Real Time Tracking of Borescope Tip Pose

Ken Martin, Charles V. Stewart*
Kitware Inc., Clifton Park, NY 12065
*Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180
ken.martin@kitware.com, stewart@cs.rpi.edu

Keywords

pose estimation, borescope inspection, industrial inspection, lens distortion

Abstract

In this paper we present a technique for tracking borescope tip pose in real-time. While borescopes are used regularly to inspect machinery for wear or damage, knowing the exact location of a borescope is difficult due to its flexibility. We present a technique for incremental borescope pose determination consisting of off-line feature extraction and on-line pose determination. The off-line feature extraction precomputes from a CAD model of the object the features visible in a selected set of views. These cover the region over which the borescope should travel. The on-line pose determination starts from a current pose estimate, determines the visible model features, and projects them into a two-dimensional image coordinate system. It then matches each to the current borescope video image (without explicitly extracting features from this image), and uses the differences between the predicted and matched feature positions in a least squares technique to iteratively refine the pose estimate. Our approach supports the mixed use of both matched feature positions and errors along the gradient within the pose determination. It handles radial lens distortions inherent in borescopes and executes at video frame rates regardless of CAD model size. The complete algorithm provides a continual indication of borescope tip pose.

1. Introduction

A borescope is a flexible tube which allows a person at one end of the tube to view images acquired at the other end. This can be accomplished with coherent fiber bundles running through the tube, or a small video camera located at the tip of the borescope and a video monitor at the other end. Borescopes are typically a centimeter or less in diameter, a few meters long, and used to inspect inaccessible regions of objects. For example, when inspecting the internal structure of a jet engine for cracks or fatigue, small openings from the outside allow the borescope to be snaked into the engine without having to drop the engine from the plane. In such an inspection, it is often difficult for the inspector to know the exact borescope tip location within the engine, making it difficult to identify the location of new cracks found or to return to previously identified trouble spots. When a crack is found, the inspector must measure its length and width which is problematic if the borescope's pose is unknown.

Traditional, non-video techniques for tracking an object do not work well within the environment a borescope typically encounters. Electromagnetic fields are corrupted by the surrounding presence of metal. Acoustic techniques are susceptible to ringing caused by metal parts and convoluted passages, and any technique that relies on a clear line of sight is by definition implausible in the borescope inspection environment. The only option is tracking based on the borescope images themselves.

2. Approach

Borescope tip tracking is essentially the camera pose problem with added constraints and difficulties. First, the light source for a borescope is collocated with the camera, so the light is moving but always coincident with the camera. Second, object recognition is non-standard: while the complete object is known, the position within the object (the jet engine) is not and only a small fraction of the complete object will appear in any single image. Third, the object contains structural repetition, making an approach based solely on dead reckoning unrealistic. Fourth, tracking (pose estimate updates) must occur at near frame-rates, preferably without specialized hardware. Combined, these constraints make borescope tip tracking a novel and challenging problem.

Our solution to these problems is a two fold approach consisting of off-line feature extraction from a CAD model and on-line pose estimation. The off-line extraction precomputes from the CAD model 3D edges consisting of

a 3D position and normal direction in the coordinate system of the CAD model. These features are computed for local regions of the model and consider visibility constraints. This off-line process need only be done once for a given CAD model. The on-line process starts from a known landmark and compares the precomputed features for that region of the model to the borescope video. From this comparison an error vector is created which is used to compute an updated pose. This process repeats for each frame of video. The performance of the on-line algorithm is unaffected by the size or complexity of the CAD model.

Beyond borescope inspection, this approach would be of value for any model-based pose-estimation problem where the camera is placed within the model.

3. Related Work

While borescope tip tracking is a novel problem, there are a number of related techniques which could be considered. One approach is to subdivide the CAD model into many smaller parts and then determine the borescope pose based on standard object recognition strategies. Unfortunately, this approach, which is employed in a different context by Kuno, Okamoto and Okada[8], suffers from aperture problems caused by the structural repetition inside a jet engine. Further, in the time required for part recognition and associated pose estimation, the borescope may have moved too far for unique localization. By contrast, our algorithm determines pose incrementally based on the image locations of precomputed features, allowing it to run nearly at frame rate.

While camera pose and object pose have been treated extensively [5][12][15], existing techniques are difficult to apply in our situation. These techniques match 3D object features with a set of 2D features extracted from a single image. In contrast, since we must estimate pose in real time, we can not extract complicated image features and the matching process must be rapid. To accomplish this, we match simple 3D edgels to 2D intensity images. We use the current pose estimate to project these 3D edgels into the image and match by searching along the projected edgel's normal direction, usually deriving only one constraint per projected edgel. This implies that the match for a 3D feature moves and changes during pose determination, making our technique similar in spirit to other techniques that determine pose without specific point matches[16].

The first of such techniques is that of Branca [1] who presents a passive navigation technique based on calculating the focus of expansion (FOE). Essentially, Branca calculates the image movement vectors between two frames, performs an iterative error minimization to find a set of feature matches between the two frames, then calculates the FOE and camera translation from them. This approach uses planar invariants to find the point matches by making the condition of planarity part of the error minimization process.

This approach is certainly novel and it doesn't require a model to work from, but it is not suitable for borescope inspection. It requires planar surfaces with at least five features on them while borescopes typically inspect curved surfaces such as the inside of a jet engine. Furthermore it only calculates camera translations (not rotation), and it has no method to prevent the incremental errors from accumulating. This last point is critical in incremental pose estimation. Many techniques use interframe differences without any form of dead reckoning. The errors from each pose estimate accumulate and create stability problems and gross errors. This is part of the motivation behind using model based pose estimation. The model can be used for dead reckoning as long as its 3D features are static. This paper does not directly address deformable models.

