# N-View Point Set Registration: A Comparison

S. J. Cunnington and A. J. Stoddart

Centre for Vision, Speech and Signal Processing

University of Surrey, Guildford, GU2 5XH, UK

`{a.stoddart,s.cunnington}@ee.surrey.ac.uk`

**Abstract**

Recently 3 algorithms for registration of multiple partially overlapping point sets have been published by Pennec [11], Stoddart & Hilton [12] and Benjemma & Schmitt [1]. The problem is of particular interest in the building of surface models from multiple range images taken from several viewpoints. In this paper we perform a comparison of these three algorithms with respect to cpu time, ease of implementation, accuracy and stability.

## 1 Introduction

Several authors have considered the problem of building complete surface models of complex objects using range images taken from several views [2, 4, 5, 6]. Since the viewpoints (or object poses) are usually not known it is necessary to register the surfaces taken from various views prior to fusion [8].

In the case of 2 views the iterated closest point algorithm (ICP) [3] may be used to register surfaces. This algorithm requires a solution to the 3D point set registration problem under rotation and translation. Several analytic solutions are available, see [10] and references contained therein.

When more than 2 views must be registered the ICP algorithm may still be used, provided that a solution for the N-view point set registration problem is available. The N-view point set registration problem may be reduced to a chain of pairwise problems and solved with the 2 view algorithm, however this is not an optimal solution. Information present in the unused overlapping view pairs should also be used for an optimal solution.

Recently 3 algorithms for registration of multiple partially overlapping point sets have been published [11, 12, 1]. The relative merits have not yet been studied. In the 2 view case a thorough evaluation of the various techniques has been performed by Eggert *et al* [7]. The purpose of this paper is similar to the Eggert work and we present a comparison of 3 N-View registration methods.

## 2 N-View Registration

Several analytical solutions exist for the 2 view point set registration problem. These methods decouple the rotation and translation, and solve for the rotation by computing the SVD of a $(3 \times 3)$ matrix [10] or the eigenvectors of a $(4 \times 4)$ matrix [9]. The translation is then usually solved by calculating the distance between rotated centroids.

We will study the methods of Stoddart and Hilton (SH) [12], Pennec [11] and Benjemaa and Schmitt (BS) [1]. All the current methods for N-view registration are iterative. However, Benjemaa and Schmitt made a significant advance insofar as they have been able to analytically decouple the rotation and translation.

## 2.1 Problem definition

Each of the three papers is presented in a distinct notational framework. We attempt to bring about some degree of consistency in our description. We begin by defining the problem following the notation of Benjemaa and Schmitt.

Benjemaa and Schmitt assume that there are $M$ point sets each taken from a different viewpoint, $S^1 \ldots S^M$, where $S^\alpha = \{p_1^\alpha \ldots p_N^\alpha\}$. The objective is to find the best rigid body transforms, $f^1 \ldots f^M$, to apply to each point set. A rigid body transform is denoted as $f^\alpha = \{R^\alpha, T^\alpha\}$, where $R^\alpha$ denotes the rotation and $T^\alpha$ denotes the translation. Hence, $f^\alpha * p = R^\alpha p + T^\alpha$.

The overlap of $S^\alpha$ with $S^\beta$ is denoted as $O^{\alpha\beta} \subset S^\alpha$ where $O^{\alpha\beta} = \{p_1^{\alpha\beta} \ldots p_{N^{\alpha\beta}}^{\alpha\beta}\}$. $O^{\alpha\beta}$ has $N^{\alpha\beta}$ points where each point $p_i^{\alpha\beta}$ is matched with $p_i^{\beta\alpha} \in O^{\beta\alpha} \subset S^\beta$. Therefore $N^{\alpha\beta} = N^{\beta\alpha}$. Schmitt also states that $O^{\alpha\alpha} = \emptyset$ and $N^{\alpha\alpha} = 0$ for convenience in subsequent formulae.

