Chapter 4

Approximate Localisation

The Approximate Localisation algorithms are based on an initial posture estimate that is subject to refinement within the vicinity of the initial posture estimate. It may not be stated that a global error minimum is achieved. One may only claim that the posture estimates are updated until a local error minimum is reached. Two different methods were developed for Approximate Localisation: Reference Transform and Error Descent.

The Reference Transform algorithm is based on matching some landmark samples in the laser scan to their counterparts in the simulated scan. When a match pair is found, the point-to-point coordinate difference expresses a fraction of the posture correction equations. The simultaneous application of this principle to several landmark pairs provides an over-determined non-linear system that can be linearised in case the posture errors are small, and solved to yield a posture update. Once the posture update is computed, the overall enhancement is assessed with the Likelihood Test and the procedure is repeated using the revised estimate as a starting value, until the Likelihood Test enhancements are negligible.

The Error Descent algorithm was designed for difficult environments where reliable landmark pairs are scarce or non-existent. It uses a massive computation approach: for every posture candidate the Likelihood Test (LT) is performed. Then, six "neighbour" postures, located at fixed steps from the candidate posture, are also tested with LT. In case any of these has lower LT cost it is considered as a better estimate. The procedure is repeated iteratively, creating a graph (or "tree") of candidate postures where the branches have always lower LT cost than its parent knots. When no further enhancements are possible, a new iteration is initiated using the final posture estimates (the "leaves") as start-up knots and a lower step value to provide a more refined estimate. The algorithm stops when the predefined number of iterations is reached. Then, the estimate with the lower LT cost is chosen as the best posture estimate.

Chapter Organisation

The Reference Transform is introduced in Section One and it is formally developed in Section Two. The Error Descent is developed in Section Three. In Section Four some compared experimental results are presented. The relative merits of the two algorithms are discussed in Section Five.

4.1 Outline of Reference Transform

The principle of operation of Reference Transform is based on identifying a set of points, common to the laser scan and a simulated scan computed as if the robot was located in the initial posture estimate, (x_I, y_I, θ_I). The initial posture estimate is "external" to Reference Transform and it is expressed in world coordinates. It may be supplied by odometry, by Frame Localisation or by the human operator using any other method.

The samples in the laser scan are expressed in a local cartesian reference associated to the robot while the samples in the simulated scan are expressed in a local cartesian reference associated to the initial posture (Figure 1). If the same scene element is identified in both scans with different coordinates (in the local references), a difference between the two references is revealed. Thus, the coordinate difference in the local references is related to the distance between the robot reference (associated to the laser scan) and the simulated scan (associated to the initial estimate).

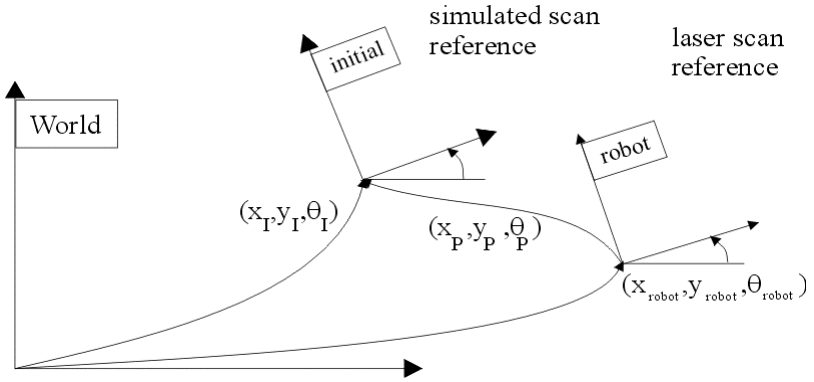


Figure 1 – Introduction to Reference Transform

The scene elements used for Reference Transform are individual range samples. Therefore, each sample is described by two coordinates, (x, y). The analysis of N matching scene elements (with $N \geq 2$) defines an over-determined system to compute the transform between the simulated scan and the laser scan references, (x_P, y_P, θ_P).

Once the reference transform is identified, the initial posture estimate, (x_I, y_I, θ_I), may be corrected by the reference transform defined by (x_P, y_P, θ_P), yielding the updated posture estimate, $(x_{robot}, y_{robot}, \theta_{robot})$.

The Reference Transform algorithm uses the laser scan and the simulated laser scan as defined for the Likelihood Test. The first step of the algorithm is to compute a simulated scan from the map data as if the robot is located at the initial posture, (x_I, y_I, θ_I). The method for creating the simulated scan was described in Chapter 3, Section 2. The current version of the simulated scan for Approximate Localisation does not use reflectance data, either.

4.1.1 Landmark identification

Once the two scans are available, the simulated scan is searched for natural landmarks. Landmarks are characteristic sequences of samples that can be uniquely identified within a given neighbourhood. The preferred landmark on two-dimensional worlds is the corner. It is very characteristic, and can be precisely located. In man made environments, perpendicular corners occur very often and the two lines intersecting at the corner may be regarded as line segments pivoting around a central point (the corner). The inner product operator applied to the two line segments provides a minimum of the its absolute value when the two segments are orthogonal. Then, the corner identification in such topologies, reduces to the validation of an inner product below a given threshold. In case there are no right corners, the corners closest to the right angles can be used, provided the validation threshold is relaxed to accommodate the actual angles in the scene.

The simulated scan is represented as an array of range samples ordered by the sweep angle, measured from a central point, the initial posture, (x_I, y_I, θ_I). With the initial posture, the range and direction data, the simulated scan can be represented as a set of (x, y) pairs by means of a canonical trigonometry transform. A sliding window of $2W+1$ samples is applied to the simulated scan, where the user-defined parameter $W \in N$ is the distance between the central sample and the samples lying at the ends of the window.

Figure 2 shows a sliding window with nine samples, *i.e.*, $W = 4$, applied at sample s_i. The L-shaped profile denotes the distribution of the scan samples in space, while the array on the bottom represents the sliding window interval along the range data array.

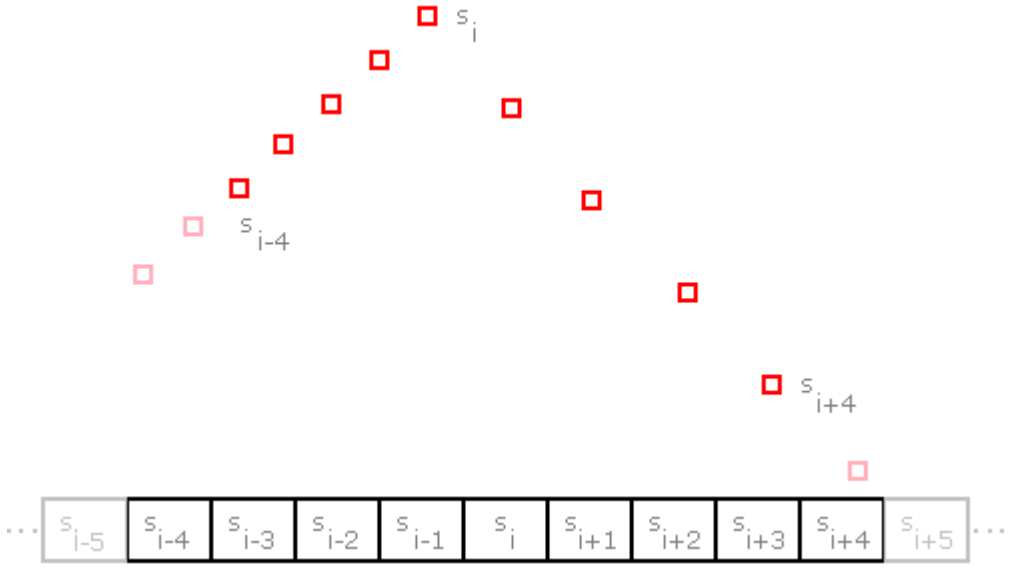


Figure 2 – The elements in the sliding window

In order to classify sample s_i as a natural landmark (a corner in this application),

the neighbouring eight samples are analysed with some edge extraction method, such as the "Edge-based initial segmentation" described in [SequeiraV_1]. For the purpose of the Reference Transform, only the location of the landmark is relevant, the parameters of the segments adjacent to it are ignored. Therefore, a simpler algorithm that measures the inner angles between s_i and the samples before s_i and s_i and the samples after s_i is used.

The centre of the sliding window, s_i, is considered a pivot point of two segments intersecting at s_i and defined from s_{i-1} to s_i and from s_i to s_{i+1}, (4.1), where $(a \to b)$ denotes the oriented line segment defined from a to b. Extending the process to the other samples in the sliding window generates additional two-segment pairs, each one defined by three samples, all intersecting at s_i, (4.2). In general, for each s_i, the sliding window data is grouped in a set of W two-segment pairs, defined by (4.3), with a common point, s_i.

$$(s_i \to s_{i-1}), (s_i \to s_{i+1}) \tag{4.1}$$

$$\begin{cases} (s_i \to s_{i-2}), (s_i \to s_{i+2}) \\ \vdots \\ (s_i \to s_{i-W}), (s_i \to s_{i+W}) \end{cases} \tag{4.2}$$

$$(s_i \to s_{i-k}), (s_i \to s_{i+k}) \quad k = 1,2,...,W \tag{4.3}$$

In Figure 3, a full simulated scan of the classroom (see Chapter 3, Section 4), is presented on the left (the invalid readings were removed; the distance between markers on the grid is 1meter).

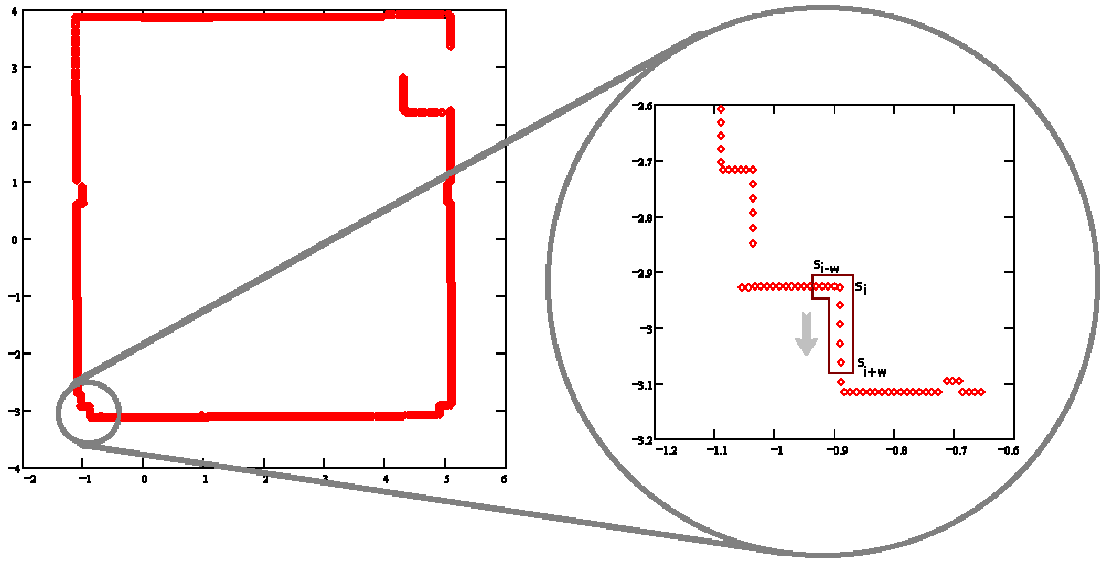


Figure 3 – Using the sliding window along the simulated scan

On the right, the lower left corner is magnified (the distance between markers on the grid is 0.1m) and a sliding window is shown. In the present example, $W = 4$ and the sliding window moves in the counter-clockwise sense.

The next step is the computation of the individual inner products between the two segments $(s_i \to s_{i-k}), (s_i \to s_{i+k})$ for every $k, 1 \le k \le W$, and their summation of the absolute values, for every s_i along the simulated scan. Since the samples are expressed in cartesian coordinates this operation is straightforward. When the summation of the inner product's absolute values for a given s_i is below a given threshold, s_i is marked as a landmark sample. The results of the search for landmark in the enlarged region of Figure 3 are illustrated in Figure 4.

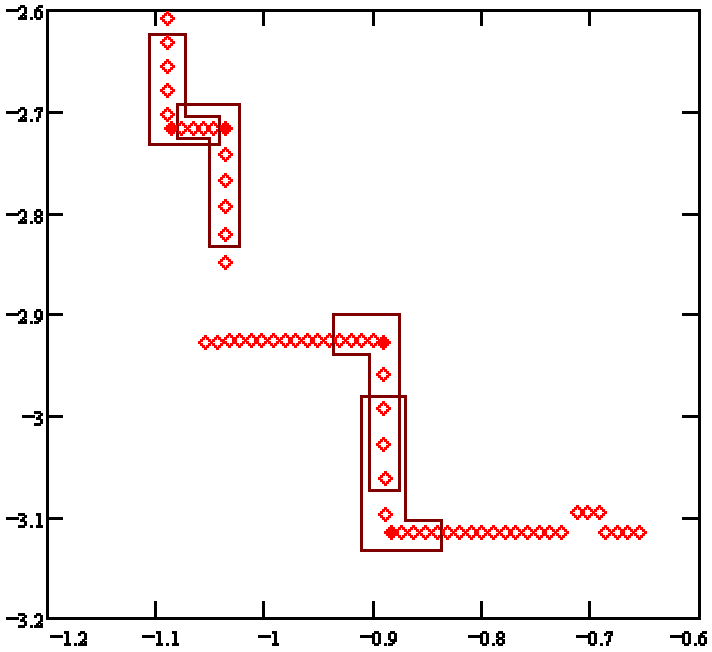


Figure 4 – Corner detection based on the sliding window

In Figure 4 the corners are filled squares (or diamonds) and the sliding window are the outlined polygons. Only the sliding windows associated to the four detected landmarks are represented.

The individual inner products are computed using the simulated scan's cartesian representation. In fact, the relevant value is the normalised inner product, *i.e.*, the inner product divided by the segments norms. This value corresponds to the cosinus of the inner angle between the two segments. In case the arc cosinus is closer to the orthogonal than the threshold, the central sample s_i is considered a corner. In Figure 5, where range data is superimposed to the simulated scan, one corner did not meet this condition; this is denoted by a filled diamond in the simulated scan without its counterpart in the range scan. The landmark selection method is repeated until a sufficient number of landmarks have been identified in the simulated scan.

Assuming the posture error is small, the laser scan and the simulated scan should have similar range profiles and their relative orientation should be approximately aligned. If a landmark in the simulated scan is centred at s_i, its counterpart in the laser scan, l_j, should lie between $i-Z$ and $i+Z$, with $Z \in N$ and $Z > W$, (Figure 5).