Another paper worth noting is Khan's [7] paper on vision based navigation for an endoscope. This is a very similar problem to borescope navigation although Khan et. al. solve it in a different manner. Instead of starting with a MR or CT model of the patient's colon, they construct the model as they go. When the endoscope enters an uncharted region of the colon they start extending the model. Navigation is accomplished by using two features that are commonly found in colonoscopies: rings of tissue along the colon wall and a dark spot (the lumen) in the image where the colon extends away from the endoscope. This approach is real-time and once it has built the model it can support dead-reckoning. It requires no modification of current endoscope hardware or precomputed models. Unfortunately its choice of features restricts its usage to inspection of ribbed tubes.

Lowe presents a model based motion technique that goes beyond pose estimation to also handle models with limited moving parts [10]. His technique is essentially a modified hypothesize & verify search tree based on line segment matches. It uses probabilities from the previous pose estimate to order the evaluation of the possible matches; this significantly improves the typically excessive computational requirements. There are three significant drawbacks to such an approach. The first is that it relies on line segments as features. As will be demonstrated later, there are few line segments in a jet engine. Second, it requires complete feature extraction from the input image which is time consuming. In his application, dedicated image processing hardware was required and yet the resulting performance was limited to three to five frames per second. Finally, it is still sensitive to the number of potential features in the model. Results were

reported for a model consisting of only a few lines. Unfortunately, pose estimation inside a complex part typically results in millions of features in the model, nearly all of which will be obscured in a given view.

Some well known related work was performed by Dickmanns and his colleagues. His work has focused on real-time, model-based pose estimation and navigation. In an early paper [2] he describes a new approach based on measuring error vectors between predicted and actual feature locations, and then using these errors as input to modify the pose estimate. In later work the state estimation is handled by a Kalman filter [3] and also incorporates non vision based inputs such as airspeed [4]. The later paper presents a good overview of his technique.

There are two significant differences between Dickmanns' work and this work. First Dickmanns uses the CAD model to develop his control mechanism, i.e. his software, feature selection, and Kalman filter are tuned to a specific task with a specific model. His approach does not support supplying a general CAD model to work from. This customization can be used to reduce the computation load and improve robustness. For example, when landing an airplane, Dickmanns uses ten predefined image features that correspond to specific parts of a runway and the horizon. Incoming images are analyzed only at the predicted locations of those ten features. Likewise those ten features represent unique defined inputs to the Kalman filter. This paper does address the issue of automatically calculating a set of pertinent features from a CAD model, and it assigns no semantic meaning to those features.

The second difference is Dickmanns' choice of features. He uses constant features such as oriented edge templates which yield one constraint. Our paper considers the interpretation of a feature as providing zero to two constraints as appropriate. This is especially important as the features are automatically generated, not manually defined as in Dickmanns' work. This impacts the central pose estimation calculation which must be adapted to the number of constraints provided by the features. There are also some minor differences such as Dickmanns' work not incorporating wide angle lens distortions as found in a borescope, but most remaining differences would require minor modifications to either approach.

One of the most closely related approaches is that of Chris Harris which uses control points from the CAD model combined with a Kalman filter [6]. The first key difference is that he doesn't indicate how his control points are generated. It is implied that they are hand selected from the 3D model which is impractical in many situations. Second, since his approach is focused on estimating the pose of an object from the outside, its visibility tests for the control points are too limited for internal pose estimation. Finally, his approach deals with features as providing zero or one constraint. As will be discussed later, it is sometimes necessary to have features provide two constraints (position) instead of just a gradient vector displacement.

4. Feature Extraction

As summarized above, our approach consists of two primary pieces: off-line feature extraction, and on-line pose determination. Off-line feature extraction must take the CAD model, which could be greater than one gigabyte in size, and produce 3D features that can be loaded quickly based on a current pose. This process is critical because just traversing the CAD model in memory would consume more time than allowed for incremental pose estimation. The major components of the feature extraction process are depicted in Figure 1 and the process is summarized in the following five steps:

1. Based on the CAD model, a list of 3D sample points is generated over the 3D region where the borescope might travel. This could be either a uniform sampling of the 3D region completely enclosing the CAD model, or it could be a non-uniform sampling of all or part of the region.
2. At each sample point, computer graphics hardware is used to render a collection of 2D images of the CAD model from that location. The view directions of these images are selected to ensure that the collected images form a mosaic completely enclosing the location in question.
3. For each synthetic image generated in step two, edge detection is used to extract edgel chains having large intensity gradients, and then a fixed size subset of edgels are selected. The selected edgels have the largest intensity gradients but also satisfy a minimum pairwise image separation.
4. The 3D location and direction on the CAD model is computed for each selected edgel. These become the 3D features for the current sample point.
5. The process repeats for each image and sample point.

The first step is to determine where the sample points will be and how many of them will be used. This operation must generate enough sample points and 3D features that there will be one near any position encountered where the borescope travels. For the results presented here, the sample points were generated simply as a regular grid.

In the second step computer graphics hardware is used to render images of the CAD model. The eventual goal is to produce features useful during on-line pose estimation. The problem then becomes determining how to "predict" a useful feature from a CAD model. From the

perspective of the on-line algorithm, a useful feature will be one that can be used to provide an image movement vector. As such, image intensity discontinuities, or edges, will be likely candidates. One possibility is that range or normal discontinuities in the CAD model will form image intensity discontinuities in the video. This is often, but not always true, and some intensity discontinuities will arise solely from markings and changes in materials. A more general and more reliable approach is to use computer rendering to create a realistic image of what the part should look like. This can then be examined for intensity edges.