The problem may be specified as minimising over the $N$ transforms $f^\alpha$ the cost $E$ where

$$E[f^1 \ldots f^M] = \sum_{\alpha=1}^{M} \sum_{\beta=1}^{M} \sum_{i=1}^{N^{\alpha\beta}} w_i^{\alpha\beta} \parallel f^\alpha * p_i^{\alpha\beta} - f^\beta * p_i^{\beta\alpha} \parallel^2 \tag{1}$$

where the weights $w_i^{\alpha\beta}$ are given. We note that the problem is undetermined up to a global transformation applied to all point sets. This can be removed by requiring that $f^1$ is the identity transform.

## 2.2 Pennec

Pennec's [11] method is by far the easiest to implement (provided that one of the 2 view algorithms is already available!) It is iterative and based on the concept of 'mean shape'. In [11] Pennec introduced a formal framework in which he was able to define the averaging of shape. Each view has corresponding points on the mean shape to which it is registered using a standard point set registration method, such as Horn et al. [9]. At the beginning of the next iteration, the new mean shape is calculated, and again the views are registered to it. This continues until convergence.

Pennec considers an object represented by a $k$-tuple $X = \{x_1 \ldots x_k\}$. A rigid transform $f$ applied on $X$ is simply $f * X = \{f * x_1 \ldots f * x_k\}$. In the real world, however, there are often missing data and thus incomplete $k$-tuples.

If we have several noisy $k$-tuples of points we may wish to compute the mean shape $\mathtt{M} = \{\mathtt{m}_1 \ldots \mathtt{m}_k\}$. Pennec provides a lengthy discussion of formal issues related to mean shape, but for our purposes it is sufficient to adopt the obvious definition of the mean shape. Suppose we have N $k$-tuples $X^1 .. X^N$ with $X^j = \{x_1^j \ldots x_k^j\}$ then the mean shape may be defined as

$$\mathtt{m}_r = \frac{\sum_{j=1}^{N} w_r^j x_r^j}{\sum_{j=1}^{N} w_r^j} \tag{2}$$

The weight $w_r^j$ may be set to zero in the case of missing data.

At first glance this may appear to be a different problem to our own but our problem may be converted into a mean shape problem. This is done by reordering the points and associating each measurement with some point $r$ in a $k$-tuple by use of a mapping $r(i, \alpha, \beta)$. Hence the mean shape is defined as

$$\mathtt{m}_{r'} = \sum_{r=1}^{k} \delta_{r'r} \frac{\sum_{\alpha=1}^{M} \sum_{\beta=1}^{M} \sum_{i=1}^{N^{\alpha\beta}} w_i^{\alpha\beta} p_i^{\alpha\beta} r(i, \alpha, \beta)}{\sum_{\alpha=1}^{M} \sum_{\beta=1}^{M} \sum_{i=1}^{N^{\alpha\beta}} w_i^{\alpha\beta} r(i, \alpha, \beta)} \tag{3}$$

Once we have a mean shape defined we can register one view at a time to this mean shape by minimising the residuals over $f^\alpha$

$$E[f^\alpha] = \sum_{r=1}^{k} \sum_{\beta=1}^{M} \sum_{i=1}^{N^{\alpha\beta}} w_i^{\alpha\beta} r(i, \alpha, \beta) \parallel \mathtt{m}_r - f^\alpha * p_i^{\alpha\beta} \parallel^2 \tag{4}$$

$f^\alpha$ is solved by using a closed-form solution such as Horn et al. [9]. The solution to the N-view registration problem is obtained by iterating over the two steps of computing mean shape and registering all views to the mean shape.

To summarise, the mean surface is first computed. Next, the optimal transforms for each view are then solved. The mean surface is then recomputed taking into account the new transforms. The transforms are again solved for, and this process continues until convergence.

## 2.3 Stoddart and Hilton

Stoddart and Hilton [12] use an iterative numerical method based on gradient descent. The problem is solved by analogy with a physical system of rigid bodies connected by springs. The friction dominated equations of motion dictate a solution that evolves over time to a local minimum in potential energy. By integrating the equations of motion over time we solve the registration problem.

To start off the process, guesses must be supplied. During the computation, all transforms associated with each view vary simultaneously. Once the computation is complete, all views are transformed so that the first view is transformed by the identity transform.

