Stabilizing Augmented Reality Overlays of Hinged and Curved Planar Regions

Robert B. Fisher

Abstract. Most Augmented Reality (AR) applications require model-to-scene feature point correspondences to estimate the graphical mapping transformation. This paper introduces a shape-based algorithm that allows registration of structures that do not have stable feature points, such as with curved regions. The technique uses a variation of the Iterated Closest Point algorithm to achieve projective (as compared to Euclidean) registration. With this technique, we show that directly registering 2D AR overlays has greater registration stability than the usual technique of estimating the 3D position of the overlay and then applying pinhole projection, which can produce noticeable frame-rate jitter of the graphical objects. Finally, we extend the technique to projectively register rigid and hinged 3D shapes projectively by using constrained 2D projective mappings. These three techniques enhance the repertoire of methods for producing high-detail, stable augmentation of video.

Keywords: Augmented Reality stabilization, homography, shape

Stabilizing Augmented Reality Overlays of Hinged and Curved Planar Regions

Abstract. Most Augmented Reality (AR) applications require model-to-scene feature point correspondences to estimate the graphical mapping transformation. This paper introduces a shape-based algorithm that allows registration of structures that do not have stable feature points, such as with curved regions. The technique uses a variation of the Iterated Closest Point algorithm to achieve projective (as compared to Euclidean) registration. With this technique, we show that directly registering 2D AR overlays has greater registration stability than the usual technique of estimating the 3D position of the overlay and then applying pinhole projection, which can produce noticeable frame-rate jitter of the graphical objects. Finally, we extend the technique to projectively register rigid and hinged 3D shapes projectively by using constrained 2D projective mappings. These three techniques enhance the repertoire of methods for producing high-detail, stable augmentation of video.

1 Introduction

One of the fundamental operations in an Augmented Reality (AR) system is the projection of the graphical objects onto a video sequence. The traditional method for this projection is to analyze the video sequence to deduce the 3D scene position of graphical object and then to project the graphical object into the video sequence using a standard camera model [3, 4, 22, 23], based on standard methods for estimating the 3D camera pose from models and 2D image features [13, 18]. While this 3D-to-2D approach is technically correct, our experience of working with 3D scenes suggests that estimating the 6 degrees of freedom of the graphical object in 3D space can be slightly unstable. This causes the graphics objects to have a frame-rate jitter which can certainly be observed in many AR applications. An alternative is to map directly from the graphical space to the image space, which is the approach being presented in this paper.

An example of where it is hard to establish correspondence is when the region to be transferred has a curved boundary, which makes it hard to find the point correspondences needed to estimate the registration. The Iterative Closest Point (ICP) matching [6] has been used for a number of 3D-to-3D and 2D-to-2D point matching applications, but is traditionally used for rigid (Euclidean) shape matching. If we extend the ICP algorithm to projective point matching (PICP), then we are able to estimate the projective registration between shapes without requiring explicit stable feature point correspondences. With this technique, we show that directly registering 2D AR overlays has greater registration stability than the more usual technique of estimating the 3D position of the overlay and then applying pinhole projection.

We then extend the technique to directly register linked and possibly hinged planes projectively by using constrained 2D projective mappings. This allows the projection of 3D shapes into an image. While this is not a true 3D projection, the projective relationships are nearly indistinguishable, making it a usable technique for scenes containing 3D planar structure.

This projection methods can be used in AR applications requiring accurate compositing, such as in special effects in video post-production, or live entertainment overlay where viewer opinion is important. Architectural applications include museum enhancement, emergency service route directions, building maintenance plan overlay, etc.

Direct projective transfer has been known for several years [15], but this work has mainly used either rectangular regions with easy-to-find boundaries, or easily locatable feature points for estimating the homography. Smith *et al* [21] explored direct image mapping to improve graphical object registration using scene constraints such as parallel lines and coplanarity of tracked features, Euclidean bundle adjustment, and estimating parameters over the whole image sequence using supplied camera projection matrices. In addition, they also explored adding connected 3D structures (a rectangular solid), by tracking the vanishing points of three sets of parallel lines to define an affine system and then estimate the camera matrices, subject again to constraints such as parallelness and coplanarity. Kutalakos and Vallino [16] demonstrated direct, but affine, mapping of 3D objects using four accurately tracked non-coplanar control points to determine the mapping of the remainder of the 3D object. Other research that has influenced the research presented here through their use of projective relationships in the tracking or alignment process includes:

- Uenohara and Kanade [23] exploited projective relationships for feature tracking.
- Feldmar *et al* [11] used the Iterative Closest Point approach [6] for feature alignment.
- Berger *et al* [5] used curve alignment as well as point alignment.

However, all used the standard 3D-to-2D projection approach for estimating overlay position.

Articulated motion has been previously analyzed and augmented, but only in the context of the 3D pose-to-2D image method [1, 10, 14, 17] or 3D-to-3D [2].

Improved registration can also be achieved using bundle adjustment [15]. That technique optimizes only the transformation, whereas the approach proposed here also optimizes the correspondences used to compute the transformation.

The research presented here extends this previous work by using perspective projection, avoiding dependence on accurate tracking of individual points or features, working with curved shapes that do not have reliably extractable correspondence points and working with multiple constrained and possibly hinged projections.

