Projective ICP and Stabilizing Architectural Augmented Reality Overlays

1 author:

Robert Fisher
The University of Edinburgh
335 PUBLICATIONS **9,851** CITATIONS

Some of the authors of this publication are also working on these related projects:

Project Dermofit View project

Project Fish4Knowledge View project

Division of Informatics, University of Edinburgh

Institute of Perception, Action and Behaviour

Projective ICP and Stabilizing Architectural Augmented Reality Overlays

by

Robert Fisher

Projective ICP and Stabilizing Architectural Augmented Reality Overlays

Robert Fisher

Abstract :

Keywords :

Projective ICP and Stabilizing Architectural Augmented Reality Overlays

Robert B. Fisher

University of Edinburgh and Trinity College Dublin

Summary. The recently developed technique of Iterative Closest Point (ICP) matching has been used for a number of 3D-to-3D and 2D-to-2D point matching applications, and has been further developed in several useful ways, as described below. Central to these applications is the notion of rigid shape matching. This paper extends the concept to projective point matching, in which shapes are related by a projective transform rather than a Euclidean transform. With this extended technique, we show that directly registering 2D Augmented Reality (AR) overlays via a projective transform has greater registration stability than the more 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. Moreover, the technique does not rely on explicit feature point correspondence and tracking. We then further extend the technique to directly register 3D shapes projectively by using mutually constrained 2D projective mappings. These two new techniques enhance the repertoire of methods for producing high-detail, stable augmentation of built scenes.

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 [1,2,12]. This approach is commonly used in architectural AR because of the straightforward 3D scene analysis. 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.

The recently developed technique of Iterative Closest Point (ICP) matching [3] has been used for a number of 3D-to-3D and 2D-to-2D point matching applications, and has been further developed in several useful ways, as

described below. Central to these applications is the notion of rigid shape matching. This paper extends the concept to projective point matching, in which shapes are related by a projective rather than a Euclidean transform.

With this extended technique, we then show that directly registering 2D Augmented Reality (AR) overlays via a projective transform has greater registration stability than the more usual technique of estimating the 3D position of the overlay and then applying pinhole projection. By using the PICP technique, transformations can be estimated without using explicit feature point correspondences. We then further extend the technique to directly register 3D shapes projectively by using mutually constrained 2D projective mappings.

This method 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.

Smith *et al* [11] have also 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 added connected 3D structures (a rectangular solid), by tracking the vanishing points of three sets of parallel lines to define an affine systems and then estimate the camera matrices, subject again to constraints such as parallelness and coplanarity.

Kutalakos and Vallino [8] 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.

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 as well as linear boundaries and working with multiple constrained projections.

1.1   The Rigid Iterative Closest Point Algorithm

ICP [3] is an iterative alignment algorithm that works in three phases: 1) establish correspondence between pairs of features in the two structures that are to be aligned based on proximity, 2) estimate the rigid transformation that best maps the first member of the pair onto the second and then 3) apply that transformation to all features 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 [5], 2) robustifying the algorithm to the influence of outliers and features lacking correspondences [14,9], 3) using a weighted least-square error metric [6], and 4) matching

between features using a metric trading off distance and feature similarity (based local shape invariances) [10]. All of these approaches assume a rigid Euclidean transformation between the corresponding features, whereas the method presented here uses projective correspondence.

2   The Projective Iterative Closest Point Algorithm

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 match. Thus, it is necessary to define a distance measure between projective points, so that we can find the 'closest' points. 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.

2.1   Projective Distance Estimation

Because a point in projective space can be represented by an infinite set of homogeneous coordinates, the normal Euclidean distance could be an unsuitable distance metric. We define a distance metric $d_p()$ between two 2D points with homogeneous representations $\boldsymbol{p}_1 = (x_1, y_1, z_1)'$ and $\boldsymbol{p}_2 = (x_2, y_2, z_2)'$ as:

$$d_p(\boldsymbol{p}_1, \boldsymbol{p}_2) = \cos^{-1}\left(\frac{\boldsymbol{p}_1 \cdot \boldsymbol{p}_2}{\| \boldsymbol{p}_1 \| \, \| \boldsymbol{p}_2 \|}\right)$$

This is the angle between the points, when considered as 3D vectors. $d_p()$ is a true metric (identity, commutativity, triangle inequality - not proved here).