Each view is associated with a rigid body having an arbitrary center of mass and moment of inertia. These parameters do have considerable effect on the rate of convergence. A sensible choice for the parameters is as follows. The center of mass should be set to the centroid of a point set and the moment of inertia is chosen as if each object were a sphere of radius equal to half the diagonal of a bounding box containing the data.

SH consider each pair of corresponding points to be connected by a spring of strength $w_i^{\alpha\beta}$ making it is possible to compute forces applied by these springs. The individual spring forces may be combined into an overall force $F_{tot}^\alpha$ acting on the center of mass and a torque $\tau_{tot}^\alpha$ around the center of mass of a rigid body for each view. In [12] it is shown how to compute these in a very efficient way.

The force and torque are inserted into a dynamical system which moves towards a potential minimum. The following friction dominated equations of motion are chosen.

$$\gamma \frac{dy_{cm}^\alpha}{dt} = F_{tot}^\alpha \tag{5} \qquad\qquad \Gamma \omega^\alpha = \tau_{tot}^\alpha \tag{6}$$

$\gamma$ resembles the mass and $\Gamma$ the moment of inertia, but here they represent the drag and

rotational drag coefficients. $\omega$ is the angular velocity (rate of change of orientation with respect to time).

The system of equations are then integrated by a simple quality controlled Euler method which can solve the dynamical system with adaptive step size. It is guaranteed to converge to a local minimum.

## 2.4 Benjemaa and Schmitt

Benjemaa and Schmitt [1] use a quaternion approach similar to that of Horn et al. [9].

The general approach is as follows. Firstly the translations are all eliminated analytically. One view, or point set, is used as a reference frame by having an identity translation and rotation associated with it which remain constant. During each iteration the rotation for all of the other point sets are solved. Only one point set at a time is allowed to be moved so that the rotation can be solved for it, while the others are kept fixed. Once that rotation is solved, the rotation for the next point set is determined. This continues until all of the optimal rotations have been obtained. This process is continued until convergence.

### 2.4.1 Optimal Translation

Benjemaa and Schmitt shows that the optimisation of rotations can be decoupled from the values of the translations. The optimal translations are then obtained by using a linear combination of differences between rotated centroids.

By ignoring the weight component of equation (1) and expressing $f^\alpha$ in its two components of rotation $R^\alpha$ and translation $T^\alpha$, it can be written as follows.

$$E = \sum_{\alpha=1}^{M} \sum_{\beta=1}^{M} \sum_{i=1}^{N^{\alpha\beta}} \parallel R^\alpha p_i^{\alpha\beta} - R^\beta p_i^{\beta\alpha} + T^\alpha - T^\beta \parallel^2$$

Schmitt rewrites the cost function as $E = E_R + E_{t,R}$, where

$$E_R = \sum_{\alpha=1}^{M} \sum_{\beta=1}^{M} \sum_{i=1}^{N^{\alpha\beta}} \parallel R^\alpha p_i^{\alpha\beta} - R^\beta p_i^{\beta\alpha} \parallel^2$$

and goes on to show that $E_{t,R}$ can be written in a matrix form. Minimising $E_{t,R}$ becomes then equivalent to the minimisation of $Q(X)$, where $X$ contains the unknown translations for the views and $A$ & $B$ are matrices computed from the data points [1].

$$Q(X) = X^T A X + 2 X^T B \tag{7}$$

By setting $R^1 = I$ and $T^1 = 0$, the first point set is fixed and equation (7) becomes,

$$Q(\bar{X}) = \bar{X}^T \bar{A} \bar{X} + 2 \bar{X}^T \bar{B} \tag{8}$$

where $\bar{X}$ and $\bar{B}$ are the vectors $X$ and $B$ deprived of their first element, and $\bar{A}$ is $A$ without the first row and column. Schmitt notes that $Q(\bar{X})$ is a quadratic form which is minimal when $\bar{A}\bar{X} = -\bar{B}$. Hence the translations can be simply obtained by the inversion of the matrix $\bar{A}$.