The contributions of this paper are:

1. **By using the full boundary shape rather than feature points, we can get augmented reality transfer with a broader range of target shapes, such as curved regions.**
2. **Direct projective transfer is stabler than using the 3D pose and pinhole projection.**
3. **The 2D direct transfer method can be extended to multiple linked or hinged planes to give the appearance of full 3D transfer.**

The research presented here has improved on the work presented in [12] by simplifying the 2D and quasi-3D transfer approaches and extending the quasi-3D transfer approach to hinged structures.

2 The Projective Iterative Closest Point Algorithm

The ICP algorithm [6] is an iterative alignment algorithm that works in three phases: 1) establish correspondence between pairs of points in the two structures that are to be aligned based on proximity, 2) estimate the rigid transformation that best maps the first members of the pairs onto the second and then 3) apply that transformation to all points in the first structure. These three steps are then reapplied until convergence is concluded. Although simple, the algorithm works quite effectively when given a good initial estimate.

The basic algorithm has been previously extended in a number of ways: 1) correspondence between a point and a tangent plane to overcome the lack of an exact correspondence between the two sets [8], 2) robustifying the algorithm to the influence of outliers and features lacking correspondences [19, 24], 3) using a weighted least-square error metric [9], and 4) matching between features using a metric trading off distance and feature similarity (based local shape invariances) [20]. All of these approaches assume a rigid Euclidean transformation between the corresponding features, whereas the method presented here uses projective correspondence.

Unlike the Euclidean case, the structures being matched don't necessarily have the same shape, because of projective distortion. However, as we are working with full projective geometry, it is still possible that the shapes can have an exact projective correspondence. Thus, it is necessary to define a distance measure between projective points, so that we can find the 'closest' points. As our chief interest is accurate image alignment, we use the reprojection error. We also need a way of estimating the homography between the set of paired 'closest' points. These are the main differences between the normal Euclidean ICP algorithm and that presented here.

We define the Projective ICP algorithm (PICP) as follows (using the notation from [20]). Let $\mathcal{S}$ be a set of N_s coplanar 2D image points $\{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{N_s}\}$ and $\mathcal{M}$ be the corresponding 2D model. Let $d_p(\boldsymbol{s}, \boldsymbol{m})$ be the projective distance between point $\boldsymbol{s} \in \mathcal{S}$ and $\boldsymbol{m} \in \mathcal{M}$. Let $\mathrm{CP}(\boldsymbol{s}, \mathcal{M})$ be the 'closest' point in $\mathcal{M}$ to the scene point $\boldsymbol{s}$, using the reprojection error. (Minimizing Euclidean reprojection error gives the Maximum Likelihood estimate of the homography [15] under the assumption of Gaussian noise on the image point measurements.)

1. Let $\mathrm{T}^{[0]}$ be an initial estimate of the homography.
2. Repeat for $k = 1..k_{\max}$ or until convergence:
   (a) Compute the set of correspondences $\mathcal{C} = \bigcup_{i=1}^{N_s}\{(\boldsymbol{s}_i, CP(\mathrm{T}^{[k-1]}(\boldsymbol{s}_i), \mathcal{M}))\}$.
   (b) Compute the new homography $\mathrm{T}^{[k]}$ between point pairs in $\mathcal{C}$ using the method of Section 2.1.

It is possible for ICP algorithms to diverge if the initial transformation estimate is not close enough to the correct alignment. This problem also arises in this algorithm, which can lead to very distorted transformations. This behavior was observed when the initial transformation estimate left the registration features close to a distinctly different part of the scene.

2.1 Projective Transform Estimation

Let $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \dots \boldsymbol{p}_n\}$ and $\{\boldsymbol{q}_1, \boldsymbol{q}_2, \dots \boldsymbol{q}_n\}$ be two sets of paired homogeneous points linked by a projective transform. The 2D projective transform $\mathbf{T}$ can be represented with a 3×3 matrix having an arbitrary scaling, and thus 8 degrees of freedom. If $n = 4$, then $\mathbf{T}$ can be solved for exactly. Here, we expect that n will be much bigger than 4 and so use the direct linear method [15] to estimate $\mathbf{T}$ such that:

$$\boldsymbol{q}_i \doteq \mathrm{T}\boldsymbol{p}_i$$

Let

$$\mathrm{T} = \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{pmatrix}$$

Define:

$$\boldsymbol{t} = (t_{11}, t_{12}, t_{13}, t_{21}, t_{22}, t_{23}, t_{31}, t_{32}, t_{33})'$$

Normalize vectors so $\boldsymbol{p}_i = (p_{ix}, p_{iy}, 1)'$ and similarly for $\boldsymbol{q}_i$. Construct the $2n \times 9$ matrix A:

$$\begin{aligned} A(2i-1,:) &= (p_{ix}, p_{iy}, 1, 0, 0, 0, -q_{ix}p_{ix}, -q_{ix}p_{iy}, -q_{ix}) \\ A(2i,:) &= (0, 0, 0, p_{ix}, p_{iy}, 1, -q_{iy}p_{ix}, -q_{iy}p_{iy}, -q_{iy}) \end{aligned}$$

The solution vector $\boldsymbol{t}$ is the eigenvector of $A'A$ with smallest eigenvalue.