2.2   Projective Transform Estimation

Let $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots \boldsymbol{p}_n\}$ and $\{\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots \boldsymbol{q}_n\}$ 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 [7] to estimate $\mathbf{T}$ such that:

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

Let

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

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:

$$A(2i-1,:) = (p_{ix}, p_{iy}, 1, 0, 0, 0, -q_{ix}p_{ix}, -q_{ix}p_{iy}, -q_{ix})$$
$$A(2i,:) = (0, 0, 0, p_{ix}, p_{iy}, 1, -q_{iy}p_{ix}, -q_{iy}p_{iy}, -q_{iy})$$

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

2.3   PICP Algorithm

Using the results from the previous subsections, we define the Projective ICP algorithm (PICP) as follows (adapted from [10]). Let $\mathcal{S}$ be a set of N_s coplanar 2D 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 projective distance defined in Section 2.1.

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 2.2.

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

3   Planar Structure Registration Using PICP for Augmented Reality

Using the theory developed in Section 2, we look at directly projecting planar graphical structures from the graphical space to the image space. In the next section, we present results for 3D graphical objects composed of planar substructures.

Assume that we are trying to map a planar curve $\mathcal{S} = \{\boldsymbol{c}(\lambda)\}, \lambda \in [0..1]$ (represented homogeneously) onto an image plane using the homography T, that is, drawing $\mathrm{T}\boldsymbol{c}(\lambda)$. We need to estimate T.

Assume that we have identified corresponding structures in the image, and have a set of points $\mathcal{P} = \{\boldsymbol{p}_i\}, i \in [1..n]$ that describe that structure. For example, this might be the boundary of the object as located by an edge detector. Using the theory in Section 2, we want to 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 $\{(T^{[k-1]}\boldsymbol{c}(\lambda_i), \boldsymbol{p}_i)\}$ at the kth iteration of the PICP algorithm, where $T^{[k-1]}$ is the $k-1^{\text{st}}$ estimate of the homography.

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

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

Noise	PICP			3D			PICPe		
	mean	std	max	mean	std	max	mean	std	max
0	0.35	0.04	0.42	0.99	0.45	2.03	0.33	0.03	0.40
10	0.35	0.04	0.49	0.90	0.35	2.34	0.33	0.04	0.44
20	0.38	0.07	0.70	1.03	0.38	2.90	0.33	0.07	0.67
30	0.43	0.11	0.86	1.33	0.44	3.29	0.37	0.10	0.80

3.1   Evaluation

We investigated the PICP approach's stability compared with two alternatives: 1) estimating the 3D transform and then projecting and 2) using the Euclidean distance rather than the projective distance in the portion of the PICP algorithm that finds the closest point. The motivation for the second alternative is that we are attempting to stabilize the graphics in the image plane, so perhaps using image plane distances might be better than the projective distance.

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 a 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.

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.

The 3D transformation used in alternative method 1 is estimated by: 1) estimating the corners of the rectangle from the image edges and 2) searching for the 3D points along the lines of sight through the corners that best fit the

model rectangle. Better 3D performance in this experiment could probably be achieved by estimating the corner positions better.

The experiments recorded two measures of stability: 1) the distance of the estimated graphical object origin from the known true origin and 2) an estimate of the average distance between the true and estimated rectangle's boundary. The former assesses the stability of a given point and the latter assesses the stability of overall shape matching and registration.

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 shows the average boundary distances and the results for the origin distances are similar. It is clear that the boundary alignment algorithms are much more accurate (comparing means) and stable (comparing standard deviations and maximum errors) than the 3D approach at all noise levels. Between the two boundary alignment algorithms, it appears that the using Euclidean distance instead of the projective distance in the feature matching stage produces slightly lower average error.
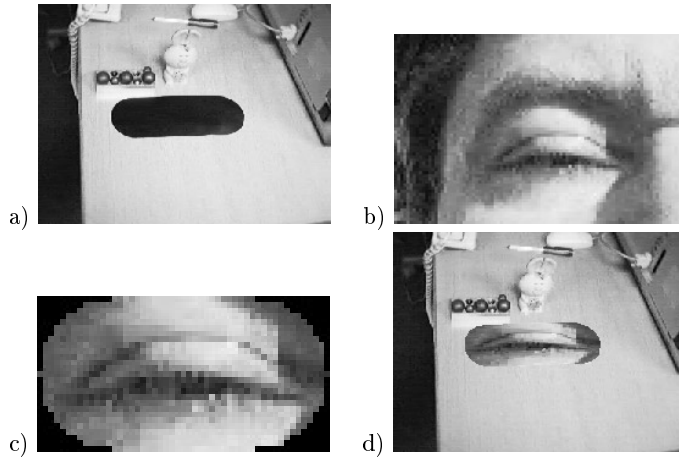

Fig. 1. Snapshot of the video transfer onto a curved boundary. a) One frame of the original sequence, b) one frame of the transfer sequence, c) cropped eye from the transfer sequence, whose boundary is mapped onto the template boundary, and d) corresponding frame from result sequence.

