An approximate EM Homographical Iterative Closest Point algorithm.

Markus Louw *Fred Nicolls*

Department of Electrical Engineering
University of Cape Town, South Africa
markus.louw@gmail.com

Abstract

This paper describes an approximately expectation-maximization (EM) formulation of a homographical iterative closest point registration approach (henceforth HICP). We show that such an EM approach allows the algorithm to converge faster, and more robustly in the presence of noise. Although this algorithm can register points transformed by a more general set of linear transformations than the original Iterative Closest Point (ICP) algorithm, it is only appropriate for use on point sets which are related by a homographical transformation, e.g. images taken of a planar scene from different angles, or images taken of a general scene by a stationary pan-tilt-zoom camera. The algorithm is tested on real and synthetic data.

1. Introduction

The Iterative Closest Point Algorithm, [2], aligns point sets by matching each point in the model point set to the closest corresponding point in the scene point set and finding the best rotation to apply to the model point set, to minimize the sum of squared distance errors between each model point and its corresponding scene point. This process is then repeated until no improvement is made. An expectation maximization approach for the ICP algorithm was then developed in [5]. Later, the homographical iterative closest point algorithm was developed in [1]. Although the authors used it to register free form curves, it was ostensibly extensible to registering point sets.

2. Iterative closest point algorithms

In the following three subsections, we describe the iterative closest point algorithm, the homographical iterative closest point algorithm, and the EM-ICP algorithm. Henceforth, we shall talk about a scene point set $\mathbf{x}_s$ which has N points and a model point set $\mathbf{x}_m$, which has M points, each stored in matrix form:

$$\mathbf{x}_s = \begin{pmatrix} \mathbf{x}_s^1 \dots \mathbf{x}_s^i \dots \mathbf{x}_s^N \end{pmatrix}^T.$$

In the algorithms discussed here, we will always attempt to register the model point set to the scene point set, i.e. the scene point set is static, while different transformations are applied to the model point set.

2.1. Formulation of the Iterative Closest Point algorithm (ICP)

The singular value decomposition (SVD) implementation of ICP, (which produces identical results to quaternion method of ICP) is described as follows:

1. Remove means μ_s and μ_m from $\mathbf{x}_s$ and $\mathbf{x}_m$ to form $\mathbf{x}_{st}$ and $\mathbf{x}_{mt}$ respectively.
2. For $i = (1..M)$, for $\mathbf{x}_{mt}^i$ find closest point in $\mathbf{x}_{st}$, which we will call $\mathbf{x}_{st}^{f(i)}$
3. Form a new matrix of corresponding points, $\mathbf{x}_{ct}$, such that
$$\mathbf{x}_{ct} = \begin{pmatrix} \mathbf{x}_{st}^{f(1)} \dots \mathbf{x}_{st}^{f(i)} \dots \mathbf{x}_{st}^{f(N)} \end{pmatrix}^T.$$
4. Calculate the rotation: $[\mathbf{U}, \mathbf{S}, \mathbf{V}] = \mathbf{SVD}(\mathbf{x}_{mt}^T \mathbf{x}_{ct})$, then $\mathbf{R} = \mathbf{UV}$
5. Apply this rotation to the points in $\mathbf{x}_{mt}$: $\mathbf{x}_{mt} \leftarrow \mathbf{R}\mathbf{x}_{mt}$.
6. If, in the previous step, the rotation is greater than a certain amount, (this could be expressed as $||\mathbf{R} - I_{2x2}|| >$ tol), go to step 2, otherwise continue to next step.
7. Algorithm has converged.