3 Registration of Planar Regions with Curved Boundaries

Using the theory developed in Section 2, we can directly project planar graphical structures from the graphical space to the image space.

Assume that we are trying to map a planar curved region model boundary $\mathcal{S} = \{\boldsymbol{c}(\lambda)\}, \lambda \in [0..1]$ (represented homogeneously) onto a corresponding set of image points on the boundary of that structure. Let the set of image points $\mathcal{P} = \{\boldsymbol{p}_i\}, i \in [1..n]$ describe that structure. These points are also represented using homogeneous coordinates. How the image points might be found is tangential to the issue considered in this paper, but they might be found by using a standard edge detector, with gating from predictions arising from target tracking. Assume that we know which model structure corresponds with each image structure. (This question is also tangential, but could use some model recognition or tracking algorithm.)

The goal is to estimate the transformation T that that maps the model curve $\boldsymbol{c}(\lambda)$ onto $\mathcal{P}$. Using the theory in Section 2, we estimate the T that best satisfies $\boldsymbol{p}_i = \mathrm{T}\boldsymbol{c}(\lambda_i)$ for the corresponding projectively closest point pairs $\{(\boldsymbol{p}_i, \boldsymbol{c}(\lambda_i))\}$. This in turn requires finding the corresponding point pairs. As we are working in the PICP framework, this search reduces to finding the closest pairs between $\{(\mathrm{T}^{[k-1]}\boldsymbol{c}(\lambda_i), \boldsymbol{p}_i)\}$ at the k^{th} iteration of the PICP algorithm, where $\mathrm{T}^{[k-1]}$ is the $k-1^{\mathrm{st}}$ estimate of the homography.

As λ_i specifies a continuous curve, finding the point in $\mathcal{P}$ closest to each $\mathrm{T}^{[k-1]}\boldsymbol{c}(\lambda_i)$ can be quite time-consuming. Fortunately, the homography is invertible, so instead we compute the λ_i for the pairs $\{((\mathrm{T}^{[k-1]})^{-1}\boldsymbol{p}_i, \boldsymbol{c}(\lambda_i))\}$, estimate $(\mathrm{T}^{[k]})^{-1}$, and then invert the estimated homography to get $\mathrm{T}^{[k]}$.

In the examples below, we estimate the homography independently for each image. The use of an image sequence is only to show the stability of the projection over the sequence, with minimal jitter. Bundle adjustment [15] or Kalman filtering could be used to optimize the transform over the whole image sequence at a computational cost. These would optimize the transformation, whereas the algorithm presented here also optimizes the shape correspondence.

3.1 Evaluation

We show here that: 1) this algorithm works well even without obvious corresponding feature points and 2) the method produces stabler AR than the traditional approach. We show first some performance results using synthetic images, and then examples with real video sequences.

We investigated the direct projection approach's stability compared with the traditional algorithm that estimates the 3D transform and then uses pinhole projection to place the graphics in the target image.

The test graphical object is a rectangle of dimensions 0.5 by 1, projected into a sequence of 20 views with a moving camera. Real structures like this simulation include a picture on an art gallery wall, a notice or advertising board, or a building side. The image background has intensity 20 and the rectangle has intensity 100. Gaussian noise of varying standard deviation is added at each pixel. (An alternative approach would have been to disturb the computed edge point positions, but varying the image noise more accurately reflects circumstances encountered in practice.)

Ten instances of the image with different noise were generated at each view, giving a total of 200 samples at each noise level. The Canny edge detector found the edge points used for registering edges of the rectangle. A sample intermediate and final view of the object is seen in Figure 1, with also the edge points from the final view.

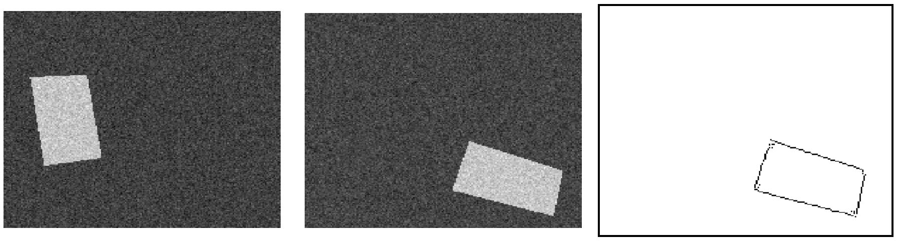

Fig. 1. Test images from an intermediate and last viewing position with image noise $\sigma = 10$. The rightmost image shows the edges from the last image.

The 3D transformation is estimated by finding the corners of the rectangle from the image edges and then searching for the 3D points along the lines of sight through the corners that best fit the model rectangle.

The PICP algorithm was allowed to run for up to 50 iterations, or terminate early if the cumulative projective distance between the registered edge points and corresponding model rectangle points differed between iterations by less than a threshold value (0.004). On average, the PICP algorithm required 15 iterations (range 4–32).

Table 1. Average, standard deviation and maximum deviation of the average boundary distance between the estimated and true graphic object rectangles for the 3D and PICP algorithms at 4 pixel Gaussian noise standard deviations.

