Temporal Registration using a Kalman Filter for Augmented Reality Applications*

Christopher Jaynes and Jing Hou
Department of Computer Science
University of Kentucky
Lexington, KY 40506
jaynes@metaverselab.org, www.metaverselab.org

Abstract

Augmented Reality uses see-through head-mounted displays to superimpose synthetically generated information on a three-dimensional scene. Information is rendered in alignment with physical objects to enhance the user's ability to perceive and interact with the world. A significant technical challenge related to Augmented Reality is determining the transformation that will correctly align the synthetic data with corresponding physical objects. This is made more difficult by the fact that the user and the scene may be in motion. Alignment must be both stable and accurate in order to produce the "illusion" that synthetic objects are an integral part of the environment. This paper presents a tracking algorithm capable of computing the three-dimensional pose of objects with respect to a head-mounted camera while both camera and object are in motion in the scene.

Using a fixed transform from the head-mounted camera to a see-through device, information can then be displayed in alignment with the current view of the object being tracked. In contrast to approaches that solve for the absolute position of the viewer within a fixed geometry [2,16], this object-centered approach to Augmented Reality models motion of both the object and the cameras-display system mounted on the user's head using a single transformation involving six parameters.

The object to be augmented is first recognized as one of the stored models in the database. Then feature points are selected and their corresponding world coordinates are extract from the object model automatically. Camera is calibrated on the base of the correspondences of the image coordinates and the world coordinates of the feature points. The intrinsic parameters (focal length and aspect ratio)are supposed to be constant once calibrated. Feature points are tracked by variable template matching so that the extrinsic parameters (pose) can be estimated in each frame. Accurate pose information is maintained using a Kalman filter that combines the predictions of a state-model and the pose of the object computed from tracked feature points. The technique is demonstrated on a database of mechanical prototype parts, using a head-mounted camera system. We explore the utility of the system in a manufacturing environment where an inspector requires access to CAD information related to a prototype part in order to facilitate qualitative decisions about the object. Results demonstrate that the algorithm is capable maintaining accurate temporal registration between the object and head-mounted system at nearly 22Hz on a Sun Ultra Sparc-20 Workstation.

*This work was funded by Honda Research and Development, Americas under the Honda Initiation Grant Program 1998 and the National science Foundation, contract No. EIA-9818332

1 Introduction

See-through display devices offer the unique ability to display information to a user who continues to perceive the environment. The primary use of these devices is to overlay or composite rendered objects with the real scene in order to generate an integrated, augmented view for the user. In this way, a user is able to access information related to the scene currently in view in a straightforward way. In addition, synthetic objects can enhance the user's ability to understand, navigate, and interact with the real world. Examples include rendering detailed product information in alignment with objects in a department store, display of virtual signposts that are specific to a particular drivers need, and the overlay of map and topological data needed by archeological researchers in the field.

There are a number of different researchers who have recognized the potential for Augmented Reality applications and have developed systems specific to several domains such as mechanical repair [4,14], virtual display of desktop environments [7, 11], technical guides projected onto complex wiring diagrams for aircraft [6,13], Augmentation of the external view of the womb with the three-dimensional ultra-sound data to provide doctors with a form of vision [4], and Image Guided Surgery in which preoperative surgical plans are transfered to the operation room and graphically overlaid onto video images of the patient[19]. Most of these applications do not solve for the relative position of the viewer with respect to the particular object to be augmented but assume that the world is static and to solve for the absolute position of the viewer in the world.

The problems related to the accurate alignment of information with the current scene are significant. In particular, the transformation that aligns the display system with the physical scene must be recovered in order to correctly augment the user's view. This is made more difficult by the fact that the user and the scene may be in motion. Alignment must be both stable and accurate in order to produce the "illusion" that synthetic objects are an integral part of the environment.

This paper presents a tracking algorithm capable of computing the three-dimensional pose of objects with respect to a head-mounted camera as they are moved in the scene. Using the fixed transform from the head-mounted camera and the see-through device, information can then be displayed in alignment with the current view of the object being tracked. In contrast to approaches that solve for the absolute position of the viewer within a fixed geometry [2,16], this object-centered approach to Augmented Reality models motion of both the object and the cameras-display system mounted on the user's head using a single transformation involving six parameters. This relative transformation explicitly encodes the mapping from the object reference frame to that of the camera at an instantaneous moment. As the object and camera move with respect to one another, this transform is updated according to observed motion and a state prediction.