2.2. Formulation of the Homographical Iterative Closest point algorithm (HICP)

The first difference between this algorithm and ICP is that the points are represented in homogeneous coordinates, so that a point (x, y, z) represents the point $(x/z, y/z)$ in $\mathbb{R}^2$. Secondly, the calculation of the rotation matrix to apply in step 5 and 6 in the previous section is replaced by a suitable method for calculating the best homography between two point sets. Methods for doing this are presented in [7]. Two appropriate methods are the direct linear transform (DLT) and the Gold Standard Method, the latter of which minimizes the sum of squares distances of the transformed model points to their corresponding scene points. We may describe the HICP algorithm as follows:

1. Remove means μ_s and μ_m from $\mathbf{x}_s$ and $\mathbf{x}_m$ to form $\mathbf{x}_{st}$ and $\mathbf{x}_{mt}$ respectively, by forming the matrices T_m and T_s:
$$T_m = \begin{pmatrix} 1 & 0 & -\mu_m(1) \\ 0 & 1 & -\mu_m(2) \\ 0 & 0 & 1 \end{pmatrix}$$
$$T_s = \begin{pmatrix} 1 & 0 & -\mu_s(1) \\ 0 & 1 & -\mu_s(2) \\ 0 & 0 & 1 \end{pmatrix}.$$
2. Set $H_{\text{cum}} = I_{3x3}$
3. For $i = (1..M)$, for $\mathbf{x}_{mt}^i$ find closest point in $\mathbf{x}_{st}$, which we will call $\mathbf{x}_{st}^{f(i)}$
4. Form a new matrix of corresponding points, $\mathbf{x}_{ct}$, such that
$$\mathbf{x}_{ct} = \begin{pmatrix} \mathbf{x}_{st}^{f(1)} \dots \mathbf{x}_{st}^{f(i)} \dots \mathbf{x}_{st}^{f(N)} \end{pmatrix}^T.$$

5. Calculate best homography between $\mathbf{x}_{mt}$ and $\mathbf{x}_{ct}$, using either DLT or Gold-Standard method.
6. Apply this homography to the points in $\mathbf{x}_{mt}$:

$$\mathbf{x}_{mt} \leftarrow H\mathbf{x}_{mt},$$

$$H_{\text{cum}} \leftarrow HH_{\text{cum}}.$$

7. If, in the previous step, the effect of the transformation is greater than a certain amount, (this could be expressed as $||(H - I_{3x3}|| >$ tol), go to step 2, otherwise continue to next step.
8. Algorithm has converged. The final homography which takes the model points to the scene points is $T_s^{-1} H_{\text{cum}} T_m$.

2.3. Expectation Maximization formulation of ICP

Next we briefly give an account of the EM-ICP method of [5] First, it is necessary to describe the match matrix, which for M model points and N scene points, has dimension MxN. It is filled with binary values, with $A_{ij} = 1$ if there is a hard correspondence between the i_{th} model point and the j_{th} scene point (there is only one hard correspondence per model point per match matrix). Each such match matrix has a probability $p(A|\mathbf{x}_m, \mathbf{x}_s, T)$, given the model points, scene points, and proposed transformation T between them. The expected match matrix $\overline{A}$ is the weighted average of each of these match matrices $p(\overline{A}) = k \sum p(A|\mathbf{x}_m, \mathbf{x}_s, T).A$, (with k a normalizing constant), which gives us an expectation value for the possible matches for each model point. The fact that A is a binary matrix allows us to write

$$p(A) = \prod_{ij} (\overline{A_{ij}})^{A_{ij}} \tag{1}$$

There are obviously a combinatorial number of possible match matrices to evaluate, but the authors of [5] show that since

$$p(A|\mathbf{x}_m, \mathbf{x}_s, T) = \prod_{ij} \left(\frac{\overline{\pi_{ij}}.p(\mathbf{x}_s^i|\mathbf{x}_m^i, T)}{\sum_{ik} \overline{\pi_{ik}}.p(\mathbf{x}_s^i|\mathbf{x}_m^k, T)} \right)^{A_{ij}} \tag{2}$$

where $\overline{\pi_{ij}}$ is a prior on any particular point match, we may through identification of the term inside the product in Eqn.1 and Eqn.2 equate

$$\overline{A_{ij}} = \frac{\overline{\pi_{ij}}.p(\mathbf{x}_s^i|\mathbf{x}_m^i, T)}{\sum_{ik} \overline{\pi_{ik}}.p(\mathbf{x}_s^i|\mathbf{x}_m^k, T)} \tag{3}$$