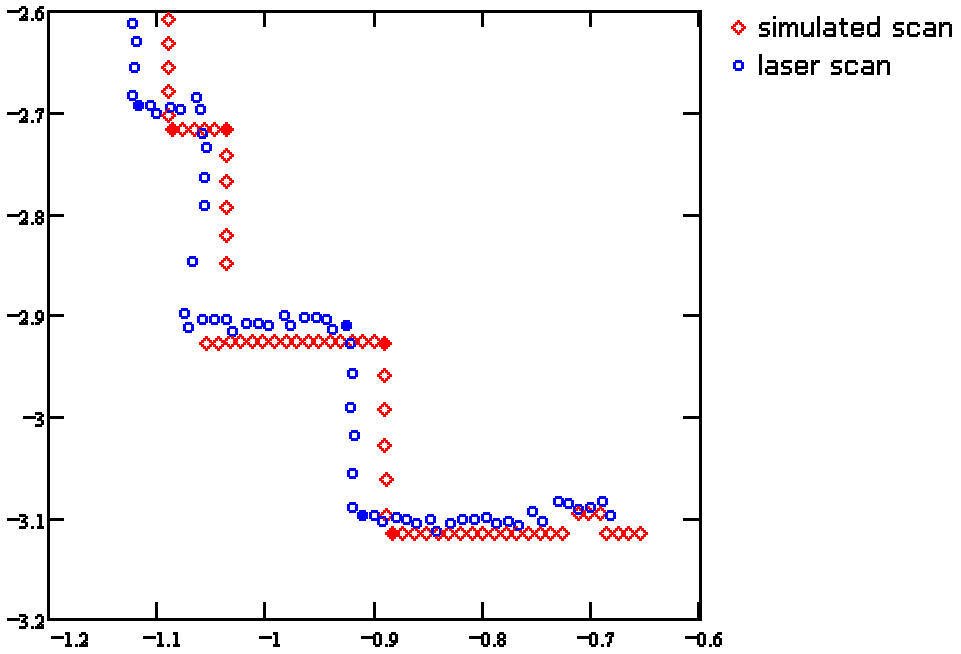


Figure 5 – Searching for landmarks in the $i \pm Z$ neighbourhood

In Figure 5, the same environment details as in Figure 3 are presented and the laser scan was superimposed (with the symbol "o"). If the corners are used as landmarks, the normalised inner product may be used as before. When a correspondence is established, the landmark pair is represented by two (x, y) points defined in cartesian coordinates, (4.4).

$$s_i = \begin{bmatrix} x \\ y \end{bmatrix}^{SimScan} \qquad l_j = \begin{bmatrix} x \\ y \end{bmatrix}^{robot} \qquad i - Z \le j \le i + Z \tag{4.4}$$

Once a sufficient number of landmarks s_i have been identified in the simulated scan and their counterparts in the laser scan, l_j, are also detected, the Reference Transform algorithm proceeds to the next step.

In this implementation, the simulated scan initiates the landmark detection while the laser scan is used to corroborate the candidate landmarks. The process could be symmetric, initiating the landmark detection with the laser scan. The former option was preferred since the laser scan is usually subject to higher noise levels and the corner definition would be more prone to errors.

It is implicit in the method that both profiles are quite similar. Since only the location of the corner is recorded and not the relative orientation of its axis, it can happen that the two corners, located approximately on the same azimuth relative to the robot, may correspond to different features. Such an example is presented in Figure 6, where the match error is due to a fault in the map, which includes an object no longer standing within the laser field of view.

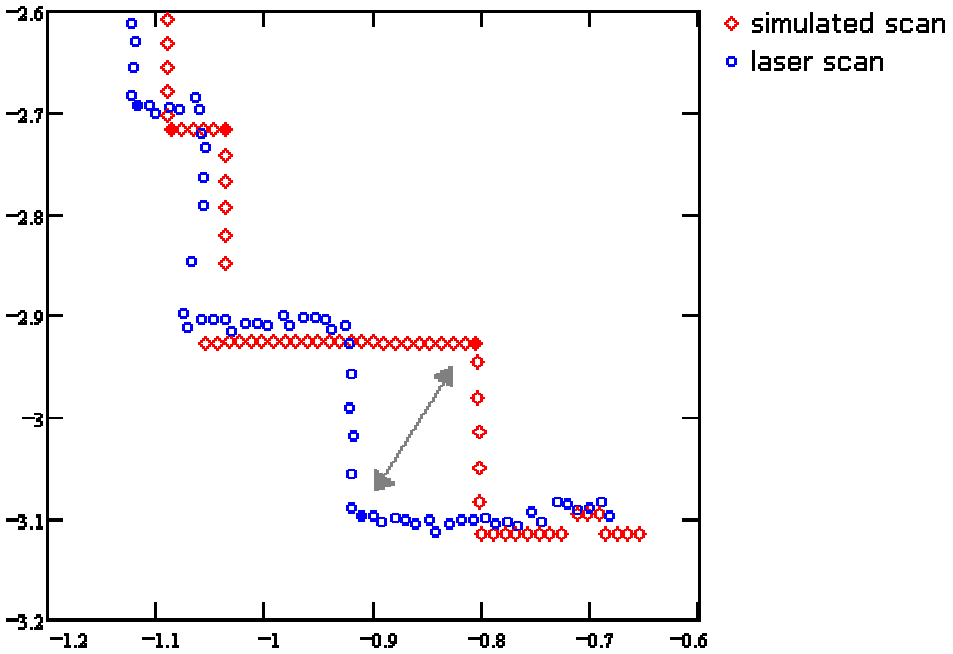


Figure 6 – Incompatible landmarks due to a map that includes an object no longer visible

To minimise this difficulty, it would be necessary to consider more data about the corner (at least its convexity status), increasing the algorithm's complexity and computation burden. Since this occasion is seldom decisive, this variant was dismissed.

4.1.2 Posture update using reference transform

The landmarks in the simulated scan are measured radially relative to the initial posture estimate, whereas the landmarks in the laser scan are measured radially relative to the actual system location. Establishing a reference transform between the simulated scan landmarks and the laser scan landmarks, results in an expression to compute the correction required to match the initial posture estimate and the actual (unknown) system posture.

Each landmark pair, expressed in cartesian coordinates, is an instance of this transform, expressed in (4.5), where (x_P, y_P, θ_P) are the three unknown parameters that represent the corrections.

$$\begin{bmatrix} x \\ y \end{bmatrix}^{robot} = \begin{bmatrix} \cos(\theta_P) & -\sin(\theta_P) \\ \sin(\theta_P) & \cos(\theta_P) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}^{SimScan} + \begin{bmatrix} x_P \\ y_P \end{bmatrix}. \qquad (4.5)$$

Since each landmark pair yields two equations, the minimum number of landmarks to solve (4.5) is two, in order to get four equations on three unknowns. However, adding a few more landmark pairs contributes to a better solution, especially if these landmarks lie on opposite sides relative to the posture in the centre of the scan, (Figure 7).

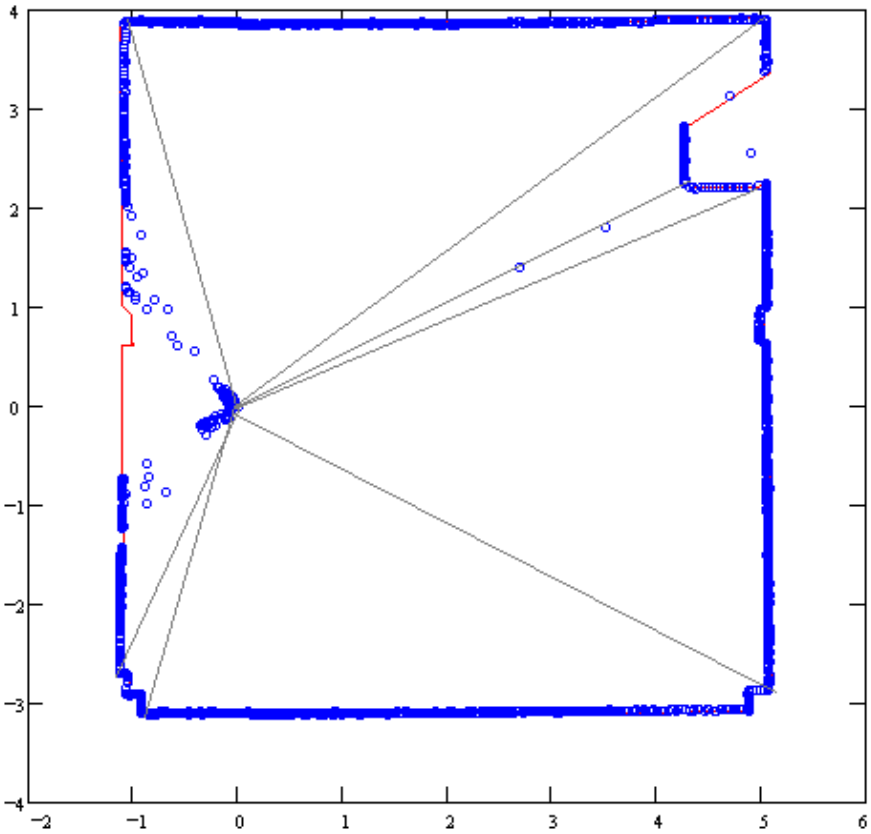


Figure 7 – Sample landmark pair distribution

The line segments in Figure 7 point from the centre of the scans to the landmarks. In case a large number of landmark pairs are available, there will be clusters of landmarks located in approximately the same direction. In case the reflectance data varies smoothly along the close landmarks, the range measurements will be similar and the posture correction induced by the landmarks in the cluster will be approximately the same for all of them. Thus, one landmark per landmark cluster suffices to the system of equations; the extra equations are nearly dependent on the selected one, and their effect reduces to a computation effort increase.

If the set of NL landmark pairs and their associated transforms (4.5) are considered simultaneously, and the terms are rearranged in order to the unknowns, (x_P, y_P, θ_P), a

non-linear over-determined system is defined, (4.6).

$$\begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \\ x_{NL} \\ y_{NL} \end{bmatrix}^{robot} = \begin{bmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ x_2 & -y_2 & 1 & 0 \\ y_2 & x_2 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_{NL} & -y_{NL} & 1 & 0 \\ y_{NL} & x_{NL} & 0 & 1 \end{bmatrix}^{SimScan} \cdot \begin{bmatrix} \cos(\theta_P) \\ \sin(\theta_P) \\ x_P \\ y_P \end{bmatrix} \tag{4.6}$$

The main postulate in Reference Transform is that the correction errors are small. This assumption is critical to the next step, where the system (4.6) is linearised around $\theta_P \approx 0$, (4.7). If the angular correction (θ_P) is lower than 0.1rad (6º), the linearisation error is less than 0.5%. Operating within this range, the linearised system, with the unknowns sorted in the usual sequence, x_P, y_P, θ_P, is defined in (4.8).

$$if\ \theta_P \approx 0 \Rightarrow \begin{cases} \cos(\theta_P) \approx 1 \\ \\ \sin(\theta_P) \approx \theta_P \end{cases} \tag{4.7}$$

$$\begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \\ x_{NL} \\ y_{NL} \end{bmatrix}^{robot} - \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \\ x_{NL} \\ y_{NL} \end{bmatrix}^{SimScan} = \begin{bmatrix} 1 & 0 & -y_1 \\ 0 & 1 & x_1 \\ 1 & 0 & -y_2 \\ 0 & 1 & x_2 \\ \vdots & \vdots & \vdots \\ 1 & 0 & -y_{NL} \\ 0 & 1 & x_{NL} \end{bmatrix}^{SimScan} \cdot \begin{bmatrix} x_P \\ y_P \\ \theta_P \end{bmatrix} \tag{4.8}$$

The over-determined linear system is solved by standard algebraic methods, detailed in Section Two. The result is the posture correction, (x_P, y_P, θ_P), representing the reference transform that translates the initial posture estimate (x_I, y_I, θ_I) into the actual posture. Thus, the actual robot posture in world coordinates may be estimated with a canonical reference transform, (4.9) and (4.10).

$$\theta_{robot} = \theta_I + \theta_P \tag{4.9}$$

$$\begin{bmatrix} x_{robot} \\ y_{robot} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta_P) & -\sin(\theta_P) & x_P \\ \sin(\theta_P) & \cos(\theta_P) & y_P \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_I \\ y_I \\ 1 \end{bmatrix} \tag{4.10}$$

The newly computed posture, $(x_{robot}, y_{robot}, \theta_{robot})$, must then be validated with the Likelihood Test (Chapter 3, Section 3). In case there is a point-to-point distance enhancement, denoted by the Number of Valid Pairs (NVP) and the zero-th to second order moments (Match Pairs, Expected Value and Dispersion), the posture $(x_{robot}, y_{robot}, \theta_{robot})$ is published as the updated posture estimate. Otherwise, the Reference Transform algorithm ends with failure.

4.1.3 Coping with Reference Transform constraints

The Reference Transform algorithm requires the following constraints to be met:

1. The sample located closest to the corner is considered to be the corner. An alternative would be to use line extraction and define the corner at the intersection of the two extracted line segments. This approach was not followed due to the problems encountered with the extraction of line segments based on few samples.
2. The laser scan resolution should be high to obtain detailed information of distant landmarks. With a laser scanner performing 2000 samples per revolution, the distance between adjacent samples at 10 meter is 0.03m. Thus, the target surface should contain no features smaller than 0.1m to 0.2m in order to obtain sufficient samples to extract the profile.
3. The point-to-point distances between the simulated scan and the laser scan near the landmarks must be small in order to assure that a proper matching can be found within a limited scope of search. Using a wide range scope would require complex extraction and matching algorithms to ensure the correspondence between simulated scan and laser scan features.
4. The angular correction should be small to allow the linearisation. The initial trials have shown that the maximum angular corrections possible with Reference Transform are below 0.1rad (6^o), within the valid approximation boundaries. Typical experimental results presented in Chapter 3, Section 5, show angular corrections, θ_P, below 0.01rad (0.6^o). Thus, the angular constraint is set by the landmark matching and not by the linearisation hypothesis.

Depending on the map data source, the laser scan and the simulated scan may exhibit different error patterns, which also constrains the landmark extraction. After each iteration, the Likelihood Test is run to assess the newly computed posture.

Given all the constraints mentioned before, the Reference Transform algorithm was implemented in closed loop, with the results of one iteration being used as the initial posture estimate of the next iteration. The algorithm ends when the pair-to-pair distance

enhancements are negligible or negative.

Apart from a limited risk of error in the initial iterations of Reference Transform, due to false identification of landmarks, laser error measurements or map faults, the algorithm converges to the best estimate within two to five iterations. The algorithm error is detected by increasing pair-to-pair distances in the Likelihood Test.

For difficult experimental conditions (limited and/or outdated maps, surfaces with poor reflectance, laser scans with insufficient resolution as described in Chapter 3, Section 5 in the "factory") a blind search algorithm was developed, the Error Descent algorithm, described in Section 3.

4.2 Formal development of Reference Transform

The data required to perform Reference Transform is an initial posture estimate, $p_I = (x_I, y_I, \theta_I)$, a laser range scan and a description of the scene obtained from off-line maps, 3D reconstructed maps or previous laser scans.