$$\bar{X}_{min} = -\bar{A}^{-1} \bar{B} \tag{9}$$

### 2.4.2 Optimal Rotation

Schmitt uses the result obtained in equation (9) and substitutes $\bar{X}_{min}$ for $\bar{X}$ in equation (8) to obtain equation (10).

$$Q(\bar{X}_{min}) = -\bar{B}^T \bar{A}^{-1} \bar{B} \tag{10}$$

$E_{t,R} = 2Q(\bar{X}_{min})$ and is now expressed in terms of rotations. Schmitt goes on to show that minimising $E$ is equivalent to maximising $H$.

$$H = \sum_{\alpha=1}^{M} \sum_{\beta=1}^{M} \sum_{i=1}^{N^{\alpha\beta}} R^\alpha p_i^{\alpha\beta} \cdot R^\beta p_i^{\beta\alpha} + \bar{B}^T \bar{A}^{-1} \bar{B} \tag{11}$$

Schmitt uses properties of quaternions to re-express the first term of H as

$$H_1 = \sum_{\alpha=1}^{M} \sum_{\beta=1}^{M} (\dot{q}^{*\beta} \dot{q}^\alpha)^T \mathbb{Q}_R^{\alpha\beta} (\dot{q}^{*\beta} \dot{q}^\alpha) \tag{12}$$

where $\dot{q}$ is a unit quaternion and $\dot{q}^*$ the conjugate of $\dot{q}$.

Schmitt shows that the 2nd term of $H$, $\bar{B}^T \bar{A}^{-1} \bar{B}$ can be expressed as

$$\bar{B}^T \bar{A}^{-1} \bar{B} = \sum_{\alpha=1}^{M} \sum_{\beta=1}^{M} (\dot{q}^{*\beta} \dot{q}^\alpha)^T \mathbb{Q}_t^{\alpha\beta} (\dot{q}^{*\beta} \dot{q}^\alpha) \tag{13}$$

The reader is advised to consult [1] for more details. Finally the problem for each view reduces to a problem of the form

$$H(\dot{q}^j) = 2\dot{q}^{j^T} N^j \dot{q}^j, \tag{14}$$

where $N^j = \sum_{\beta=1, \beta \neq j}^{M} Q^{*\beta^T} \mathbb{Q}^{j\beta} Q^{*\beta}$, and where Q is the quaternion matrix form.

$H(\dot{q}^j)$ is a quadratic form, and so the optimal unit quaternion which maximises this function is the eigenvector corresponding to the highest eigenvalue of the matrix $N^j$.

## 3 Results

To characterise the three methods a series of numerical experiments were performed to determine the rate of convergence, accuracy, stability, and computational time required by the methods.

The reader may also wish to consider implementation issues alongside other criteria. In our subjective opinion having implemented all three algorithms the Pennec algorithm is by far the easiest to implement. The other two algorithms are both complex algorithms needing significant effort to implement.

In addition the SH algorithm has several free parameters which are chosen heuristically. The present implementation is based on a quality controlled Euler routine which requires some parameters to be set. In contrast the Pennec algorithm requires no parameters to be chosen other than the termination criterion and threshold. The same applies to Benjemaa and Schmitt.

## 3.1 Creation of Synthetic Data Sets

Synthetic data sets were generated from 3D surface models by a process intended to emulate a multiple view range data acquisition.

We begin by selecting a 3D model. We choose some number of points at random on the surface. For each of the $\alpha = 1..M$ views a view direction was chosen and the subset of the total set was chosen that was visible from the specified view direction. This results in the point sets $S^\alpha$.

It is in principle possible for every point set to overlap with every other point set. We specified overlaps to take place only when a point was simultaneously visible from 2 views. Thus we obtain the overlap sets (also called correspondence sets) $O^{\alpha\beta} = O^{\beta\alpha}$. The number of correspondence sets generated will depend on the number of views specified and the characteristics of the 3D model used.

When adding a point to a correspondence set, each view will have an identical copy of that point. Hence the views within each correspondence set are perfectly registered. After adding noise it will no longer be true that $O^{\alpha\beta} = O^{\beta\alpha}$.