Objects to be augmented must first be recognized as one of several stored within a database. Recognition of mechanical objects has been the subject of our work in the past [9,10] as well as that of other researchers [20]. In this paper, we focus on the robust tracking of a recognized object and encourage the interested reader to refer to [10] for more detail on the mechanical part recognition problem. Once recognized, appropriate feature points are extracted from the image for tracking and auxiliary information, associated with the object model is rendered into the current view. Accurate pose information is maintained using a Kalman filter that combines the predictions of a state-model and the pose of the object computed form tracked feature points and a three-dimensional pose recovery algorithm. The technique is demonstrated on a database of mechanical prototype parts, using a head-mounted camera system.

We explore the utility of our algorithms in a manufacturing environment where an inspector requires access to CAD information related to a mechanical part in order to facilitate both qualitative decisions about the object. The accurate design and manufacture of mechanical parts is a critical component of the engineering process. Typically this process involves a computer-aided design (CAD) stage followed by the creation of a physical prototype. The prototype must be inspected in order to guarantee that the design meets particular tolerances, the manufacturing process is capable of producing a part to specification, and to insure that the designed object will satisfy the needs of the overall system.

Due to the costly and critical nature of prototype verification, there have been a number of different attempts automate the process using active vision sensors [2,5,8,10]. Our approach is to make use of the inspector's inherent knowledge while improving his/her ability to make decisions through the seamless integration of auxiliary information and the inspector's current view of a prototype part. Object-centered Augmented Reality approach involves three image processing components: 1) automatic part recognition, 2) object tracking, and 3) the recovery of the relative position, or pose, of the part with respect to the viewer. This paper focuses on temporal registration of the object with the display device. The next section introduces our approach to temporal registration using a Kalman filer which includes a prediction phase, a feature tracking component. Experimental results are presented in section 3. Section 4 makes a conclusion and states the majority of our research in the future.

2 Temporal Registration Using a Kalman Filter

The purpose of temporal registration is to maintain an accurate estimate of the object pose over time. The computation of the relative object-sensor geometry is critical to the correct annotation of the scene through the user's display. Information retrieved from the CAD database such as material density overlay, through-hole widths, and error tolerances must be presented in correct alignment with the object under inspection in a meaningful and unambiguous manner.

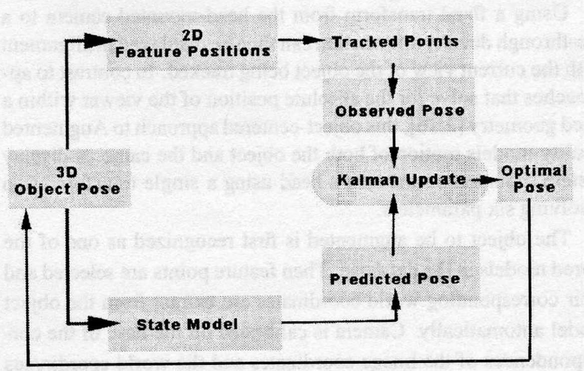


Figure 1: General temporal registration approach using a Kalman filter. Pose in each frame is both predicted by a state model and computed from tracked 2d feature points and their known corresponding 3d positions on the object.

The tracking algorithm requires that the initial correspondence of image features and model points on the object is known. For the experimental system presented in this paper, this initial object pose is supplied by an object recognition algorithm that determines the correct alignment of features from a CAD model and the current view. Recognition is not the focus of the work here and the interested reader is referred to [10] for a full treatment of the algorithm.

Given an initial pose of the object, the system maintains temporal registration through the combination of a state model that predicts that motion of the object and the observed pose that is derived from tracked features in the image. A Kalman filter [11] incorporates information about the error in both the prediction and the observations to derive an optical pose estimate for the new frame. The overall approach is depicted in Figure 1.

2.1 State Model and Pose Prediction

In general, the Kalman filter incorporates information from a state model of a physical system with observations of that system. Given information about the accuracy of both the model and observations, an optimal state of the system is estimated over time. Initially, we model the relative position and orientation of the object with respect to the camera as the state vector:

$$X = [xyz\widehat{xyz}w\phi\kappa\widehat{w\phi\kappa}] \tag{1}$$

Where x, y, z is the displacement vector from the camera's coordinate to the object and w, ϕ, κ are three rotation angles that encode relative rotation of the object frame to the camera. Velocity of position and orientation (denoted with a)is also represented in system state vector. In the absence of other information, such as inertial models of the object and arm displacing it (see section 5), we model the motion of the object as random, white noise accelerations with zero mean. This yields a 12×12 state update matrix Θ, corresponding to the 12-element state vector, with the ith element given by:

$$\Theta_i(t) = \begin{bmatrix} x_i(t-\delta t) + \widehat{x}_i(t-\delta t) \\ \widehat{x}_i(t-\delta t) + \varepsilon(t-\delta t) \end{bmatrix} \tag{2}$$

