applied sciences

MDPI

Article

An Interactive and Personalized Erasure Animation System for a Large Group of Participants

Hua Wang [1], Xiaoyu He [1] and Mingge Pan [2,*]

[1] School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450000, China; wanghua@zzuli.edu.cn (H.W.); hexiaoyu_zzuli@163.com (X.H.)

[2] School of Arts and Design, Zhengzhou University of Light Industry, Zhengzhou 450000, China

[*] Correspondence: pmg-1@163.com; Tel.: +86-185-3853-0628

Received: 3 October 2019; Accepted: 14 October 2019; Published: 18 October 2019

Abstract: This paper introduces a system to realize interactive and personalized erasure animations by using mobile terminals, a shared display terminal, and a database server for a large group of participants. In the system, participants shake their mobile terminals with their hands. Their shaking data are captured by the database server. Then there are immersive and somatosensory erasure animations on the shared display terminal according to the participants' shaking data in the database server. The system is implemented by a data preprocessing module and an interactive erasure animation module. The former is mainly responsible for the cleaning and semantic standardization of the personalized erasure shape data. The latter realizes the interactive erasure animation, which involves shaking the mobile terminal, visualizations of the erasure animation on the shared display terminal, and dynamic and personalized data editing. The experimental results show that the system can realize various styles of personalized erasure animation and can respond to more than 2000 shaking actions simultaneously and present the corresponding erasure animations on the shared display terminal in real time.

Keywords: interactive; personalized; erasure animations; mobile terminals; shared display terminal; a large group of participants

1. Introduction

With the rapid development of virtual reality technology, mixed reality technology, and human–computer interaction technology, an increasing number of businesses are working from various angles to realize erasure animations in mixed reality so that their users' participations can be improved. Some examples are Fruit Ninja and The Swords. In these systems, limits exist in the number of face-to-face participants and the unnatural interactive devices. How to realize rich somatosensory interactive erasure animations for a large group of face-to-face participants by some popular interactive devices has great influence on the participations.

The technologies of image matting and interactive erasing are working to develop users' participations in erasure animations. Existing image matting methods are mainly used for two-dimensional (2D) scenes and their mask images and background images are all 2D data. They do not support customizations of three-dimensional (3D) scenes [1,2]. Some 3D simulation software systems, such as Unity3D and Unreal Engine 4, use their powerful shader functions to perform texture transparency blending on the image which contains the information of erasure shapes and background scenes, and thus reveal the scenes. The erasure can be performed in every position. However, the visualization of multiple erasure actions simultaneously requires to create and load multiple shaders, which is very time-consuming. Therefore, they are not suitable for a larger number of participants. In the field of interactive erasing technology, some sensor-based, and vision-based gesture recognition methods are used for interactions [3,4], for example, virtual reality glasses, Kinect, cameras. Some of them are not easy in implementation and some of them suffer from serious occlusion of a larger number participants.

To tackle the aforementioned challenges, we design a system to realize interactive and personalized erasure animations by using mobile terminals, a shared display terminal and a database server. The system is implemented by a data preprocessing module and an interactive erasure animation module (Figure 1). The data preprocessing module is mainly responsible for preprocessing the input erasure shape data, including cleaning the personalized shape data and semantic standardizations. The interactive erasure animation module consists of three parts: shaking mobile terminals, visualization of the erasure animations in the shared display terminal, and dynamic and personalized data editing in the database server. In our system, users shake their mobile terminals continuously and simultaneously with their hands, and their valid shaking data are captured and saved in the database server. Then the shared display terminal accesses the database server and shows visualizations of real-time erasure animations according to the data. Note that the system can only show a continuous animation for many shakings: one valid shaking occurs when a user shakes his mobile terminal, then the shared display shows an erasure shape; another valid shaking occurs when another user shakes his mobile terminal, then the shared display shows two erasure shapes; more valid shakings occur and then the shared display shows more erasure shapes.

The main contributions of our system are as follows:

- We introduce a novel interactive erasure animation system based on a shared display terminal and mobile terminals (mobile phone/tablet computer), the implementation of which is very easy for a larger number of participants.
- In our system, the shared display terminal can respond to a larger number of shaking actions from participants in real time and show an immersive and somatosensory erasure animation, the erasure shapes which are highly consistent with the participants' shaking actions (position, travel distance, angle, etc.).
- Our system supports real-time personalized erasure animations. The erasure shape, mask data, scene data, and so on can be customized on our backend management platform. The scale, rotation, and translation of each shape are personalized and determined by the corresponding shaking action.

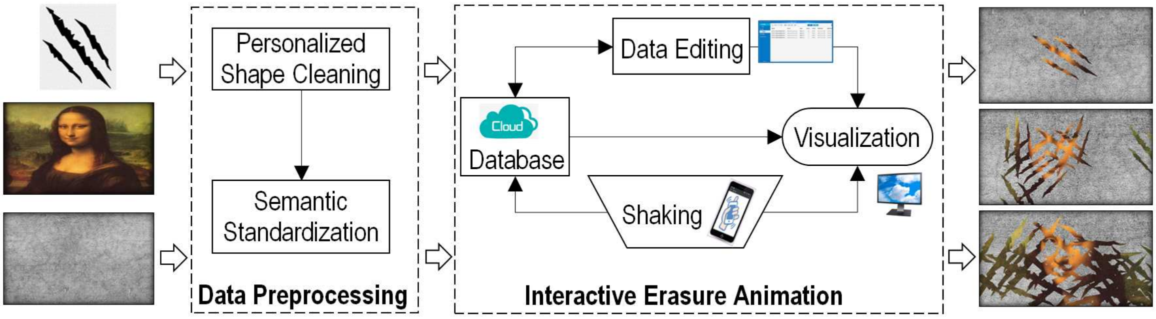

Figure 1. The framework of our system.