The EM-ICP algorithm is then formulated as the iterative minimization of the expectation of a cost function. At each iteration the average match matrix is calculated, then the best transformation calculated. The cost function used is:

$$C_{ICP}(T, A) = -\log(\mathbf{x}_s, A|\mathbf{x}_m, T)$$
$$= \sum_{ij} A_{ij} \left(-\log p(\mathbf{x}_s^i)|\mathbf{x}_m^j, T) - \log(\overline{\pi_{ij}}) \right)$$

And the expectation of this cost function, which is the criterion to be minimized, is

$$C(T) = E_A(C_{ICP}(T, A))$$
$$= \sum_{ij} \overline{A_{ij}} \left(-\log p(\mathbf{x}_s^i)|\mathbf{x}_m^j, T) - log(\overline{\pi_{ij}}) \right) \tag{4}$$

2.4. Expectation Maximization formulation of HICP

Although the formulation used for EM-ICP will work equally well for HICP, it can also be done by noting the similarity between this point registration problem (under the assumption of Gaussian noise) and that of training Gaussian Mixture Models (GMMs). Under the assumption of Gaussian noise, each point in the model point set may be regarded as a Gaussian center in a GMM, which may be responsible for any of the scene points (taking into consideration its location and covariance). Therefore we may use the GMM update equations, to find the MAP estimates for the new locations of the model points (although these will not be their actual values after the transformation, these values are determined by the results of the DLT or Gold Standard methods for finding the best homography between the model and scene points). The GMM update equations for training GMMs using the EM method are available in [3] and are as follows, where α_l is the prior on the l_{th} GMM center, μ_l is the l_{th} GMM center's mean, and Σ_l its covariance matrix.

$$\alpha_l^{new} = \frac{1}{N} \sum_{i=1}^{N} p(l|x_i, \Theta^g), \tag{5}$$

$$\mu_l^{new} = \frac{\sum_{i=1}^{N} x_i p(l|x_i, \Theta^g)}{\sum_{i=1}^{N} p(l|x_i, \Theta^g)}, \tag{6}$$

$$\Sigma_l^{new} = \frac{\sum_{i=1}^{N} p(l|x_i, \Theta^g)(x_i - \mu_l^{new})(x_i - \mu_l^{new})^T}{\sum_{i=1}^{N} p(l|x_i, \Theta^g)}, \tag{7}$$

where Θ^g is $(\alpha_l^{old}, \mu_l^{old}, \Sigma_l^{old})$, i.e. the priors, means and standard deviations from the previous iteration, and l is a mixture center number, (so $p(l|x_i, \Theta^g)$ is the probability that mixture center l was responsible for datum x_i).

Since we calculate the homography after taking each expectation, we omit the calculation of the new covariances, since they will not be used, and Eqn.7 assumes the means are updated according Eqn.6, which they are not. Similarly, we don't put priors on the likelihood of any model point being a cause of the scene point measurements, so only Eqn.6 is used in our algorithm. The same equation, written using our notation for point sets, is:

$$\mathbf{x}_m^{j\text{new}} = \frac{\sum_{i=1}^{N} \mathbf{x}_s^i p(\mathbf{x}_m^j|\mathbf{x}_s^i, \Theta^g)}{\sum_{i=1}^{N} p(\mathbf{x}_m^j|\mathbf{x}_s^i, \Theta^g)}, \tag{8}$$

We also experimented with using a covariance measure for each model point with respect to the scene points, as is sometimes used (e.g. [6]) after k-means clustering but before GMM training:

1. For each model point, find the scene points which are closest to that model point, and collect them into a matrix $B_i = (\mathbf{x}_s^g(1) \dots \mathbf{x}_s^g(n))^T$ (if there are n such points for the i_{th} model point). If $n = 0$ then set $\Sigma_i = \sigma.I_{2x2}$ and return, if $n > 0$ continue to step 2.
2. Form the matrix D_i, with $\mathbf{x}_m^i$ subtracted from B_i so that $D_i = B_i - I_{nx1}(\mathbf{x}_m^i)^T$
3. $\Sigma_i = D_i^T D_i / n$

This allows us to calculate a better posterior solution for the locations of the mixture means (or model points, in our case). In our experimental results, this version of the algorithm will be referred to as EM-HICP-K. In the simple EM-HICP algorithm, σ is initialized and decays it each iteration, and all covariance matrices are simply $\Sigma_i = \sigma I_{2x2}$.