Noise	3D			PICP		
	mean	std	max	mean	std	max
0	0.99	0.45	2.03	0.33	0.03	0.40
10	0.90	0.35	2.34	0.33	0.04	0.44
20	1.03	0.38	2.90	0.33	0.07	0.67
30	1.33	0.44	3.29	0.37	0.10	0.80

The experiments measured the average distance between the true and estimated rectangle's boundary, which assesses the stability of overall shape matching and registration. Table 1 shows the average boundary distances. It is clear that the boundary alignment algorithm is much more accurate (comparing means) and stable (comparing standard deviations and maximum errors) than the 3D approach at all noise levels.

This experiment required approximately 2 seconds per iteration with about 200-230 edge points on a 270 Mhz Sun workstation and unoptimized Matlab code. This suggests one can achieve overlay on a 2 GHz PC at real-time video with optimized C/C++ code.

To demonstrate the performance on a real video sequence, observe the animated GIF at: `http://www.dai.ed.ac.uk/homes/rbf/PICP/eye.htm`. This shows the transfer of a video sequence (88 frames) of an eye blinking onto an office interior sequence containing a target with a curved boundary. The source eye was manually edited to select the eye window with shape a scaled version of the tracked template. The PICP algorithm was then used to register the shapes for transfer of the winking eye into the tracking sequence. Figure 2 shows a single frame from the animated sequence.

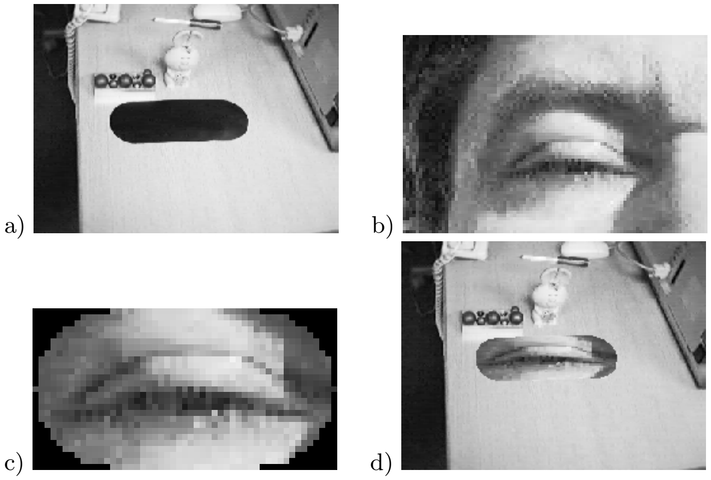

Fig. 2. Snapshot of the video transfer onto a curved boundary. a) One frame of the original sequence, b) one frame of the transfer sequence, c) cropped eye from the transfer sequence, whose boundary is mapped onto the template boundary, and d) corresponding frame from result sequence.

A second interior sequence can be seen at: `http://www.dai.ed.ac.uk/homes/rbf/PICP/corr.htm`. This shows the augmentation of a corridor scene with navigation instructions, such as might be presented to an emergency service person on a future head-mounted AR display. In this case, the overlay lies in the same projective plane as the registration features, which were matched to the interior edges of the O, but did not use any of the registration features. Figure 3 shows one frame of the result sequence (11 frames) and the overlay plane. Note that the transfer still is stable even though much perspective distortion is now appearing.

4 Quasi-3D Structure Registration Using Constrained Projection

If the graphical object to be projected contains 3D structure, then one can use the normal approach of estimating the full 3D transformation and then applying image projection using a camera model. As both intrinsic and extrinsic camera

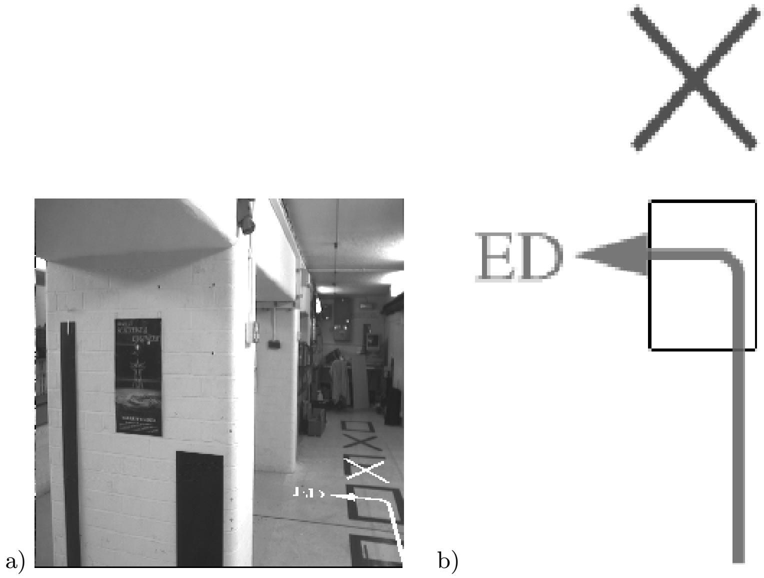

Fig. 3. Snapshot of the transfer onto a corridor scene: a) One frame of the result sequence and b) the transfer overlay, including the registration features.

model estimation can have instabilities, then the graphical object might jitter around the video object.

One can alternatively apply an extension to the method of Section 3 if the 3D object consists of connected planar segments (*e.g.* a polyhedral or triangulated model). For example, the planar models could be different faces of an object or walls of a building. The extension presented here is applicable for both rigid and hinged objects, such as doors.

