Vision-based Visual/Haptic Registration for WYSIWYF Display

Yasuyoshi Yokokohji *, Ralph L. Hollis, and Takeo Kanade
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213, U.S.A
yokokoji@mech.kyoto-u.ac.jp, rhollis@cs.cmu.edu, tk@cs.cmu.edu
http://www.cs.cmu.edu/~msl
*Currently with Department of Mechanical Engineering, Kyoto University, Kyoto 606, JAPAN

Abstract

We have been working on developing a visual/haptic interface for virtual environments. In the previous work, we have proposed a WYSIWYF (What You See Is What You Feel) concept which ensures a correct visual/haptic registration so that what the user can see via a visual interface is consistent with what he/she can feel through a haptic interface. The key components of the WYSIWYF display are (i) vision-based tracking, (ii) video keying, and (iii) physically-based simulation. The first prototype has been built and the proposed concept was demonstrated. It turned out, however, that the original system had a bottleneck in the vision tracking component and the performance was not satisfactory (slow frame rate and large latency). To solve the problem of our first prototype, we have implemented a fast tracker which can track more than 100 markers in video-rate. In this paper, new experimental results are shown followed by the improvements of the vision-based tracking component.

1 Introduction

Haptic interfaces have been recognized as important input/output channels to/from the virtual environment[10][11][16]. Usually a haptic interface is implemented with a visual display interface such as a head-mounted display or a stereoscopic display screen. Correct registration of visual and haptic interfaces, however, is not easy to achieve and has not been seriously considered. For example, some systems have a graphics display simply beside the haptic interface resulting in a "feeling here but looking there" situation as shown in Fig.1. Poor visual/haptic registration could result in inter-sensory conflicts that leads to a wrong sensory rearrangement[15].

One of the most important potential applications of VR systems is training and simulation. For training visual-motor skills, correct visual/haptic registration is important because a visual-motor skill is composed of tightly coupling visual stimuli (associated with task coordinates) and kinesthetic stimuli (associated with body coordinates). If there is an inconsistency between the two kinds of stimuli, there would be no significant skill transfer[8], or in an even worse case, the

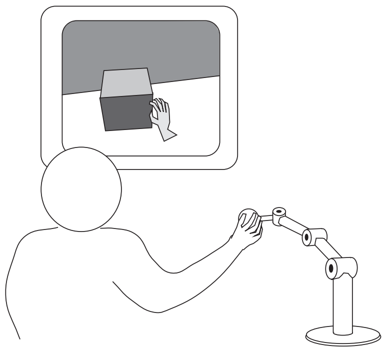


Figure 1: "Feeling here but looking there" situation

training might negatively hurt performance in real situations (negative skill transfer).

In our previous paper, we have proposed a WYSIWYF (What You See Is What You Feel) concept[19]. The proposed concept ensures correct visual/haptic registration so that what the user can see from the visual interface is consistent with exactly what he/she can feel through the haptic device. A vision-based object tracking technique and a video-keying technique are used to get the correct visual/haptic registration. The first prototype was built by using a color liquid crystal display (LCD) panel and a CCD camera for the visual interface component and a PUMA 560 robot for the haptic interface component. It turned out, however, that the original system had a bottleneck in the vision tracking component and the performance was not satisfactory (low frame rate and large latency). To solve the problem of our first prototype, we have implemented a fast tracker which has a capability to track more than 100 markers in video-rate (30 Hz).

The rest of the paper is organized as follows. First the concept of the WYSIWYF display is presented in section 2. Our prototype system and some new experimental results are shown in section 3, and finally some improvements of the vision-based tracking component including the video-rate tracker is shown in section 4.