Where $\varepsilon(t-\delta t)$ is a white noise process noise vector (of 12 elements) with a normal distribution and zero mean. Given the error model, a error covariance matrix Q(t) is computed as

$$Q(t) = E[\varepsilon(t-\delta t)\varepsilon(t-\delta t)^T] \tag{3}$$

The predicted motion of the object then is computed by multiplying the state update matrix with the current state vector derived from the initialization of the tracking algorithm, or the previous optimal estimate provided by the filter. A new error covariance is the estimated and used by the filter to attenuate the contribution of the predicted state in the 3D object pose.

2.2 Two-dimensional Feature Tracking

We use a template-based approach to track object feature points along the object contour that are visible and known to be stable. In general our tracking algorithm is a both 2D and 3D approach. Feature points are matched in subsequent frames via a local, 2D image search. However, the position and size of search regions is computed by projecting a three-dimensional region about each 3D feature based on the covariance error and optimal pose estimate provided by the Kalman filter (See equation (15) in section 2.4). This error covariance is determined by the Kalman filter through a weighted combination of predicted state, observed state and known error (for details see section 2.4).These search regions are then correlated with the appropriate templates to provide a new measurement and are sufficient to compute a new three-dimensional pose estimate for the process to continue. Figure 2 shows conceptually how the 2D search regions are determined based on the known 3D error in the current position of the object with respect to the camera.

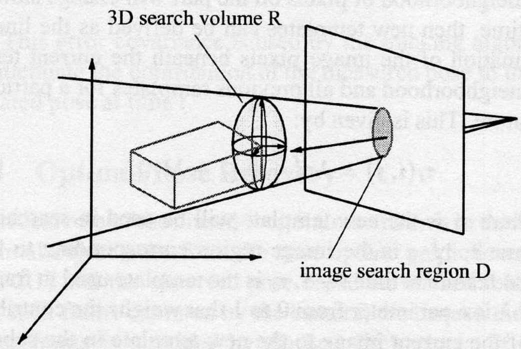


Figure 2: Error about each feature is derived from the state covariance matrix P, computed by the predictive filter (See equation (15) in section 2.4).This error, represented as a three-dimension space region R whose size along its axis is determined by the eigenvalues of P, is perspectively projected onto the image to determine an appropriate search region of size D for the feature in the next frame.

Subsequent frames are searched for the position of each new template. Each template,τ , is compared to the image at local positions i and j through a correlation measure given by:

$$\tau \times I(i,j) =$$

$$\frac{2 * \sum_{m,n} \tau(m,n) * I(i-m, j-n)}{N_1 * \sum_{m,n} I(i-m, j-n)^2 + N_2 * \sum_{m,n} \tau(m,n)} \tag{4}$$

$$N_1 = \sum_{m,n} \frac{\tau(m,n)}{I(i-m, j-n)}$$

$$N_2 = \sum_{m,n} \frac{\tau(i-m, j-n)}{T(m,n)}$$

This measure is similar to the standard cross-correlation score but has properties of steerable filters [8]. Normalized cross-correlation degrades under relative rotation between the image and the template. However, the measure above is shift-invariant in Cartesian space and hence is invariant for 2D rotations. The measure will degrade for significantly non-planar rotations, however, and we must account for this by tracking only the features that are stable in the current viewing aspect. Note that appropriate features to track can be determined from the correct known pose and the object model itself. For example stable features such as corners that are both visible from the camera and are moving in a approximate plane are more desirable than features likely to disappear from view.

As the part is repositioned in the scene, radiometric properties of the scene will change the appearance of a tracked feature in the image. If we assume that the appearance of a fixed neighborhood of pixels on the part will change slowly over time, then new templates can be derived as the linear combination of the image pixels beneath the current template neighborhood and all previous templates for a particular feature. This is given by:

$$\tau_t(i,j) = \lambda \tau_{t-1}^I + (1-\lambda)\tau_{t-1} \tag{5}$$

where τ_t is the new template will be used in searching in frame t, τ_{t-1}^I is the image region corresponding to the tracked feature at time $t-1$, τ_t is the template used in frame t, and λ is a parameter from 0 to 1 that weighs the contribution of the current image to the new template in the subsequent frame. For the experiments shown here, lambda was fixed at 0.89. For non-lambertian surfaces and out of plane rotations, image intensity changes nonlinearly. We are exploring more sophisticated predictive adaptive techniques to address these issues (see section 4).