Applying the method of Section 3 directly to the individual planar segments works, but it gives individually estimated homographies that might cause the shared edges of the graphical objects to no longer align when projected. Hence, this section looks at how to estimate the individual surface homographies subject to the constraint that shared model vertices are coincident in the projected image (which also guarantees that the shared edges - the intersection of the two respective planes - are coincident).

This is not true 3D transfer because each plane can have a quite different projection. However, from the observer's viewpoint, the constrained points and connecting edge do align, as well as the rest of the model shape to the image edges. Hence, the appearance transfers approximately as expected. To improve internal alignment of the transferred shapes, additional model points g along the shared edge can be constrained to project to the same point g'. (See Figure 4.) Care must be taken to ensure that the numerical representations are exact so that the transfer estimates do not degenerate to satisfy the projection constraints.

Formally, let $\mathcal{M}_1 = \{\boldsymbol{c}_1(\lambda_1)\}$ and $\mathcal{M}_2 = \{\boldsymbol{c}_2(\lambda_2)\}$, $\lambda_i \in [0..1]$ be two planar curves (represented homogeneously) mapped by the homographies T_1 and T_2 into a common image plane containing image feature points $S_1 = \{\boldsymbol{p}_{11}, \boldsymbol{p}_{12}, \ldots \boldsymbol{p}_{1n_1}\}$ and $S_2 = \{\boldsymbol{p}_{21}, \boldsymbol{p}_{22}, \ldots \boldsymbol{p}_{2n_2}\}$. (See Figure 4.) We assume that the image feature points have already been segmented into sets corresponding to the appropriate graphical object, by some process not considered here. Some feature points of S_1 and S_2 will be shared; these are the vertices and curves common to both sets.

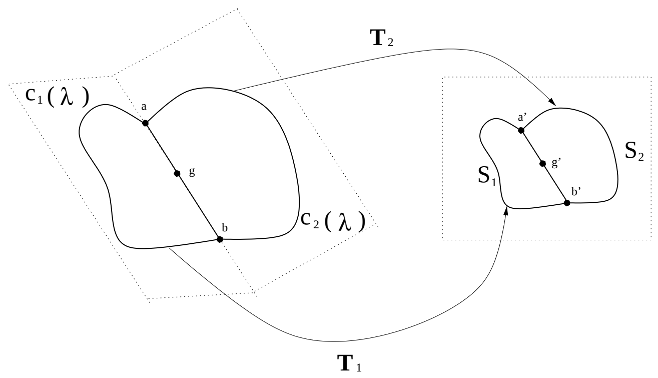

Fig. 4. Projection of 2 non-coplaner curves $\boldsymbol{c}_i(\lambda)$ via homographies T_i into a common image mapping shared point a to a' and b to b'.

Suppose that $\boldsymbol{c}_1(\lambda_{1k})$ and $\boldsymbol{c}_2(\lambda_{2k})$, $k = 1..K$, map to the same image point (*e.g.* points a and b in Figure 4). For example, these are the vertices at the end of a shared line segment.

Then, the problem can be formulated as: Find the T_1 and T_2 that minimize the mapping distance of $\{\boldsymbol{c}_1(\lambda_1)\}$ and $\{\boldsymbol{c}_2(\lambda_2)\}$ onto S_1 and S_2 respectively such that

$$\mu_k \mathrm{T}_1 \boldsymbol{c}_1(\lambda_{1k}) = \mathrm{T}_2 \boldsymbol{c}_2(\lambda_{2k})$$

where μ_k are new variables reflecting the difference in homogeneous scaling.

To solve the problem, we observe that: 1) each homography can represented by the mapping of four 'control' points from the model to the image plane and 2) two of the control points can be shared between the two homographies as the points have to lie in both planes. Thus, we can reformulate the constrained transformation process to be:

Let:
a, b be the shared control points (see Figure 4)
c_1, d_1 be the extra control points for plane 1
c_2, d_2 be the extra control points for plane 2
g_i be any other shared points

g_i are the additional points on the shared edge that improve the internal alignment of the transfer during the quasi-3D transfer.

The initial control points are defined by the mapping:

$$[a, b, c_i, d_i]' = \mathrm{T_i}[m_{i1}, m_{i2}, m_{i3}, m_{i4}]' \tag{1}$$

using the initial estimates of $\mathrm{T_i}$, and an arbitrary set of model points m_{ij}. We want to find the a, b, c_1, d_1, c_2, d_2 that give the $\mathrm{T}_1(a, b, c_1, d_1)$ and $\mathrm{T}_2(a, b, c_2, d_2)$ that minimize the target shape boundary and shared constraint point reprojection error:

$$E(\mathrm{T}_1, \mathrm{T}_2) = \sum_{i=1}^{n_1} d_p({\mathrm{T}_1}^{-1}\boldsymbol{p}_{1i}, \mathcal{M}_1) + \sum_{i=1}^{n_2} d_p({\mathrm{T}_2}^{-1}\boldsymbol{p}_{2i}, \mathcal{M}_2) + \frac{n_1 + n_2}{5} \sum_{i=1}^{Q} d_p(\mathrm{T}_1\boldsymbol{q}_{1i}, \mathrm{T}_2\boldsymbol{q}_{2i})$$

A standard numerical method is used to update the values of a, b, c_1, d_1, c_2, d_2. From these $\mathrm{T_i}$ can be recalculated using Eqn (1).