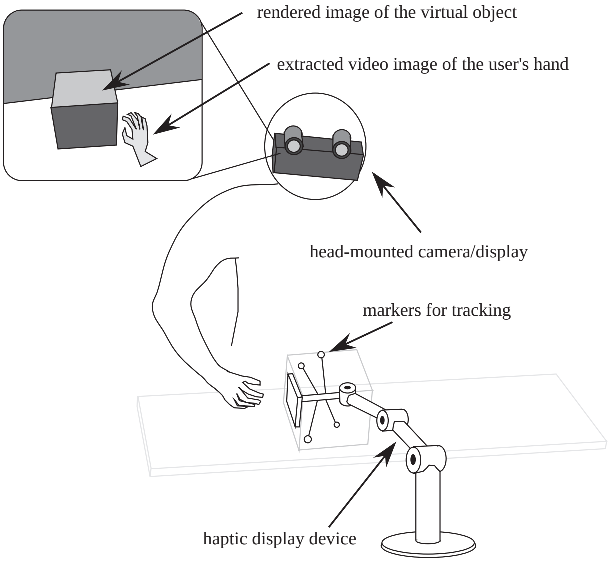


Figure 2: WYSIWY Fdi spla y

2 WYSIWY FDisplay Concep t

2.1 Realizing WYSIWF

Fi g2ir le l u st threpa rt oepss ecodnc eopftt hve i - s u a l /hi atnpe tr if Aahceea. d - tmeo ch uanmet rr ks ac ma re ka st th at dare n c to e rt ye p dh a p d ei v ic e (t h me e a n io feg n c to e rt y e p d wi lb d e xp l a i n e d l a ta enrel) s t i ma r n ate esp o t s b e we e tn hu es e r 's h e a d a n td h he a p d ei v ic i Th e v i r to p hea cil ts e g i s - te r te od h he a p d ei v ij c u d t s e t a k u g n t ee m d e a y l i f a p p l i ca an ti di ti o n sm a sig, de i sy p ol t d a to hu es ev r a a h e a d - tmeo dh uins y.p Um l e i a k u g n t ee m d e a y , l i t h ow e ev r t , ht e hr e e s a d e i m s e od t i sy p o l d e a c x t e e p h t e u s e hr a ' ns d T h e hu es e hr a ' n si d m a i g s e x t r a c t e d f r o ct mhr e e s a d e t n a e k d n b y t h he e a d - tmeo dh uam m - e r a n , id s ue pr i on sp e o d t hs e yt nh es t c ie oe n fe h e v ir te n u a l r o tn S n e e F n i g . f5 (b a rc n) c t e u x a d a m - p l e .

I f th ve i r to p h b ae l c m a ig s e o r r e e c g t l s y t to e r r e d t h he a p d ei v it c hu es , e hr a ' n s d a r n e h a t ch he a p t i c d e v ie x a c w th le y h m i s / h a e n r de h a e ct s hv ei r t u a l o b je c T t h s p a t io at l w i b e n tr e l u es e f r e c h ha sp - t is ce n s a t th ir o t mu hg ht e le v i is c e o n s ti w s i t ttel hn e s p a t i o a t l o i fo h h v ei r to p h b a e p l e tr ce e d i h v r o u g h t hv ei s d u i a s l y. p W ha a t t hu es e s r i e e s x a c w th la y t h e / s f h e e e T h s e r e t fh o W r Y e S I W Y S F t u a i t s i o n r e a l i z e d .

T h r e k e e y c o m p n t e s o f t h W Y S I W Y d F s y p l a y a r (e i v) i s i o n t - r h k a i c a s (g e i v d i) d k e e p y i a n g (d i i i) p h y s i c a l l s y - u n b l a t s E a d c h c o m p n t e w i n l b e l e x p l a b i r i n e b e f l l y w o F o r m o r d e e t a s i e [1 l 9 s] . ,

Vision based head tra c king

U s i t n h g e v i s i o n t - r h k a i c a s t e g h d n i q t u h e u s e r ' s h e a p o d s c e a b l e e s t i m b y a t t r e k a d c n a t g a r o g j b e e t c t i n t h c e a m e v r i a e T w h . t e a r o g j b e e t c c o t u b l e e d i t h e r t h he a p d ei v o i r n a fi x e o d j b e c i c t n h w o r k e i n g v i r o tn s m u h c a n s a t a b T h l e e r e l a b t e i v t e e o e n t n h e h a p d ei v c t i c a p n t d h w e o r k ei n v n i g r o tn c m a e h n e n o b t a b y n j e o t d s r e n s o o f t r h s h e a p d e i v c e .

A d a v t n a g e p o o i u t s n s o f v i s i o n t - r h k a i c s o n v g e d r e m i t t e r t - r h k a i c r s t g e h d n i q s u u h c a s m a g n e t i c t r k a e c r a s n d a c o u s t i k c e a c s r (e i v) i s i o n - b a s e d m e t d h c o a t n r k a a c y n t h i w h h l g i c s v i s f b l t t e h e c a m e v r h a e r t e l h a e s m i t t e r n - e b l t h d c s o a e t n d r k a c o n l t h y o e j b e w i t t a h l m a g n e t i c r / a c e o c e u r i s a t d i c (i t i r) k a i c n a g c u o f a r c y i s o n m - e b t a h s o e e d u b l e d i p p i x l e e l w h h l i c s n o u f g o t r h d e i s y p l u a p s e a n i d s f r f e e r d e n m r o m a g n a t e e t r i c a l s .

Video keying

T h C E C D c a m e f r o m a r e h t r k a i c n g a a n l s e a o p t u r e t h w e o r k e i n v n i g r o t n i m e e c n l t u d h u e s n e g h r a ' n s l d f . we c o u e l x d t r t a h p e o t r t o i f o h l u e s e h r a ' n s i d m a g e , we c o u s l u d e p r i o n s p t e h i s n a g t c e h o t h g e r a p h i c a l i m a g f e t h v e i r t e u n a i l r o t n i m r e s n t o e f r a e d n d e r - i n a g s y t n h e p o i l c y g h o n a n e l l m a g T e h . e a s i e s t way t o e x t r t a h u e s e t e h r a ' n s i d m a i g t e h " C h r o m a Ke y i n t g h g n i q w y h h l i c s o m m o u n s l y e i d w e a t h e r f o r e c a T s V t p t r i o n g g r U a s m i s t n h g l e e i e v s e h r a ' n s d i m a g l e w l s o s t o e l i m i a d r a t e a v e - g e l d s s b i e k n s i n g d e v i c e .