2.5. Algorithm outline

1. Form initial transformation matrices which bring barycenters of model and scene points to the origin:

$$T_m = \begin{pmatrix} 1 & 0 & -\mu(x_m) \\ 0 & 1 & -\mu(y_m) \\ 0 & 0 & 1 \end{pmatrix}$$

$$T_s = \begin{pmatrix} 1 & 0 & -\mu(x_s) \\ 0 & 1 & -\mu(y_s) \\ 0 & 0 & 1 \end{pmatrix}$$

2. Initialize σ and $H_{\text{cum}} = \mathbf{I_{3x3}}$

3. Convert points to non-homogeneous representation, and create M virtual scene points:

$$\mathbf{v}_m^j = \frac{\sum_{i=1}^{N} \mathbf{x}_s^i p(\mathbf{x}_m^j | \mathbf{x}_s^i, \Theta^g)}{\sum_{i=1}^{N} p(\mathbf{x}_m^j | \mathbf{x}_s^i, \Theta^g)},$$

4. Convert all points back to homogeneous representation, and calculate the best fitting homography H_t between the model points and the virtual points.

5. Apply the homography H_t to the set of model points, to find the model points for the next iteration: $\mathbf{m}_i \leftarrow H_t \mathbf{m}_i$

6. Update the H_{cum} matrix: $H_{\text{cum}} \leftarrow H_t H_{\text{cum}}$

7. Reduce σ using a decay factor. In our experiments we used $\sigma \leftarrow \sigma f$ where f is a decay factor between 0 and 1.

8. If the algorithm has converged, (if $||H_t - I_{3x3}|| < \text{tol}$), go to step 9, otherwise go to step 3.

9. We calculate the final homography between the two point sets as $H_{fin} = T_s^{-1} H_{\text{cum}} T_m^{-1}$

2.6. Divergence from pure EM

We can see that at stage 5, the above algorithm diverges from a pure expectation maximization algorithm, since the homography which is calculated does not itself maximize the likelihood of the scene and the matches given the model points and transformation, but rather satisfies the constraint that the new model points must lie within a projective homographical transformation of the model points at the previous iteration, in the lest squares sense. Therefore, it is not the expectation of the posterior likelihood of the scene given the model and transformation which is maximized, but rather a compromise between this expectation and constraints imposed by the projective homography restriction. Each method for calculating a homography (DLT, Gold-Standard) will in fact minimize a different criterion in its attempt to calculate the best homography, (algebraic error, Sampson error, etc.), but this is expedient in terms of the speed and ease of implementation of such an algorithm.

3. Experimental Results

To verify the EM-HICP algorithm, we exposed it to real and synthetic data.

σ (noise)	% conv. HICP	% conv. EM-HICP	% conv. EM-HICP-K
0	42	75	78
1	16	42	39
2	31	75	77
3	39	79	83

σ (noise)	ave. iters HICP	ave. iters EM-HICP	ave. iters EM-HICP-K
0	15.2	15.8	11.3
1	24.1	15.3	12.7
2	14.6	15.9	16.1
3	15.2	15.4	16.3

Table 1: Tables of convergence results for the HICP and the EM-HICP algorithm, using synthetic data. Ave. iters is the average number of iterations over the successfully convergent trials, to converge. 150 trials were done for each value of σ, which is the standard deviation of the Guassian noise added to the coordinates of the points in the scene point set.