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 the possibility of overlay at around 10 frames per second on a 1 Ghz PC, and real-time video with optimized C/C++ code.

   To demonstrate the performance on a real video sequence, observe the animated GIF at URL: `http://www.dai.ed.ac.uk/homes/rbf/PICP/picp.htm`. This shows the transfer of a video sequence (88 frames) of an eye blinking onto an office interior sequence containing a template 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 1 shows a single frame from the animated sequence.

   A second interior sequence can be see from the same URL, which 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 2 shows one frame of the result sequence (11 frames) and the overlay plane. Note that the transfer still is stable even though perspective distortion is now appearing.
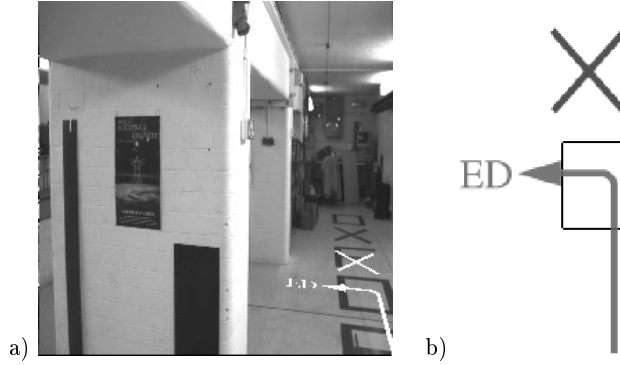

a)   b)

Fig. 2. Snapshot of the transfer onto a corridor scene: a) One frame of the result sequence and b) the transfer overlay, including the registration features.

4   3D Structure Registration Using Constrained PICP

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 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 problem with applying the method of Section 3 directly to the individual planar segments is the individually estimated homographies 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 are coincident).

Formally, let $\mathcal{M}_1 = \{c_1(\lambda_1)\}$ and $\mathcal{M}_2 = \{c_2(\lambda_2)\}$, $\lambda_i \in [0..1]$ be two planar curves (represented homogeneously) mapped by the homographies T_1 and T_2 into a common image plane containing image feature points $S_1 = \{p_{11}, p_{12}, \ldots p_{1n_1}\}$ and $S_2 = \{p_{21}, p_{22}, \ldots p_{2n_2}\}$. (See Figure 3.) 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.
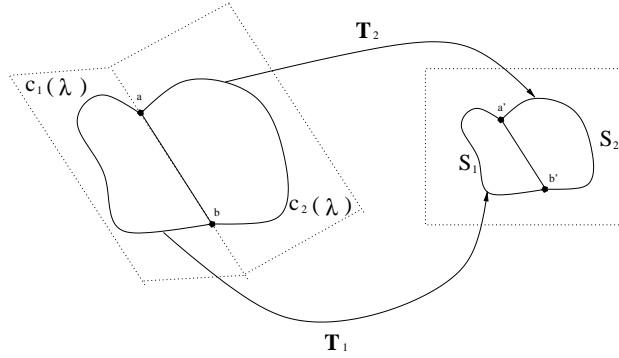

Fig. 3. Projection of 2 non-coplaner curves $c_i(\lambda)$ via homographies T_i into a common image mapping shared point a to a' and b to b'.

Suppose that $c_1(\lambda_{1k})$ and $c_2(\lambda_{2k})$, $k = 1..K$, map to the same image point. 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 minimizes the mapping distance (using $d_p()$) of $\{c_1(\lambda_1)\}$ and $\{c_2(\lambda_2)\}$ onto S_1 and S_2 respectively such that

$$\mu_k T_1 c_1(\lambda_{1k}) = T_2 c_2(\lambda_{2k})$$

μ_k are new unknown variables for the difference in homogeneous scaling.