The algorithm used is:

1. Compute initial estimates of ${\mathrm{T}_1}^{[0]}$ and ${\mathrm{T}_2}^{[0]}$ independently using the PICP method of Section 3.
2. Construct a 6*2 parameter vector $\boldsymbol{s} = (a_x, a_y, b_x, b_y, \ldots, d_{2y})'$ representing the 2 shared control points and 2 independent control points each for the two planes by using the initial estimates of T_1 and T_2.
3. Loop M (*e.g.* $M = 5$) times:
   (a) Minimize $E(\mathrm{T}_1(\boldsymbol{s}), \mathrm{T}_2(\boldsymbol{s}))$.
   (b) Recompute correspondences between closest model and data edge points, as used when computing $E()$. (In theory we should recompute the correspondences in the minimization step above, as the transformations are modified. However, the transformations are already close to optimal, so the correspondences do not change much. As this computation is very expensive, it is placed in the outer loop rather than in the optimization loop, resulting in a significant decrease in running time.)

If more than two planar segments are in the model, the method generalizes easily by adding additional control points: 2 for a new plane linked to only one previous plane, 1 for a new plane linked to 2 previous independent planes.

4.1 Evaluation

To evaluate the constrained projection method, we show first some performance results using synthetic images, and then an example with a real image.

The synthetic results used two constrained semicircles linked perpendicularly as seen in Figure 5a. The individual semicircle edges are registered to the image edges shown in Figure 5b, and then constrained to share the same four semi-circle vertices. The resulting constrained mapping projects the model edges onto the image as seen in Figure 5c. This problem was constrained in about 15 minutes using Matlab on a 270 Mhz Sun. Matlab's inbuilt `fminsearch` optimization was used, which suggests recoding could easily produce a 100 factor speedup.

Regenerating this image with different image noise ($\sigma = 10$) 5 times each over a tracking sequence with 10 positions gave an mean average boundary distance error of 0.47 pixels, with standard deviation 0.033 and maximum average error of 0.53 pixels. Thus, the process is stable to image noise well below the level of integer pixel edge data.

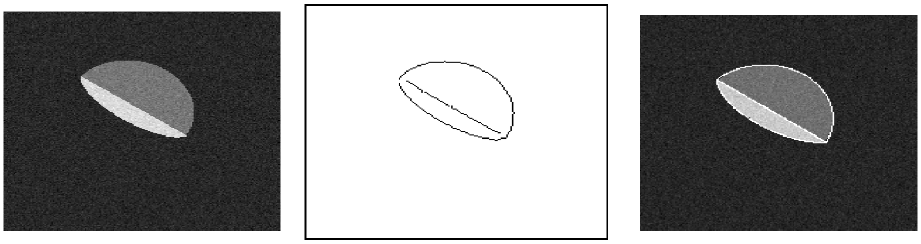

Fig. 5. Test images for 3D registration, showing: a) the raw image, b) the edges used for image capture c) the two registered semicircle models projected onto the raw image.

We show here the constraint method applied to a real image sequence, namely for overlaying a hinged butterfly shape, over which we overlay a more ornate image. Figure 6 shows a) one raw image, b) the edges from that image, c) the butterfly model fitted to both sides of the image data and d) the corresponding frame from result sequence with the model projected onto the image.

The performance on the real video sequence can be seen at: `http://www.dai.ed.ac.uk/homes/rbf/PICP/bfly.htm`. The wings are hinged and vary in position by about $\pm\frac{\pi}{4}$ radians. The matched edges are not exactly found as in the synthetic example, but the butterfly does transfer both stably and reliably, with the two sides of the shape always perfectly joined. Figure 6b shows typical detected edges, which do not have the perfect model shape. Because of the optimization step, the computation took about 15 minutes per image (Matlab on a 360 Mhz Sun).

We next show the method on a second real sequence, where we overlay an apple wedge somewhat similar to the synthetic example. Figure 7 shows a) one raw image, b) the edges from that image, c) the apple model fitted to both sides of the slice and d) the corresponding frame from result sequence with the model projected onto the image. The video performance can also be seen at: `http://www.dai.ed.ac.uk/homes/rbf/PICP/act.htm`.
In this example, the matched edges are not as reliably and stably found, but the apple does not have the spherical wedge as in the synthetic example. Figure 7b shows typical detected edges, which do not have the expected shape. None-the-less, the tracking is reasonable but not as stable; however, the constraint is always satisfied.