2. Related Work

Our work has a cross-section of research in the fields of image matting and interaction technique. In this section, we first briefly review the previous work in image matting and then introduce the interaction in mixed reality.

2.1. Image Matting

With the development of digital multimedia technologies, image matting has gained increasing interest from both academic and industrial communities. Existing image matting methods are simply divided into three types: sampling-based methods, propagation-based methods, and hybrid methods that combine the sampling-based methods and propagation-based methods.

The basic idea of sampling-based methods [5–7] is as follows: It is assumed that the pixels of an unknown area can be estimated according to the nearby foreground color and background color. The value of the transparency can be calculated using these known sample pixels, and thus the pixels of the unknown area are partitioned. For example, the probability and statistics method [7], the Bayes image matting algorithm [5]. The quality of image matting yielded by the sampling-based method depends heavily on the transparency of pixels of the unknown area. Propagation-based methods [8–13] make assumptions based on the statistical properties of the image and describe the relationship between neighboring pixels based on color, spatial position, and so on. Closed-form matting [10], Poisson matting [13], and KNN matting [14] are some famous propagation-based methods. To better combine the advantages of the above two methods, some researchers have proposed hybrid method that combines the sampling-based and propagation-based methods [15–17], such as the shared-matting method [15], the global matting method [16], and robust mapping method [17].

The purpose of image matting is to precisely extract the foreground objects with arbitrary shapes from an image or a video frame [6]. Both of their foreground data and background data are 2D raster data. They cannot be used for 3D interactive erasure animations, which can provide immersive and somatosensory animations.

2.2. *Interaction in Mixed Reality*

Mixed reality content can be interacted with via different devices, such as mobile terminals, virtual reality glasses, and Kinect [18]. Simon et al. [4] introduced an interaction paradigm for co-located collaboration in large projection-based display systems. Their system provides a shared experience and collaboration for a large number of users by handling the same shared display terminal, which inspires us to deal with large number of participants. On the one hand, the system was designed for tracking controls in the shared display terminal and can not be used for interactive erasures. On the other hand, our interactive erasures can be realized by hand gestures, which play an important role in our day to day communication.

2.2.1. Hand Gesture Recognition

Hand gestures play a natural and intuitive communication model for all human interactive erasures. The ability of a computer or any processing system to understand the meaning of these gestures is referred to the hand gesture recognitions. According to the difference of input devices, there are two methods for hand gesture recognitions: the sensor-based method and the vision-based method.

In the sensor-based method, a user wears a sensor terminal or a colored glove, which serves as an interface to communicate with the computer. The method makes it easy to collect hand movement data and can give accurate results. However, it is cumbersome and unnatural. It hinders the ease and naturalness with which the user can interact with the computer [4,19]. So, the key is to get accurate hand gesture results by some light and portable terminals. The vision-based method overcomes the drawback of the sensor-based method as it serves as a natural means of interaction [19]. In this method, the movement of the hand is recorded by video cameras. This input video is decomposed into a set of features [20]. The method adopts computer vision and machine learning algorithms for recognizing the features. However, obtaining highly accurate results is a challenging task for the vision-based method [21], especially when large number of participants and serious occlusions occur.

Here we introduce a database server to bridge the mobile terminals and the shared display terminal to realize interactive erasure animations, the implementation of which is very easy for a larger number of participants. There are no occlusion problems in our system.

2.2.2. 3D Animation

The key problem in the mixed reality interaction is the animation efficiency. Researchers in the fields of computer graphics have proposed lots of methods to improve animation efficiency, such as point-based method, image-based method, and the Level Of Detail (LOD) method [22–24]. The LOD method [22] decreases the complexity of a 3D model representation as it moves away from the viewer. The reduced visual quality of the model is often unnoticed because of the small effect on object appear.

In this study, we combined the ideas of image matting, LOD method, and the hand gesture interaction to realize the interactive erasure animations.

3. Data Preprocessing

Data preprocessing refers to preprocessing the erasure shape data to reduce the computing complexity during the following erase animations.

In our system, the input erasure shape data is image data. The purpose of data preprocessing is to extract the erasure shape from the image data and semanticize it, so that it can be used conveniently in erasure animations. The input image data which contains the erasure shape data must be a noise-free grayscale image (Two colors in the image, one is the foreground color and the other is the background color), which is created by an image matting method [6].

We transform the above erasure shape data to the vector data with a semantic standardization to make a template for the subsequent real-time animation process. The process mainly includes the following three steps (Figure 2).

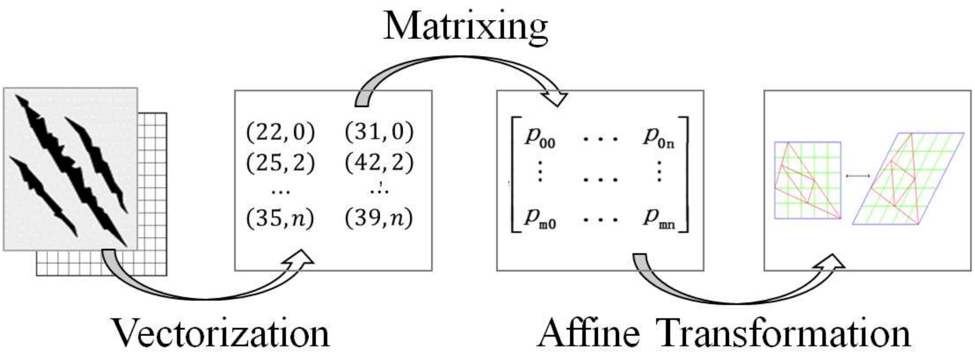

Figure 2. The semantic standardization process.