To solve the problem, we eliminate the μ_k to form six new scalar equalities as follows. Let T_i^j be the jth row of homography T_i. Then isolating μ_k in the

above constraints produces:

$$\mu_k = \frac{T_2^j \boldsymbol{c}_2(\lambda_{2k})}{T_1^j \boldsymbol{c}_1(\lambda_{1k})}$$

for shared points $k = 1..K$ and homogeneous coordinates $j = 1, 2, 3$. One of $j = 1, 2, 3$ can be derived from the other two so we just use $j = 1, 2$. Equating these for μ_k and simplifying gives the constraints:

$$C_{jk}(T_1, T_2) = [(T_2^j \boldsymbol{c}_2(\lambda_{2k}))(T_1^{3-j} \boldsymbol{c}_1(\lambda_{1k})) - (T_2^{3-j} \boldsymbol{c}_2(\lambda_{2k}))(T_1^j \boldsymbol{c}_1(\lambda_{1k}))]^2$$

The goal is to estimate T_1 and T_2 that minimizes

$$E(T_1, T_2) = \sum_{i=1}^{n_1} d_p(T_1^{-1} \boldsymbol{p}_{1i}, \mathcal{M}_1) + \sum_{i=1}^{n_2} d_p(T_2^{-1} \boldsymbol{p}_{2i}, \mathcal{M}_2)$$

subject to $C_{jk}(T_1, T_2) = 0$ for $k = 1..K$ and $j = 1, 2$.

We solve this minimization using Werghi's technique [13] for multiply constrained minimization. Define a cost function

$$F(\rho, T_1, T_2) = E(T_1, T_2) + \rho \sum_{k=1..K; j=1,2} C_{jk}(T_1, T_2)$$

Then we:

1. Compute initial estimates of $T_1^{[0]}$ and $T_2^{[0]}$ independently using the PICP method of Section 3.
2. Set ρ to a small value (1.0).
3. Recompute correspondences between closest model and data edge points, as needed for computing $E()$.
4. Minimize $F()$ using a standard numerical method with $T_1^{[t-1]}$ and $T_2^{[t-1]}$ as the initial estimate to get $T_1^{[t]}$ and $T_2^{[t]}$
5. Increase ρ (50%) and return to step 3 until the desired degree of constraint is achieved. In this case, we want $T_1 \boldsymbol{c}_1(\lambda_{1i})$ and $T_2 \boldsymbol{c}_2(\lambda_{2i})$, $i = 1, 2$ to lie within 0.1 pixel.

If more than two planar segments are in the model, the method generalizes by adding extra constraints for each shared point.

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 4a. The individual semicircle edges are registered to the image edges shown in Figure 4b, and then constrained to share the same straight edge. The resulting constrained mapping projects the model

edges onto the image (Figure 4c). This problem was constrained in about 15 minutes using Matlab on a 270 Mhz Sun.

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 0.53 pixels, with standard deviation 0.087 and maximum average error of 0.71 pixels. Thus, the process is stable below the level of integer pixel edge data to image noise.
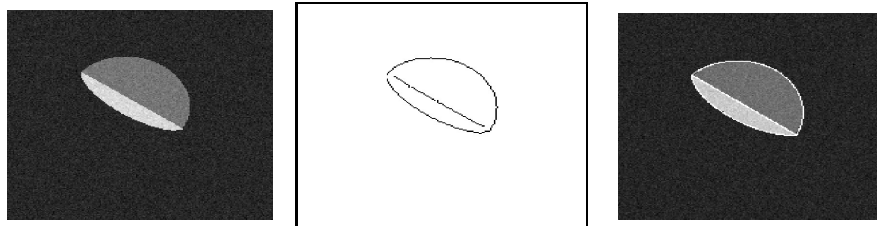

Fig. 4. Test images for 3D registration, showing: a) the raw image, b) the edges used for image capture c) the two registered semicircle models projected onto the raw image.

We also applied the constraint method to a real image, namely for tracking a real apple wedge somewhat similar to the synthetic example. Figure 5 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.

To demonstrate the performance on a real video sequence, observe the animated GIF at URL: `http://www.dai.ed.ac.uk/homes/rbf/PICP/picp.htm`. Here, the matched edges are not as reliably and stably found, nor does the apple have the spherical wedge as in the synthetic example. Thus, the tracking is reasonable but not as stable; however, the constraint is always satisfied. Because of the optimization step, the computation took about 10 minutes per image (again in Matlab on a 270 Mhz Sun). This technique is easily applicable to architectural structures because of the many connected planes.