In practice, step two involves rendering six images for each sample point. These images are rendered with a view angle of ninety degrees and oriented along the positive and negative directions of the three cartesian axes to form an image cube. To obtain the best features possible, the specular and diffuse material properties of the CAD model are manually adjusted so that the computer rendered images closely resemble video of the part to be inspected. This requires manually selecting material properties for rendering that match the properties of the physical part.

In step three, standard techniques are used to extract edgel chains from each rendered image. From the edgel chains up to fifty features are extracted with a minimum angular separation of three degrees relative to the borescope tip. That gives fifty features for each view, six views for each sample point yielding up to 250 features per sample point. If the synthetic images contain few edgels then there will be fewer than 250 features. Likewise more features could be used if available. Limiting the number of features selected effectively limits the processing time required for the on-line pose estimation.

These image features are then converted to 3D edges by casting a ray from the sample point, through the edgel in the image plane and into the CAD model. The closest ray-polygon intersection provides the 3D position of the edgel. The gradient in 3D can be found in a similar manner. These 3D positions and orientations are recorded in the global coordinate system of the CAD model so they can be projected onto any image, not just the ones from which they were generated.

Overall, this off-line process is very time consuming, necessitating use of ray-polygon acceleration techniques, such as spatial hashing [11].

5. On-line Pose Determination

The on-line pose estimation algorithm combines an initial pose estimate, the 3D features computed off-line, and the live video from the borescope to produce a running pose estimate. This algorithm uses the current pose estimate to select the appropriate subset of the 3D features extracted during preprocessing, project these features into the image coordinate system, match these features to the borescope image, and refine the pose estimate based on differences between the projected and matched feature positions. By using precomputed features, by avoiding the need for explicit feature extraction in the borescope images, and by iterative pose refinement, the algorithm achieves video rate pose determination. An important feature of this pose determination algorithm is that a projected 3D feature (edgel) may either (a) be matched exactly, giving a 2D position error vector, or (b) be matched along the gradient direction, giving only the 1D component of position error along this direction, or (c) be ignored entirely as an outlier. The approach also accounts for the significant lens distortions of a borescope. An overview of the algorithm follows:

1. Obtain the previous borescope location and orientation. Initially this comes from the operator positioning the borescope at a known location or landmark. Subsequently, it is taken from the results of the estimated pose for the previous borescope image.
2. Determine the 3D sample point closest to the previous borescope location.
3. Determine which features for that point would be visible to the borescope based on its previous pose.
4. Repeat the following three steps until the change in the pose estimate falls below a threshold or the interframe time has expired.
    5. Project the N 3D features selected in step three onto a 2D image coordinate system based on the current pose estimate (initially the estimate from step one), computing both the 2D position and gradient (normal) direction of each feature.
    6. For each feature, estimate the error in its projected position by finding the position of the best match between the projected feature and the borescope image region near the projected position. The difference in position between projected and matched image positions forms a 1D or 2D error term for each feature, depending on the results of the matching process.
    7. Use the N error terms to update the borescope pose estimate.
8. Return to step one and start working on the next frame of video.

In step one, there typically are a few predefined landmarks for the inspector to choose as a starting point. Step two is a very simple calculation to find the closest sample point to the current pose. Step three selects only the features that could be seen by the borescope in its previous pose. Since the features are 3D locations, this involves

determining if they are in the borescope's view frustum. This set is further restricted to eliminate features near the edges of the view frustum by selecting an angle slightly smaller than the borescope's view angle. Step four starts an iterative error minimization process to determine the optimal pose estimate. This process is limited to the inter-frame time of the borescope. In practice the optimal pose may not have been found when the time has elapsed, but the current pose estimate is typically close enough to the optimum to provide a suitable pose for projecting features for the next frame. In addition, first order motion prediction is used to aid the projection of features for the next frame. In step five the 3D features are projected onto the current image plane yielding 2D edgels. This projection is the standard pinhole perspective projection followed by a second order radial lens distortion (see below).

Steps six and seven warrant additional detail. Step six starts with N 2D edgel locations, their corresponding 2D edgel gradients, and the current borescope image. For each 2D location, the normalized cross-correlation between a one dimensional step function template (as shown in Figure 2) oriented along the edgel gradient, and the video at that location (see Figure 3) is measured. This process is repeated at locations along the positive and negative edgel gradient direction up to a maximum distance determined from the camera's uncertainty. If the maximum correlation found is greater than a threshold, then the feature is considered a gradient feature. It provides one constraint to the pose refinement computation --- the optimal gradient displacement. This displacement is calculated using a weighted average of the correlations, essentially giving a subpixel location to the matched position. In practice the gradient direction will not be axis aligned as in Figure 2, so calculating the normalized cross-correlation involves resampling the step function template onto the pixel array. For maximum performance a set of step function templates can be precomputed for a set of discrete orientations and sub-pixel positions. This is what is actually done in the implementation to avoid the cost of on-line resampling. The step function template selected provides a balance between discrimination and noise resistance for the 300x300 pixel images produced, but other templates could be used.

The initial template matching is restricted to image positions along the feature's gradient direction because each feature consists of a location and a direction --- there is nothing to limit tangential movement. Only if a correlation above the threshold isn't found along the gradient direction, is the search extended to include tangential displacements. For example, if the feature shown in Figure 3 were moved to the left, eventually the gradient direction search would no longer intersect the edge it previously had. In this situation the tangential search can provide the necessary error vector --- a two component error vector as opposed to the earlier one component gradient displacement error vector. This error vector will drive incremental pose estimation toward a pose where the projected feature location will produce a match along its gradient in subsequent iterations of steps five through seven.