The next step in creation of simulated data is the creation of a number of views. The views were chosen to get the maximum coverage of the 3D model. For each experiment the same views were used. The number of views chosen were 2, 3, 6 and 18 which allows a sequence of tests of increasing difficulty.

The views $(0, 0, 1)$ and $(0, 1, 0)$ were used in the two view case. The three view case used the additional view $(0, 0, -1)$, and for the six view case, the additional views $(-1, 0, 0)$, $(0, 1, 0)$ and $(0, -1, 0)$ were used. The eighteen view case was generated by rotating the six views around the x,y & z axes individually by $45°$.

In order to test the algorithms we need to start from some erroneous position. We choose these positions as follows. The transform for the first view (view 0) is always null, and thus can be used as a reference frame. For each view the rotation and translation is incremented. The rotation for the second view (view 1) is $1°$ and for each subsequent view the angle is increased by $1°$. The rotation axis is always $(1, 1, 1)$. The translation for view 1 is $(0.2, 0.2, 0.2)$, and for subsequent views the $x$, $y$ and $z$ components are incremented by 0.2. Hence each view has a unique rotation and translation associated with it.

Finally we add zero mean Gaussian noise with rms $\sigma$ to each coordinate of the synthetic measurements. [ This corresponds to isotropic noise with rms $\sqrt{3}\sigma$ when considering the rms error on vectors.]

The noise is set in terms of a percentage $p$ of the diagonal of the bounding box of the noise free data, $B$ as follows

$$\sigma = pB/100 \tag{15}$$

## 3.2 Quantitative Measures Used

There are two quantitative measures that we can use to evaluate the result of registration. We can consider the errors between the ground truth and the estimated rotations and translations for each view. For convenience we report the error for the last view denoted $\delta\theta$ and $\delta T$, the former in units of degrees.

The second measure is the residuals between corresponding points after registration.

This should be a weighted average over all the point pairs and is given by

$$e = \frac{\sum_{\alpha}^{M} \sum_{\beta}^{M} \sum_{i}^{N^{\alpha\beta}} w_i^{\alpha\beta} \parallel f^{\alpha} * p_i^{\alpha\beta} - f^{\beta} * p_i^{\beta\alpha} \parallel^2}{\sum_{\alpha}^{M} \sum_{\beta}^{M} \sum_{i}^{N^{\alpha\beta}} w_i^{\alpha\beta}} \qquad (16)$$

Care should be taken to compute $e$ from the above expression as we have found that some mathematically equivalent expressions are inaccurate for small $e$ due to roundoff errors.

Since we have added a known amount of noise $\sigma$ to each component of the point pair we expect that

$$e = \sqrt{3}\sqrt{2}\sigma \qquad (17)$$

The $\sqrt{3}$ takes account of the 3 components $(x, y, z)$, and the $\sqrt{2}$ accounts for the fact that noise has been added to both points.

## 3.3   Convergence

We begin by considering data sets with no noise added. This is an artificial problem as it is possible to solve the problem by registering views in a pairwise manner. However this case is very useful for determining the rate of convergence of the algorithm.

The dataset is derived from a surface model of an icosahedron with unit radius. Firstly we examine the 2 view case with 50 random points per view. Figure 1 shows the convergence of $e$ as a function of iteration number. We see that BS converges in 1 step, Pennec converges in 1 step and SH converges in 45 steps. The one step convergence of BS is expected since in the 2 view case it is equivalent to existing analytic methods. As a purely numerical method the convergence of SH is as expected. The method of Pennec is somewhat faster than might be expected in this case, but we recall that it too contains a 2 view analytic method within. For this problem we expect $e$ to converge to zero, we observe that all methods converge to a number in the region of $10^{-15}$, in other words the algorithms all converge to a number close to full machine precision.