3.1. Vectorization of the Erasure Shape Data

The erasure shape is usually irregular, and it may even have multiple connection graphs. Inspired by the idea of the scan line algorithm, we obtain the vector data of the erasure shape boundaries from the image raster data through a row-by-row scanning process to realize the vectorization of the image data. The process consists of three steps: finding the intersections, sorting the intersections, and pairing the intersections.

Finding the intersections: The scan lines are intersected with the edges of the erasure shape during row-by-row scanning and then the coordinates of each intersection points can be calculated (Figure 3).

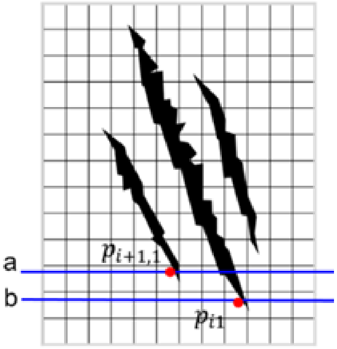

Figure 3. The process of row-by-row scanning.

Sorting the intersections: We sort the intersection points according to their x-coordinates.

Pairing the intersections: We pair the sorted points in the following way according to their x-coordinates: 1st and 2nd, 3rd, and 4th, and so on. Each pair of intersection points represents the left and right boundary of the erasure shape.

We then obtain a vector data set P^1 represented by a 2D coordinate sequence to describe the boundary of the erase shape:

$$P^1 = \left\{ \begin{matrix} (x_{11}\ y_1), & (x_{12}\ y_1), & \ldots, & (x_{1,2s_1}\ y_1), \\ (x_{21}\ y_2), & (x_{22}\ y_2), & \ldots, & (x_{2,2s_2}\ y_2), \\ \cdots & \cdots & & \cdots \\ (x_{t1}\ y_t), & (x_{t2}\ y_t), & \ldots, & (x_{t,2s_t}\ y_t) \end{matrix} \right\}$$

Here $(x_{i,2j}\ y_i)$ $(i \in [1,t], j \in [1,s_i])$ represents the left and right boundary of the erasure shape. t is the number of scan lines, $2s_i$ represents the number of intersection points between the i-th $(i \in [1,t])$ scan line and the boundary of the erasure shape.

Let $p_{ij} = (x_{ij}\ y_i)$ $(i \in [1,t], j \in [1, 2s_i])$, then

$$P^1 = \left\{ \begin{matrix} p_{11}, & p_{12}, & \ldots, & p_{1,2s_1}, \\ p_{21}, & p_{22}, & \ldots, & p_{2,2s_2}, \\ \cdots & \cdots & & \cdots \\ p_{t1}, & p_{t2}, & \ldots, & p_{t,2s_t} \end{matrix} \right\} \tag{1}$$

Let $\{p_{i1}, p_{i2}, \ldots, p_{i,2s_i}\}$ $(i \in [1,t])$ be the i-th line. It is easy to prove that the number of elements in each line in Equation (1) is different, that is, $s_i \neq s_j, (i \neq j,\ \ i,j \in [1,t])$. The reason is that the erasure shape is irregularity and complexity (for example, in Figure 3, the number of elements obtained by the scan line a is 4, and the number is 2 by the scan line b).

To facilitate computations of the subsequent operations, it is necessary to use a matrix to depict P_1.

3.2. Matrixing of the Vector Data

Let the matrix of P_1 be $P = P_{m \times n} = (p_{ij})_{m \times n}$, then

$$m = t$$
$$n = max\{2s_i \mid i \in [1,m]\}$$

For $\forall i \in [1,m]$, if $j \in (2s_i, n]$, let $p_{ij} = (-10,000, -10,000)$.

It is easy to see that $p_{i,j}$ and $p_{i+1,j}$ $(i \in [1, m-1],\ j \in [1,n])$ are never the left boundary of a connection graph (Figure 3). In order to realize an efficient searching in animations, we use position-shifting operations to ensure that $p_{i,j}$ and $p_{i+1,j}$ $(i \in [1, m-1],\ j \in [1,n])$ are the left boundary of one connection graph. Algorithm 1 is as follows:

Algorithm 1: Position-shifting operations for P.

```
$For(i = 2; i <= n; i++)$
    $For(j = 1; j < s_i; j++)$
      $If(x_{i-1,2j-1} > x_{i,2j})$
          $For(h = i-1; h >= 1; h--)$
              $For(k = 2s_i; k >= 2j+2; k--)$
                  $x_{h,k} = x_{h,k-2}; y_{h,k} = y_{h,k-2};$
              $End$
              $x_{h,k} = -10,000; y_{h,k} = -10,000;$
              $x_{h,k-1} = -10,000; y_{h,k-1} = -10,000;$
          $End$
      $End$
      $Else$
          $For(k = 2s_i; k >= 2j+2; k--)$
              $x_{i,k} = x_{i,k-2}; y_{i,k} = y_{i,k-2};$
          $End$
          $x_{i,k} = -10,000; y_{i,k} = -10,000;$
          $x_{i,k-1} = -10,000; y_{i,k-1} = -10,000;$
      $End$
    $End$
$End$
```

3.3. *Affine Transformation of the Matrix Data*

It is difficult to use the above shape data to realize an erasure animation. The reasons are as follows:

- The area of the erasure shape is larger than the area of the shared display terminal. We have to shrink the size of the shape to make sure that it can be drawn on the shared display terminal.
- The coordinate values of the erasure shape are too large/small. We have to translate all of the coordinates of the erasure shape to make sure it can be drawn on the shared display terminal.