The simulated scan data, defined in Chapter 3, Section 2, is expressed as an array of N range samples associated to the initial posture, p_I, uniformly distributed in the angular interval between bearings $b_{\min}$ and $b_{\max}$, $S = \{s_i, i = 1,2,...,N \mid p_I\}$. The angular boundaries are defined in the laser scan parameters.

The samples in the simulated scan, *S*, are expressed in cartesian coordinates, relative to the initial posture estimate, p_I, (4.11).

$$\begin{bmatrix} x \\ y \end{bmatrix}^{SimScan} = \begin{bmatrix} s_i \cdot \cos(b_{\min} + \Delta \cdot i) \\ s_i \cdot \sin(b_{\min} + \Delta \cdot i) \end{bmatrix} \qquad (4.11)$$

$$where\ i = 1,...,N \quad and \quad \Delta = \frac{b_{\max} - b_{\min}}{N}$$

The proposed landmark selection method assumes that corners are used as landmarks. The identification of the natural landmarks is performed with a sliding window of size $2W+1$, centred on sample s_i, $SW(i)$, (4.12). The sliding window sweeps the array from $i = W+1$ to $i = N-W$. The size of the window, *W*, is defined off-line by the user. The normalised inner product, $NIP(i,k)$, measures the degree of co-linearity of the two oriented line segments defined by from $(i \to i-k)$ and $(i \to i+k)$, (4.13), with $k = 1,...,W$. This test is bounded to the elements within the sliding window. For every sample within the sliding window, the individual normalised inner products, $NIP(i,k)$, are computed, (4.14).

$$SW(i)=\left\{\begin{bmatrix}x_k\\y_k\end{bmatrix}^{SimScan},\ k=i-W,i-W+1,...,i,...,i+W-1,i+W\right\} \tag{4.12}$$

$$\left(\begin{bmatrix}x_i\\y_i\end{bmatrix}^{SimScan}\rightarrow\begin{bmatrix}x_{i-k}\\y_{i-k}\end{bmatrix}^{SimScan}\right),\left(\begin{bmatrix}x_i\\y_i\end{bmatrix}^{SimScan}\rightarrow\begin{bmatrix}x_{i+k}\\y_{i+k}\end{bmatrix}^{SimScan}\right) \tag{4.13}$$

$$NIP(i,k)=\frac{(x_{i-k}-x_i)\cdot(x_{i+k}-x_i)+(y_{i-k}-y_i)\cdot(y_{i+k}-y_i)}{\sqrt{(x_{i-k}-x_i)^2+(y_{i-k}-y_i)^2}\sqrt{(x_{i+k}-x_i)^2+(y_{i+k}-y_i)^2}} \tag{4.14}$$

The normalised inner products (NIP) correspond to the cosinus of the inner angle of the oriented line segments. The absolute values of $NIP(i,k)$ are added from $k=1$ to $k=W$. If the average normalised absolute inner product value is lower than a given threshold, (4.15), the sample $\begin{bmatrix}x_i & y_i\end{bmatrix}^T$ is a candidate landmark.

$$\frac{\sum_{k=1}^{W}\left|NIP(i,k)\right|}{W}<InnerThr \tag{4.15}$$

When the simulated scan array is completely swept, the set of candidate landmarks is analysed. In case there are two or more landmarks within the distance of the sliding window, the landmark with lower average NIP is kept and the others are discarded. The landmark set with *NL* elements is stored, keeping the original order and the radial index associated to the landmark, (4.16).

$$Landmarks_{SimScan}=\left\{l_m=\left\{\begin{bmatrix}x_i\\y_i\end{bmatrix}^{SimScan},i\right\},\ \begin{matrix}m=1,2,...,NL_{SimScan}\\ i\in\{1,2,...,N\}\end{matrix}\right\} \tag{4.16}$$

The next step of the algorithm is to search for similar landmarks in the laser scan. The laser scan is a set of laser samples with range and reflectance data (although the reflectance is not used), defined from $b_{\min}$ and $b_{\max}$. The first step is the expression of the laser scan in cartesian coordinates, similar to (4.11). The same angular parameters must be used, to assure accuracy and correspondence between the simulated scan and the laser scan:

$$\begin{bmatrix}x\\y\end{bmatrix}^{robot}=\begin{bmatrix}rng_j\cdot\cos(b_{\min}+\Delta\cdot j)\\rng_j\cdot\sin(b_{\min}+\Delta\cdot j)\end{bmatrix} \tag{4.17}$$

$$where\ i=1,...,N\quad and\quad \Delta=\frac{b_{\max}-b_{\min}}{N}$$

Once the two scans are expressed in the same form, the laser counterparts of the simulated scan landmarks can be searched within the vicinity of the landmark indexes stored for each $l_m^{SimScan}$, (4.16). The method of the normalised inner product is used again, restricted to the vicinity of sample s_i, associated to $l_m^{SimScan}$. The search window comprehends $2Z+1$ samples. For each simulated scan landmark, $l_m^{SimScan}$, the associated index, *i*, defines the index of its laser counterpart, l_n^{robot}, (4.18), located within Z samples of *i*. If no landmark can be found in the vicinity of *i*, the candidate landmark $l_m^{SimScan}$ is discarded.

$$l_n = \left\{ \begin{bmatrix} x_j \\ y_j \end{bmatrix}^{robot}, j : Z - i \le j \le Z + i \right\} \tag{4.18}$$

$$Landmarks_{robot} = \left\{ l_n = \left\{ \begin{bmatrix} x_j \\ y_j \end{bmatrix}^{robot}, j \right\}, \quad \begin{matrix} n = 1,2,...,NL_{robot} \\ j \in \{1,2,...,N\} \end{matrix} \right\} \tag{4.19}$$

If more than one laser landmark, l_n^{robot}, is defined within the sliding window size, *W*, the landmark with the lower average NIP is selected and the remaining ones are discarded. Finally, the set of natural landmarks (4.19) associated to the robot is restricted to the pairs where a simulated scan landmark (4.18) was detected.

Beyond this point, the number of landmarks, NL, is reduced to the cases where the instances in the set of $NL_{SimScan}$ landmarks associated to the Simulated Scan have their corresponding pairs in the set of NL_{robot} landmarks associated to the laser scan.

In case the number of landmark pairs exceeds 10, a set of five to nine landmark pairs is selected from the whole set, following a maximum distance algorithm described in Appendix E. Using a reduced number of landmark pairs reduces the computation effort and minimises the linear dependence of the equations associated to landmark pairs located at short distances from each other.

The set of *NL* landmarks pairs is expressed in the form of (4.8), where the linearisation hypothesis ($\theta_P \approx 0$) is assumed. The over-determined system of $2NL$ equations on three unknowns is expressed as (4.20).

$$\begin{bmatrix} x_1^{robot} - x_1^{SimScan} \\ y_1^{robot} - y_1^{SimScan} \\ x_2^{robot} - x_2^{SimScan} \\ y_2^{robot} - y_2^{SimScan} \\ \vdots \\ x_{NL}^{robot} - x_{NL}^{SimScan} \\ y_{NL}^{robot} - y_{NL}^{SimScan} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -y_1^{SimScan} \\ 0 & 1 & x_1^{SimScan} \\ 1 & 0 & -y_2^{SimScan} \\ 0 & 1 & x_2^{SimScan} \\ \vdots & \vdots & \vdots \\ 1 & 0 & -y_{NL}^{SimScan} \\ 0 & 1 & x_{NL}^{SimScan} \end{bmatrix} \cdot \begin{bmatrix} x_P \\ y_P \\ \theta_P \end{bmatrix} \tag{4.20}$$

This system is of the general form of (4.21), where M is defined as (4.22) and Q is defined as (4.23). The matrix dimension is indicated in subscript.

$$M \cdot \begin{bmatrix} x_P \\ y_P \\ \theta_P \end{bmatrix} = Q \tag{4.21}$$

$$M = \begin{bmatrix} 1 & 0 & -y_1^{SimScan} \\ 0 & 1 & x_1^{SimScan} \\ 1 & 0 & -y_2^{SimScan} \\ 0 & 1 & x_2^{SimScan} \\ \vdots & \vdots & \vdots \\ 1 & 0 & -y_{NL}^{SimScan} \\ 0 & 1 & x_{NL}^{SimScan} \end{bmatrix}_{(2NL\times 3)} \tag{4.22}$$

$$Q = \begin{bmatrix} x_1^{robot} - x_1^{SimScan} \\ y_1^{robot} - y_1^{SimScan} \\ x_2^{robot} - x_2^{SimScan} \\ y_2^{robot} - y_2^{SimScan} \\ \vdots \\ x_{NL}^{robot} - x_{NL}^{SimScan} \\ y_{NL}^{robot} - y_{NL}^{SimScan} \end{bmatrix}_{(2NL\times 1)} \tag{4.23}$$

Multiplying (4.21) on the left by M^T, the over determined system can be solved in the form of (4.24), which is a regular (3x3) linear system.

$$M^T_{(3\times 2NL)} \cdot M_{(2NL\times 3)} \cdot \begin{bmatrix} x_P \\ y_P \\ \theta_P \end{bmatrix} = M^T_{(3\times 2NL)} \cdot Q_{(2NL\times 1)} \tag{4.24}$$

The system coefficients in (4.24) are defined in (4.25) and (4.26).

$$M^T M = \begin{bmatrix} NL & 0 & -\sum_{i=1}^{NL} y_i^{SimScan} \\ 0 & NL & \sum_{i=1}^{NL} x_i^{SimScan} \\ -\sum_{i=1}^{NL} y_i^{SimScan} & \sum_{i=1}^{NL} x_i^{SimScan} & \sum_{i=1}^{NL} \left[\left(x_i^{SimScan} \right)^2 + \left(y_i^{SimScan} \right)^2 \right] \end{bmatrix}_{(3\times 3)} \tag{4.25}$$

$$M^T Q = \begin{bmatrix} \sum_{i=1}^{NL} (x_i^{robot} - x_i^{SimScan}) \\ \sum_{i=1}^{NL} (y_i^{robot} - y_i^{SimScan}) \\ \sum_{i=1}^{NL} \left[x_i^{SimScan} \cdot (y_i^{robot} - y_i^{SimScan}) - y_i^{SimScan} \cdot (x_i^{robot} - x_i^{SimScan}) \right] \end{bmatrix}_{(3\times 1)} \tag{4.26}$$

The method used to solve (4.24) is the standard Gauss elimination. The result, $\begin{bmatrix} x_P & y_P & \theta_P \end{bmatrix}^T$, represents the transform parameters relating the actual robot posture associated to the laser scan to the initial posture estimate (x_I, y_I, θ_I) associated to the simulated scan. The posture update is performed according to equations (4.9) and (4.10).

The updated posture $(x_{robot}, y_{robot}, \theta_{robot})$ is submitted to the Likelihood Test. If the number of valid pairs (NVP) and/or match pairs (MP) increases, relative to the initial posture, (x_I, y_I, θ_I), and the expected value (EV) does not increase, (4.27), the updated posture is validated. In addition, if the NVP and MP do not decrease significantly while the EV and/or the dispersion (Disp) decrease, the newly computed posture is validated, (4.28), where δ and ε are two small positive integers. Otherwise, the update is discarded.

$$\left[(NVP_{robot} > NVP_I) \vee (MP_{robot} > MP_I) \right] \wedge (EV_{robot} \leq EV_I) \tag{4.27}$$

$$\begin{aligned} &\left[(EV_{robot} \leq EV_I) \vee (Disp_{robot} < Disp_I) \right] \wedge \\ &\qquad \wedge (NVP_{robot} - NVP_I > \delta) \wedge (MP_{robot} - MP_I > \varepsilon) \end{aligned} \tag{4.28}$$

While the update is valid, the algorithm is repeated, using the updated posture as the new initial estimate. When the update fails, the most recent initial posture is again submitted to the Likelihood Test with the final validation thresholds. The result of the algorithm is defined by its results, as described in Chapter 3.

4.3 Error Descent

4.3.1 Introduction

During the project development, the need for extending Localisation to more difficult operating conditions arose. First, the 3D Reconstruction map was replaced by a poorer off-line map; then the possibility of using previous laser scans to supply map data was considered (see Chapter 3, Section 2). In parallel, trials were conducted in very difficult environments such as the "factory" (Chapter 3, Section 4). Both lines of work led

to the conclusion that a new algorithm, with major emphasis on robustness, was required to tackle the new challenges.

The Frame Localisation and the Reference Transform algorithms, described previously, condense, in an early stage, all the laser and map data into short amounts of information that are submitted to further processing. The Likelihood Test is an extensive distance measure, designed to encompass all the data samples and validate the candidate postures.

Using an opposite approach, a third algorithm was created to use the available data extensively, at a "raw" stage, without building models of the data. Based on the Likelihood Test, it tracks the evolution of the point-to-point distance parameters, starting at an initial posture and sliding along a path of diminishing cost (see Chapter 3, Section 3 and (4.29)). Therefore, it is termed as Error Descent.

This algorithm presents some advantages and a few disadvantages too:

1. It requires no landmarks or particular characteristics on the laser or map data. All it needs is a form of simulated scan, based on a 3D Reconstruction map, an off-line map or previous laser data and a laser scan of the actual system posture.
2. It is quite robust, even if the initial posture estimate is distant of the best possible estimate. In the field trials, corrections of 0.4m in position (x, y) and/or 0.15 rad (9º) in orientation (θ), were commonly achieved, a figure far beyond the capabilities of Reference Transform.
3. It is very intuitive: following a diminishing cost function is easily understood and implemented. It is a classical problem in control and robotics, with many algorithms and strategies available.
4. It takes much longer to process the data than the other algorithms. When this kind of approach was first considered, it was abandoned due to excessive computation time (estimated to vary between 30 to 120 seconds with an Intel Pentium-class computer, or comparable, to process range scans with 2000 samples). With current hardware, the computation time was cut by five to ten and it varies now from 2 to 10 seconds, depending on the parameters and resolution required.
5. High quality results, especially in difficult environments, depend on the Likelihood Test parameters, primarily the maximum match distance threshold, T_{Map}, and the last slot equivalent distance, $\Delta_{\max}$ (see Chapter 3, Section 3).

4.3.2 Description of Error Descent

The Likelihood Test values used for Error Descent are the number of valid pairs (NVP), match pairs (MP) and expected value (EV). These three values are involved in the cost function introduced in Chapter 3, Section 3, and repeated below:

$$\cos t(p_n) = ExpectValue(p_n) + \alpha \cdot \frac{ValidPairs(p_n) - MatchPairs(p_n)}{ValidPairs(p_n)} \qquad (4.29)$$

In (4.29), p_n is any candidate posture and α is a parameter, with the dimension of distance, which weighs the cost of non-match pairs.