Physically-based simulation

Mo s h a p t r i e c n d e a r l i g n g r i a t r l b e m a s s e o d h i m e p d a n c c o t e n r o [l 1 1] w [l 1 e 6] t r , h a p o s i t a i n o d n t h e v e l c o y t o f t h l e a p t d e v i a c r m e e a s u a r n e d d a n a p p r o j f r o i r i a c s t e i e s y p o l t d a t c t h u e s e W i . t h t h a i s p p r h o , a f c o r c c a e b e g e n e r o a n t l w e y h d e a n p e n e t r o a c t c i u b e e n s t w e e t w o o j b e c T o t s r . e n d e r a r i g o i j b d e w i t t t h l i e m e p d a r n d c i e s y p a l p a p r h o , a c t h s e t i f f p i o e a s r s a n s e h t o e t u r e l l d a r e g n e o u a g n h a d , h i g s h a m p l r i a n t i e s e q u i o r t d h e , d r t w h i s e s u e n h a - t i t o e n t d o d s e u n s t a b l e [5] .

I n t h i e t n e r a e c t o i m p y u g t r e a r p f h e i l d B s e r a f f [2] h a p s r o p s e a p d h y s i c a l l d y - n b a s n s i u e m l d a t i o n m e t d h f o o n r o n e - n p e t r r a i t g b o d i d g e W s e a p p l i e d h i a s l g o r i t d t h h e m p r t e i n d e T h i r e a n d g a v n a g e s o f a p p l y h i a s n p g p r h o t a t c h l e a p t r e i n d e a r r i e n g (i i) p p r i n c a r i p l b g o i d d c y a b e r e n d e a r n e d (i d i) i p p r a c t l h t e i e s m , e t s p i o z f e h s e i u n h a t i i s n o r n t c r i t e i e v r o f a o s l r i u m l a t r i i n g b g o d d i e s .

2.2 Encoun tered type haptic displa y

M c N e e l y [l a s 1 4] s h a i p f e t e d i d s y p s i l t u a t o h f e o l - l w o i n t g h r t e y e p e s : (i w) o r y n - p t , (i h i e) l y d p - , t a n d (i i e i n) c t o e u r n y e p - d t E x a m p d o l e w s o r n - t y p s a r e e x o e s l k e t m o n s t e a r n [d 3] t a v e w g i l t o h a c t u a t o U n r s i e [r 4 s h] a) n c d o t n r o l l a e n r f d [o 1 r 0] c e f e e d b t j a y s c k [t a G r t y p i e c x a d m p o l f t h e s l e e l d - t y p h a p t d e i v A i c w e o . r y n - p t o r h e l y d p - h t a p - t i i d n e r f a l a e s v o o r h k s a r i n p d u e t v t i o m n e e a s u r e t h u e s e m r o ' t s i r o e n q , u i t h i r a e t h d g e e v i a k a e y w e s b e p s h y s i c c a o l n l n y e t c t c o t e m p e a r t o f t h u e s e b r o d y s (y t p i c a a r l h l a l , y a n y r o d f i n g e T r h s i r) s e q u i t r l e i m e n e n t i r t t s h u e s e w o ' r s k s p a n n d o l b s t r t u h e s t e s f r r ' e s e m o t i A o n d . i s t e i d n h i c f f e i n r b e e t w u e c e e t n h w o r n -

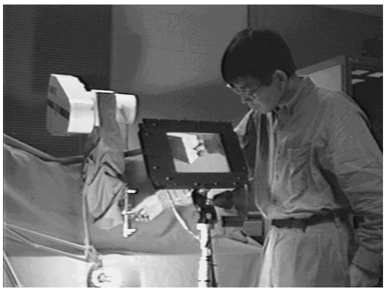


Figure 3: System overview in use

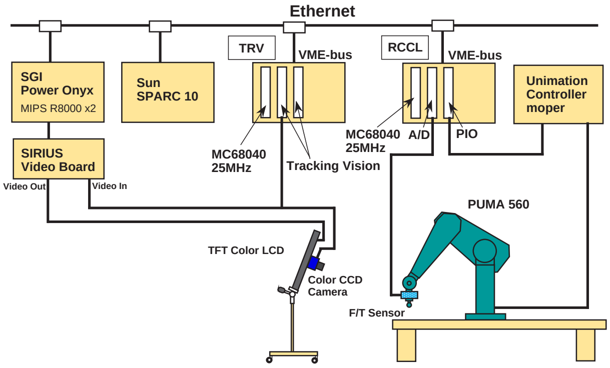


Figure 4: Prototype system configuration

type and the held-type is that the held-type device is "grounded" whereas the worn-type device is not.

With the encountered-type, on the other hand, the user need not keep holding the haptic device. Instead, the system tracks the motion of the user's hand and places the haptic device in the appropriate location, waiting for the user to "encounter" it (*surface display mode*). Once the user encounters the device, it responds to the forces exerted by the user, based on the virtual object model (*admittance display mode*). McNeely[14][7], Hirota and Hirose[9], and Tachi et al.[17] have already proposed and implemented this encountered-type approach, where they called *robotic graphics*, *surface display*, and *haptic space* respectively.

The encountered-type approach is well suited to our WYSIWYF concept as shown in Fig.2. But WYSIWYF display concept is not limited to the encountered-type haptic display.

3 Prototype WYSIWYF Display

3.1 System configuration

Figure 3 shows a system overview in use. Figure 4 illustrates our current prototype system configuration. Although a head-mounted camera/display would be ideal for WYSIWYF, we decided to use an existing LCD panel for our first prototype. A color CCD camera is attached at the back plane of the LCD panel. The LCD/camera system is mounted on a movable stand so that the user can move it around to change his/her viewpoint.