2.3 Relative Pose Recovery

Given a set of tracked points in the image, the three-dimensional rotation and position of the object, with respect to the camera, must be recovered. This observed pose is then used by the full object tracking algorithm to compute and optimal pose of the object based on known error, and the predicted position of the object based on the state model (see Figure 1). Using the relative position of the camera and object, information can be rendered into the view based on this known viewing transform. It is important to note, however, that the geometry of the display device and the camera must be known. In particular, we envision a system in which the positions of the see-through display, and head-mounted cameras are rigidly fixed and precomputed. The goal of the pose determination algorithm, then, is to robustly compute the unknown position of the object as it is repositioned during the inspection process.

Using the known model to image correspondences, we have developed a pose registration algorithm that is capable of computing the observed object pose, in the closed-loop predictive tracking system at 22Hz on a Sun Ultra Sparc-10 workstation (see Section 3). Pose is represented as a 3x3 rotation matrix R, whose entries are written as $r_{i,j}$, and a displacement vector T= [Tx, Ty,Tz]. In addition, the effective focal length f_x (focal length divided by the horizontal pixel size) and the aspect ratio α of the lens must be known in order to compute accurate pose. Point correspondence yield

$$x = -f_x \frac{r_{11}X^w + r_{12}Y^w + r_{13}Z^w + T_x}{r_{31}X^w + r_{32}Y^w + r_{33}Z^w + T_z} \tag{6}$$

$$y = -f_x \frac{r_{21}X^w + r_{22}Y^w + r_{23}Z^w + T_x}{r_{31}X^w + r_{32}Y^w + r_{33}Z^w + T_z} \tag{7}$$

Since (7) (8) have the same denominator and $\alpha = f_x/f_y$, for each pair of correspondence, we have an equation for the eight unknowns $V = [v_1, v_2, ..., v_8]$:

$$x_i X_i^w v_1 + y_i Y_i^w v_2 + x_i Z_i^w v_3 + x_i v_4 - y_i X_i^w v_5$$
$$-y_i Y_i^w v_6 - y_i Z_i^w v_7 - y_i v_8 = 0 \tag{8}$$

where,

$$v_1 = r_{21} \qquad v_5 = \alpha r_{11}$$
$$v_2 = r_{22} \qquad v_6 = \alpha r_{12}$$
$$v_3 = r_{23} \qquad v_7 = \alpha r_{13}$$
$$v_4 = T_y \qquad v_8 = \alpha T_x$$

This leads to a system of N linear equations of the form

$$AV = 0 \tag{9}$$

for N correspondences between image points (x_i, y_i) and model points (x, y, z). Given greater than seven corresponding point pairs that are not coplanar, the system (8) has a nontrivial solution which can be determined from the SVD of A.

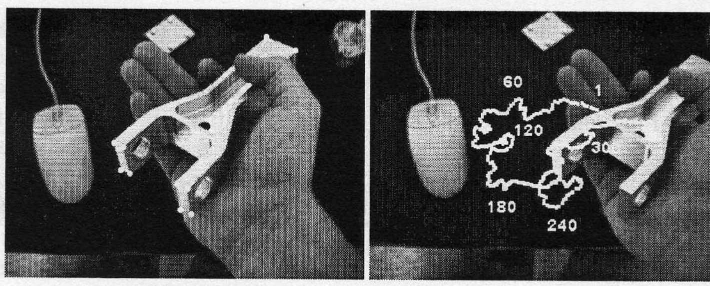
The rotation matrix (R), the first two elements (T_x, T_y) of the translation vector and aspect ratio α are computed from V under the orthogonality constraint of rotation matrix R: $RR^T = I$, $r_{11}^2 + r_{12}^2 + r_{13}^2 = 1$, $r_{21}^2 + r_{22}^2 + r_{23}^2 = 1$ and the third row of R is the vector product of the first two rows thought of as 3-D vectors.