After calculating the normalized cross-correlation over the entire region the maximum value may still be quite small. This corresponds to the situation where an edge is expected within the region but nothing suitable is found. In this case no error vector is produced but the feature still provides information by its lack of an error vector. Its low correlation will impact the average correlation which is used as a confidence measure for the algorithm. This process is repeated independently for all N features, each contributing zero, one or two components to the error vector $\vec{E}_t$ for the overall local error at iteration t. This is the error vector to be minimized by the delta pose calculation. Step seven, calculating the change in pose based on the error vector $\vec{E}_t$, is described in the next section.

6. Delta Pose Calculation

The change in pose is computed from the error vector $\vec{E}_t$. For simplicity in deriving the delta pose estimate we first consider the case where each of the features are 2D matches each producing two constraints. This will then be extended to handle all three conditions. We start with the following definitions:

$\vec{P}_t$ = the 6D boroscope pose vector at iteration t

$\vec{u}_{it}$ = the 2D image position for feature i at iteration t

$\vec{x}_i$ = the 3D position of feature i

$\vec{F}$ = the boroscope projection function

Where for $\vec{F}$ we are using the simple perspective camera model with known intrinsic parameters (This is extended to handle radial lens distortion below.) Starting from the equation

$$\vec{u}_{it} = \vec{F}(\vec{P}_t, \vec{x}_{it})$$

we can derive an expression for the change in the feature's image coordinates based on changes in the borescope pose as follows:

$$\begin{aligned}\Delta\vec{u}_{it} &= \vec{u}_{it} - \vec{u}_{i(t-1)} \\ &= \vec{F}(\vec{P}_{(t-1)} + \Delta\vec{P}, \vec{x}_i) - \vec{F}(\vec{P}_{(t-1)}, \vec{x}_i) \\ &= J_i(\vec{P}_{(t-1)}, \vec{x}_i)\Delta\vec{P} + \text{H.O.T.}\end{aligned}$$

Where J_i is the Jacobian. Dropping the higher order terms, yields a constraint on the pose for each error vector.

We can combine the constraints for all N >= 3 matches to solve for the pose. First we combine the $\Delta\vec{u}_{it}$ vectors into a 2N error vector $\vec{E}_t$. Likewise we combine the Jacobians, J_i, into a 2N by 6 matrix J. Then, we determine the pose error $\Delta\vec{P}_t$ by minimizing the following error norm:

$$\left\| J\Delta\vec{P}_t - \vec{E}_t \right\|^2 \text{ yielding } \Delta\vec{P}_t = (J^t J)^{-1} J^t \vec{E}_t$$

The resulting $\Delta\vec{P}_t$ provides an error vector for the borescope pose and can be computed using singular value decomposition (SVD). For a given frame this technique is applied in an iterative manner to account for non-linearity and feature rematching. The Jacobians can be computed along the lines of the technique described in Lowe[9]. For a given borescope image frame this technique is applied in an iterative manner to account for non-linearities and feature rematching, as discussed above.

This technique is easily extended to handle the radial lens distortion typical of borescopes. The following distortion model [13], solves for the distorted coordinates as a function of the non-distorted coordinates.

$$\tilde{u} = u(1 + kr^2)$$

$$\tilde{v} = v(1 + kr^2)$$

$$\tilde{r} = r(1 + kr^2)$$

In this u and v are the non-distorted pixel coordinates in consideration, r is the non-distorted radius of that point (measured form the center of the image), and k is the distortion constant computed off-line during camera calibration. This is a second order approximation and it can easily be extended to higher orders. This model is less commonly used than the traditional formulation which computes non-distorted coordinates from distorted values. Our motivation is that this formulation is easily differentiated and hence can be incorporated into the above Jacobian calculations using the chain rule. The derivatives are:

$$\frac{\partial\tilde{u}}{\partial u} = 1 + 3ku^2 + kv^2 \qquad \frac{\partial\tilde{v}}{\partial u} = 2kuv$$

$$\frac{\partial\tilde{u}}{\partial v} = 2kuv \qquad \frac{\partial\tilde{v}}{\partial v} = 1 + 3kv^2 + ku^2$$

The second extension to the above derivation handles the situation where some features provide one constraint ("gradient features"), others provide two constraints ("position features") and the remainder are ignored. Starting with the equation for gradient displacement:

$$\Delta d_{it} = \hat{g}^t \Delta\vec{u}_{it}$$

where $\hat{g}$ is the unit gradient direction, we can construct an error vector

$$\vec{E}_t = \begin{bmatrix} \Delta d_{it} \text{ for all i that are gradient features} \\ \Delta\vec{u}_{it} \text{ for all i that are position features} \end{bmatrix}$$

The error norm becomes $\left\| HJ\Delta\vec{P}_t - \vec{E}_t \right\|^2$ where we have introduced the matrix H constructed as follows:

$$H = \begin{bmatrix} g_{1u} & g_{1v} & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & g_{2u} & g_{2v} & \dots & 0 & 0 & 0 \\ & & & & \dots & & & \\ 0 & 0 & 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{bmatrix}$$

If a is the number of gradient features then H will have a rows similar to the first two shown. These rows map 2D displacements into gradient displacements essentially by performing a dot product with the unit gradient. Likewise if b is the number of position features then the bottom right corner of H will be the $2b$ identity matrix. The resulting size of H will be $a + 2b$ by 2N. We can then solve this in the same manner as before yielding:

$$\Delta\vec{P}_t = ((HJ)^t (HJ))^{-1} (HJ)^t \vec{E}_t$$

While $H^t H$ is not invertible, in general $(HJ)^t (HJ)$ is.

