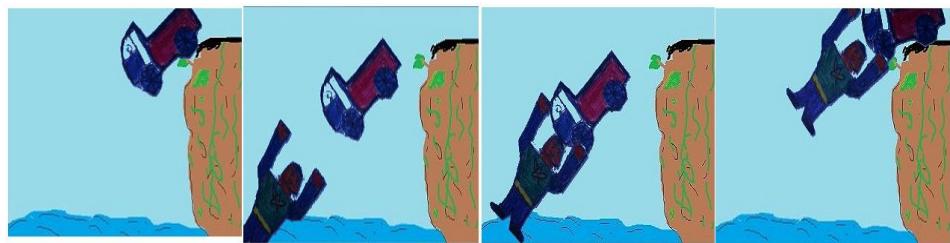


FIGURE 5

6. LIMITATIONS

Articulated animation is not possible with this simple approach. Because there is no information about the relation between character parts which are controlled by the animator. We know only foreground and background. Another drawback is when two characters come closer or even overlap (when there is a situation like cross over) the magnets may attract or may ripple causing a distortion in the animation sequence.

7. CONCLUSION AND FUTURE WORK

We presented a new interface which uses magnets (PMagnets and CMagnets) for controlling animation. Up to the best of our knowledge this is the first system which is designed by using the concept of Magnets. The advantage of using our system is that it can be designed in a cost effective manner. Our system have several advantages over existing system [1] like we are not using any object tracking or object detection algorithm which is computationally expensive. Computational cost never increases as the number of characters increases. There is no need to maintain a separate database. Partial visibility of a physical character is possible and simple. Even size of physical character (puppet) is not an issue in our approach. There is no limit on the speed at which a physical character is moved. In future we will try to work on the articulated animation of physical puppets. We will also try to add effects discussed in [1] in our system.

8. REFERENCES

- [1] Barnes, Connelly, David E. Jacobs, Jason Sanders, Dan B. Goldman, Szymon Rusinkiewicz, Adam Finkelstein, and Maneesh Agrawala. "Video puppetry: a performative interface for cut out animation." In ACM Transactions on Graphics (TOG), vol. 27, no. 5, p. 124. ACM, 2008.
- [2] Lowe, David G. "Object recognition from local scale-invariant features." In Computer vision, 1999. The proceedings of the seventh IEEE international conference on, vol. 2, pp. 1150-1157. Ieee, 1999.
- [3] Sinha, Tomasi, Carlo, and Takeo Kanade. "Detection and tracking of point features". School of Computer Science, Carnegie Mellon Univ., 1991.
- [4] Sohn, Eisung, and Yoon-Chul Choy. "Sketch-n-Stretch: sketching animations using cutouts." Computer Graphics and Applications, IEEE 32.3 (2012): 59-69.
- [5] Qian, Richard J., and M. Ibrahim Sezan. "Video background replacement without a blue screen." In Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on, vol. 4, pp. 143-146. IEEE, 1999.
- [6] Horprasert, Thanarat, David Harwood, and Larry S. Davis. "A statistical approach for real-time robust background subtraction and shadow detection." In IEEE ICCV, vol. 99, pp. 1-19. 1999.