Finally, compute Tz, and fx from a least-squares solution of equation (6) which can be rewritten

$$x_i(r_{31}X_i^w + r_{32}Y_i^w + r_{33}Z_i^w + T_z) =$$
$$-f_x(r_{11}X_i^w + r_{12}Y_i^w + r_{13}Z_i^w + T_x) \tag{10}$$

by solving the over-constrained system of N linear equations

$$A \begin{pmatrix} T_z \\ f_x \end{pmatrix} = b \tag{11}$$



(a) (b)

Figure 3: Example of 2D tracking. (a): Tracking points initialized by the object recognition scheme and used for temporal registration shown overlaid in first frame of tracking sequence. (b): Motion of part over 450 frame tracking sequence. Motion is depicted as the object's center of mass (derived from the 3d tracker) re-projected into the final frame. Numbers indicate frame numbers along the motion path.

Our approach differs from the traditional calibration problem in that the effective focal length and aspect ratio are computed to minimize error in the first several frames of the tracking sequence. Once computed, these intrinsic parameters remain fixed, and improve the robustness of the tracker by reducing the number of parameters to be estimated in each subsequent frame.

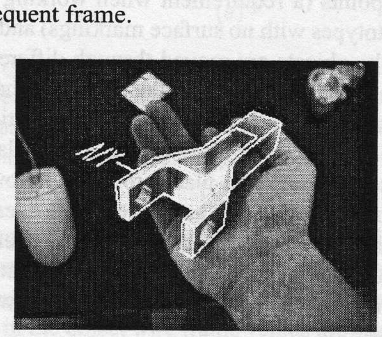


Figure 4: CAD model re-projected into a single frame based on the recovered viewing transform. Text at left represents the part label associated with the object and is projected into the view in alignment with the object.

Finally, the measured pose is represented as a measurement vector z , that, in principle, has measured the state of the system via the measurement matrix:

$$z(t) = H(t)X(t) + \eta(t) \tag{12}$$

where, $\eta(t)$ is the measurement error vector. This measurement error is related to the accuracy of the 2D tracking algorithm and is computed as the inverse of the mean of the maximum correlation values, multiplied by a scale factor:

$$\eta_t = \lambda\left(\frac{\sum_i Max[\tau_i \times I(m,n)]}{i}\right) \tag{13}$$

The covariance of the measurement error at time t is given by:

$$R(t) = E\left\lfloor \eta(t)\eta(t)^T \right\rfloor \tag{14}$$

This error covariance is used by the tracking algorithm to attenuate the contribution of the measured pose to the estimated pose at time t.

2.4 Optimal Pose Recovery

A critical role of the closed-loop Kalman filter is to derive an optimal object pose from the state update prediction (Section 2.1) and the observed pose based on tracked features. For each frame captured by the camera, the time to the previous frame is provided by the host PC's clock, δt. For each discrete time interval, three-dimensional pose parameters are computed via the state update matrix. A state error covariance Q is computed (see equation (3) in Section 2.1). In addition, the tracking and pose recovery algorithms provide an observed state and provide the corresponding error covariance matrix R.

Given these, an overall error covariance is measured by propagating the previous error through the system state combined with the current predicted state covariance:

$$\widehat{P}(t) = \Theta(t-\delta t)P(t-\delta t)\Theta(t-\delta t)^T + Q(t) \tag{15}$$

This error is combined with the measurement covariance, to compute the Kalman gain of the controller:

$$K(t) = \widehat{P}(t)H(t)^T[H(t)\widehat{P}(t)H(t)^T + R(t)]^{-1} \tag{16}$$

The kalman gain, weights the contribution of the measured pose with that of the predicted pose of the object based on the covariance measures to derive the optimal pose estimate.

$$\widehat{X}(t) = \Theta(t-\delta t)\widehat{X}(t-\delta t) + K(t)[z(t) - H(t)\Theta(t-\delta t)\widehat{X}(t-\delta t)] \tag{17}$$

For example, in cases where correlation scores are high, the measurement covariance, in turn, will be small leading to a large contribution to the optimal state form the tracked points and recovered pose. In the inverse case, the tracker may provide inaccurate measurements and the filter will attenuate the contribution of the measured pose to the optimal estimate. In this case, the optimal pose will be a small, random change in the position of the object (based on the random acceleration state model).