Here we use the following affine transformations to solve above problems:
Scaling transformation:

$$P_1 = PA \tag{2}$$

Here A is a $n \times n$ diagonal matrix, in which principle diagonal elements are k. k denotes the scaling coefficient:

$$k = \tau max\{\frac{w}{max\{x_{i,2s_i}|i\in[1,m]\}}, \frac{h}{y_m}\}$$

Here w and h are the length and width of the shared display terminal, respectively. τ is constant and $\tau \leq 1$. If $\tau \equiv 1$, the area of the erasure shape is nearly same as the area of the shared display terminal. The area of the erasure shape is smaller than the area of the shared display terminal if the value of τ is smaller.

Translating transformation:

$$P_2 = [P_1 \;\; 1]B \tag{3}$$

Here B is a translation matrix,

$$B = B_{(n+1)\times n} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & \vdots & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & 0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & 1 \\ p^0 & p^0 & p^0 & \cdots & p^0 \end{pmatrix}$$

Here $p^0 = (x^0\ y^0)$. x^0 is the translation distance on the x-coordinate, y^0 is the translation distance on the y-coordinate, $x^0 = \frac{\sum_{i=1}^{n}\sum_{j=0}^{2s_i} x_{ij}}{2s_t * n}$, and $y^0 = \frac{\sum_{i=1}^{n} y_i}{n}$.

After the above translating transformation process, the center position of the erasure shape is near the center of the shared display terminal. P_2 is the template of the erasure shape for the subsequent real-time animation process.

4. Interactive Erasure Animation

In this section, we introduce an interactive erasure animation process based on the above-mentioned erasure shape data.

To conduct a rich somatosensory interactive erasure animation process for all participants, one shared display terminal is used for all of them. These participants continuously and simultaneously shake their mobile terminals with their hands. Note that all popular mobile terminals, such as iphone mobile phones, android based mobile phones, tablet computers, can be directly used in our system. Then our system shows visualizations of real-time erasure animations on the shared display terminal (for example, a large screen), the shapes of which are highly consistent with the participants' shaking actions: the rotation angle of the erasure shape is consistent with the angle that the mobile terminal is turned during shaking, and the size of the erasure shape is proportional to the displacement of the mobile terminal, etc.

Now we have to process multiple sets of shaking data in parallel in the shared display terminal, in which we also have to do time-consuming erasure animations. There are no ready sensors to capture the shaking data and transmit them to the shared display terminal. The vision-based method [19] cannot be used because there are serious occlusions among participants. Here we introduce a database server to bridge the mobile terminals and the shared display terminal. Participants firstly connect their mobile terminals to the database server. The database server then captures and saves their shaking data through the accelerometers of their mobile terminals. The shared display terminal accesses the database server in real time to show visualizations of the interactive erasure animations.

From a user's point, there are three steps to realize an interactive erasure animation process: shaking mobile terminals, visualization of the erasure animations in the shared display terminal, and dynamic and personalized data editing in the database server.

4.1. Shaking the Mobile Terminal

Participants can apply the interaction permission via mobile terminals by submitting their personal information to the database server through scanning a QR code. After their permissions are successfully obtained, connections between their mobile terminals and the database server are built. They shake their mobile terminals using their hands. The database server captures all of their shaking actions through the accelerometers of their mobile terminals in real time and then judges the validity of the shaking actions. The valid shaking data (shaking angles, shaking amplitudes, and shaking positions) are saved into the database for erasure animations (Figure 4).

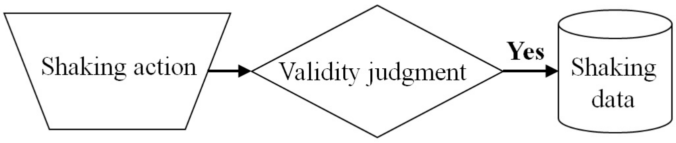

Figure 4. How to capture and save shaking data in our system.

A valid shaking action is defined as follows:

- The time interval between two adjacent shaking actions for a user is larger than the interval threshold *T*_threshold. In our system, *T*_threshold = 0.1s.
- The shaking speed is larger than the speed threshold *v*_threshold. In our system, *v*_threshold = 2 m/s.

Algorithm 2, about how to capture the shaking action of a user and judge its validity, is as follows:

Algorithm 2: How to capture the shaking action of a user and judge its validity.

```
$diffTime = curTime - lastTime;$
$If(diffTime >= deltaT)$
    $dis = \sqrt{(x - last\_x)^2 + (z - last\_z)^2};$
    $If(dis/diffTime >= v_{threshold})$
        $\theta = getRotaAng(x - last\_x, z - last\_z);$
        $addMSG(dis, \theta);$
        $lastTime = curTime;$
    $End$
    $last\_x = x;$
    $last\_y = y;$
    $last\_z = z;$
$End$
```

Here curTime is the current time and lastTime is the starting time of the last valid shaking action. (x, y, z) and (last_x, last_y, last_z) are the coordinates of the center of gravity of the mobile template at the moment of curTime and lastTime. *dis* represents the amplitude of the shaking action. $getRotaAng()$ returns the angle value of the 2D vector, which is the angle value of the vector (x-last_ x, z-last_ z). We use $addMSG()$ to save the valid shaking action data into the database. The coordinate system of the above coordinates is described as follows:

- The origin is the central point of the display template.
- The x-axis direction and the z-axis direction are parallel to the horizontal direction and vertical direction of the display template, respectively.
- The coordinate system complies with the right-hand rule.