The Error Descent algorithm adopts a graph search approach. It starts with a single posture, which is the root of the graph (or tree), (x_1, y_1, θ_1). Then, the candidate postures are computed iteratively, following a path of diminishing cost along a regular grid. The grid is defined in three independent dimensions, X, Y, θ, and the steps between adjacent samples, $\delta_X, \delta_Y = \delta_X$ and δ_θ are defined off-line by the human operator. In addition to its cartesian coordinates, each posture is represented by three integer coordinates relative to the root. To translate these into world coordinates the step parameters are used.

To detect a path along a diminishing cost function, a set of neighbours must be considered around each posture, p_i, (the dark central cube in Figure 8).

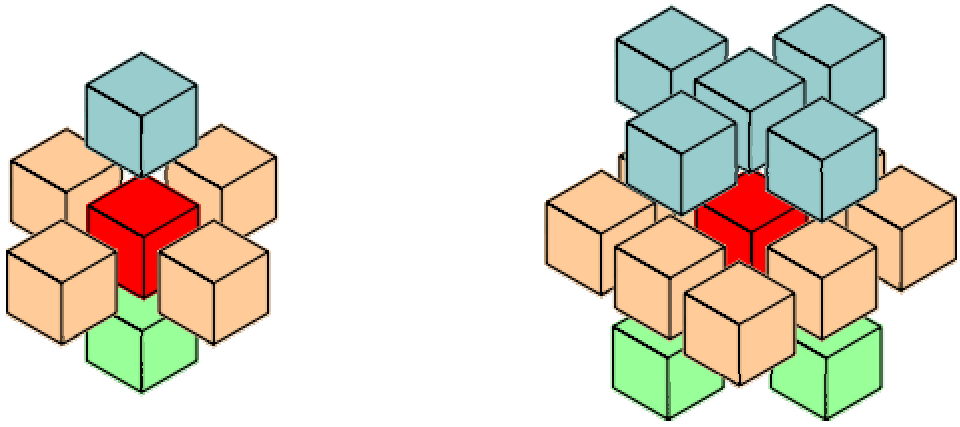


Figure 8 – Two type of neighbourhood in three dimensions:
6 neighbours and 18 neighbours

If the posture p_i is regarded as the centre of a 3x3x3 cube, the neighbourhood could consist of six postures, in case only one axis varies between p_i and its immediate neighbours, or eighteen neighbours, in case two axes vary between p_i and its neighbours, or twenty six neighbours, in case the three axes vary between p_i and its neighbours.

In all cases, every point in the grid can be reached in successive steps, whilst the number of required steps to move between any two given points increases as the number of neighbours decreases. In most experiments, the cost (4.29) varies smoothly as the path progresses towards a given direction, except when the steps are very short and the variations are small. The six-neighbourhood variant, (4.30), was chosen in order to minimise the number of candidates aiming at directions where no cost reduction is expected. This approach reduces the number of posture submitted to the Likelihood Test, which is the

most time consuming operation, while augmenting the number of steps in the path.

In order to operate within the error descent grid the postures' metric representation is replaced by an equivalent representation in integer coordinates, $q_k = (n_{Xk}, n_{Yk}, n_{\theta k})$, relative to the origin (p_I), where n_{Xk}, n_{Yk} and $n_{\theta k}$ are the number of steps progressed in each direction along the grid. The six neighbours are defined in (4.31) and the conversion between the two coordinate systems is defined in (4.32).

7

$$\begin{aligned} p_{i,1} &= (x_i + \delta_X, y_i, \theta_i) & p_{i,2} &= (x_i, y_i + \delta_Y, \theta_i) & p_{i,3} &= (x_i, y_i, \theta_i + \delta_\theta) \\ p_{i,4} &= (x_i - \delta_X, y_i, \theta_i) & p_{i,5} &= (x_i, y_i - \delta_Y, \theta_i) & p_{i,6} &= (x_i, y_i, \theta_i - \delta_\theta) \end{aligned} \tag{4.30}$$

$$\begin{aligned} q_{i,1} &= (n_{Xi} + 1, n_{Yi}, n_{\theta i}) & q_{i,2} &= (n_{Xi}, n_{Yi} + 1, n_{\theta i}) & q_{i,3} &= (n_{Xi}, n_{Yi}, n_{\theta i} + 1) \\ q_{i,4} &= (n_{Xi} - 1, n_{Yi}, n_{\theta i}) & q_{i,5} &= (n_{Xi}, n_{Yi} - 1, n_{\theta i}) & q_{i,6} &= (n_{Xi}, n_{Yi}, n_{\theta i} - 1) \end{aligned} \tag{4.31}$$

$$p_{i,k} = (x_i, y_i, \theta_i) = (x_I + n_{Xi} \cdot \delta_X, y_I + n_{Yi} \cdot \delta_X, \theta_I + n_{\theta i} \cdot \delta_\theta)$$

$$q_{i,k} = \left(n_{Xi} = \frac{x_i - x_I}{\delta_X}, n_{Yi} = \frac{y_i - y_I}{\delta_X}, n_{\theta i} = \frac{\theta_i - \theta_I}{\delta_\theta} \right) \tag{4.32}$$

The diminishing cost path follows every neighbour with lower cost than its root. When a posture has lower cost than all its neighbours, the search in that sub-path terminates and the posture is marked with a flag. The output of the algorithm is the sequence of steps (the paths) from the initial set of posture estimates to the best possible estimates, which are the postures with lower costs than its neighbours.

The Error Descent procedure is straightforward. On each iteration, repeat the following steps with all the elements on the set of candidate postures:

1. Compute the cost, (4.29), associated to the candidate posture, $p_k = (x_k, y_k, \theta_k)$, represented in integer coordinates as $q_k = (n_{Xk}, n_{Yk}, n_{\theta k})$;
2. Establish a set of six neighbours around q_k;
3. Verify if any of the six neighbours was previously considered and thus, already present in the set of candidate postures. In such cases, the repeated neighbour is discarded;
4. Convert q_k's to p_k's and compute the costs, (4.29), associated to the remaining neighbours;
5. Add every candidate with a cost lower than the initial one, $\cos t(p_k)$, to the candidate posture set and mark it as a new candidate posture;

6. If no neighbour has a cost lower than $\cos t(p_k)$, mark the corresponding q_k as a leaf (a termination of the graph or tree);
7. Proceed to the next posture in the set of candidate postures.

When the set of candidate postures has been fully browsed, resume the operation with the newly computed candidates. The algorithm terminates when all the newer candidates have lower weights than their respective neighbours do.

The outcome of Error Descent is a graph (tree) starting from the initial posture, $q_I = (0,0,0)$, growing along multiple directions (branches) until the terminations (leaves), (Figure 9).

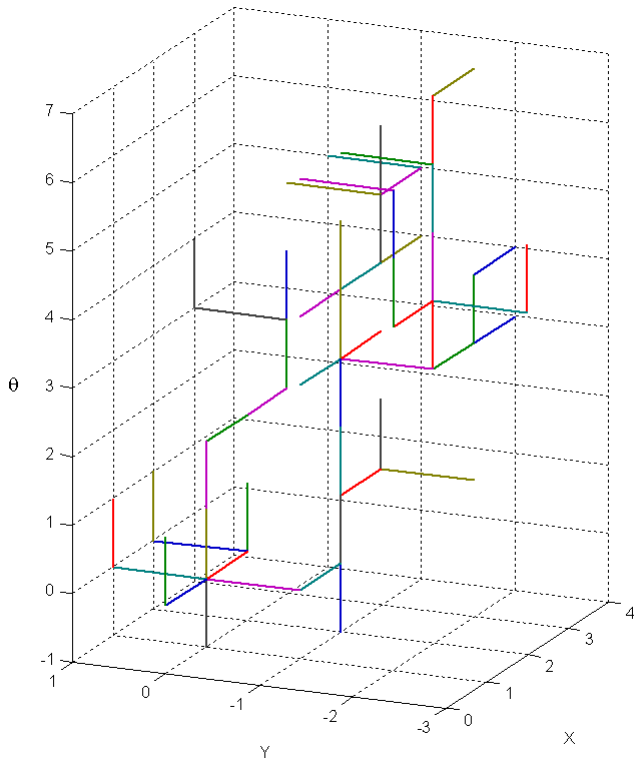


Figure 9 – A sample Error Descent graph (tree)

In Figure 9, the paths along the grid are represented. Only the candidates along the diminishing path until the leaves are shown. The three axes represent the three directions and the grid coordinates represent the number of steps in each direction. The algorithm starts at $p_I = (x_I, y_I, \theta_I) \Leftrightarrow q_I = (0,0,0)$. In this experiment, all the neighbours around q_I have lower cost, thus six paths leave from $(0,0,0)$. Two iterations later, only two paths are still promising:

$$(0,0,0) \rightarrow (0,0,1) \rightarrow (0,0,2) \rightarrow (1,0,2) \rightarrow (2,0,2) \rightarrow (2,0,3) \rightarrow \ldots$$

$$(0,0,0) \rightarrow (0,1,0) \rightarrow (1,1,0) \rightarrow (1,1,1) \rightarrow (1,1,2) \rightarrow (1,1,3) \rightarrow \ldots$$

Notice that the paths do not intersect each other, since each neighbour is compared to all candidates in the set before being analysed. Notice also that, in this case, the general direction is similar for both paths. The two paths are scanning the whole grid in the $+\theta$ sense progressing almost parallel to each other. The best result is the one that achieves the leaf with the lower cost, (4.29).

The main pitfall of Error Descent is local minima. A short displacement in the coordinates might lead to the inclusion or exclusion of some samples in the last slots of the point-to-point distribution, inducing significant changes in the Likelihood Test parameters. A second cause of point-to-point distance changes, albeit less important, is the discrete nature of the histogram: when moving the candidate posture, some samples may fall in another slot, changing the distribution by discrete amounts. Using only six neighbours reinforces the effects of these two issues: it is possible that the cost diminishes between q_k and some of its twenty-six neighbours, and yet it increases when the steps in each direction are considered separately.

A second difficulty is the determination of an optimal step. If the step is too small, the algorithm progresses very slowly. If the step is too large there is a poor resolution between adjacent postures, and the algorithm might stop at a given posture after it had "jumped" over a better result.

To overcome the two problems a multiple resolution variant was implemented. In the first run, the steps between a candidate and its neighbours are large (e.g., 0.05m and 0.005rad). This sows some candidates "far" away of the initial posture, jumping over possible local minima and accelerating the exploration of the grid. The second run discards the intermediate branches leading to the leaves and starts from the set of posture candidates (leaves) computed on the first run but the steps are divided by two. In order to keep the integer coordinates coherent, the values of the leaves q_i are multiplied by two before the start of the second run.

The third and following runs repeat the procedure until very small steps are considered (e.g., 0.001m and 0.001rad). The initial steps and the number of runs may be defined by the user to match the experiment conditions, the hardware computation speed and the mission requirements.

When the path search is complete, the candidate posture (leaf) with the lower cost, $q_{\min} = (n_{Xk=\min}, n_{Yk=\min}, n_{\theta k=\min}) : \cos t(q_{\min}) < \cos t(q_k)\ \forall k, k \neq \min$, is selected as the best posture estimate. The grid coordinates are converted into the world reference metrics, (4.33).

$$p_{\min} = (x_I + n_{Xk=\min} \cdot \delta_X, y_I + n_{Yk=\min} \cdot \delta_X, \theta_I + n_{\theta k=\min} \cdot \delta_\theta) \qquad (4.33)$$

If the cost is lower than the accuracy requirements, the best posture is published. Otherwise, the algorithm ends with failure.

4.4 Experimental Results

This section presents the comparative results of the two Approximate Localisation algorithms, based on a common set of experimental data. The scenarios chosen are the same as in Chapter 3, Section 4: the classroom and the office. The "factory" is not presented since the Reference Transform algorithm fails in that environment due to the poor map, and reduced number of reliable natural landmarks within the range of operation of the laser scanner. The Error Descent results, with a limited degree of success were presented before (see Chapter 3, Section 4).

The results of Reference Transform are presented first, followed by the results computed with Error Descent. Finally, the point-to-point distance distribution is compared for the two Approximate Localisation algorithms.

4.4.1 The classroom

In Figure 10 the laser scan of the classroom is presented with circles, superimposed on the simulated scan defined from the initial posture estimate, represented by the thin continuous line. The invalid range readings were removed.

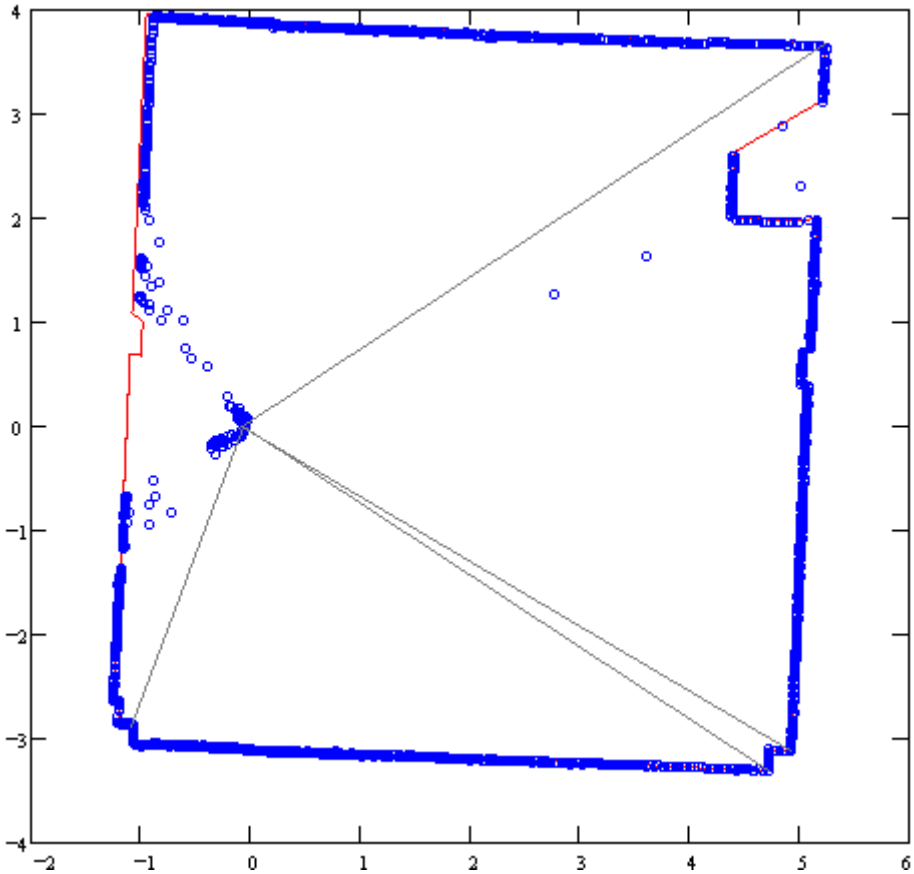
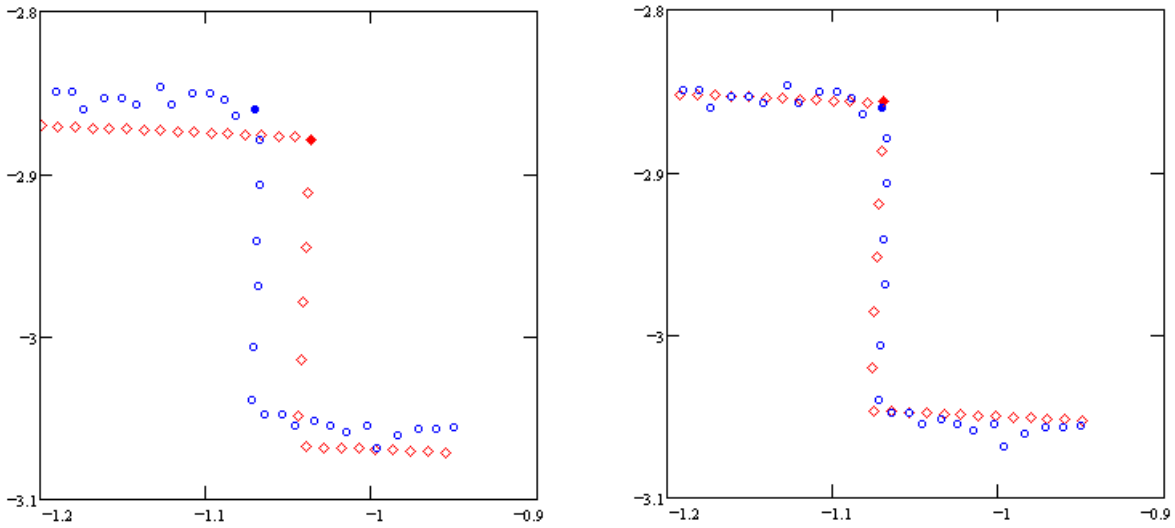