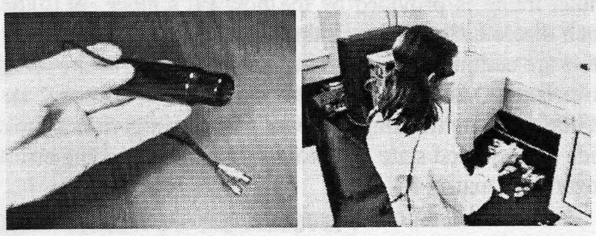
Finally, the filter computes a new overall covariance, required by the next iteration of the sequence :

$$P(t) = (I - K(t))\hat{P}(t-\delta t)(I - K(t))^T + K(t)R(t)K(t)^T \quad (18)$$

Empirically, the use of the filter improves the overall temporal registration algorithm dramatically. Figure 4 depicts a single frame from a tracking sequence. A wire frame model and part identification text are re-projected onto the object based on the optimal pose provided by the filter.

3 Experimental Results

A simple prototype system was developed in order to test the accuracy of the Augmentation system. A lightweight camera, shown in Figure 6a, was connected to a simple bracket and headstrap to be worn by a user. The camera provides a 320x240- resolution image and was tethered to a frame grabber in a host PC for image capture. The overall setup of the system is depicted in Figure 6b.



(a) (b)

Figure 6: Prototype system used in experiments. (a) Mini video camera is lightweight and inexpensive. (b) Overall setup. A user wears the camera that is tethered to a frame grabber in a host PC. Objects can then be selected and inspected in the users work environment. Augmentation is currently overlaid to the digital video on the workstation and not a see-through display.

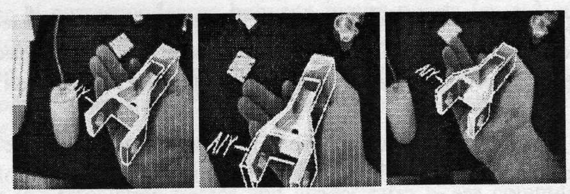
In order to isolate errors in the temporal registration system, a see-through display was not used in the experiment. The wire-frame CAD model corresponding to recognized parts was projected into each frame of the incoming video based on the relative transform of the object and camera. In order to use a see-through system, an additional, fixed transformation that maps coordinates in the image frame to that of the display device would be precalibrated.

System accuracy was tested using eight different mechanical objects. The objects are representative of mechanical prototypes that are manufactured for inspection and refinement purposes. Each of the eight objects were modeled in a CAD database. In addition to the geometric CAD model, information about appropriate tracking features was also stored. The object recognition algorithm (see [10]), supplies the tracking algorithm with the location of these features once the part has been recognized and initializes the temporal registration process.

The user was instructed to select objects from random on the worktable and move them in an approximately planar manner for 10-20 seconds and then place the object back on the table. The process is repeated until each object was selected by the user, recognized, tracked, and then replaced.

Accuracy of the system was measured by re-projecting the CAD wire frame into the video stream on the workstation using the recovered relative pose. Figures 7 and 8 depict the general behavior of the system. By showing single frames near the start, middle, and end of the sequence. In order to best understand the behavior of the system, the reader is referred to www.metaverselab.org/ research/augmented_reality/ in order to view sample video tracking sequences.

The system is capable of tracking objects using only a few feature points (a requirement when working with mechanical prototypes with no surface markings) and performs robustly as the objects are moved through different illuminations. A primary restriction of the system is that it is unable to robustly track full three-dimensional rotations of the object. This is due to the failure of the 2D template matching technique to model non-planar rotations. However, the Kalman framework that we have presented does not preclude the introduction of a mechanism that will select which features should be tracked based on the current state vector and its derivatives. This is a major focus of our work in the future (see Section 4)



(a) (b) (c)

Figure 8: (See www.metaverselab.org/research/augmented-reality for video). Three frames from a tracking sequence of the Y-bracket part (approximately 20 seconds long). Augmentation can consist of any information that is registered with the CAD models coordinate frame. Here the model is augmented with the wire frame and an object label that was stored in the database. (a) Frame near the start of the sequence. (b) Middle frame. (c) Frame near the end as the user begins to place the object on back on the table.