Pose estimation was originally performed by a SGI PowerOnyx with an optional SIRIUS Video Board in the first prototype[19]. It turned out, however, that the original system had a bottleneck in the vision tracking component and the performance (frame rate and latency) was not satisfactory. To solve the problem of our first prototype, we have implemented a video-rate tracker (Fujitsu Tracking Vision) which has a capability to track more than 100 markers in video-rate (30 Hz). More details about Tracking Vision will be described in the next section.

Rendering the virtual scene is performed by a SGI PowerOnyx and final images are sent to the LCD panel via SIRIUS Video Board. A PUMA 560, 6 DOF industrial robot, is used for the haptic device. We put an aluminum plate with four markers, small incandescent lamps covered by translucent lenses, at the tip of the PUMA for tracking.

Physically-based simulation is performed on a VME-bus-based MC68040 CPU board (Motorola MVME162) with the VxWorks realtime OS. RCCL/RCI[12], realtime C libraries for controlling PUMA, has been installed on our VxWorks system. A SPARC 10 workstation is used for the VxWorks and RCCL host machine. A six-axis force/torque sensor is attached to the PUMA. The Unimation controller and the VxWorks system are connected by a parallel cable. Original VAL in the Unimation controller has been replaced by a special communication software called "moper".

Due to the computational performance limitation, the physically-based simulation algorithm runs at 50 Hz (20 msec/cycle). The simulation module gives the current position/orientation of the virtual object to the RCCL/RCI module as a setpoint. RCCL/RCI then interpolates these points and generates a smooth trajectory. The generated trajectory data are sent to the Unimation controller via parallel lines. Moper receives the trajectory data and distributes the data to each joint servo loop module in the controller. The lowest joint level servo loop in the Unimation controller runs at 1000 Hz.

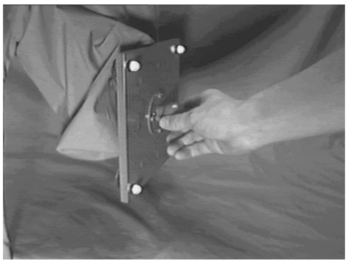
The working environment was covered by blue cloth for Chroma Keying. The forearm and wrist portions of the PUMA including the F/T sensor, were wrapped up by blue cloth as well.

3.2 Experimental results

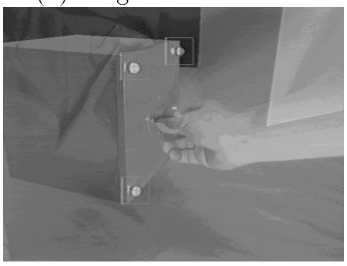
3.2.1 A cube

A simple frictionless virtual environment was built, where a 20 cm×20 cm×20 cm cube is on top of a flat table. Figure 5 shows the tracking and video blending process. The overlaid image in Fig.5 (b), which is not shown to the user, shows how well the virtual cube is registered to the real marker plate. Small square windows in the image indicate searching windows for the marker tracking. One of the four markers is occluded by the user's hand.

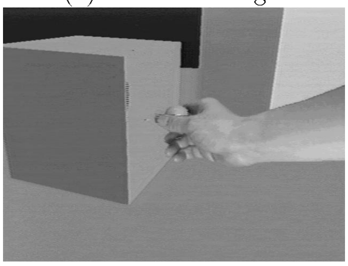
In this system, the sensor knob is the only permitted portion of the haptic device for the user to access, which corresponds to the knob attached to the virtual cube (see Fig.5(c)).



(a) Original video scene

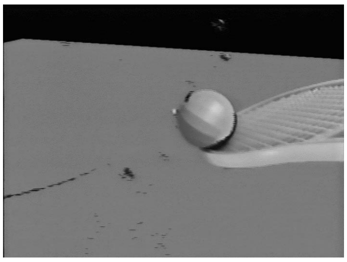


(b) Overlaid image

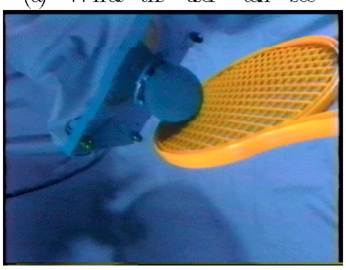


(c) Final blended image

Figure 5: Results of registration and blending



(a) What the user can see



(b) What the user is actually doing

Figure 6: Virtual tennis

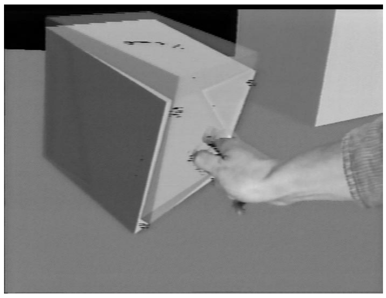


Figure 7: An example of skill training

Although the PUMA is controlled by conventional high-gain position servos and we are updating the set-point every 20 msec, which is a relatively slow rate, the system can keep stable and can render reasonably convincing haptic sensations.

3.2.2 Virtual tennis

Figure 6 is another example, virtual tennis. Note that the ball is a virtual image but the racket is a real image. This example demonstrates that the user can interact with a virtual environment not only with his/her own hand but also with other real tools.

