In annotation overlay applications using augmented reality (AR), view management is widely used for improving readability and intelligibility of the annotations. In order to recognize the visible portions of objects in the user’s view, the positions, orientations, and shapes of the objects should be known in the case of conventional view management methods. However, it is difficult for a wearable AR system to obtain the positions, orientations and shapes of objects because the target object is usually moving or non-rigid. In this paper, we propose a view management method to overlay annotations of moving or non-rigid objects for networked wearable AR. The proposed method obtains positions and shapes of target objects via a network in order to estimate the visible portions of the target objects in the user’s view. Annotations are located by minimizing penalties related to the overlap of an annotation, occlusion of target objects, length of a line between the annotation and the target object, and distance of the annotation in sequential frames. Through experiments, we have proven that the prototype system can correctly provide each user with annotations on multiple users of wearable AR systems.

Index Terms— Augmented reality, Wearable computer, Annotation, View management

1. INTRODUCTION

Augmented reality (AR) can enhance real world by superimposing virtual objects on the user’s view. Annotation overlay applications give us additional information for the real world [1, 2, 3]. Annotations are generally overlaid by using a 3D model or a 2D image as shown in Fig. 1.

It is absolutely essential for annotations to be understood as an information about a target object in a real-world scenario. Therefore, it is important to improve the readability and intelligibility of the annotations in the user’s view. In annotation overlay applications, view management makes it possible to appropriately overlay annotations for a user so as to intuitively understand annotations [4, 5]. One critical issue of managing a view of annotations for a target object is the avoidance of an overlapping of one annotation with others [6, 7]. The unexpected occlusion of objects by other annotations must also be prevented. In order to overcome these problems, an annotation overlay system needs to estimate the visible portions of target objects in the user’s view. In conventional view management methods, position, orientation, and shape of objects are known. However, it is difficult for a wearable AR system to obtain the position, orientation, and shape of objects if the objects are moving or non-rigid. In this paper, we propose a view management method for networked wearable AR to overlay annotations of moving or non-rigid objects. The wearable AR system obtains positions and shapes of target objects via a network shared database framework [3] in order to estimate the visible portions of target objects in the user’s view. The annotations are overlaid by minimizing the penalty function that considers overlaps of annotations, occlusion of target objects, distance between the annotation and the target object, and distance of the annotation in sequential frames.

2. AR SCENE FOR TESTBED

We have prepared an indoor environment as a testbed for proposed wearable AR system. The environment is conducive to the use of wearable AR systems that generate annotation overlay images by estimating the user’s position and orientation using a localization method [8] and obtain annotation information via a wireless network by using a framework [3]. In this environment, wallpapers with pre-printed invisible markers on a ceiling are used in the localization method [8]. Further, the position and orientation errors of the method are 1.3cm and 0.5°. This paper assumes that a wearable computer user is a moving or a non-rigid object. We have prepared 3D object annotation for displaying navigation information and 2D image annotation for displaying user’s location and per-
sonal information. The 3D object annotation is overlaid as objects fixed in the real world as shown in Fig. 1(a). On the other hand, the 2D image annotation is overlaid by using the proposed view management method as shown in Fig. 1(b).

3. VIEW MANAGEMENT ALGORITHM
The proposed method uses a position and a shape of wearable computer users and follows the three steps given below.

- Step 1) Estimation of existing region of the users
- Step 2) Calculating penalty
- Step 3) Annotation overlay

A detailed description of each step is given in the following sections.

3.1. Estimation of existing region of the users
In this step, the wearable AR system estimates an existing region of a user. The wearable AR system creates a probability map which is used for calculating the penalty for an annotation. Fig. 2 shows an example of a probability map. This method is designed using the concept of probability density, and the location of the user in the real environment is approximated by using primitive models. The details of each step of the estimation are given below.

In the first step, the position of the user in the view of the AR system is estimated by using the position data obtained via a network. In this report, the position of the user’s camera is considered to be the user’s position. In the next step, the existing region of the user is estimated by using primitive models. This approximation is based on the assumptions that the position of the top of the head is the user’s position and that the user is positioned vertically on the floor. In this paper, the model user consists of three parts (head, upper body, and lower body), and rectangular solids are used as the primitive models. Each part of the model is represented as a set of rectangular solids that are arranged radially in horizontal direction.

Each part has an existing probability as a parameter. We have concluded that the head has a high probability density because the camera is set on the head. In contrast, the lower body has a low probability density.

Next, each primitive model is drawn by off-screen rendering in order to estimate the probability in the image from the user’s view. Each primitive model is drawn with a permeation rate (alpha value) in inverse proportion to the existing probability. The location of the users in the view of the AR system is estimated as described above.

Finally, the probability map is generated using the result of off-screen rendering. This map includes the location of the users as well as that of the 3D object annotations. The regions of 3D object annotations are rendered using colored textures and are not transparent. The color of a point on the image generated by off-screen rendering can be treated as an existing probability of the users and fixed annotations at that point.

3.2. Calculating penalty
Many factors need to be considered while managing the overlaying annotations. As the first step of this work, we handle three factors related to readability and intelligibility: the amount of overlapping, the length of a line between the user and its annotation, and distance of the annotation in sequential frames. Annotations of the users are overlaid on the 2D user’s view. A penalty when an annotation is overlaid at a position \((x, y)\) is defined as

\[
P(x, y) = w_1 P_1(x, y) + w_2 P_2(x, y) + w_3 P_3(x, y)
\]  

where \(P_1\) is a penalty related to overlapping, \(P_2\) is a penalty related to the amount of distance between annotations in sequential frames on the view plane. Further, \(w_1, w_2,\) and \(w_3\) are weight parameters. Default of each penalty is described as follows.