3.1. Synthetic data

In a single trial using synthetic data, the following procedure was followed: a random point set of 150 points is generated in the range $(x, y) \in ([0, 100], [0, 100])$. A random homography matrix H_r is generated from the product of homographies $H_r = H_s * H_a * H_p$, such that

$$H_s = \begin{pmatrix} s\cos(\theta) & -s\sin(\theta) & t_x \\ s\sin(\theta) & s\cos(\theta) & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

$$H_a = \begin{pmatrix} k1 & k2 & 0 \\ 0 & k3 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and

$$H_p = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ v1 & v2 & 1 \end{pmatrix}$$

and the elements in these matrices were drawn from the following distributions: s $[0.25, 0.75]$, θ $[0, 0.2\pi]$, kl $[0.9, 1.1]$, $k2$ $[0]$, $k3$ $[0.9, 1.1]$, $v1$ $[-0.001, 0.001]$, $v2$ $[-0.001, 0.001]$. We then generate our scene points by multiplying H_r with the set of model points (which are represented in homogeneous coordinates).

$$\mathbf{p}_i = H_r \mathbf{m}_i$$

It should be noted that a homography matrix generated from values outside of these ranges tend to produce synthetic scene points which cannot be registered using either the HICP or the the EM-HICP method, and are are not a useful way of comparing the two algorithms. Finally, we corrupt the newly generated scene points with Guassian noise, so that

$$\mathbf{s}_i = \mathbf{p}_i + n,$$

where n is a sample from a Gaussian distribution with standard deviation σ.

The HICP, and EM-HICP algorithms are then run on the this synthetic data, and the convergence results are shown in Table 1.

Image number	iters (HICP)	iters (EM-HICP)	iters (EM-HICP-K)
1	10	10	10
2	6	5	5
3	DNC	40	38

Table 2: Table of convergence results for the HICP and the EM-HICP algorithm, using real data. Iters is the number of iterations to converge. if convergence was correct. DNC indicates lack of correct convergence.

3.2. Real data

In the experiments involving real data, some real image pairs from a pan-tilt-zoom camera were taken, and some images of (approximately) planar scenes were taken. This guarantees that corresponding points between the two images are related by a planar homography. To extract corners from each of the image pairs, the Harris corner detector [4] was used. Obviously the performance of the corner detector/feature extractor, and the reliability with with it will extract the same points in corresponding scenes affects the outcome of any ICP algorithm quite seriously. For example, the Harris detector performs unreliably when a straight edge changes orientation between images, where it is prone to selecting corners from this edge in one image more than in the other. Therefore, where the HICP registration algorithm has failed, it is usually the failure of the feature detector to produce consistent features across images. A table of convergence results is shown in Table 2.

4. Discussion

Results show that EM-HICP-K performs slightly better than the EM-HICP, and each of them perform better than HICP, in terms of successful convergence. The average number of iterations per successful convergence could be improved for EM-HICP and EM-HICP-K if at some appropriate point the algorithm was forced into choosing hard correspondences for each model point, since many iterations are wasted close to the correct local minimum.

5. Conclusions

We can see that the EM-HICP algorithm is faster than the HICP algorithm to converge to the true solution, and is more robust to converge to the correct solution when Gaussian noise is added to the synthetic data. In the real data, which consists of corner features extracted from image pairs taken by pan-tilt-zoom cameras and of planar scenes under general intrinsic and extrinsic parameters, the EM-HICP algorithm performed better than the HICP algorithm. Moreover, when covariances are estimated for each model point, the algorithm (referred to previously as EM-HICP-K) becomes slightly more reliable than when the same spherical covariance matrix is used for each point for an iteration.

6. References

[1] Y. Doytsher A. Akav, G. H. Zalmanson. Linear feature based aerial triangulation. *ISPRS*, 2004.

[2] P.J. Besl. A method for registration of 3d shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1992.

[3] J.A. Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *Technical report*, 1998.

[4] M. Stephens. C. Harris. A combined corner and edge detector. *Proc. Alvey Vision Conference*, pages 189–192, 1988.

[5] S. Granger and X. Pennec. Multi-scale em-icp: A fast and robust approach for surface registration. *ECCV 2002*, 2002.

[6] Ian T. Nabney. *NETLAB: Algorithms for Pattern Recognition*. Springer, 2001.

[7] A. Zisserman R. Hartley. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.