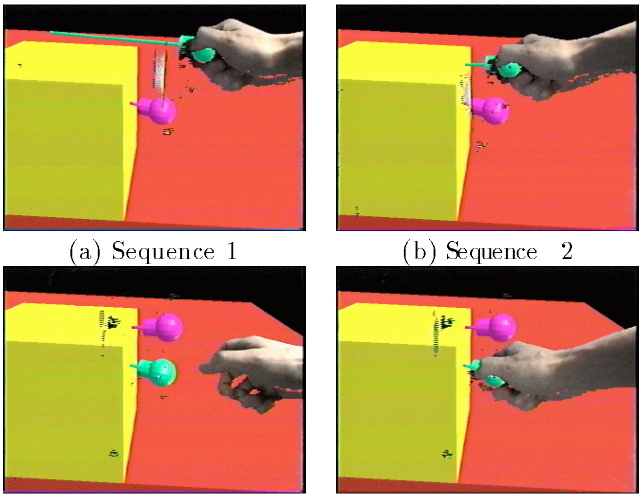
3.2.3 Training

One potential application of this system is for the training of visuo-motor skills, such as medical operations. Figure 7 shows a simple example of training. The user is trying to follow the pre-recorded motion of the expert displayed by a transparent cube. A weak position servo can guide the user to the reference motion. Unlike just watching a video, the trainee can feel the reaction forces from the virtual environment while following the reference motion.

3.2.4 Handling multiple tools

As discussed in 2.2, our WYSIWYF display adopts the encountered-type haptic display. In the previous examples, the user could manipulate only one virtual object (a cube or a ball). In such a case, the haptic device can simply stay at the location where the virtual object exists. If there are more than one virtual object, however, the device has to change its location according to the user's choice.

Figure 8 shows an example of handling multiple tools. There are two tools sticking in a piece of "virtual cheese". When the user decides to change the tool, the haptic device changes its location so that he/she can encounter the next tool. In this example, the user manually selects one of these tools by a toggle switch. When the selected tool gets ready to be encountered, its color changes from red (dark color in the figures) to green (bright one). Ideally the system should detect his/her selection automatically by tracking the motion of his/her hand.



(a) Sequence 1 (b) Sequence 2

(c) Sequence 3 (d) Sequence 4

Figure 8: Handling multiple tools

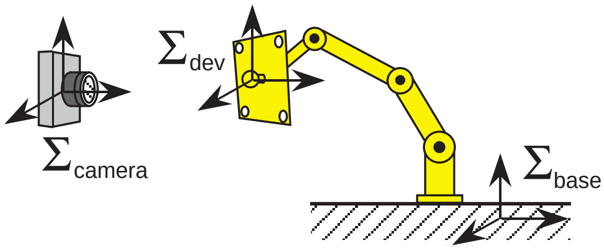


Figure 9: Coordinate frames

4 Vision-based visual/haptic registration

4.1 Registration method

Vision-based tracking is a key component of the WYSIWYF display. In the computer vision field, several techniques have been developed for tracking the object in the video image. Uenohara and Kanade[18] have developed a real-time image overlay system with vision-based object registration and tracking techniques. Bajura and Neumann[1] have implemented a similar method for the head tracking application, to compensate registration errors induced by the magnetic head-tracking sensor.

As we discussed in section 2, markers may be placed either on the haptic device or on the fixed environment. In our prototype system, only one CCD camera is used and the camera will almost always be headed to the haptic device. If the markers attached on the fixed environment, they may be occluded by the haptic device or may be out of the camera view. We therefore decided to attach the markers at the tip of the haptic device.

Iterative pose estimation with least squares minimization[13] is an efficient way, assuming that the relative motion between the camera and the target object is so small in each subsequent video frame that the relation can be linearized. We first implemented this method. This technique is simple but sensitive to the measurement noise. In our case, the estimated pose tends to be shaky even when the markers are stationary in the camera view. We then implemented the pose estimation algorithm based on the extended Kalman Filter (EKF) by Gennery[6]. Gennery's algorithm uses quaternion to represent the orientational component. Unlike Euler angles and Roll-Pitch-Yaw angles, quaternion has no singular representation.

At least three markers are necessary to estimate the pose of the target object (position/orientation in three dimensional space). More than three markers should be put on the target object to cope with the marker occlusion.

4.2 Some improvements of vision-based tracking

4.2.1 Tracking moving markers from a moving camera

For the following discussion, let three coordinate frames, $\boldsymbol{\Sigma}_{camera}$, $\boldsymbol{\Sigma}_{dev}$, and $\boldsymbol{\Sigma}_{base}$, be defined. These frames are attached to the camera, the tip of the haptic device, and the fixed environment respectively as shown in Fig. 9. Let ${}^{A}\boldsymbol{T}_{B}$ denote a 4×4 homogeneous transformation matrix from $\boldsymbol{\Sigma}_B$ to $\boldsymbol{\Sigma}_A$, representing the position and orientation of $\boldsymbol{\Sigma}_B$ with respect to $\boldsymbol{\Sigma}_A$. To render the virtual object and the background, we need ${}^{camera}\boldsymbol{T}_{dev}$ and ${}^{camera}\boldsymbol{T}_{base}$ respectively. Reading the joint sensor information of the haptic device, one can get accurate ${}^{base}\boldsymbol{T}_{dev}$. What the camera can see is 2D projection of the markers' location ${}^{camera}\boldsymbol{T}_{dev}$. ${}^{camera}\boldsymbol{T}_{dev}$ can be decomposed as follows:

$$ {}^{camera}\boldsymbol{T}_{dev} = \left({}^{base}\boldsymbol{T}_{camera}\right)^{-1} {}^{base}\boldsymbol{T}_{dev} \qquad (1) $$

Eq. (1) means that the markers' motion in the camera view could be caused by both the camera motion and the device motion. To reconstruct the 3D pose, we have to track the moving markers from the moving camera.