A more meaningful test of the algorithm is a case with more than 2 views. The next case we consider has 200 points sampled from the icosahedron and 6 views. There were 12 overlap sets. No noise was added. The convergence is illustrated by the graphs in figure 2. As can be seen all methods show geometric convergence but SH and Pennec converge faster than Schmitt. The results are summarised in table 1. We see that all methods converge to full machine precision. The fastest method is SH.
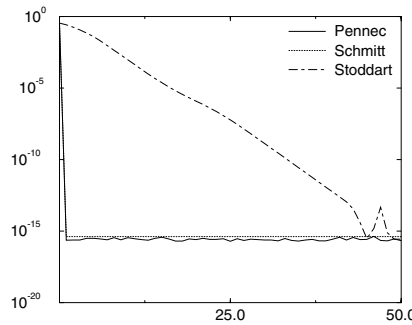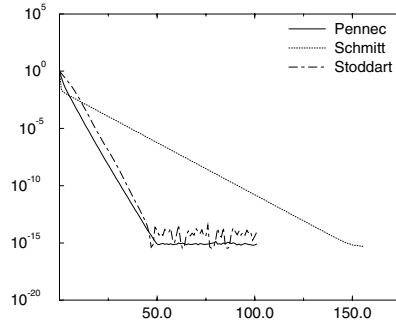


Figure 1: $e$, 50 Points, 2 Views, No Noise

Figure 2: $e$, 200 Points, 6 Views, No Noise

Table 1: 200 Points, 6 Views, No Noise

| method | iterations | cpu | $e\ 10^{-16}$ | $\delta\theta\ 10^{-14}$ | $\delta T\ 10^{-16}$ |
|---|---|---|---|---|---|
| Pennec | 51 | 4.99 | 7.81 | 2.99 | 9.93 |
| Schmitt | 156 | 0.80 | 5.60 | 2.62 | 5.44 |
| Stoddart | 48 | 0.24 | 81.99 | 32.28 | 25.51 |

In the next case we add noise equivalent to 0.5% of the diagonal of the bounding box. The results are summarised in figure 3 and table 2. The predicted value for $e$ is 0.0353 which is consistent with the result in the table.
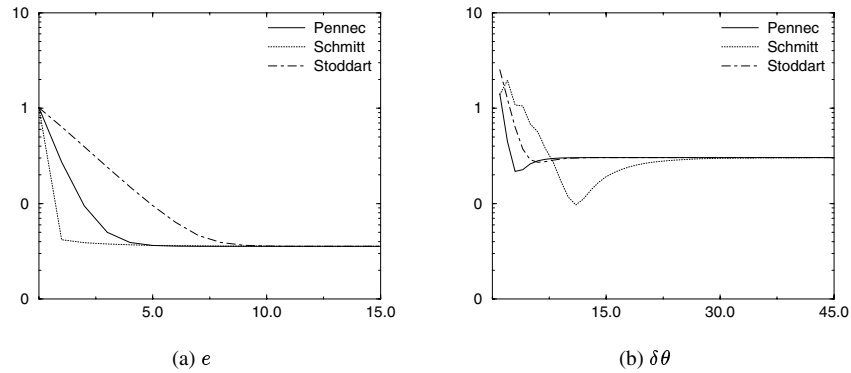


(a) $e$      (b) $\delta\theta$

Figure 3: 200 Points, 6 Views, 0.5 Noise

Table 2: 200 Points, 6 Views, 0.5 Noise

| method | iterations | cpu | $e$ | $\delta\theta$ | $\delta T$ |
|---|---|---|---|---|---|
| Pennec | 10 | 1.00 | 0.0356877 | 0.303154 | 0.00305913 |
| Schmitt | 12 | 0.06 | 0.0356877 | 0.303154 | 0.00305913 |
| Stoddart | 15 | 0.07 | 0.0356877 | 0.303154 | 0.00305913 |

An unexpected result is the overshoot of BS in the angle graph which is not visible in the graph of $e$. It does seem that Schmitt is more affected by increasing number of views as can be seen in a figure of $\delta\theta$ convergence for the 200 point 18 view case shown in figure 4.
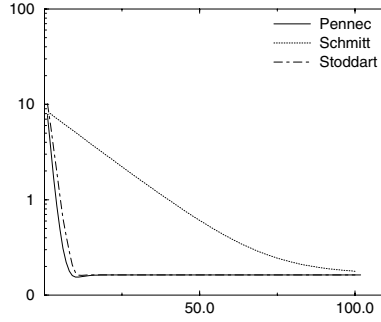
241

Figure 4: $\delta\theta$, 200 Points, 18 Views, 0.5 Noise

## 3.4 Highly nonspherical models

The results in the previous section are representative of the overall behaviour of the various methods as applied to a dataset that comes from a regular approximately spherical shape.