In addition to understanding the general behavior of the system through observing the augmented video stream, the accuracy of the temporal registration algorithm was characterized by measuring the actual image overlap of the pro-

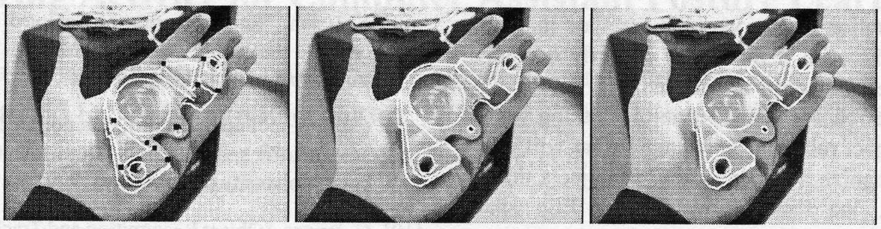


Figure 5: Temporal registration using the 3D Kalman filter. (a) 2D Template tracking may temporarilly miscorrelate due to clutter, severe lighting changes, and occlusions. This results in a pose estimate that is incorrect. (b) The predicted object poseis based on the state model and previous motion of the part. (c) Optimal estimate of pose generated by the Kalman filter algorithm. The estimate is a nonlinear combination of the observed and predicted pose and takes potential tracking error into account.

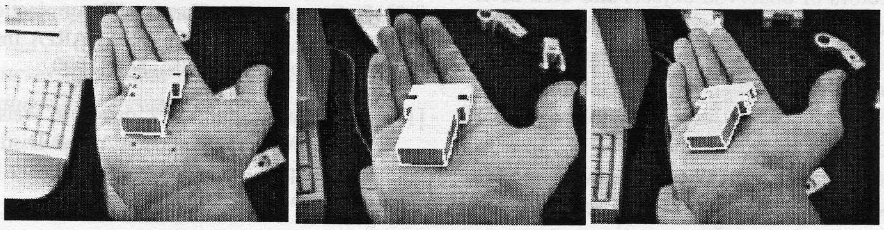


Figure 7: (See www.metaverselab.org/research/ augmented-reality for video). Three frames from a tracking sequence of the T-Bracket object. (a) The object was recognized in the 23 frame. Small white squares (on the object corners) indicate the measured location of tracked features. Grey squares indicate the predicted location of the features in the subsequent frame. Because the object was just recognized, the Kalman filter has not yet converged and the predictions are far from measured feature locations. (b) Middle frame of the sequence. (c) Frame near the end of the sequence.

jected wire frame and object in every 10th frame of the tracking sequence. An overlap percentage was measured by counting the number of object pixels that were correctly contained in the object wire frame versus the number of pixels that were either outside of the wire frame, or in the case that the projected wire frame model is too large (for example Figure 8b), the number of pixels within the wire frame that were not the object.

The measure was computed for every 10th frame in a tracking sequence where the object was recognized and tracked. A mean value was computed by averaging the overlap values across for a single object. Table 3 below reports these results for 10 complete tracking sequences. The table reports the worst sequence, the best sequence and the mean overlap percentage achieved by running the experiment over 10 trials.

Object Name	Overlap % (Worst)	Overlap % (Best)	Overlap % (Mean)
Y-Bracket	0.793	0.969	0.958
H-Plate	0.694	0.873	0.732
Mount-Plate	0.703	0.932	0.860
Cylinder-Cap	0.882	0.952	0.911
Screw-Plate	0.794	0.950	0.923
T-Bracket	0.910	0.989	0.984
L-Bar	0.899	0.973	0.952
End-Plate	0.914	0.976	0.958

Table 1: Worst, best, and mean overlap percentage over 10 tracking trials.

4 Conclusions and Future Work

Our initial experiments have demonstrated the feasibility of the Object-Based Augmented Reality paradigm. By making use of viewing constraints and information contained in the CAD database (such as the aspect graph) we are able to both recognize and track a class of objects without the complex calibration assumptions made in earlier work [8]. An additional advantage is that we compute the relative pose of the object with respect to the sensor, rather than an absolute viewing position in each frame. Using the approach, motion of both the users view and the object itself can be modeled as six parameters. This approach avoids problems encountered by other researchers that attempt to estimate absolute position[18] by assuming a static scene, or modeling object motion independently.

An integrated tracker that dynamically determines 2D image search region based on the 3D error covariance for a particular feature was introduced. This tracker is insensitive to planar rotation due to variable template matching algorithm. It works even if the object is partly occluded, since only a few feature points are needed.

A primary research objective within the coming year is to explore new object recognition strategies that will allow the system to recognize a wider variety of objects than is currently possible. We plan on developing an Eigenspace approach to mechanical object recognition by constructing a classification space based on the features present in mechanical objects such as drill-holes, corners, and beveled edges. The space is then reduced to a set of orthogonal vectors that are capable of representing the set of objects in any given database. Recognition, using this approach, is based on building feature vectors of observed objects and then determining their position within the classification space. In addition, we are investigating the use of a stereo head-mounted device that provides three-dimensional information, to improve our recognition algorithms based on an analysis of the differential properties of visible surfaces. A more accurate, scalable, and robust recognition system will improve the tracking results by providing better initial feature points to be tracked.