We first took the relative pose between the camera and the device, ${}^{camera}\boldsymbol{T}_{dev}$, as the state variable of the EKF. Figure 10 (a) illustrates this first configuration. In this configuration, EKF outputs the estimated ${}^{camera}\boldsymbol{T}_{dev}$ as follows:

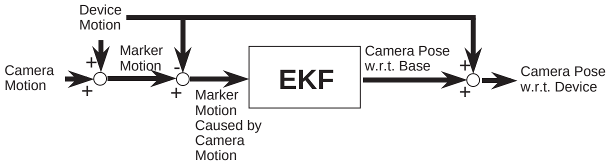
$$ {}^{camera}\tilde{\boldsymbol{T}}_{dev} = EKF({}^{camera}\boldsymbol{T}_{dev}) \qquad (2) $$

The estimated pose of the fixed environment with respect to the camera is then calculated using ${}^{base}\boldsymbol{T}_{dev}$.

$$ {}^{camera}\tilde{\boldsymbol{T}}_{base} = {}^{camera}\tilde{\boldsymbol{T}}_{dev}\left({}^{base}\boldsymbol{T}_{dev}\right)^{-1} \qquad (3) $$

In eq. (2), $EKF()$ means a transfer function of the estimator. As discussed in [6], EKF can be approximated to a second-order recursive filter. The larger the covariance matrix of noise is set, the larger the time

(a) Our original EKF configuration



(b) Modified EKF configuration

Figure 10: Two configurations for pose estimations using EKF

constant of the filter becomes. In general if the input signal X contains low frequency components only, the filter output $\widetilde{X} = EKF(X)$ is nearly equal to X, i.e. $\widetilde{X} \simeq X$. If the input signal contains a high frequency component, this component is filtered out and its phase is delayed, resulting in $\widetilde{X} \neq X$. Note that this is just a qualitative discussion.

With our prototype system using a LCD panel, users tend to keep the camera/display system at his/her preferred location, meaning that ${}^{base}\boldsymbol{T}_{camera}$ is often constant while ${}^{base}\boldsymbol{T}_{dev}$ always changes. In the first configuration, the estimator cannot know whether the markers are moved due to the haptic device motion or due to the camera motion. Since ${}^{camera}\boldsymbol{T}_{dev}$ contains the device motion, the high frequency component is filtered out and ${}^{camera}\widetilde{\boldsymbol{T}}_{dev} \neq {}^{camera}\boldsymbol{T}_{dev}$. Consequently from eq. (3), ${}^{camera}\widetilde{\boldsymbol{T}}_{base} \neq {}^{camera}\boldsymbol{T}_{base}$, meaning that if the user moves the haptic device up and down or rotates back and forth, the background image tends to be shaky even when the camera/display system is stationary.

Since we know exactly the motion of the haptic device, we can exclude the contribution of the haptic device motion from the state variable of the estimator. The modified method does so.

$$ {}^{camera}\boldsymbol{T}_{base} = {}^{camera}\boldsymbol{T}_{dev} ({}^{base}\boldsymbol{T}_{dev})^{-1} \quad (4) $$

Equation (4) means that although the markers are attached to the tip of the device, they can be regarded as markers attached to the fixed base, where the markers' locations are not fixed but known. Fig. 10(b) shows this modified configuration. In this modified configuration, the EKF estimates the fixed environment pose with respect to the camera as follows:

$$ {}^{camera}\widetilde{\boldsymbol{T}}_{base} = EKF({}^{camera}\boldsymbol{T}_{base}) \quad (5) $$

and then the haptic device pose with respect to the camera is obtained.

$$ {}^{camera}\widetilde{\boldsymbol{T}}_{dev} = {}^{camera}\widetilde{\boldsymbol{T}}_{base} {}^{base}\boldsymbol{T}_{dev} \quad (6) $$

If the camera is stationary or moving slowly (containing low frequency components only), the estimator can output a nearly true value, i.e. ${}^{camera}\widetilde{\boldsymbol{T}}_{base} \simeq {}^{camera}\boldsymbol{T}_{base}$. Consequently a nearly correct pose between the camera and the device is also obtained by eq. (6), i.e. ${}^{camera}\widetilde{\boldsymbol{T}}_{dev} \simeq {}^{camera}\boldsymbol{T}_{dev}$. Of course, the timing of taking a image for marker tracking and the timing of reading the joint sensor should be exactly synchronized. After this modification, the background image became stable even when the marker moves in the camera view due to the the haptic device motion.

Lowe pointed out that smoothing the motion with EKF is not effective for tracking objects which may be bumped or collide[13]. This is true in the first configuration because the device motion may not be smooth when the virtual object collides. In the modified configuration, however, the EKF estimates only the camera motion component which is usually smooth, and smoothing by EKF is reasonable.