4.2. Visualization of the Erasure Animations

Our system accesses the shaking data from the database server and shows a real-time visualization of the erasure animations on the shared display terminal.

In our system, we use the orthographic projection theory [25] and the LOD method [22] to realize the visualization of the erasure animations. Specifically, we draw the environment in the direction of the observer's sight in the 3D space created by orthographic projection and then draw a mask image in front of it. In the erasure animation, let the alpha channel of the erasure area be 0 (i.e., completely transparent), and the alpha channel of the rest area be 1 (i.e., completely opaque). The erasure area is the set of erasure shapes, which is created by all users participating in the interaction. Let the direction of the observer's sight in the 3D space be the positive direction of the z-axis and the position where the mask image draws be z_0.

Let the total number of valid shaking actions be N in the database and the projection coordinates of the shaking position of the kth valid shaking action in N be $(w_k, h_k), k \in [1, N]$. Algorithm 3 for the process of calculating the erasure area R is as follows:

Algorithm 3: The process of calculating the erasure area R.

```
$R = \varnothing$
$For(k = 1; k <= N; k++)$
    $\lambda = \lfloor dis_k / thre\_dis \rfloor;$
    $\cos\theta = 1 - \theta_k^2 * 0.5;$
    $\sin\theta = \theta_k * (1 + \theta_k^2 * 0.5 * 0.167);$
    $For(j = 1; j < 2_{S_i}; j = j + 2)$
        $For(i = 1; i <= n; i = i + L_{LOD})$
            $x_1 = w_k + \lambda(x_{i,j}\cos\theta - y_{i,j}\sin\theta);$
            $y_1 = h_k + \lambda(y_{i,j}\cos\theta - x_{i,j}\sin\theta);$
            $x_2 = w_k + \lambda(x_{i,j+1}\cos\theta - y_{i,j+1}\sin\theta);$
            $y_2 = h_k + \lambda(y_{i,j+1}\cos\theta - x_{i,j+1}\sin\theta);$
            $x_3 = w_k + \lambda(x_{i+L_{LOD},j+1}\cos\theta - y_{i+L_{LOD},j+1}\sin\theta);$
            $y_3 = h_k + \lambda(y_{i+L_{LOD},j+1}\cos\theta - x_{i+L_{LOD},j+1}\sin\theta);$
            $x_4 = w_k + \lambda(x_{i+L_{LOD},j}\cos\theta - y_{i+L_{LOD},j}\sin\theta);$
            $y_4 = h_k + \lambda(y_{i+L_{LOD},j}\cos\theta - x_{i+L_{LOD},j}\sin\theta);$
            $R = R \cup P\_Rect(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4);$
        $End$
    $End$
$End$
```

Here dis_k is the shaking amplitude, which is calculated in Section 4.1. $thre\_dis$ is the amplitude threshold, which is 0.2 m in our system. L_{LOD} is a positive integer and is determined by the distance between the shared display terminal and the mobile terminal. θ_k is the shaking angle, which is calculated in Section 4.1. $\cos\theta$ and $\sin\theta$ are the first-order Taylor expansions of $\cos(\theta_k)$ and $\sin(\theta_k)$, respectively. $(x_{i,j}\ y_{i,j}), (i \in [1, N], j \in [1, 2s_i])$ is the boundary of the erasure shape, which is calculated in Section 3. $P\_Rect(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$ represents the polygon defined by coordinates $(x_1, y_1), (x_2, y_2), (x_3, y_3)$, and (x_4, y_4).

4.3. Dynamic and Personalized Data Editing

A lot of data have to be used in each erasure animation frame in our system, such as scenes, mask data, templates of erasure shapes, users, shaking data. We save the data in the database server for better management. We develop a backend management platform to manage and customize these data. In our system, we can customize the following data in real time: scene data, mask image, erasure shape, erasure sound, and the interaction permissions of each user, and so on.

The scene data can be 2D vector data, images, videos, and 3D scenarios. We can modify the scene data information in real time, and the shared display terminal responds to the data change in real time. The mask image is an image. All common image file formats (JPG, PNG, BMP, TIFF, etc.) can be used in our system directly. The erasure shape data are obtained in Section 3. We can change the erasure shape by modifying the erasure template data. If we choose the erasure template that has been made in Section 3, the shared display terminal responds to the data change instantly. If we choose a new erasure shape, we should firstly create the erasure template according to the method described in Section 3. The erasure sound shows the sound in the shared display terminal when a valid shaking action occurs in our system. When the erasure sound is changed, the shared display terminal will show the new sound when a user creates a valid shaking action. We can also decide the interaction permissions of each user. The shaking action of users without the interaction permission will not yield any erasure animation on the shared display terminal.

5. Results

We introduce a system to realize interactive and personalized erasure animations. In our experiments, a large screen was shared by users. These users stood (or sat down) in front of the screen. There are serious occlusions among these users. They applied the interaction permission firstly and then shook their smartphones using their hands. What follows are results about kinds of personalization/interaction and performance analysis.

5.1. Kinds of Personalization and Interaction

We can customize erasure shapes, scene data, and so on. In addition, there are high interactions between the shaking actions of users and the visualization of erasure animations. Different shaking actions introduce different erasure animations.

5.1.1. Erasure Shape

Almost all kinds of shapes can be used in our system, including single connection graphs, multiple connection graphs, convex polygons, concave polygons, etc. Figure 5 shows some erasure animations when different shapes (including ellipse, pentagram, flag, and multi-connection concave polygon) were used (The outcome is highlighted in the supplementary video (http://v.qq.com/vplus/f9ca5d28f9c7c8696f356522325b0902)). Note that if the shape has been made into an erasure template, we can use it directly in erasure animations. Otherwise, we should firstly create the erasure template according to the method described in Section 3.