It is our belief that there are several situations where the behaviour of the algorithm may be much worse. We have tested one such case in which the data comes from a highly non spherical object.

The object is generated from the previously used icosahedron by scaling two axes by a factor of 1000. The result is a long thin cigar shaped object. The reader will recall that when points are collinear in the two view case registration is not possible.

We consider the 6 view case with 200 points and no noise. The results are shown in figure 5 and table 3. It is clear that the SH method now fails completely and BS produces a significantly worse answer than Pennec.
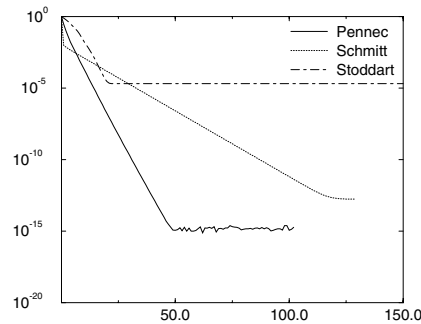


Figure 5: Degenerate - $e$, 200 Points, 6 Views, No Noise

Table 3: Degenerate - 200 Points, 6 Views, No Noise

| method | iterations | cpu | $e$ | $\delta\theta$ | $\delta T$ |
|---|---|---|---|---|---|
| Pennec | 50 | 5.44 | 1.889e-15 | 1.186e-10 | 2.927e-12 |
| Schmitt | 129 | 0.70 | 1.729e-13 | 9.183e-09 | 2.268e-10 |
| Stoddart | 22 | 0.13 | 1.978e-05 | 2.848 | 0.070 |

As expected the angular error has become much worse ($10^{-10}$) due to the fact that we have begun to approach a degenerate case.

242

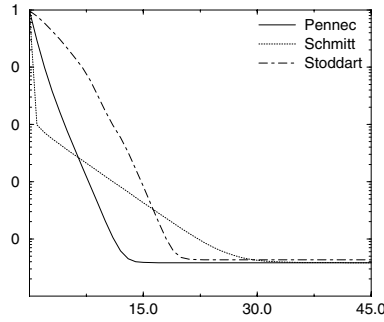If we add noise of 0.001% of the bounding diagonal we get the results shown in figure 6 and table 4.



Figure 6: Degenerate - $e$, 200 Points, 6 Views, 0.001 Noise

Table 4: Degenerate - 200 Points, 6 Views, 0.001 Noise

| method | iterations | cpu | $e$ | $\delta\theta$ | $\delta T$ |
|---|---|---|---|---|---|
| Pennec | 18 | 1.94 | 3.847e-05 | 0.0266727 | 0.0006627 |
| Schmitt | 41 | 0.21 | 3.847e-05 | 0.0266725 | 0.0006627 |
| Stoddart | 24 | 0.14 | 4.308e-05 | 2.84937 | 0.0703454 |

In figure 7 we also show the behaviour of $\delta\theta$ under noise for 3, 6 and 18 views. Some unusual convergence behaviour is visible for the BS method but it does make steady progress to the solution as measured by $e$.
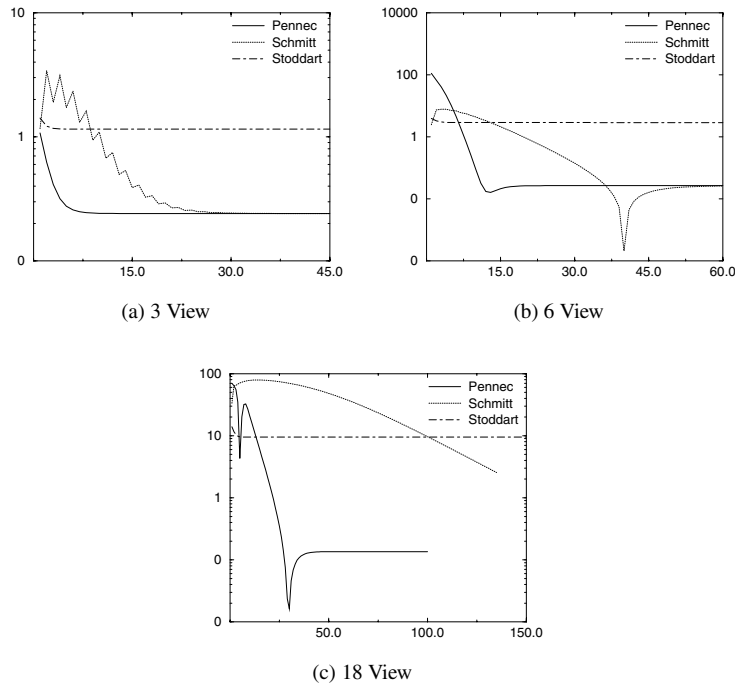