4.2.2 Implementation of video-rate tracker

In the first prototype system, marker tracking and video blending were performed by a SGI Power Onyx with a SIRIUS Video board[19]. Although the SIRIUS Video Board has a built-in video keying circuitry, a somewhat disappointing design specification of the SIRIUS Video prevents us from using this circuitry as long as we use the video-import for marker tracking. Alternatively one has to do the chroma keying by software. The estimated frame rate is about 5 Hz with software chroma keying, which is far from the satisfactory (30 Hz or more). In addition to the low frame rate, the latency is also large (about 0.5 sec in the worst case).

To avoid this annoying low frame rate and large latency we introduced camera fixed mode in which marker tracking is disabled so that chroma keying can be done by the built-in circuitry. The estimated frame rate of camera fixed mode is 30 Hz. In the camera fixed mode, there is no noticeable latency. Figures 6 and 7 are examples of this mode.

Of course, camera fixed mode is not a fundamental solution; although the frame rate is satisfactory, the camera/display system must be fixed. To achieve fast marker tracking, we have implemented a video-rate tracker, FUJITSU Tracking Vision (TRV), which has a capability to track more than 100 markers in video-rate (30 Hz). As shown in Fig. 4, Tracking Vision needs a CPU board running VxWorks. Tracking Vision system and SGI Power Onyx communicate via socket. TRV also enables us to use the built-in video keying curcuitry.

Tracking Vision has a Motion Estimation Processor (MEP) which can perform template matching based on the sum of absolute difference (SAD). One limitation of the Tracking Vision is that the template images of the markers can not be updated during the tracking. Usually an image is taken from a known marker location beforehand, and this image is used for templates. If the camera gets closer to the markers or far away from them, the tracking may fail. To solve

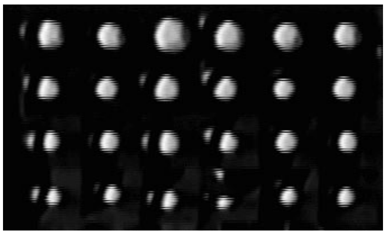


Figure 11: Templates of markers

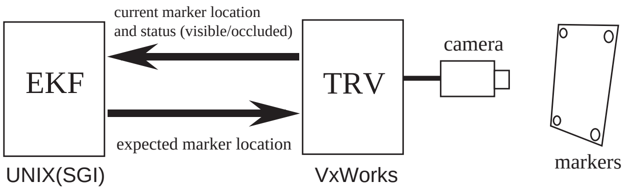


Figure 12: Data communication between EKF and TRV modules

this problem, we have taken six images from the different camera locations that covers the expected camera motion range. Figure 11 shows the taken templates. Each row of the image is a set of six templates for each marker taken from six different poses. In each tracking cycle, six templates are applied for determining the current marker location. The current marker location is determined by the template which matches to the current marker image most. When even the best matching score is over a certain threshold, we judge that this marker is occluded by something (e.g. the user's hand).

Tracking Vision system tracks marker in video-rate. As shown in Fig. 12, SGI send a request to TRV along with the expected marker location data based on the estimated pose. TRV sends back the most recent marker locations to SGI. When TRV detects occlusion of markers, it send this status to SGI. SGI can then ignore the occluded marker for the next pose estimation. TRV can use the expected marker location to recover the tracking of occluded markers.

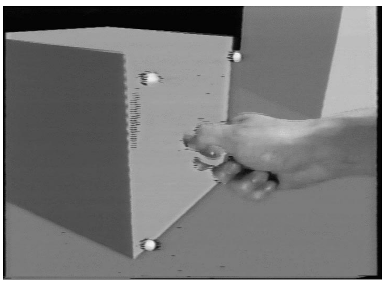


Figure 13: Tracking mode using a fast tracker

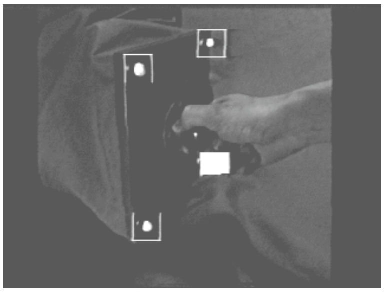


Figure 14: Monitor image of the Tracking Vision

Figure 13 shows a display scene of tracking mode with the Tracking Vision. Figure 14 shows the tracking status of the Tracking Vision. Three markers are correctly tracked, while one marker is occluded by the user's hand. A white window indicates that the tracker correctly detects this occlusion and the window location represents the expected location of the occluded marker.

The estimated frame rate was about 20 Hz. Although Tracking Vision itself can track the markers in video-rate (30 Hz), there is an overhead of socket communication between TRV and the SGI Power Onyx. Even after introduced TRV, there is a noticeable latency (about two or three frames). Figure 13 indicates this latency In this figure, the user is moving the cube downward. Since we are using hardware chroma keying circuitry, the image of the user's hand and the markers is displayed without delay. The cube image, on the other hand, is displayed based on the estimated pose which has two or three frames delay, resulting in a noticeable registration error. There are three sources of this noticeable latency: (i) Tracking Vision itself, (ii) socket communication, and (iii) EKF. Even though Tracking Vision can track the markers in video-rate, the tracked location data is available one frame later, resulting in one frame latency at this point. A field-rate tracker rather than frame-rate would be preferable. A local Ethernet connection rather or more tight connection between TRV system and SGI such as bus-to-bus connection will be necessary

5 Conclusions

This paper has introduced our new concept of a visual/haptic interface device, namely a WYSIWYF display, which ensures correct visual/haptic registration. There are three key components: (i) vision-based tracking for pose estimation, (ii) superimposing the user's live hand image with video-keying, and (iii) encountered type haptic rendering with the physically-based simulation.