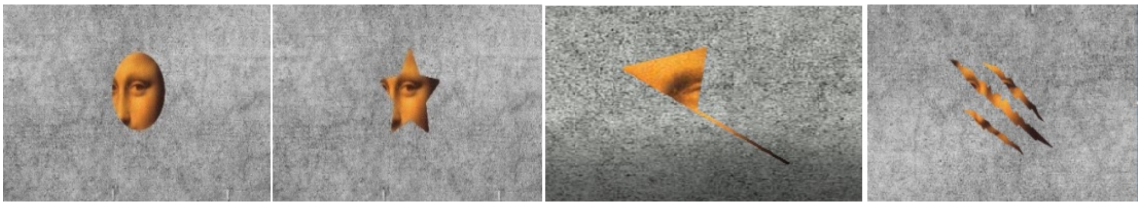

Figure 5. Erasure animations with different erasure shapes.

5.1.2. Scene Data and Mask Image

We can update the scene data and mask image in real time in erasure animations. The scene data can be videos, 3D scenarios, 2D images, 2D vector data, etc. The file formats of mask image data can be JPG, PNG, BMP, TIFF, etc. There is no constraint for the mask image features. Figure 6 shows some visualizations of the erasure animations with a 3D scenario. Figure 7 shows visualizations with a video/2D image scene data and different mask image data. As shown, our system is capable of showing erasure animations no matter what types of scene data and mask image data are used, which ensures a large scope of application.

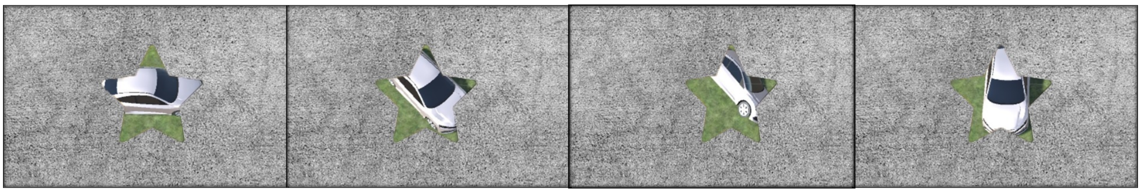

Figure 6. Erasure animations with a 3D scene data.

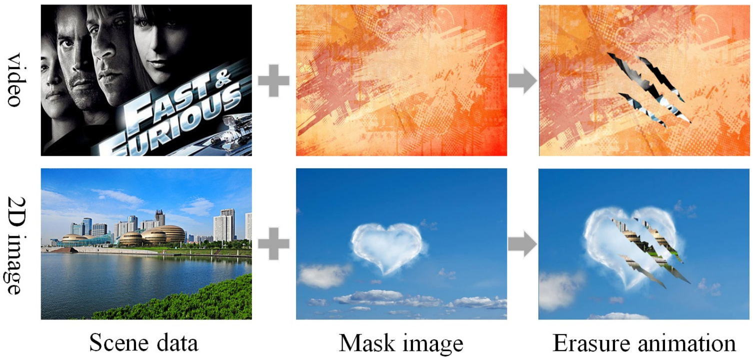

Figure 7. Erasure animations with different scene data and mask image data.

5.1.3. Erasure Animation

In Section 4.2, we show how to realize a visualization of the erasure animations on the display terminal. One erasure shape corresponds to one valid shaking action. The position of the erasure shape for the kth valid shaking action is (w_k, h_k), which is obtained from the database server. The size and the angle of the erasure shape for the kth valid shaking action are determined by dis_k (the amplitude of the shaking action) and θ_k(the rotation angle of the shaking action), which are also obtained from the database server.

Figure 8 shows some erasure animations with different rotation angles, different positions, different amplitudes, and different times of shaking actions. The left-most subfigure of Figure 8 shows the erasure animation of a valid shaking action ($N = 1$, $w_1 = 0, h_1 = 0$, $dis_1 = 0.1$, $\theta_1 = \pi/4$ in Section 4.2). Figure 8a shows the erasure animation of one valid shaking action with a different rotation angle ($\theta_1 = -\pi/4$). Figure 8b shows the erasure animation of one valid shaking action at a different spatial position ($w_1 = 1200, h_1 = 0$). Figure 8c shows the erasure animation of a valid shaking action with a different amplitude ($dis_1 = 0.02$). Figure 9 shows some snapshots of the erasure animation as the number of valid shakings increases. More valid shaking actions occur and then the shared display shows more erasures shapes. The shapes (scale, rotation, translation, and number) are highly consistent with the users' shaking actions, which indicate natural interactions between users and the display terminal.

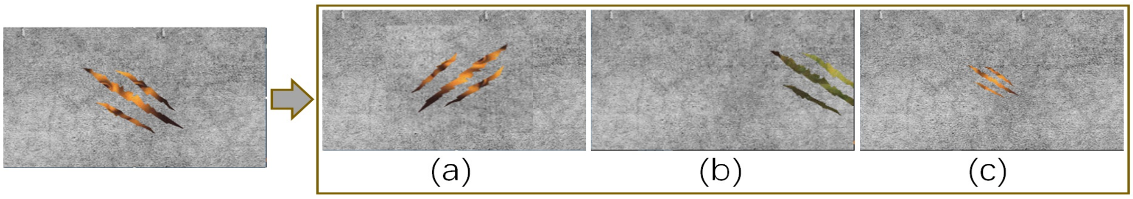

Figure 8. Erasure animations with different rotation angles, different positions, different amplitudes, and different times of shaking actions.