Calculation of penalty related to overlapping \(P_1\): The penalty related to overlapping, \(P_1\), includes two types of penalties. One is a penalty \(P_{anns}\) related to overlapping between annotations, and the other is the penalty \(P_{users}\) related to overlapping between the annotation and users. \(P_1(x, y)\) is thus defined as

\[
P_1(x, y) = P_{anns}(x, y) + P_{users}(x, y)
\]  

Fig. 3 shows three patterns of overlaying annotations that include the penalty related to overlapping between annotations. Fig. 3(a) shows an overlapping between 2D images. Fig. 3(b) shows overlapping lines. Fig. 3(c) shows an overlapping between the 2D image and the line. The penalty related to overlapping between annotations is the amount of overlap that is simply calculated as an area of the common area. In contrast, a penalty related to overlapping between the annotation and the users is calculated as a sum of probabilities in the common area.

Fig. 4 shows two patterns of overlaying annotation that include the penalty related to overlapping between the annotation and the users. When the annotation is overlaid at a point.
The penalty related to the line between the annotation and the user.

Fig. 4. Penalty related to overlapping between the annotation and users.

\[(x, y), P_{user}(x, y) \text{ is defined as} \]
\[P_{user}(x, y) = \sum_{i \in A(x,y)} C_i \]  

where \(i\) is a point on the probability map, \(C_i\) is the brightness value of the point \(i\), and \(A(x,y)\) is a set of points that are occluded if an annotation is placed at \((x, y)\).

**Calculation of penalty related to line \(P_1\):**
The penalty related to the line between the annotation and the user \(P_1\) is proportional to the length of the line. In this study, in order to simplify the regulation of penalty, \(P_1\) is defined as \(P_1(x, y) = P^l\), where \(l\) is the length of the line between the annotation and the user.

**Calculation of penalty related to distance \(P_d\):**
\(P_d\) is a penalty related to the distance of the annotation on the view plane in sequential frames. As described in Fig. 5, \(P_d(x, y) = |B_t - A_t|\). \(A_{t-1}\) is the position of the annotation in the previous frame, \(A_t\) is the position in the current frame calculated by keeping the relative position between the annotation and the user in the previous frame, and \(|B_t - A_t|\) is the distance between \(B_t\) and \(A_t\).

3.3. Annotation overlay
Annotations are overlaid by minimizing the penalty defined in section 2.3. In order to minimize the penalty, the following steps are used.

**A : Configuration of size and shape of annotation**

**B : Decision of an arrangement of annotations**

A detailed description of these steps is given below.

4. EXPERIMENT

In this experiment, one user of wearable AR system and three wearable computer users are introduced in the environment described earlier. The users’ positions are estimated by their wearable PC (CPU: 933 MHz, Memory: 1.0 GB) and an original infrared camera (Captured image: 640 × 480 pixels, Captured rate: approximately 29 fps) and are transmitted to the wearable AR system. The infrared camera unit composed of an infrared camera and a video camera.

The infrared camera (Captured image: 1024 × 768 pixels, Captured rate: approximately 30 fps) is used for estimating...
the user’s position and orientation. The video camera (Captured image: $800 \times 600$ pixels, Captured rate: approximately 15 fps) is used for capturing the user’s view. In this experiment, the size of the generated AR images is $640 \times 480$ pixels; the annotations are rectangle-shaped and have a fixed size of $81 \times 41$ pixels. For this paper, both the upper and lower surfaces of the rectangular solids are square-shaped and measure 5 cm on each side. The number of solids used for representing each part are as follows: head: 25, upper body: 81, and lower body: 169. We have set the parameters of each zone as follows: head: 0.7, upper body: 0.3, and lower body: 0.1.

Fig. 6 shows the images generated when the four users are walking in the area and the user of the AR system receives the location of the other users around him at a rate of approximately 20 times per second. The annotations for the users in Fig. 6(a) are dynamically placed by the view management, and in Fig. 6(b), the annotations are placed by fixing the relative position between the annotation and the user. As shown in Fig. 6(a), the proposed method overlays annotations preventing overlaps in this result. For a faster penalty calculation in this experiment, the penalty is calculated only in an area of $21 \times 21$ pixels around the point $A_t$ in Fig. 5 and 16 points around the user’s position for preventing a local minimum. The distance between the 16 points and the user is $450 \div l$, where $l$ is the distance between the target user of the annotation and the user of the wearable AR system in the real world. The AR system worked at a rate of 8-12 fps in this experiment.

Fig. 7 shows the comparative result between an annotation overlay image and a penalty distribution. Fig. 7(b) shows penalties of all points on the image when an annotation for the user on the left is overlaid in Fig. 7(a). As shown in Fig. 7, the penalty is high in the area of the 3D object annotation and the users, and the annotation for the user on the left is placed at a position whose penalty is low.

Fig. 7. Comparative result between annotation overlay image and penalty distribution.

5. SUMMARY

This paper proposes a view management method for annotations of moving or non-rigid objects. By using the proposed method, the users of the AR system were intuitively able to understand personal information. We confirmed the feasibility of the method by conducting experiments in an indoor environment. As a future work, we will consider the user’s attention in the real world for improving the view management.

6. REFERENCES