7. Results

We have implemented and tested the feature extraction and pose determination algorithms on both real and synthetic data. The off-line feature extraction was developed on a UNIX platform using the Visualization Toolkit[14] as a framework. The on-line system is PC based which, with a simple frame capture card, maintains pose updates at over ten frames per second independent of the size of the CAD model.

The top image in Figure 4 shows a portion of a CAD model from a F110A exhaust duct and liner. The liner is an undulating surface with numerous cooling holes. The 3D features extracted for a portion of this CAD model are shown as small white spheres typically surrounding the cooling holes. Note that there are no line segments or flat surfaces to use for complex features. The bottom image in Figure 4 shows a computer rendering of a simple CAD model with some of the 3D feature positions shown as white spheres. Note that some of the features are located at changes in material properties, not just at range or normal discontinuities.

Figure 5 shows two sample images from a Welch Allen VideoProbe 2000 borescope. The image quality suf-

fers from the borescope's small lens, narrow depth of field, and relatively low resolution pixel array. These images are typical of the input to the on-line pose estimation algorithm.

To test the robustness of the on-line algorithm a set of twelve video test sequences were captured that included a wide variety of borescope motions within a CAD part. An effort was made to collect a variety of movements without consideration for how the pose estimation would perform. For each frame of each sequence the probable errors in position and orientation were recorded along with the average correlation of the projected 3D features. The probable error is calculated from the residuals of the SVD solution. The data from sequences two and three are shown in Figure 6. In sequence two the positional error averages one millimeter while the rotational error averages a quarter degree (the volume being inspected is about one cubic decimeter). The average confidence is about 0.07 which may seem low given that an ideal match would be 1.0. This low correlation isn't due to misalignment but rather the small gradients found in typical borescope images. The results for sequence three were similar except that near the end the confidence starts to fall while the positional and rotational errors start to climb. Eventually this test sequence failed due to a lack of features as the borescope was brought up against a wall.

Of the twelve test sequences, the on-line algorithm was able to track nine of them without difficulty. Of the three failed runs, one failed due to an interframe movement too large for the algorithm to handle, the other two failed due to a lack of features. First consider the failure due to interframe movement. The on-line algorithm uses first order motion prediction to handle borescope velocities that could otherwise cause it to fail. However, given a large enough acceleration, the algorithm will fail because the predicted feature locations are simply too far away from where they were expected. The other two failures, which includes sequence three, occurred as the borescope was moved up to a wall. As expected, when the borescope's input video contains little or no variability, there is no information to use in determining the pose. Given that the borescope's pose has six degrees of freedom the on-line algorithm will require at least six "edgels" in the input video.

To validate the resulting pose estimates, videos were made containing the original borescope video overlaid with the CAD model according to the pose estimate. One frame of an overlaid video sequence is shown in Figure 7. The overlaid CAD model is shown as thick white lines. Generating the overlaid image consists of rendering the CAD model from the pose estimate, extracting edgel chains from the rendered image, warping the edgel chains according to the lens distortion model, and finally rendering the warped edgels chains as white lines on top of the borescope video. On-line robustness measures include display of the SVD's condition as well as the average normalized cross-correlation of the projected 3D features onto the video.

8. Conclusions

This paper has made three contributions to the area of model based pose estimation. The first is its use of computer graphics hardware to generate local features for pose estimation. While there has been work on directly extracting features from a CAD model, it has focused on categorizing this information based on the direction of view, not on location. It has also been based on visibility constraints and geometric features. This paper goes further to consider view direction, view position, material properties and lighting conditions in determining features. The novel use of computer graphics hardware makes this computationally feasible for the large models found in industry. This work has shown that computer renderings can be used as a predictor of features for pose estimation. This approach can be extended to provide more accurate feature prediction by using more advanced rendering and lighting techniques.

The second contribution is in the on-line pose estimation, where a number of weaknesses in the current pose estimation literature have been addressed. First, neither high order features nor explicit feature extraction is required from the input image. Instead, features precomputed from the CAD model and predicted from the previous pose are projected into the current image and matched using simple, efficient correlation techniques. Based on the matching results, each feature provides from zero to two constraints on the pose estimate. The framework incorporates radial lens distortions and the algorithm runs at near frame rates on a personal computer.

The third contribution is in the consideration of large industrial datasets. Many model based pose techniques have not been designed to scale to models composed of a million or more polygons. Issues include the combinatorics of matching, the reliability of matching, and the small percentage of model features typically visible in any single view. Even though the requirements for borescope inspection differ from other applications, many of these issues are universal when dealing with CAD models of industrial parts. The techniques presented in this paper allow for on-line pose estimation without any performance or accuracy degradation for large, more complex, CAD models.

A few open issues arise from this work. The first is how to improve feature selection from the rendered images. When backprojecting detected image features onto the CAD models using ray-polygon intersections, the

stability of the 3D position of the feature can be determined. For example, features caused by tangency between the camera line of sight and a surface are unstable. These unstable features should be removed from the feature set. In addition, the selection of features could consider the distribution of gradient directions in addition to the gradient intensities. This can prevent singular matrix problems that occur when the gradient directions are closely aligned.