(a) 3 View



(b) 6 View



(c) 18 View

Figure 7: Degenerate - $\delta\theta$, 200 Points, 0.001 Noise

# 4  Conclusion

It is clear that Pennec's method is by far the easiest to implement. There are no parameters to choose. Its rate of convergence is geometrical. It is the only method that consistently gives high accuracy solutions.

We have also seen that Pennec's method is by far the slowest and in some applications we can imagine the additional cpu time would not be a major disadvantage. That it is the slowest is a inevitable consequence of the fact that the other algorithms use cpu time proportional to the number of points *added* to the number of iterations whereas Pennec uses time proportional to the number of points *multiplied* by the number of iterations.

The BS method is harder to implement and suffers from a slight loss of accuracy for the near degenerate case. [ This may be a flaw in our implementation. ] If speed is the most important criterium it is the best algorithm.

The SH method has the disadvantage of requiring additional parameters to be chosen. It fails in the near-degenerate case.

# References

[1]  R. Benjemaa and F. Schmitt. A solution for the registration of multiple 3d point sets using unit quaternions. In *Fifth European Conference on Computer Vision*, pages 34–50, Freiburg, Germany, 1998.

[2]  R. Bergevin, D. Laurendeau, and D. Poussart. Registering range view of multipart objects. *Computer Vision and Image Understanding*, 61(1):1–16, 1995.

[3]  P.J. Besl and N.D. McKay. A method for registration of 3D shapes. *IEEE Trans. Pattern Analysis and Machine Intell.*, 14(2):239–256, 1992.

[4]  G. Blais and M. D. Levine. Registering multiview range data to create 3D computer objects. *IEEE Trans. Pattern Analysis and Machine Intell.*, 17(8):820–824, 1995.

[5]  Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, 1992.

[6]  C. Dorai, J. Weng, and A.K. Jain. Optimal registration of multiple range views. In *12th Int. Conference on Pattern Recognition*, pages A569–571, Jerusalem, Israel, 1994.

[7]  D. W. Eggert, A. Lorusso, and R. B. Fisher. Estimating 3D rigid body transformations: a comparison of four major algorithms. *Machine Vision and Applications*, 9:272–290, 1997.

[8]  A. Hilton, A. J. Stoddart, J. Illingworth, and T. Windeatt. Implicit surface based geometric fusion. *Computer Vision and Image Understanding*, 69(3):273–291, 1998.

[9]  B. K. P. Horn, H. M. Hilden, and S. Negahdaripour. Closed form solution of absolute orientation using orthonormal matrices. *J. of Optical Society of America*, A5:1128–1135, 1988.

[10]  K. Kanatani. Analysis of 3D rotation fitting. *IEEE Trans. Pattern Analysis and Machine Intell.*, 16(5):543–549, 1994.

[11]  X. Pennec. Multiple registration and mean rigid shapes: Application to the 3D case. In *16th Leeds Annual Statistical Workshop*, pages 178–185, Leeds, U.K., 1996.

[12]  A. J. Stoddart and A. Hilton. Registration of multiple point sets. In *13th Int. Conference on Pattern Recognition*, pages B40–44, Vienna, Austria, 1996.