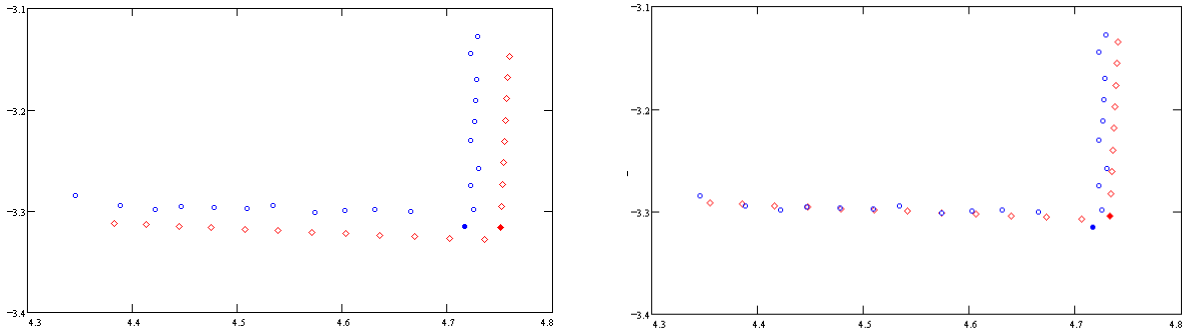
Figure 10 – Detected landmarks for Reference Transform

The Reference Transform algorithm was run with a sliding window of size $W = 4$. The maximum shift between the simulated scan and the laser scan is $Z = 8$. The average inner threshold is $InnerThr < 0.05$, corresponding to an angle of $\pm \pi/2 \pm 0.05$ ($\pm 90° \pm 3°$). Under these conditions, four corresponding landmark pairs were found in the simulated scan and the laser scan, (Figure 10). The initial posture estimate was computed by Frame Localisation.

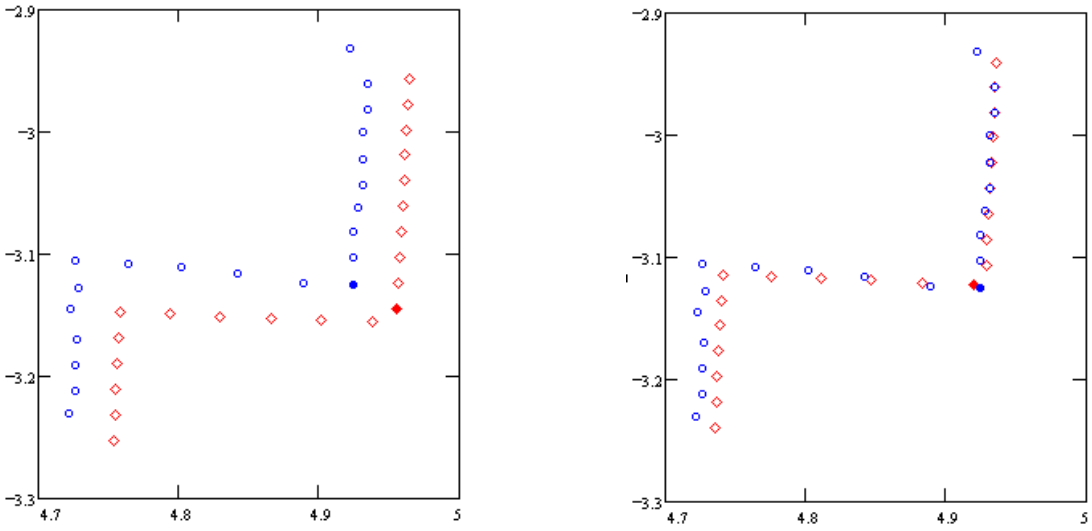
In the figures below the simulated scan and the laser scan around the four landmark pairs used with Reference Transform are presented before (left) and after (right) the algorithm was run using simultaneously the four landmark pairs (Figure 10).



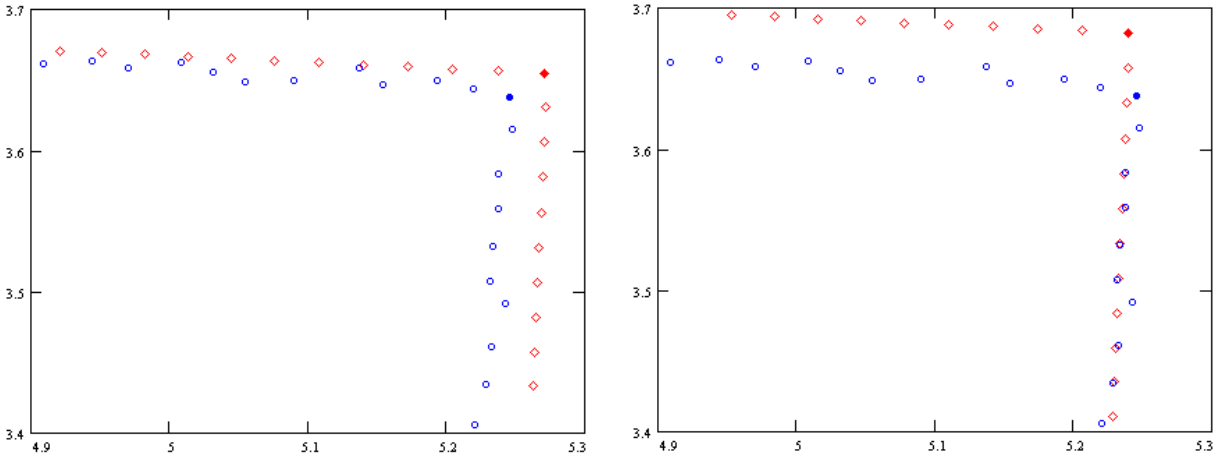
(a) – First landmark (bottom left corner of Figure 10)



(b) – Second landmark (bottom right corner of Figure 10)



(c) – Third landmark (bottom right corner of Figure 10)



(d) – Fourth landmark (top right corner of Figure 10)

Figure 11 – Adjusting the simulated scan to the laser scan with Reference Transform

Since the spatial resolution of the laser scanner reduces as the range augments, the scale used in Figure 11 varies to embrace the full landmark pair. However, the distance between markers is constant (0.1m) to improve the compared analysis. It should be noticed that the simulated scan exhibits small variations, since the estimate posture relative to the scenario is updated in the process.

The convention in the legend is the same as before: squares (or diamonds) represent the simulated scan, while circles represent the laser scan. From visual inspection only of the left columns in Figure 11, one concludes that the two scans are aligned and therefore the required angle correction is negligible. Regarding the position in the XY-axes, the simulated scan should move to the left and upwards to match the laser scan although the fourth landmark (Figure 11d) suggests that a displacement to the left suffices to match the two scans.

The experimental results, illustrated in Figure 11 at the right column, confirm this impression. They were obtained after a single iteration of Reference Transform, using the four landmarks. The correction vector computed with Reference Transform is

$$(x_P, y_P, \theta_P) = (0.0248\text{m}, -0.0188\text{m}, -7.80\times10^{-4}\,\text{rad}).$$

The initial and refined estimates are presented in Table 1 with the Likelihood Test (LT) moments (see Chapter 3, Section 3). The number of valid pairs is 1753 in both cases.

Algorithms	Posture Estimates			Likelihood Test moments		
	x [m]	y [m]	θ [rad]	MP	EV [m]	$\sqrt{Dispersion}$ [m]
Frame Localisation	1.0992	3.1116	0.0469	1457	0.02838	0.04045
Reference Transform	1.12645	3.0920	0.04612	1456	0.02811	0.05217
correction	0.02725	-0.01963	-0.78×10^{-3}	-	-	-

Table 1 - Approximate Localisation results with Reference Transform

The point-to-point distance distributions are shown in Figure 12, with the Reference Transform in the foreground (lighter) and the Frame Localisation in the background (darker).

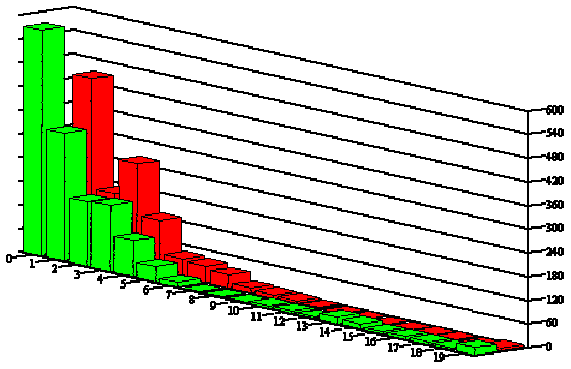


Figure 12 – Comparing Reference Transform (lighter) with Frame Localisation (darker)

The correction is more apparent in the distribution (Figure 12) than in the Expected Value (EV), (Table 1). This is due to the fact that the original posture estimate, computed by Frame Localisation is already a good estimate; the expected value is 0.02838m, while the typical expected value due to the laser only is around 0.02m. In addition, since the fitting is based on four samples only, it may occur that the point-to-point distance

increases in some cases, denoted by the increased number of hits in slots 12 to 19 with Reference Transform (see Figure 12). Since the pairs with higher distance weigh more to the Expected Value and Dispersion, the posture enhancement may appear smaller.

The Error Descent algorithm was applied to the same initial posture estimate supplied by Frame Localisation. The Error Descent was run with an initial step of 0.05m and 0.005rad (0.3º) and seven iterations (the final step is less than 0.0008m and 0.00008rad (0.0045º)).

A summary of the results of the iteration sequence is presented in Table 2. For each iteration, the number of leaves, the posture estimate coordinates, both in grid coordinates and metric coordinates, and the associated Expected Value (EV) are presented. The Expected Value is used instead of the cost, (4.29), because the difference between valid pairs and match pairs is constant throughout the iterations.

The grid coordinates (q_k's) are associated to the scale of the last iteration (with the smaller steps) to create a uniform measure of distance and allow the comparison between iterations.

iteration	# leaves	optimal q_i			optimal p_i [m, m, rad]	optimal EV [m]
initial	-	0	0	0	(1.0992, 3.1116, 0.0469)	0.02838
1	1	0	0	0	(1.0992, 3.1116, 0.0469)	0.02838
2	4	32	0	32	(1.1242, 3.1116, 0.0494)	0.02298
3	12	32	-16	16	(1. 1242, 3.0991, 0.04815)	0.02278
4	69	32	-8	0	(1. 1242, 3.10535, 0.0469)	0.02270
5	313	28	-8	12	(1.12108, 3.10535,0.04784)	0.02236
6	1553	24	-14	38	(1.11795, 3.10066, 0.04987)	0.02195
7	6933	35	-1	-6	(1.12654, 3.11082, 0.04643)	0.02189

Table 2 - Approximate Localisation results with Error Descent

Not surprisingly, the overall results are better than with Reference Transform. In fact, since the measure of quality is established by the Likelihood Test, any method based on optimising the LT moments would lead to the optimal estimate.

At the end of each iteration, the number of remaining leaves initiates the next iteration. Thus, in general, there is not a inheritance relation between the optimal estimate at iteration *k*, and the optimal estimate at iteration *k+1*.

The point-to-point distance distribution for the optimal posture is shown in Figure 13, with Error Descent in the foreground and Frame Localisation in the background.

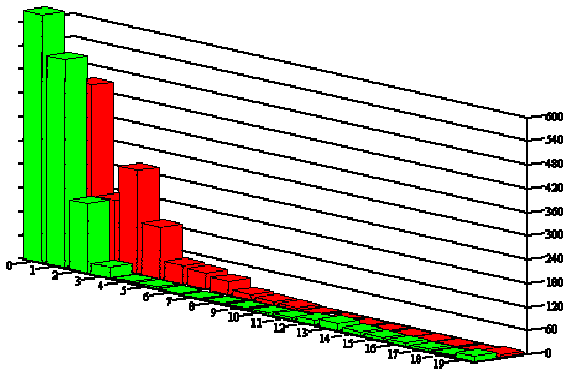


Figure 13 – Comparing Error Descent (lighter) with Frame Localisation (darker).

It should be noticed that after the second iteration, where the EV drops to 0.2298 and the steps are $\delta_X = \delta_Y = 0.025\text{m}$ and $\delta_\theta = 0.0025\text{rad}$ (0.14º), the enhancements are negligible – about one millimetre less in EV. Most often, these enhancements correspond to adjusting the data to fit some samples with high pair-to-pair distance, a side effect of the Likelihood Test form. It would be necessary to reduce the weight granted to postures located at the right of the histogram to minimise this effect.

All subsequent operations, which take about 90% of the run time of Error Descent, are of little relevance for localisation purposes. Although the optimal step and number of operations are difficult to define, the proposed number of iterations is exaggerated. The number of iterations was chosen for illustration purposes only. In regular field trials, four iterations suffice for simple environments and five are used for scenarios like the "factory". Any further operations will have little effect on the estimate quality.

Finally, the refined estimates yielded by the two Approximate Localisation algorithms are compared in Table 3. The point-to-point distance distributions are compared in Figure 14, where Reference Transform is in the background (darker) and Error Descent is in the foreground (lighter).