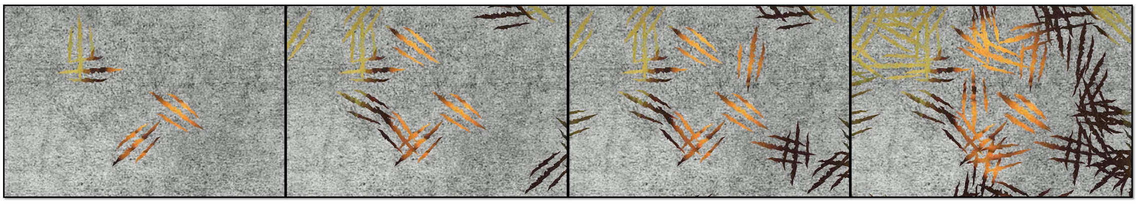

Figure 9. Some snapshots of the erasure animation as the number of valid shaking actions increases.

5.2. Performance Analysis

We evaluate the performance of our system from two parts: the efficiency of data preprocessing and the efficiency of the erasure animations. These results were collected on an Intel(R) Core 8 Xeon (TM) i7-4790 CPU with a 3.60 GHz processor and 4.0 GB of RAM.

5.2.1. Efficiency of Data Preprocessing

The efficiency of data preprocessing is determined by the size of the input image and the size of the erasure shape in the image.

Firstly, we zoomed in on a 256 × 256 image, the circumference of the erasure shape in which is 980 pixels. Figure 10 shows the relationship between the size of the image and the total data preprocessing time. It shows that the data preprocessing time increased as the image size increases. The reason is that the size of the image affects the whole data preprocessing process, including the vectorization of the erasure shape data, the matrixing of the vector data, and the affine transformation of the matrix data. Note that the side length of the image is usually about 1000 pixels. The data preprocessing process can be completed in a matter of seconds, which is completely acceptable to users.

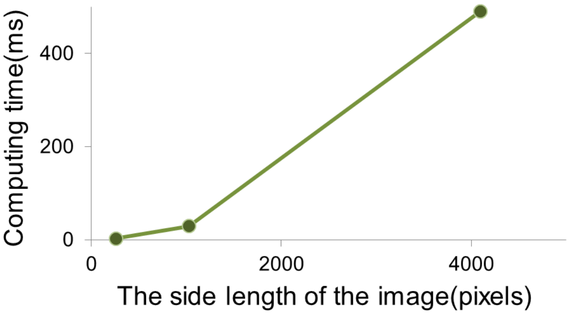

Figure 10. The relationship between the size of the image and the total data preprocessing time.

Secondly, we changed the circumference of an erasure shape in a 1024 × 1024 image. Figure 11 shows the relationship between the circumference of the erasure shape and the total data preprocessing time. It shows that the data preprocessing time increased as the circumference of the erasure shape increases. The reason is that the size of the erasure shape affects the matrixing process of the vector data and the affine transformation process of the matrix data.

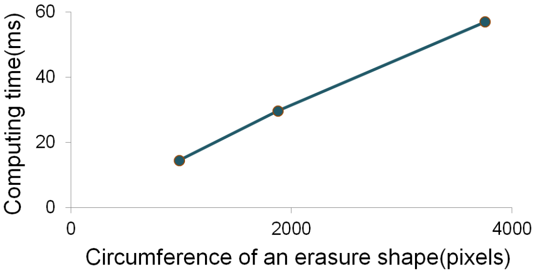

Figure 11. The relationship between the circumference of the erasure shape and the total data preprocessing time.

5.2.2. Efficiency of Erasure Animations

We showed our system in a meeting with new students to evaluate the performance of our erasure animations. There were about 1500 students and all of them had obtained the interaction permissions. They shook their mobile terminals using their hands. The largest number of valid shaking actions captured by our system is 620 in a short time interval (50 ms). Our system can respond to those actions simultaneously and present the corresponding erasure animations on the shared display terminal in real time.

We also used our system to a simulation experiment to evaluate the relationship between the number of shaking actions and the total computing time of the erasure animations. In the experiment, we used an erase shape with a perimeter of 1192 pixels. The experimental results are shown in Figure 12. It shows that our system could respond to more than 2000 shaking actions. Note that in this experiment, we did not take into account of the duration of shaking data transmissions from mobile terminals to our database server.

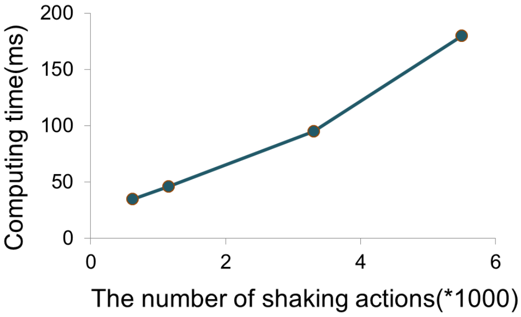

Figure 12. The relationship between the circumference of the erasure shape and the total data preprocessing time.

6. Conclusions

We introduced a novel and easy-to-use interactive and personalized erasure animation system by using mobile terminals, a shared display terminal, and a database server for a large group of participants. We do not need to face the occlusion problems and the unnatural sensor terminals in our system. Users apply for the interaction permissions and then shake their mobile terminals. Their valid shaking actions are captured and saved in the database server of our system. The shared display terminal then shows 3D visualization of erasure animations according to the shaking actions. The experimental results show that there are natural interactions between the user and the shared display terminal. Our system can respond to more than 2000 shaking actions simultaneously and present the corresponding erasure animations on the shared display terminal in real time. In the future, we would like to introduce a continuous animation for each erasure shape to realize a richer somatosensory interactive erasure animation. We also would like to make the erasure shape consistent with the user's shaking rather than the erasure template in our animation.