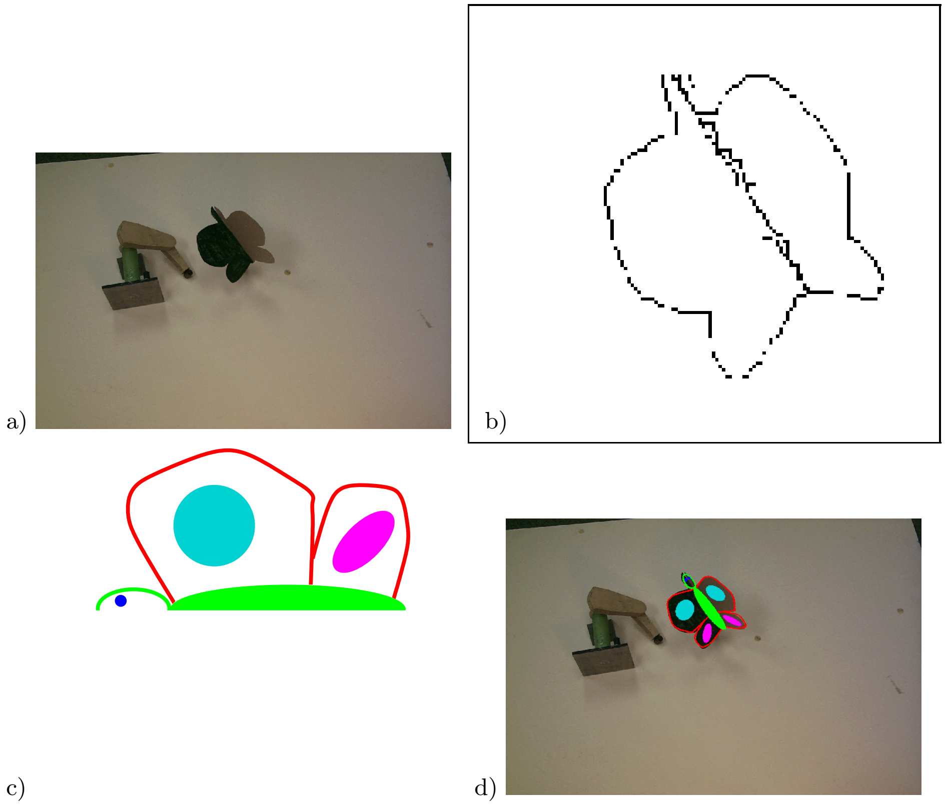

Fig. 6. Snapshot of the video transfer onto a hinged object with a curved boundary. a) One frame (9) of the original sequence (10 frames), b) the edges that are being fit by the two instances of the model, c) the transfer model and d) the corresponding frame from result sequence.

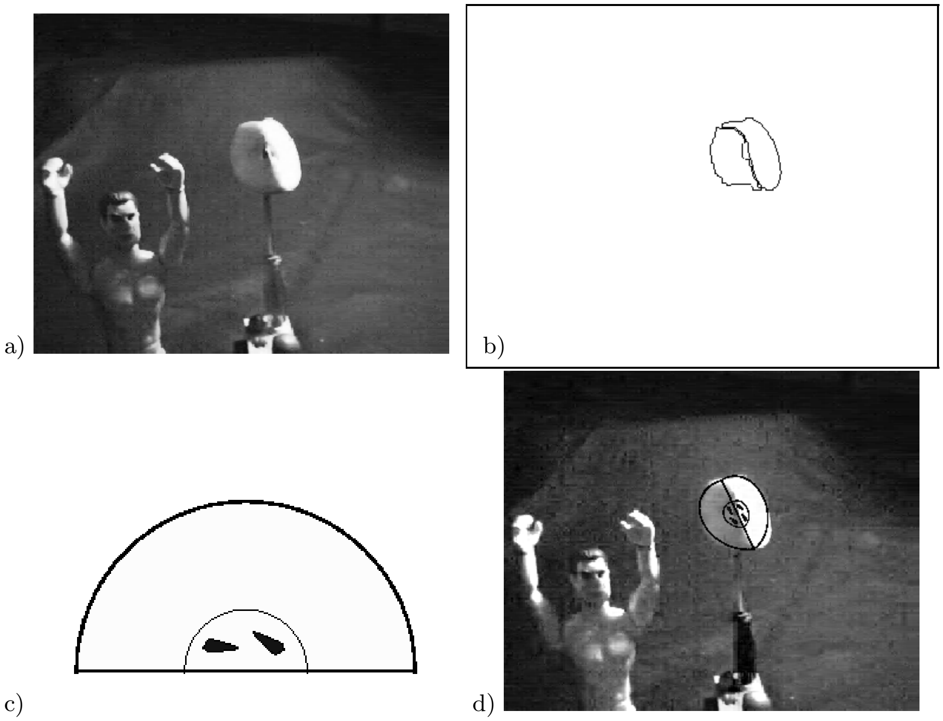

Fig. 7. Snapshot of the video transfer onto a curved boundary. a) One frame (13) of the original sequence (60 frames), b) the edges that are being fit by the two instances of the model, c) the transfer model and d) the corresponding frame from result sequence.

5 Conclusions

This paper has presented a projective Augmented Reality registration algorithm that does not require explicit image feature point correspondences. With this algorithm, we showed that it can be used for more stable registration of augmented reality graphics on top of video, by directly registering to the image edges rather than via a 3D pose estimation. Further, we extended the single plane projection method to incorporate multiple constrained and possibly hinged planes, thus allowing simultaneous apparent projection of 3D structures. Both of the techniques presented here (projective point alignment and constrained alignment) have much potential in AR applications, particularly in man-made environments because of their many individual and joined planar structures. The constrained linkage approach could potentially be used in more general AR applications (such as ensuring objects lie on a groundplane or road), or where the projected object has independently registerable subcomponents, but the approach requires that the projection matrix can be directly estimated. Note that the corresponding points need not be real features, but could be defined by *e.g.* local texture distributions.