Algorithms	Posture Estimates			Likelihood Test moments		
	x [m]	y [m]	θ [rad]	MP	EV [m]	$\sqrt{Dispersion}$ [m]
Reference Transform	1.12645	3.0920	0.04612	1456	0.02811	0.05217
Error Descent	1.12654	3.1108	0.04643	1456	0.02189	0.04589
difference	0.00009	0.0188	0.31×10^{-3}	0	-0.00622	-0.00628

Table 3 - Comparing Error Descent with Reference Transform

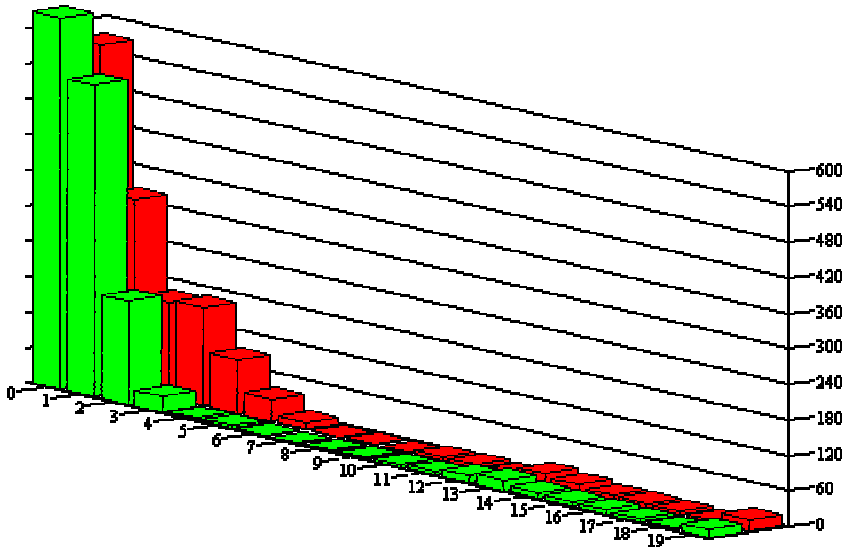


Figure 14 – Comparing Error Descent (lighter) with Reference Transform (darker)

In the posture estimates, only the difference in Y is significant. This had been suggested by the misalignment of the fourth landmark in Reference Transform, (Figure 11d). Perhaps, there is a systematic error in a large extension of the laser scan that is minimised with Error Descent but not in Reference Transform. This effect is visible in the columns from 2cm to 6 cm (Figure 14). Notwithstanding, both algorithms met the accuracy specifications and proved to be adequate to this environment.

4.4.2 The office

A large number of experiments were run in the office room. Two of them were illustrated in Chapter 3, Section 3.4.2. In this section, the first experiment presented in trial (a) is resumed with the two Approximate Localisation algorithms. The second experiment (trials (b) and (c)) are not mentioned since the Reference Transform results are based on the same features as trial (a). Instead, a new experiment – trial (d) - is introduced with laser data measured from a different location.

The Reference Transform algorithm was run with a sliding window of size $W = 4$. The maximum shift between the simulated scan and the laser scan is $Z = 16$. The average inner threshold is $InnerThr < 0.1$, corresponding to an angle of $\pm \pi/2 \pm 0.1$ ($\pm 90° \pm 6°$). In this experiment, only two landmarks were detected (Figure 15). This is a result of the office structure: two walls are covered with windows, and few corners are visible. Notwithstanding, the Reference Transform updated successfully an initial estimate computed by Frame Localisation.

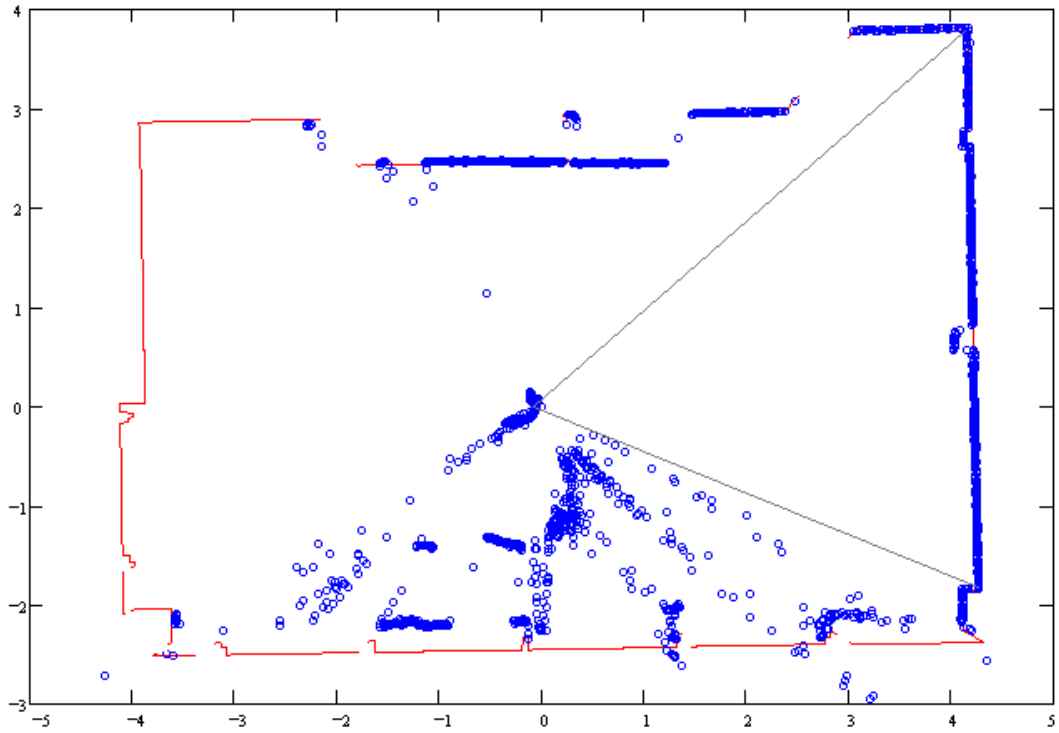
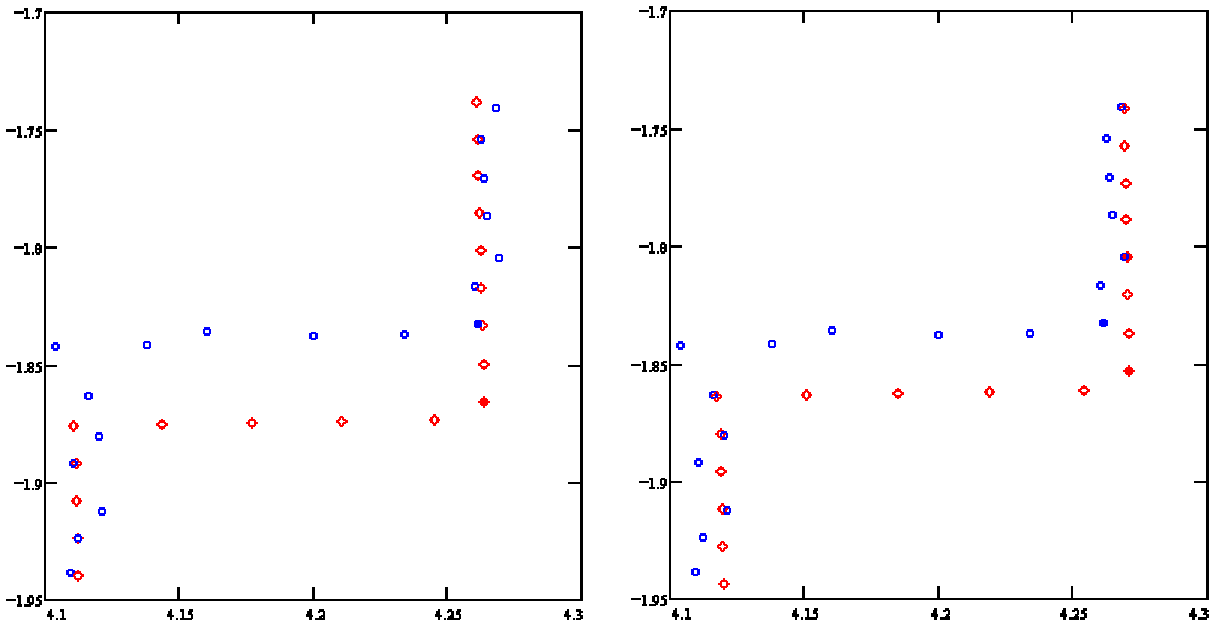
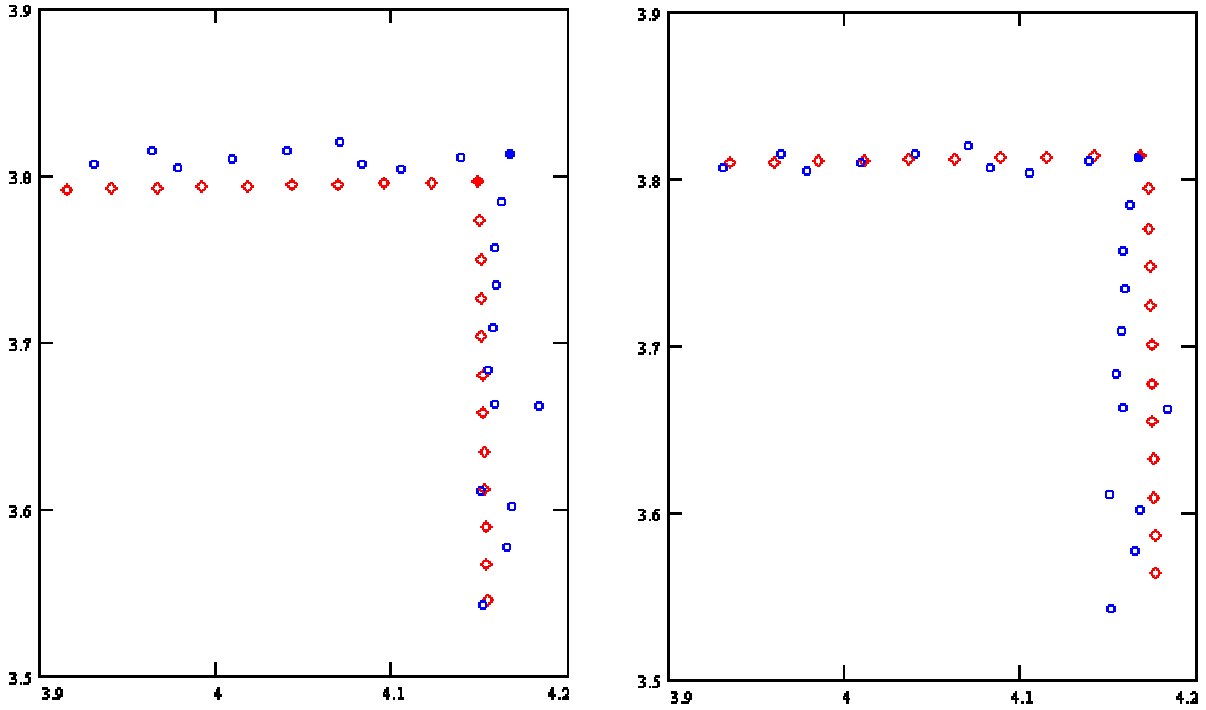


Figure 15 – Trial (a): detected landmarks for Reference Transform

The figures below represent the laser scan and the simulated scan around the two landmark pairs before Reference Transform in the left column and afterwards in the right column. In Figure 16 the diamonds represent the simulated scan while the circles represent the laser scan. The distance between markers is 0.05m.



(a) – First landmark (bottom right corner of Figure 15)



(b) – Second landmark (top right corner of Figure 15)

Figure 16 – Trial (a): adjusting the simulated scan to the laser scan

By examining the figures, it is clear that the two scans are nearly aligned and the correction in position is small. Because there are only two landmarks, the system solution led to an almost exact match (Figure 16b), while the other accommodates the error due to the residual differences between the laser scan and the simulated scan.

The correction posture is

$$(x_P, y_P, \theta_P) = (-0.0043\text{m}, -0.0395\text{m}, -3.54\times10^{-3}\,\text{rad}).$$

The initial and refined estimates are presented in Table 4, with the Likelihood Test moment. The number of valid pairs is 1738.

Algorithms	Posture Estimates			Likelihood Test moments		
	x [m]	y [m]	θ [rad]	MP	EV [m]	$\sqrt{Dispersion}$ [m]
Frame Localisation	4.1368	2.4759	-0.0186	956	0.05876	0.11125
Reference Transform	4.1237	2.4510	-0.0151	977	0.05807	0.11454
correction	-0.0131	-0.0249	-3.5×10^{-3}	-	-	-

Table 4 - Trial (a): Approximate Localisation results with Reference Transform

There is an enhancement in the number of Mathc pairs (MP) and in the Expected Value (EV). The additional dispersion is due to the increased number of MP detected in Reference Transform. Since these pairs have high point-to-point distance, their contribution to the distribution dispersion is significant.

The point-to-point distance distributions in Figure 17 show the Reference Transform histograms in the foreground (lighter) and the Frame Localisation histogram in the background (darker). There are some enhancements in the first classes (from 1cm to 8cm), at the expense of an increased number of samples in the last slot.

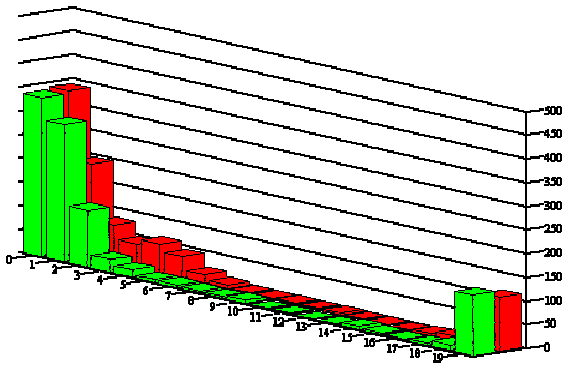


Figure 17 – Trial (a): comparing Reference Transform (lighter) with Frame Localisation (darker)

The overall enhancement is modest due to the reduced number of landmarks and uneven landmark distribution around the robot.

The Error Descent was run with an initial step of 0.05m and 0.005rad (0.3º) and five iterations (the final step is 0.003125m and 0.0003125rad (0.018º)). It should be noticed that the q_i's in this experiment correspond to steps four times as large as the steps in the classroom. The algorithm evolution is traced in Table 5. The number of Valid Pairs is 1738 and the number of Match Pairs increases from 956 to 967 as the algorithm progresses.

iteration	# leaves	optimal q_i			optimal p_i [m, m, rad]	optimal EV [m]
initial	-	0	0	0	(4.1368, 2.4759, -0.0186)	0.05876
1	1	0	0	0	(4.1368, 2.4759, -0.0186)	0.05876
2	3	0	-8	16	(4.1368, 2.4509, -0.0136)	0.05527
3	6	0	-8	16	(4.1368, 2.4509, -0.0136)	0.05527
4	16	0 0	-6 -6	12 10	(4.1368, 2.4572, -0.01485) (4.1368, 2.4572, -0.01548)	0.05484
5	90	0	-7	11	(4.1368, 2.4540, -0.01516)	0.05482

Table 5 – Trial (a): Approximate Localisation results with Error Descent

In the fourth iteration, there are two close candidate postures (leaves) with approximately the same weight. Accordingly, the best posture estimate found in the fifth iteration lies between them. The point-to-point distance histograms are presented in Figure 18. The enhancement with Error Descent is significant, although the number of instances in the last slot also increases, albeit less than with Reference Transform.