A related issue in the feature extraction is determining where the sample points should be located and how many of them to use. While we have used a simple regular grid, this is inefficient for the structural variety found in CAD models. A more advanced approach would start with a sparse set of sample points and then locally subdivide the volume based on some measure of feature quality. This is basically an octree decomposition of the volume where the error measure is based on the feature variability across the octent. One measure of the feature variability is the test presented in the previous paragraph. If the ratio of the features rejected to the total number of features exceeds a specified threshold, then the octent should be subdivided. This would provide an adaptive distribution of sample points throughout the model.

An open issue with the on-line pose estimation algorithm involves the weighted average used in the feature matching. Currently if a predicted feature lies between two viable matches, the resulting error vector from the weighted average will be roughly zero. This is not really correct because the true error vector is the vector to one of the two possible matches. What is correct, is to say that the feature has no meaningful contribution to the error vector, only to the overall confidence. As the weighted average transitions from a bimodal to uni-modal distribution the relevance of the error vector will increase. Somehow this change in distribution must be taken into account in addition to the final result. One possibility is to modify the error metric to include a weighting vector that represents the confidence in the elements of the error vector.

While there are open issues to consider, this paper has contributed a practical approach for model based pose estimation. This approach handles industrial CAD models, runs at frame rates, and requires no special hardware. These techniques developed to handle borescope and endoscope inspection are valuable to most pose estimation applications involving industrial CAD models.

Acknowledgments

This work was performed while the first author was employed at General Electric Corporate Research and Development. The second author would like to acknowledge the financial support of the National Science Foundation under grants IRI-9217195 and IRI-9408700.

References

[1] Branca A, Stella E, and Distante A, Passive Navigation using Focus of Expansion, *Proc. Third WACV* (IEEE Computer Society Press, 1996) 64-69.

[2] Dickmanns E D, An Integrated Approach to Feature Based Dynamic Vision, *Proc. CVPR '88 (*IEEE Computer Society Press, 1988) 820-825.

[3] Dickmanns E D, Mysliwetz B, and Christians T, An Integrated Spatio-Temporal Approach to Automatic Visual Guidance of Autonomous Vehicles, *IEEE Transactions on Systems, Man, and Cybernetics*, 20(6) (1988) 1273-1284.

[4] Dickmanns E D, and Schell F R, Autonomous Landing of Airplanes by Dynamic Machine Vision, *Proc. WACV (*IEEE Computer Society Press, 1992) 172-179.

[5] Gilg, A and Schmidt G, Landmark-Oriented Visual Navigation of a Mobile Robot, *IEEE Transactions on Industrial Electronics*, 41(4) (1994) 392-397.

[6] Harris C, Tracking with Rigid Models, in: Blake A, Yuille A, ed., *Active Vision*, (MIT Press, Cambridge, 1992) 59-73.

[7] Khan G N, and Gillies D F, Vision based navigation system for an endoscope, *Image and Vision Computing*, 14 (Elsevier Science, 1996) 763-772.

[8] Kuno Y, Okamoto Y, and Okada S, Robot Vision Using a Feature Search Strategy Generated from a 3-D Object Model, *IEEE PAMI*, 13(10) (1991) 1085-1097.

[9] Lowe D G, Three-Dimensional Object Recognition from Single Two-Dimensional Images, *Artificial Intelligence*, 31 (1987) 355-395.

[10] Lowe D G, Robust Model-based Motion Tracking Through the Integration of Search and Estimation., *International Journal of Computer Vision*, 8(2) (Kluwer Academic Publishers, 1992) 113-122.

[11] Mortenson M E, *Geometric Modelling*, (John Wiley & Sons, New York, 1985).

[12] Phong T Q, Horaud R, Yassine A, Pham D T, Optimal Estimation of Object Pose from a Single Perspective View, *Proc. Fourth ICCV* (1993) 534-539.

[13] Puskorius G V, and Feldkamp L A, Camera Calibration Methodology Based on a Linear Perspective Transformation Error Model, *Proc. IEEE International Conference on Robotics and Automation* (IEEE Computer Society Press, 1988) 1858-1860.

[14] Schroeder W, Martin K, Lorensen B, *The Visualization Toolkit: An Object Oriented Approach to 3D Graphics* (Prentice-Hall, 1996).

[15] Shakunaga T, Robust Line-Based Pose Enumeration From A Single Image *Proc. Fourth ICCV* (1993) 545-550.

[16] Viola P, and Wells W M III, Alignment by Maximization of Mutual Information, *Proc. Fifth ICCV* (1995) 16-23.

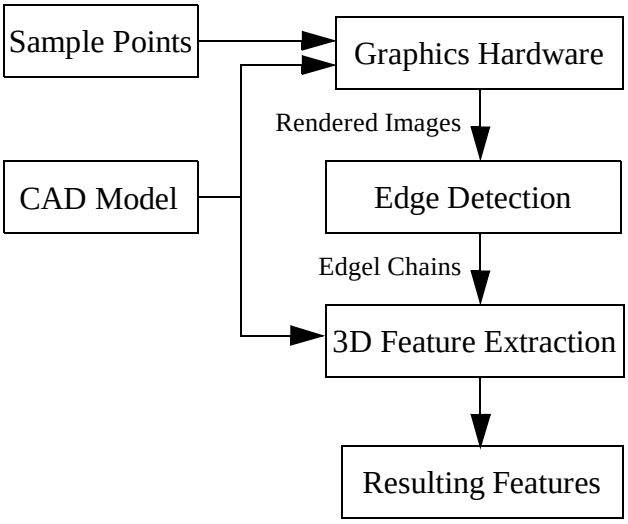


Figure 1. Feature extraction process.