If the registered contours are ambiguous, thus producing alternative correspondences, then higher level processing would be needed to resolve the ambiguity. This is a common failing of ICP-like algorithms, which depend on being initially close enough to a solution that correct convergence happens.

Because of the independent frame transformation estimation, the graphics can still have some residual jitter in each frame. This could be smoothed over a video sequence, *e.g.* by Kalman filtering; however, raw video also jitters due to capture electronics instability and human jitter during capture. Thus, the projected graphics needs to be able to track the actual video, rather than an idealized version.

The matlab implementation performance of a few minutes per step suggests that recoding the algorithm in C/C++ and using a more efficient optimization algorithm on a fast PC would give real-time performance.

References

1. J. K. Aggarwal, Q. Cai, W. Liao, B. Sabata. Nonrigid Motion Analysis: Articulated and Elastic Motion. Comp. Vis. and Image Unders., 70(2), pp. 142-156, May 1998.
2. A. P. Ashbrook and R. B. Fisher, “Constructing Models of Articulating Objects: Range Data Partitioning”, Proc. Int. Conf. on Recent Advances in 3-D Digital Imaging and Modeling, Ottawa, Canada, pp 164–172, May 1997.
3. M. Bajura, U. Neumann. Dynamic registration correction in video-based augmented reality systems. IEEE Comp. Graphics and Applic. 15(5), pp 52-60, 1995.
4. P. Beardsley, P. Torr, A. Zisserman. 3D model acquisition from extended image sequences. Proc. ECCV, LNCS 1604/1065, pp 683-695, Springer-Verlag, 1996.
5. M. O. Berger, B. Wrobel-Dautcourt, S. Petitjean, G. Simon. Mixing Synthetic and Video Images of an Outdoor Urban Environment, Machine Vision and Applications 11(3), pp. 145-159, 1999.

6. P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. IEEE Trans. Pat. Anal. and Mach. Intel. 14(2), pp 239-256, Feb 1992.
7. C. S. Chen, Y. P. Hung, J. B. Cheung. RANSAC-based DARCES: a new approach to fast automatic registration of partially overlapping range images. IEEE Trans. Pat. Anal. and Mach. Intel. 21(11), pp 1229-1234, Nov. 1999.
8. Y. Chen, G. G. Medioni. Object modelling by registration of multiple range images. Image and Vision Comp. 10(3), pp 145-155, 1992.
9. C. Dorai, J. Weng, A. K. Jain. Optimal registration of object views using range data. IEEE Trans. Pat. Anal. and Mach. Intel. 19(10), pp 1131-1138, Oct 1997.
10. T. Drummond, R. Cipolla. Real-time Tracking of Multiple Articulated Structures in Multiple Views. Proc. 6th Eur. Conf. on Comp. Vis, Vol II. pp 20–36, 2000.
11. J. Feldmar, N. Ayache, F. Betting. 3D-2D Projective Registration of Free-Form Curves and Surfaces. Comp. Vision and Image Under., 65(3), pp. 403-424, March 1997.
12.
13. D. Gennery. Visual Tracking of Known Three Dimensional Objects. Int. J. Computer Vision, 7(3), pp 2430270, 1992.
14. W. E. L. Grimson. On the recognition of parameterized 2D objects. Int. J. Comp. Vis. 2(4), pp 353–372, 1989.
15. R. Hartley, A. Zisserman. Multiple view geometry in computer vision. Cambridge ; New York : Cambridge University Press, 2000.
16. K. N. Kutalakos, J. R. Vallino. Calibration-free augmented reality. IEEE. Trans. Visualization and Comp. Graphics, 4(1), pp 1-20, 1998.
17. D. G. Lowe. Fitting Parameterized 3-D models to images. IEEE Trans. Pat. Anal. and mach. Intel., 13(5), pp 441-450, 1991.
18. D. G. Lowe. Robust Model-Based Motion Tracking Through the Integration of Search and Estimation. Int. J. Computer Vision, 8(2), pp 113-122, 1992.
19. T. Masuda, N. Yokoya. A robust method for registration and segmentation of multiple range images. Comp. Vision and Image Under. 61(3), pp 295-307, May 1995.
20. G. C. Sharp, S. W. Lee, D. K. Wehe. Invariant features and the registration of rigid bodies. Proc. IEEE Int. Conf. on Robotics and Autom., pp 932-937, 1999.
21. R. A. Smith, A. W. Fitzgibbon, A. Zisserman. Improving augmented reality using image and scene constraints. Proc. Brit. Mach. Vis. Conf, pp 295-304, 1999.
22. M. Tuceryan, D. S. Greer, R. T. Whitaker, D. E. Breen, C. Crampton, E. Ross, K. H. Ahlers. Calibration requirements and procedures for a monitor-based augmented reality system. IEEE Trans. Visualization and Comp. Graphics, 1(3), pp 255-273, 1995.
23. M. Uenohara, T. Kanade. Vision-Based Object Registration for Real-Time Image Overlay. Proc. Computer Vision, Virtual Reality and Robotics in Medicine, CVRMed95(pp). Lecture Notes in CS, Volume 905, Springer-Verlag, pp 14-22, 1995.
24. Z. Y. Zhang. Iterative point matching for registration of free-form curves and surfaces. Int. J. of Computer Vision, 13(2), pp 119-15, Oct. 1994.