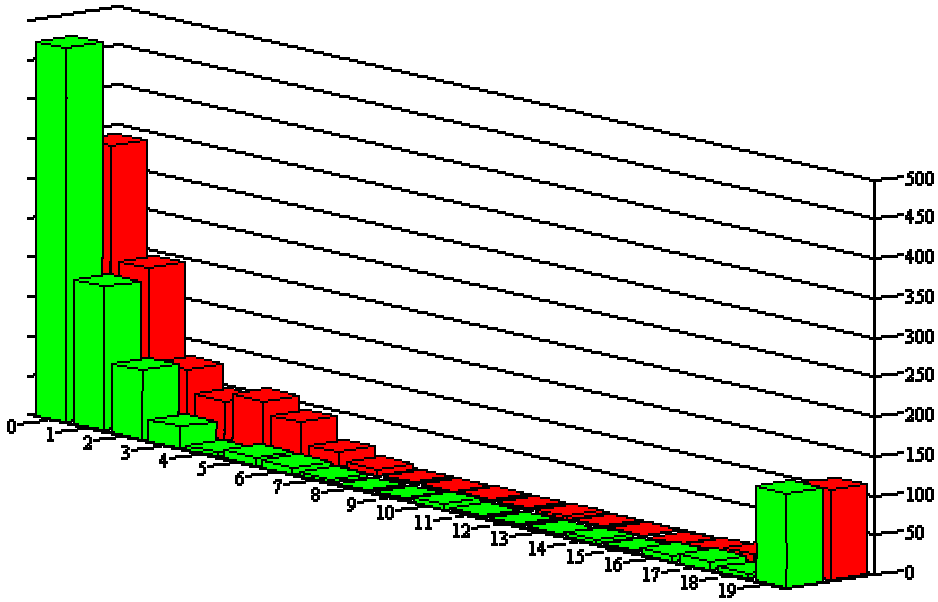


Figure 18 – Trial (a): comparing Error Descent (lighter) with Frame Localisation (darker)

The comparative analysis of the two Approximate Localisation algorithms is illustrated in Table 6 and in Figure 19.

Algorithms	Posture Estimates			Likelihood Test moments		
	x [m]	y [m]	θ [rad]	MP	EV [m]	$\sqrt{Dispersion}$ [m]
Reference Transform	4.1237	2.4510	-0.01506	977	0.05876	0.1145
Error Descent	4.1368	2.4540	-0.01516	967	0.05482	0.1118
difference	0.0131	0.003	0.1×10^{-3}	-10	-0.00394	-0.00628

Table 6 - Trial (a): comparing Error Descent with Reference Transform

The difference in the X-axis is due to the poor discrimination of error in the horizontal axis in Reference Transform. Notice that in Figure 15 the two landmark pairs are located at approximately the same coordinate in the X-axis. Thus, correcting the x-estimate based on these two samples only emphasises the effects of systematic laser errors.

The difference in the Expected Value (EV) and dispersion moments is manly due to the extra instances in the last slot. The Reference Transform includes 128 instances while the Error Descent includes only 119. The extra "cost" in the EV due solely to these nine instances is 0.00274.

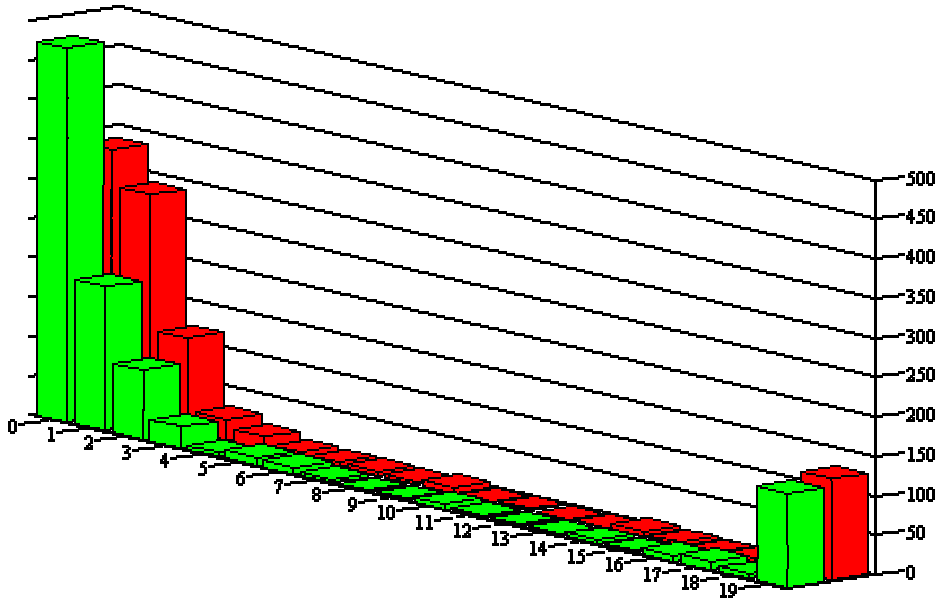


Figure 19 – Trial (a): comparing Error Descent (lighter) with Reference Transform (darker)

In this experiment, the Reference Transform was pushed to its limits. Although the Localisation is possible with two landmarks only, it is prone to errors that might prevent the desired posture estimate refinement. On the other hand, the Error Descent results suggest that the computation effort may be reduced by a large fraction (approximately 80% to 90%) without significant loss of accuracy. Therefore, the main cause restraining the use of Error Descent loses importance.

In the second trial, termed Trial (d) to avoid confusion with previous trials presented in Chapter 3, Section 4.2, the robot was located close to the file cabinets in the west wall, (Figure 20). The parameters in this trial are the same as before. Three landmark pairs are detected from this location.

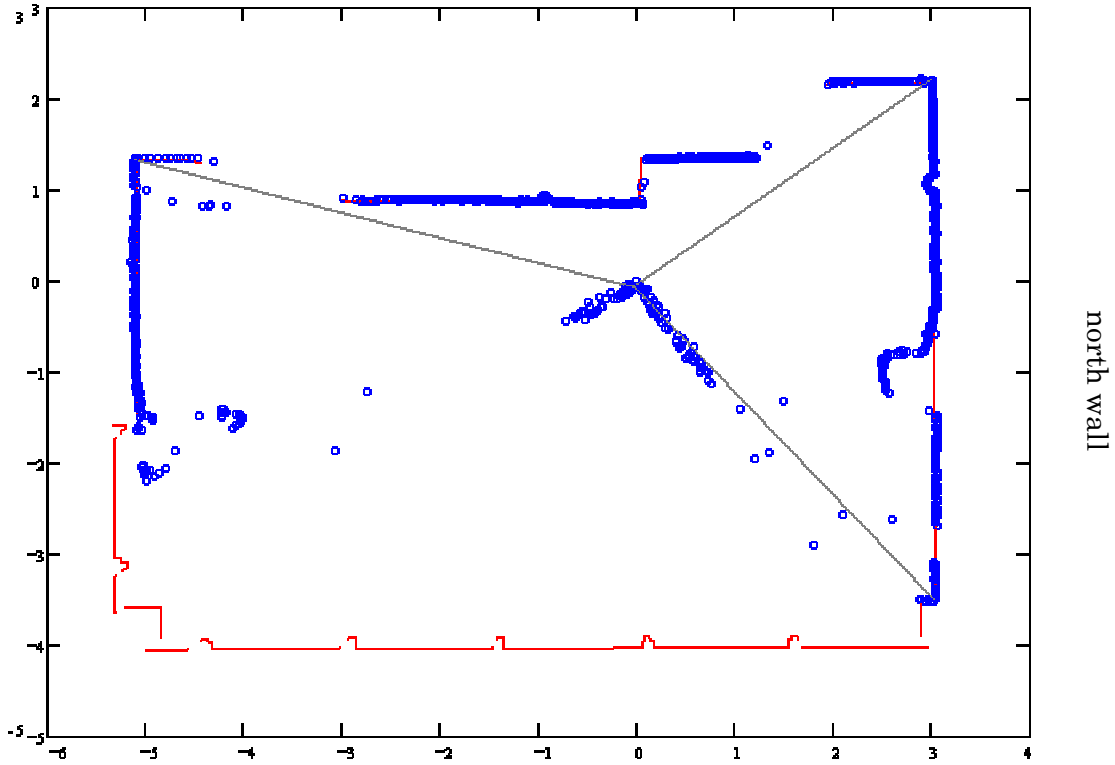
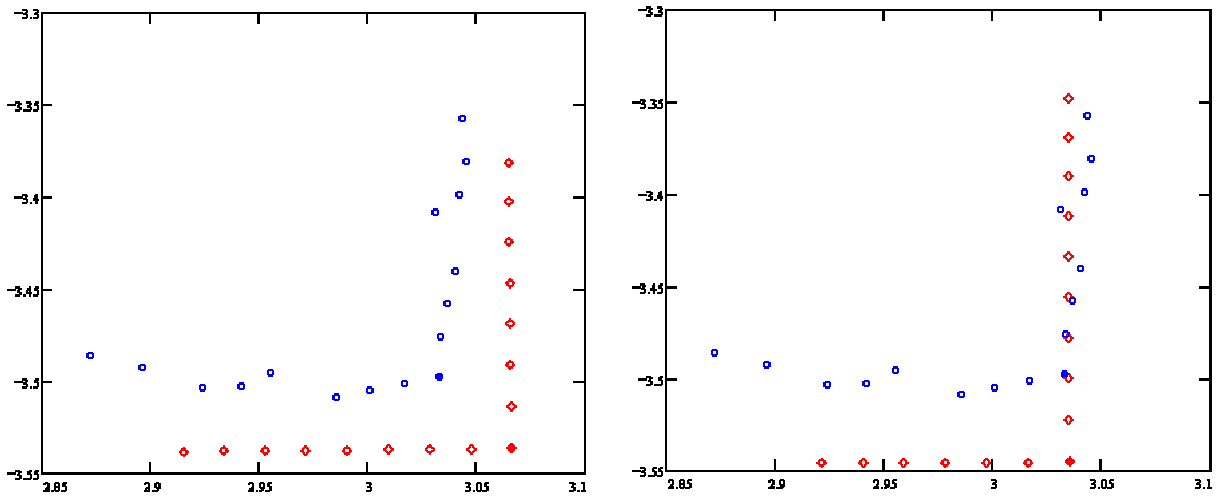
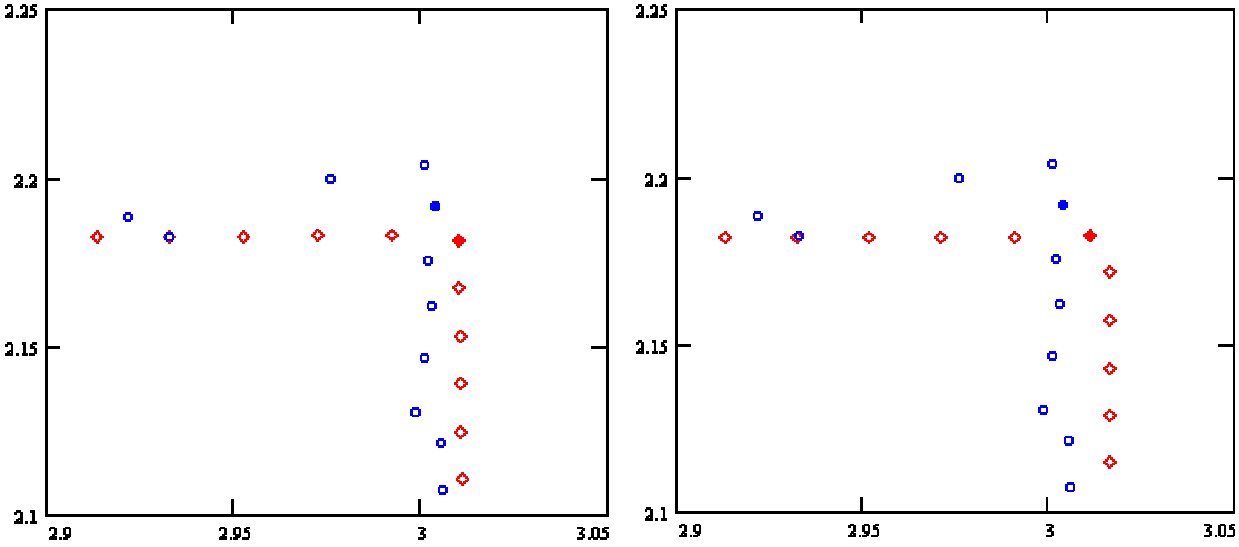


Figure 20 – Trial (d): detected landmarks for Reference Transform

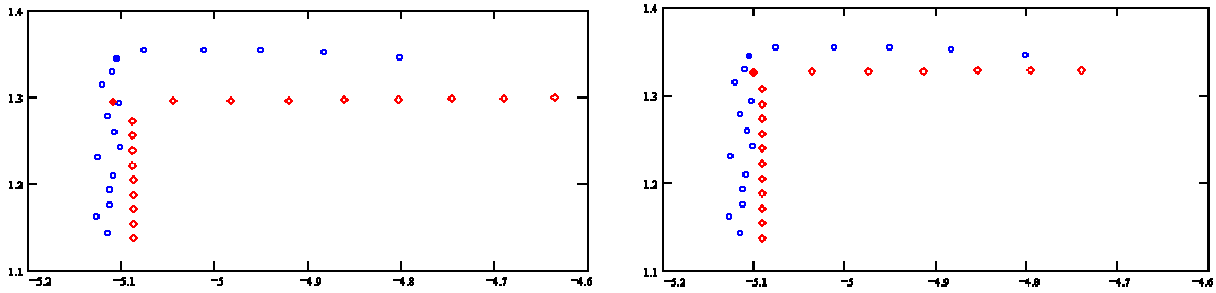
The first relevant note is the increased accuracy and significance, relative to the previous trial, of the initial posture estimate computed by Frame Localisation, due to the extra data on the south wall. This corresponds to the metallic blind that was lowered during the trial and was added to the scene map, as described in Trial (c), see Chapter 3, Section 4. The figures are summarised in Table 7 and illustrated in Figure 22.



(a) – First landmark (bottom right corner of Figure 20)



(b) – Second landmark (top right corner of Figure 20)



(c) – Third landmark (top left corner of Figure 20)

Figure 21 – Trial (d): adjusting the simulated scan to the laser scan with Reference Transform

The three landmark pairs are presented in Figure 21 as in previous trials. The simulated scan and laser scan before Reference Transform are presented on the left while the simulated and laser scans after the algorithm are presented on the right.