Supplementary Materials: Supplementary Materials are available at http://www.mdpi.com/2076-3417/9/20/4426/s1.

Author Contributions: Conceptualization, H.W.; writing–original draft preparation, H.W.; writing–review and editing, X.H.; visualization M.P.

Funding: This research was funded by the National Natural Science Foundation of China [Grant no. 61602425].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, J.; Agrawala, M.; Cohen, M.F. Soft scissors: An interactive tool for realtime high quality matting. In Proceedings of the International Conference on Computer Graphics and Interactive Techniques, San Diego, CA, USA, 5–9 August 2007; p. 9.
2. Luo, C.; Yang, W.; Huang, P.; Zhou, J. Overview of image mapping algorithm based on affine method. *J. Comput.-Aided Des. Comput. Graph.* **2016**, *28*, 678–693.
3. Ullah, S.; Khan, D.; Ur Rahman, S.; Alam, A. Marker based interactive writing board for primary level education. *Pak. J. Sci.* **2016**, *68*, 366.

4. Simon, A. First-person experience and usability of co-located interaction in a projection-based virtual environment. In Proceedings of the ACM Symposium on Virtual Reality Software and Technology, Monterey, CA, USA, 7–9 November 2005; pp. 23–30.
5. Chuang, Y.; Curless, B.; Salesin, D.; Szeliski, R. A Bayesian approach to digital matting. In Proceedings of the Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December 2001; pp. 264–271.
6. Zhu, Q.; Shao, L.; Li, X.; Wang, L. Targeting accurate object extraction from an image: a comprehensive study of natural image matting. *IEEE Trans. Neural Netw.* **2015**, *26*, 185–207.
7. Ruzon, M.A.; Tomasi, C. Alpha estimation in natural images. In Proceedings of the Computer Vision and Pattern Recognition, Hilton Head Island, SC, USA, 15 June 2000; pp. 18–25.
8. He, K.; Sun, J.; Tang, X. Fast matting using large kernel matting Laplacian matrices. In Proceedings of the Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2165–2172.
9. Lee, P.G.; Wu, Y. Nonlocal matting. In Proceedings of the Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 20–25 June 2011; pp. 2193–2200.
10. Levin, A.; Lischinski, D.; Weiss, Y. A closed-form solution to natural image matting. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 228–242. [CrossRef] [PubMed]
11. Rother, C.; Kolmogorov, V.; Blake, A. "GrabCut": Interactive foreground extraction using iterated graph cuts. In Proceedings of the International Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, USA, 8–12 August 2004; pp. 309–314.
12. Singaraju, D.; Rother, C.; Rhemann, C. New appearance models for natural image matting. In Proceedings of the Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 659–666.
13. Sun, J.; Jia, J.; Tang, C.K. Poisson matting. *ACM Trans. Graph.* **2004**, *23*, 315–321. [CrossRef]
14. Chen, Q.; Li, D.; Tang, C. KNN matting. In Proceedings of the Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 869–876.
15. Gastal, E.S.L.; Oliveira, M.M. Shared sampling for real-time alpha matting. *Comput. Graph. Forum* **2010**, *29*, 575–584. [CrossRef]
16. He, K.; Rhemann, C.; Rother, C.; Tang, X.; Sun, J. A global sampling method for alpha matting. In Proceedings of the CVPR 2011, Colorado Springs, CO, USA, 20–25 June 2011; pp. 2049–2056.
17. Wang, J.; Cohen, M.F. Optimized color sampling for robust matting. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; pp. 1–8.
18. Battisti, C.; Messelodi, S.; Poiesi, F. Seamless bare-hand interaction in mixed reality. In Proceedings of the International Symposium on Mixed and Augmented Reality Adjunct, Munich, Germany, 16–20 October 2018; pp. 198–203.
19. Choudhury, A.; Talukdar, A.K.; Sarma, K.K. A review on vision-based hand gesture recognition and applications. In *Intelligent Applications for Heterogeneous System Modeling and Design*; IGI Global: Hershey, PA, USA, 2015; pp. 256–281.
20. Garg, P.; Aggarwal, N.; Sofat, S. Vision based hand gesture recognition. *Int. J. Comput. Electr. Autom. Control. Inf. Eng.* **2009**, *3*, 186–191.
21. Wario, R.; Nyaga, C. A survey of the constraints encountered in dynamic vision-based sign language hand gesture recognition. In Proceedings of the International Conference on Human–Computer Interaction, Orlando, FL, USA, 26–31 July 2019; pp. 373–382.
22. Osullivan, C.; Cassell, J.; Vilhjalmsson, H.H.; Dingliana, J.; Dobbyn, S.; Mcnamee, B.; Peters, C.E.; Giang, T. Levels of detail for crowds and groups. *Comput. Graph. Forum* **2002**, *21*, 733–741. [CrossRef]
23. Zwicker, M.; Pauly, M.; Knoll, O.; Gross, M.H. Pointshop 3D: An interactive system for point-based surface editing. In Proceedings of the International Conference on Computer Graphics and Interactive Techniques, San Antonio, TX, USA, 23–26 July 2002; pp. 322–329.

24. Lippman, A. Movie-maps: An application of the optical videodisc to computer graphics. In Proceedings of the International Conference on Computer Graphics and Interactive Techniques, Seattle, WA, USA, 14–18 July 1980; pp. 32–42.
25. Lanman, D.; Hauagge, D.C.; Taubin, G. Shape from depth discontinuities under orthographic projection. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision Workshops, Kyoto, Japan, 27 September–4 October 2009; pp. 1550–1557.