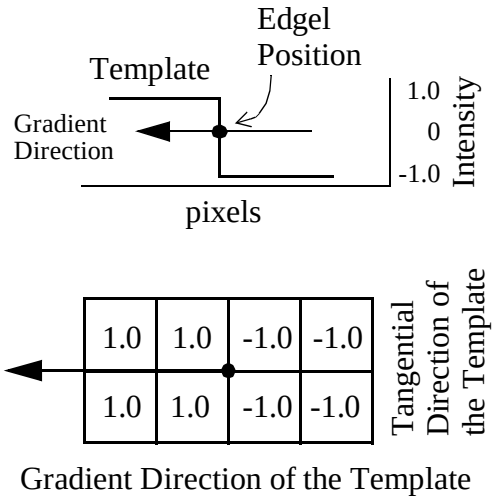


Figure 2. Step function feature template.

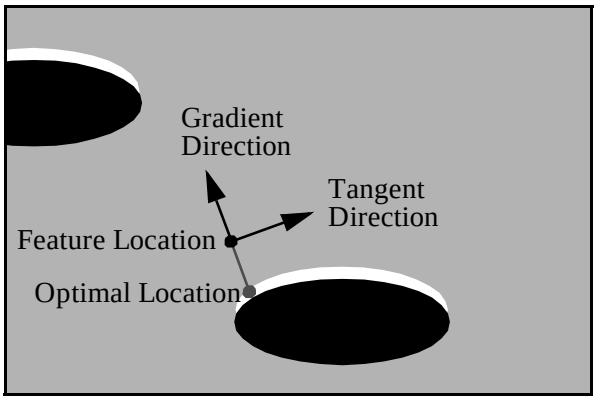


Figure 3. A simple example of matching a feature to an image.

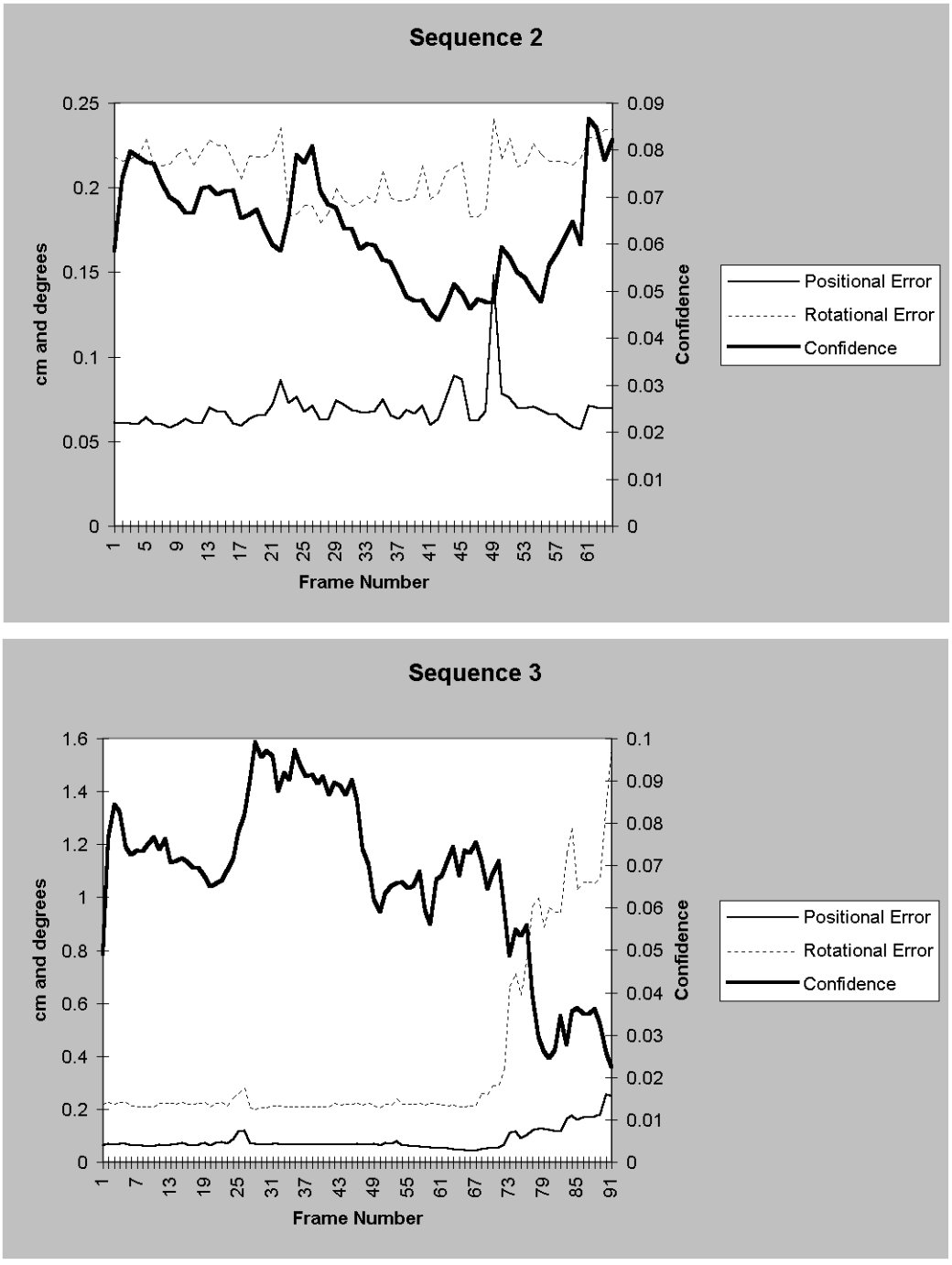
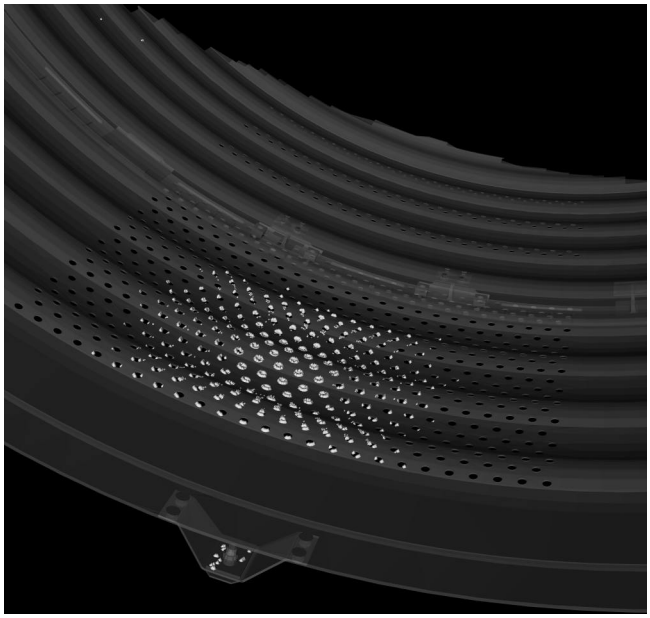