The correction posture computed with one iteration of Reference Transform and using simultaneously the three landmark pairs is:

$$(x_P, y_P, \theta_P) = (-0.028695\text{m}, -0.035778\text{m}, 4.835\times10^{-3}\,\text{rad})\,.$$

Running additional iterations doesn't increase the estimate's quality since the three landmarks do not require the same correction. The Likelihood Test parameters are presented in Table 7. The number of Valid Pairs is 1758. The point-to-point distance histograms are illustrated in Figure 22.

Algorithms	Posture Estimates			Likelihood Test moments		
	x [m]	y [m]	θ [rad]	MP	EV [m]	$\sqrt{Dispersion}$ [m]
Frame Localisation	5.3274	4.0349	1.5637	1300	0.02778	0.056
Reference Transform	5.3365	4.0248	1.5685	1301	0.02604	0.0527
correction	0.0091	-0.0101	4.835×10^{-3}	-	-	-

Table 7 - Approximate Localisation results with Reference Transform

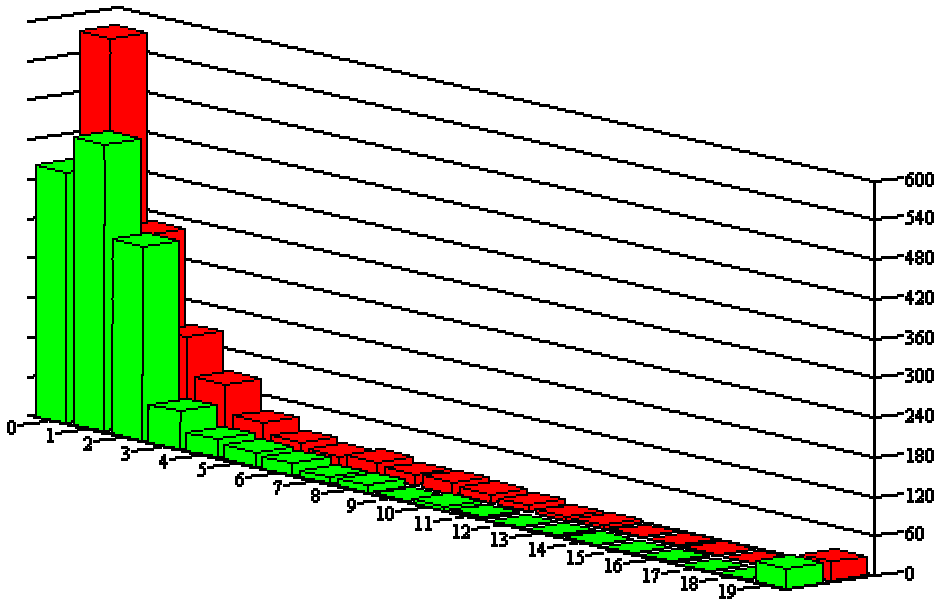


Figure 22 – Trial (d): comparing Reference Transform (lighter) with Frame Localisation (darker)

In spite of the high quality of the Frame Localisation estimate, the Reference Transform successfully refined the posture. Apart from the robot location, the main difference from Trial (a) to Trial (d) is the addition of lowered blind, both in the scan and in the map data. This offers an extra landmark pair in the opposite wall in Trial (d). Comparing Trial (d) with Trial (a), the Expected Value (EV) and Dispersion were reduced to half the values, approximately. In addition, the number of match pairs increased from 956 to 1300. A more adequate positioning of the laser sensor and the inclusion of the blinds in the map description contributed to this result. It also contributed to the extra landmark pair detected by Reference Transform in the top left corner.

This new positioning reveals the difficulties of Reference Transform in presence of noisy laser data. In particular, the second landmark (Figure 21b) shows an irregular profile just after the detected landmark that prevented the corner detection. To proceed with the algorithm, the first sample after the landmark was manually classified as an

outlier and the algorithm discarded it. The posture update embedded in (4.24) automatically weighs the various conflicting corrections determined by each of the equations in (4.20). The major correction is performed in orientation. The combined effects of orientation plus translation update results in larger shift in the X-axis direction for the first landmark (Figure 21a), whereas the third landmark was shifted mainly in the Y-axis direction (Figure 21c) and the second landmark appears to have little correction, since it is closer to the robot location, where the orientation correction represents a lower spatial correction.

The Error Descent algorithm was run with the same parameters as in the previous trial. The number of Valid Pairs is 1758 and the number of Match Pairs varies from 1300 to 1303. The progression of the algorithm is shown in Table 8 and the point-to-point distance histograms are shown in Figure 23.

iteration	# leaves	optimal q_i			optimal p_i [m, m, rad]	optimal EV
initial	-	0	0	0	(5.3274, 4.0349, 1.5637)	0.02778
1	2	0	0	16	(5.3274, 4.0349, 1.5687)	0.02702
2	3	0	0	16	(5.3274, 4.0349, 1.5687)	0.02702
3	17	4	-4	16	(5.3399, 4.0224, 1.5687)	0.02504
4	61	0	-2	18	(5.3274, 4.0287, 1.5693)	0.02497
5	223	1	-4	14	(5.3305, 4.0224, 1.5681)	0.02481

Table 8 - Approximate Localisation results with Error Descent

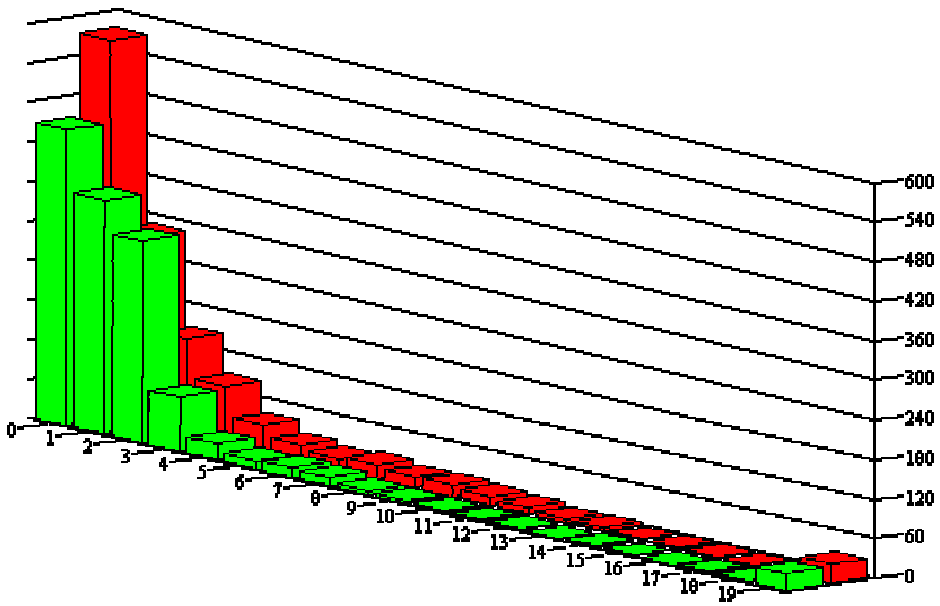


Figure 23 – Trial (d): comparing Error Descent (lighter) with Frame Localisation (darker)

The correction required to enhance the Expected Value (EV) is concentrated in the orientation (0.044rad = 0.25º). The histograms in Figure 23 show that a significant weight in the first slot (125 instances less) is moved towards the right as a necessary cost to reduce the number of samples in the middle of the histogram (between slots 3cm to 14cm, the number of instances is reduced by 98). This is a typical result of Approximate Localisation, although the effect in the first slot is more apparent than usual. The overall result is a moderate enhancement in the Likelihood Test parameters.

In Trial (d), the results of the two Approximate Localisation algorithms are similar, as it can be verified in Table 9 and Figure 24.

Algorithms	Posture Estimates			Likelihood Test moments		
	x [m]	y [m]	θ [rad]	MP	EV [m]	$\sqrt{Dispersion}$ [m]
Reference Transform	5.3365	4.0248	1.5685	1301	0.02604	0.05274
Error Descent	5.3305	4.0224	1.5681	1303	0.02481	0.05033
difference	-0.006	-0.0024	0.4×10^{-3}	2	-0.00123	-0.002409

Table 9 - Comparing Error Descent with Reference Transform

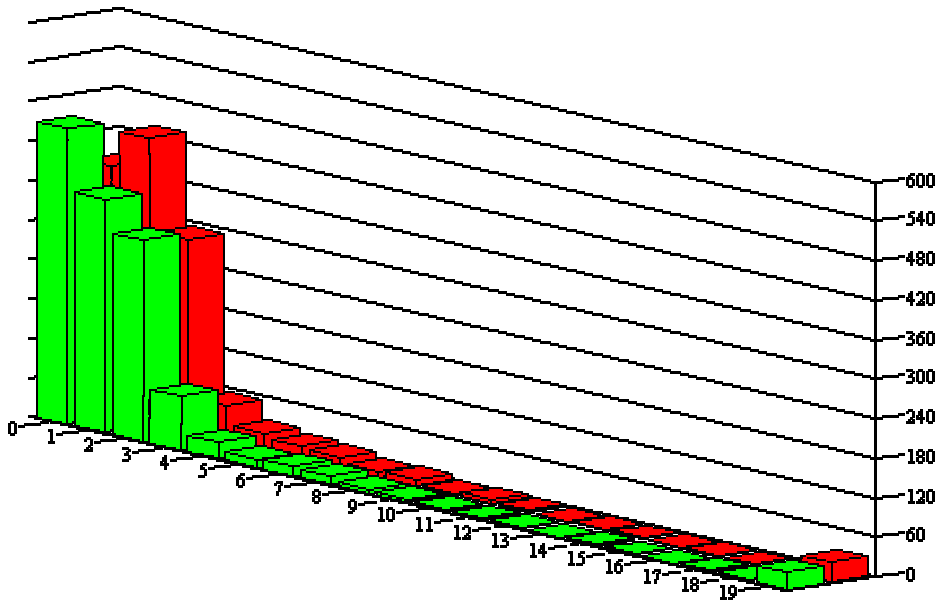


Figure 24 – Trial (d): comparing Error Descent (lighter) with Reference Transform (darker)

4.5 Comparing the Approximate Localisation algorithms

The two Approximate Localisation algorithms follow opposite approaches to the problem of posture estimate refinement. Although the Reference Transform is more efficient and aesthetically more elegant, the Error Descent is far more robust even though requiring a computation time that can be 1000 times longer.

In good operation conditions, the quality of the results is similar. The Reference Transform is less tolerant to adverse conditions, while the Error Descent very seldom fails to provide some posture enhancement. As the RESOLV project progressed, the latter trials were conducted in conditions where the Reference Transform failed due to the absence of adequate maps and landmarks or because the laser data was too noisy. The noise in the laser data depends on the laser but also on the reflectance of the scanned surfaces. In addition, the laser was operated close to its operation range, which reduces the quality of the estimates. On the other hand, using fewer iterations, the computation effort associated to Error Descent may be reduced by a factor of five to ten with negligible effects on the estimate's accuracy.

Due to its superior performance, the automatic implementations of Approximate Localisation use only the Error Descent algorithm, and the experimental results presented in Chapter 3 are all based on Error Descent, since they were obtained in actual field trials.

The parameters to be defined by the user in Reference Transform are: the length of the sliding window, the length of the search window for matching landmark pairs and the inner product threshold for detecting a landmark. The linear approximation boundaries are not controlled by the user since its tolerance is much wider than the overall tolerance of the algorithm.

The length of the sliding window is determined by map constraints. If the sliding window is too long, it might encompass more than one scene feature, concealing the landmark. On the other hand, if the sliding window is too short, the results are too sensitive to angle of incidence variations and minor features in the scene map, especially in 3D reconstructed maps. In most experiments, the sliding window varies from nine ($W = 4$) to fifteen ($W = 7$) samples.

The length of the search window is determined by the distance between the laser scan and the simulated scan. If the two scans are very close – measured by the point-to-point distance distribution - or if the surfaces vary rapidly, a small search distance will be adequate to detect the exact match. If the two scans are quite different or if the surfaces are smooth, the search window should be wider. In case the scans are very different and Reference Transform detects few landmark pairs, it might be worthwhile running the algorithm again with a wider window. Typical search interval range from twelve ($Z = 6$) to forty ($Z = 20$) samples.

The inner angle threshold depends on the nature of the landmarks and the quality

of the map data and the scan data. If the room has square angles, the threshold is determined by the quality of the map data and the scan data. Otherwise, the threshold is determined by the angle in the surfaces meeting at the selected natural landmarks. In case the map or scan data are poor, the threshold must be relaxed to accommodate higher errors, as in the office experiments.

The Error Descent algorithm has fewer parameters left to the operator for tuning. The number of iterations controls the accuracy and the computation time, while the linear and angular steps control the maximum leap between adjacent postures in the first run. These three parameters are far more intuitive to the operator, therefore the results are less exposed to tuning errors. Again, Error Descente is superior to Reference Transform.

In spite of all the advantages mentioned, the Error Descent has a subtle pitfall. Because the algorithm is based on the cost measure, it always converges to the lower cost, hence yielding an apparent high quality estimate. However, the Likelihood Test does not react to the scan differences in the same manner as the human operators. Humans learn to discard outliers and irrelevant errors, while Likelihood Test grants extra emphasis to the samples with larger point-to-point distance. The Likelihood Distance extension tries to cope with this issue using a modulator factor to reduce the relevance of higher distance instances.

Notwithstanding, the Error Descent algorithm might “anchor” in an estimate that is optimal regarding the Likelihood Test parameters, but are wrong from the human operator’s point of view. This is the result of an effort to minimise the point-to-point distance associated to some particular instances that are clearly outliers for the trained human operator, as in Chapter 3, Section 4, Trials (b) and (c) in the office. This occurs most often in the presence of non-mapped features “fit” in the available map by the algorithm. Although this solution is correct because the existing map is the ultimate ground truth, this result is clearly undesirable for the human operator.

There are two ways to overcome this difficulty. The best one is to add the feature to the map, when necessary. This was the path followed in the experiment illustrated in Chapter 3. The second one is to analyse the graphical results yielded by the algorithm, search for intermediate steps, and determine the best posture estimate from visual inspection of the graphics and Likelihood Test moments.

The final step of Approximate Localisation is a validation with Likelihood Test using parameters that are more stringent than the ones used in the validation of Frame Localisation. If the test passes, the solution is published and the algorithm terminates with success. However, if the test fails the Approximate Localisation algorithm ends with failure. In case Frame Localisation has not been run because the initial posture estimate was supplied by an external source, it is called at this stage.

In automatic implementations, if the validation tests run after Approximate Localisation fail, the posture estimate is always published. This solution serves the overall RESOLV mission. The other modules are informed by the Localisation termination state that the estimate is poorer than required but they still have a posture estimate to work with.