5   Conclusions

This paper has introduced the PICP registration algorithm that acts in projective, rather than Euclidean space. One of the key advantages of the algorithm is that it does not require explicit 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 planes,
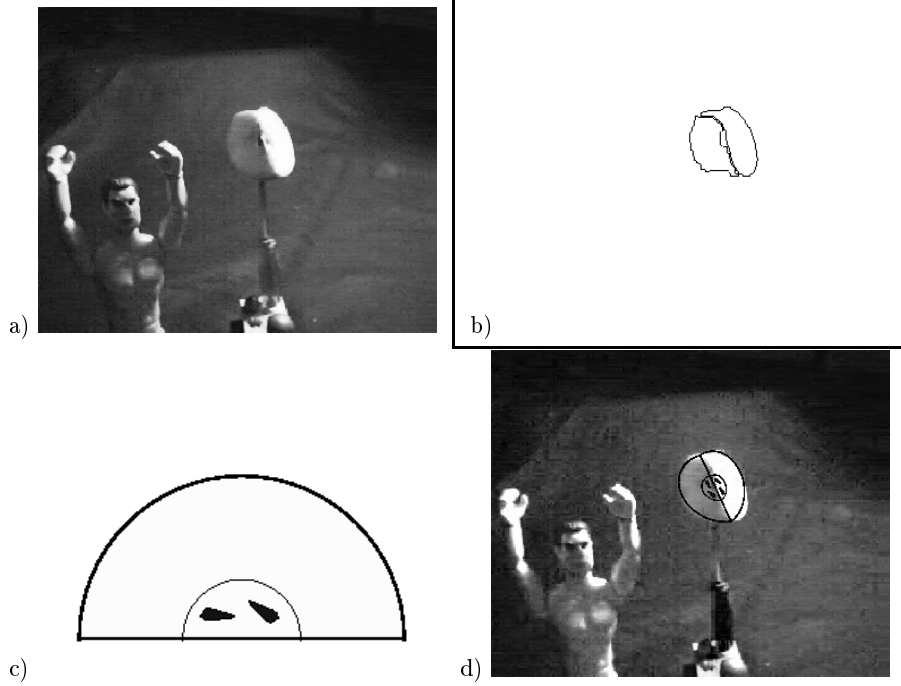
Fig. 5. Snapshot of the video transfer onto a curved boundary. a) One frame (13) of the original sequence (60 frames), b) the edges that are being fit by the two instances of the model, c) the transfer model and d) the corresponding frame from result sequence.

thus allowing simultaneous registration of 3D structures. Both of the techniques presented here (projective point alignment and constraints in alignment) have much potential in AR applications, particularly in man-made environments because of their many individual and joined planar structures. The constraint 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 PICP approach is limited to applications where the projection matrix can be directly estimated from the point correspondences. Note that the corresponding points need not be real features, but could be defined by *e.g.* local texture distributions.

If the registered feature is matched to an occluding contour, then the approach will fail as the transformation requires exact correspondences. However, if the scene has some other features, such as internal markings, that can be used for registration, then estimating the homography might be possible. If 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.

The iterative method demonstrated here has some similarity to bundle adjustment [7]. Bundle adjustment only optimizes the transformation, whereas the algorithm presented here also optimizes the point correspondences. Bundle adjustment can optimize over the whole image sequence, which was not done here. This could be incorporated here as well, at a computational cost.

Because of the independent frame transformation estimation, the graphics can still have some residual jitter in each frame. This could be smoothed, *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.

Acknowledgements

References

1. M. Bajura, U. Neumann. Dynamic registration correction in video-based augmented reality systems. IEEE Comp. Graph. and Appl. 15(5), pp 52-60, 1995.
2. P. Beardsley, P. Torr, A. Zisserman. 3D model acquisition from extended image sequences. Proc. ECCV, LNCS 1604/1065, pp 683-695, Springer-Verlag, 1996.
3. 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.
4. 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.
5. Y. Chen, G. G. Medioni. Object modelling by registration of multiple range images. Image and Vision Comp. 10(3), pp 145-155, 1992.
6. 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.
7. R. Hartley, A. Zisserman. Multiple view geometry in computer vision. Cambridge ; New York : Cambridge University Press, 2000.
8. K. N. Kutalakos, J. R. Vallino. Calibration-free augmented reality. IEEE. Trans. Visualization and Comp. Graphics, 4(1), pp 1-20, 1998.
9. T. Masuda, N. Yokoya. A robust method for registration and segmentation of multiple range images. Comp. Vision and Image Under. 61(3), 295-307, 1995.
10. 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.
11. R. A. Smith, A. W. Fitzgibbon, A. Zisserman. Improving augmented reality using image and scene constraints. Brit. Mach. Vis. Conf, pp 295-304, 1999.
12. 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.
13. N. Werghi, R. B. Fisher, A. Ashbrook, C. Robertson. Object reconstruction by incorporating geometric constraints in reverse engineering. Computer-Aided Design, Vol 31(6), pp 363-399, 1999.
14. 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.