To solve the low frame rate and large latency problem of the first prototype system, we have implemented a fast video tracker. Although the system performance has been improved, further improvements will be necessary

Potential applications of this system would be a teleopration system with large time delay and a rehearsal and training system for visual motor skills such as medical operations.

Acknowledgments

The authors would like to express their grateful thanks to Mr. Michihiro Uenohara, visiting research scientist of CMU, for his comment on vision-based tracking and his help for making the LCD/camera system. Dr. David Baraff, assistant professor of CMU, gave them many valuable comments on physically-based simulation. Dr. John Lloyd helped them for implementing RCCL/RCI to their system. Mr. Mike Blackwell, senior research engineer of CMU, kindly maintained the SGI Power Onyx and installed the SIRIUS Video. Dr. Alfred Rizzi, postdoctoral research associate of CMU, helped them to solve a malfunction of the PUMA power amplifier. Mr. Toshihiko Morita, research engineer of FUJITSU LABORATORIES LTD., suggested an idea of using multiple templates for each marker. Finally they would express their appreciation to SHARP CORPORATION for providing the LCD panel, and to FUJITSU LABORATORIES LTD. for providing the Tracking Vision system.

References

[1] M. Bajura and U. Neumann, "Dynamic Registration Correction in Augmented-Reality Systems", In *Proceedings, IEEE Virtual Reality International Symposium '95*, pp. 189–196 (1995)

[2] D. Baraff, "Fast Contact Force Computation for Nonpenetrating Rigid Bodies", In *Proc., SIGGRAPH'94*, pp. 23–34 (1994)

[3] M. Bergamasco et al., "An Arm Exoskeleton System for Teleoperation and Virtual Environments Applications", In *Proceedings, 1994 IEEE International Conference on Robotics and Automation*, pp. 1449–1454 (1994)

[4] G. C. Burdea, "A Portable Dextrous Master with Force Feedback", Presence, vol. 1, no. 1, pp. 19–28 (1992)

[5] J. E. Colgate and J. M. Brown, "Factors Affecting the Z-Width of a Haptic Display", In *Proc, 1994 IEEE Int. Conf. on Robotics and Automation*, pp. 3275–3210 (1994)

[6] D. B. Gennery, "Visual Tracking of Known Three-Dimensional Objects", Int. J. of Computer Vision, vol. 7, no. 3, pp. 243–270 (1992)

[7] P. E. Gruenbaum et al., "Implementation of Robotic Graphics For a Virtual Control Panel", In *VRAIS-95 Video Proceedings* (1995)

[8] M. Hammerton and A. H. Tickner, "Transfer of training between space-oriented and body-oriented control situations", British Journal of Psychology, vol. 55, no. 4, pp. 433–437 (1964)

[9] K. Hirota and M. Hirose, "Simulation and Presentation of Curved Surface in Virtual Reality Environment Through Surface Display", In *Proc., VRAIS'95*, pp. 211–216 (1995)

[10] H. Iwata, "Artificial Reality with Force-Feedback: Development of Desktop Virtual Space with Compact Master Manipulator", Computer Graphics, vol. 24, no. 4, pp. 165–170 (1990)

[11] T. Kotoku, K. Komoriya and K. Tanie, "A Force Display System for Virtual Environments", In *Proc., IEEE Int. Workshop on Robot and Human Communication*, Tokyo, Japan, 1-3 September, 1992.

[12] J. Lloyd and V. Hayward, "Multi-RCCL User's Guide", McGill University (1992)

[13] D. G. Lowe, "Robust Model-Based Motion Tracking Through the Integration of Search and Estimation", International Journal of Computer Vision, vol. 8, no. 2, pp. 113–122 (1992)

[14] W. A. McNeely, "Robotic Graphics: A New Approach to Force Feedback for Virtual Reality", In *Proc., VRAIS'93*, pp. 336–341 (1993)

[15] J. P. Rolland et al., "Quantification of Adaptation to Virtual-Eye Location In See-Thru Head-Mounted Displays", In *Proc., VRAIS'95*, pp. 56–66 (1995)

[16] S. E. Salcudean and T. D. Vlaar, "On the Emulation of Stiff Walls and Static Friction with a Magnetically Levitated Input/Output Device" In *Proc., International Mechanical Engineering Congress and Exposition*, Chicago, November, 1994, pp. 303–309 (1994)

[17] S. Tachi et al., "A Machine that Generates Virtual Haptic Space", In *VRAIS-95 Video Proceedings* (1995)

[18] M. Uenohara and T. Kanade, "Vision-Based Object Registration for Real-Time Image Overlay", Computers in Biology and Medicine, vol. 25, no. 2, pp. 249–260 (1995)

[19] Y. Yokokohji, R. L. Hollis, and T. Kanade, "What You can See Is What You can Feel - Development of a Visual/Haptic Interface to Virtual Environment-", In *Proc., IEEE Virtual Reality Annual International Symposium (VRAIS'96)*, pp. 46–53 (1996)