Figure 6. Results from test sequences two and three. The horizontal axis is the frame number of the test run. The positional error is the probable error in centimeters for the borescope's position as determined by the singular value decomposition. The rotational error is the total probable rotational error for the three axes as determined by the singular value decomposition. The confidence is the average correlation of the projected features to the image. (Note that the graphs have different scales)



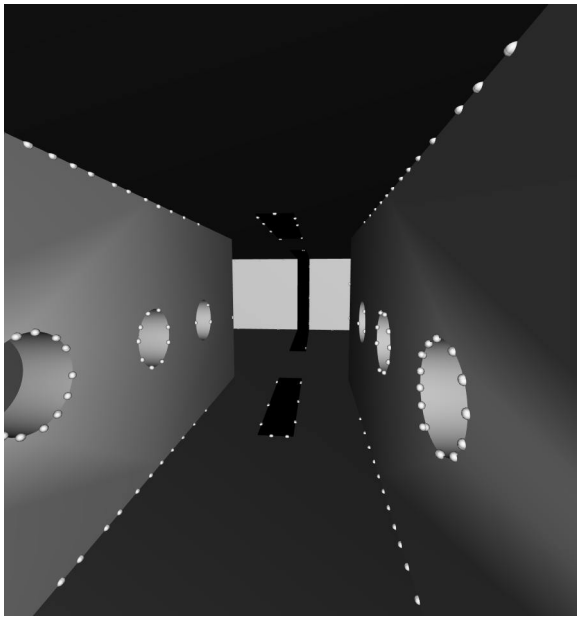


Figure 4. Computer rendered images (partial views from inside) of two CAD models with extracted 3D feature locations overlaid as small white spheres. The top image is from a F110-A exhaust duct and the bottom is from a test phantom

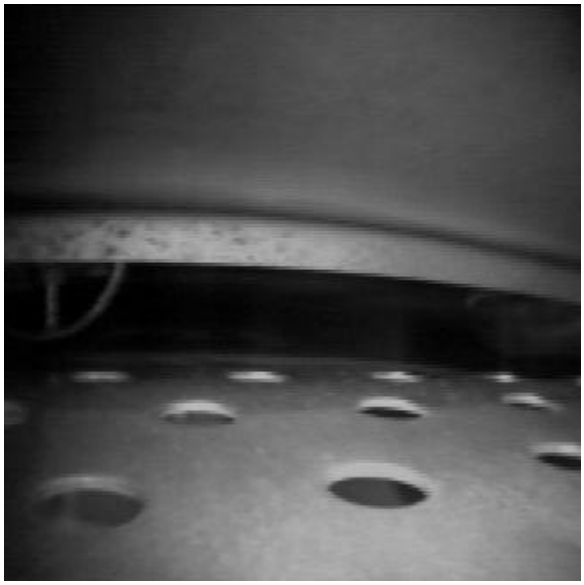


Figure 5. Two sample images from a Welch Allyn borescope placed inside two different objects. The top image is from a F110-A exhaust duct and the bottom is from a test phantom.

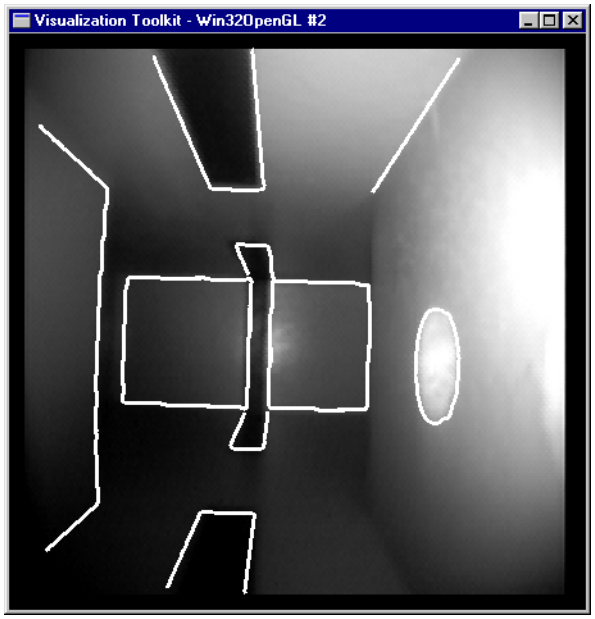


Figure 7. CAD model overlaid onto borescope video.