The majority of our future work focuses on the improvement of the tracking algorithm to include full, full-dimensional relative rotation of the object with respect to the view. This will involve a control mechanism that selects appropriate feature points, based on the known 3D pose, to track as the current points are rotated out of plane.

References

[1] P. Allen. Robotic Object Recognition Using Vision and Touch, Kluwer Academic Publishers, Norwell MA, 1987.

[2] R. Azuma, "Tracking Requirements for Augmented Reality", CACM 36, 7 (July 1993), 50-51.

[3] M. Berger. "Resolving Occlusion in Augmented Reality: a Contour Based Approach without 3D Reconstruction", Proc. Computer Vision and Pattern Recognition, pp. 91-96, 1997.

[4] P. Caudell, and D. Mizell, "Augmented Reality: An application of Head-Up Display Technology to Manufacturing Processes." Proc. Of the Hawaii International Conference on System Sciences, 1992.

[5] S. Dickenson, P. Jasiobedzki, G. Olofsson, and H. Christensen. "Qualitative Tracking of 3-D Objects using Active Contour Networks" Proc. of Computer Vision and Patten Recognition, pp 269-280, 1990.

[6] M. Friedmann, T. Starner, and A. Pentland, "Device Synchronization Using an Optimal Filter", Proc. 1992 Symposium on Interactive 3D Graphics

[7] D. Gennery. "Tracking known three-dimensional objects", International Journal of Computer Vision, 7(3):243-270, 1992.

[8] Y. Hsieh. "Reconstruction of Sculpted Surfaces using Coordinate measuring machines" Masters Thesis, Mechanical Engineering Dept., University of Utah, June 1993.

[9] C. Jaynes. "Fast Feature Extraction of Mechanical Parts in Motion" Technical Report, Department of Computer Science. University of Kentucky, 1999.

[10] C. Jaynes, "Object Recognition and Tracking for Augmented Reality Design", Proc. 1998 Honda Research Symposium, Columbus, OH, 1998.

[11] R. E. Kalman, and R. Bucy, "New Results in Linear Prediction Theory", Trans. ASME, J. Basic Eng. Series 83D, 95-108, Mar. 1961.

[12] R. Kumar and A Hanson. "Determination of Camera Position and Orientation", Proc. DARPA Image Understanding Workshop, pp. 870-881, May 1990.

[13] S. Ravela, B. Draper, J. Lim, and R. Weiss. "Adaptive Tracking and Model Registration Across Distinct Aspects". Technical Report, Department of Computer Science, University of Massachussetts, 1996.

[14] D. Sims, "New Realities in Aircraft Design and Manufacture", IEEE Conputer CG&A, 14,2 1994.

[15] T. Sobh, J. Owen, C. Jaynes, M. Dekhil, and T. Henderson. "Industrial Inspection and Reverse Engineering", IEEE 2nd CAD-Based Vision workshop, Feb. 1994.

[16] M. Ward, R. Azuma, R. Bennet, S. Gottschalk, and H. Fuchs, "A Demonstrated Optical Tracker with Scalable Work Area for Head-Mounted Display Systems", Proc. 1992 Symposium on Interactive Computer Graphics, pp. 43-52, Cambridge, 1992.

[17] Z. Zhang, Iterative point matching for registration of free-form curves and surfaces." Int. Journal of Computer Vision, 13(2):119-152, Dec. 1994.

[18] Welch, G., G. Bishop, L. Vicci, S. Brumback, K. Keller, D. Colucci, "The HiBall Tracker: High-Performance Wide-Area Tracking for Virtual and Augmented Environments," Proc. of the ACM Symposium on Virtual Reality Software and Technology 1999 (VRST), London, Dec. 20-22, 1999.

[19] W.E.L. Grimson, G.J. Ettinger, S.J. white, T. Lozano-Perez, W.M. Wells III, R. Kikinis, "An Automatic Registration Method for Frameless Stereotaxy, Image Guided Surgery, and Enhanced Reality Visualization",Trans. on Medical Imaging, 1996.

[20] Joachim M. Buhmann, Jitendra Malik and Pietro Perona, "Image Recognition: Visual Grouping, Recognition and Learning", Proc. National academy of Science, Vol. 96, No 25, pp. 14203-14204, Dec. 7, 1999