

Università degli Studi di Roma "*La Sapienza*"
Facoltà di Ingegneria

Relazione finale di Tirocinio
Ingegneria Automatica e dei Sistemi di Automazione
a.a. 2003/2004

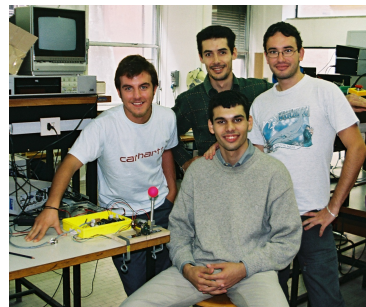
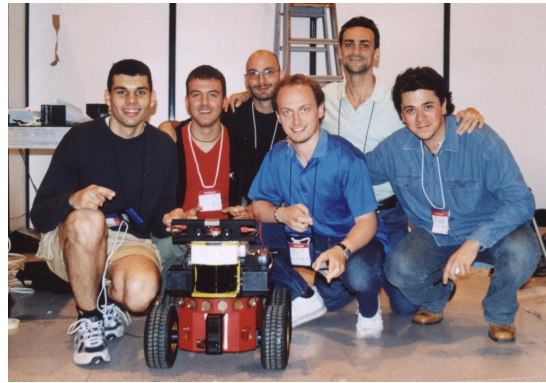
Scan Matching in the Hough Domain

Relatore:
prof. Daniele Nardi

Candidato:
Andrea Censi

Ringrazio Nardi-sensei per avermi dato l'opportunità di lavorare con una compagnia mica male: Giorgio Grisetti, che volendomi insegnare lo SLAM e a bere come un vero uomo, è riuscito in uno solo dei due intenti; Luca Iocchi, che ha reso comprensibile questo lavoro; Leone-san con il quale ho condiviso freddi pomeriggi al CNR; Shahram Bahadori, uomo di mondo; il saggio Calisi e il compagno Farinelli.

Vorrei inoltre ringraziare tutte le altre persone che in questi anni mi hanno quotidianamente sopportato; i colleghi di corso, fra cui Massimo Ferri e Luca Marchionni, e tutti i buoni maestri che ho trovato nell'università e nella vita.



In alto, pronti alla prima gara della Robocup Rescue a Lisbona. Da sinistra: Luca Marchionni, Giorgio Grisetti, Daniele Calisi, Luca Iocchi, Shahram Bahadori. In basso a sinistra, a Kobe, Riccardo Leone-san e Nardi-sensei. Una rara foto del Farinelli nel riquadro. A destra, davanti ad un avanzato apparato sperimentale, Luca Marchionni, Massimo Cefalo e Massimo Ferri.

Indice

1	Introduzione	7
1.1	Il problema dello scan matching	9
1.2	Il nostro approccio	10
2	Scan matching	13
2.1	Related work	14
2.2	Our approach	17
3	Properties of the Hough Transform	19
3.1	Hough Transform for lines	20
3.2	Properties of the parameter space for lines	22
3.3	The Discrete Hough Transform	24
3.4	The Hough Spectrum	26
4	The Hough Scan Matcher	29
4.1	Spectra correlation and ϕ estimation	30
4.2	T estimation	31
4.3	Complexity	34
5	Experiments	37
5.1	A somewhat limit case	37
5.2	Systematic experiments	37
6	Conclusions and future work	43

Capitolo 1

Introduzione

Il problema della localizzazione autonoma consiste nello stimare la posizione di un robot mobile all'interno di una mappa nota a partire dai dati sensoriali. Se la mappa non è nota e deve essere costruita anch'essa il problema prende il nome di SLAM (dall'inglese *simultaneous localization and mapping*).

Lo *scan matching* (in italiano *allineamento di scansioni*) è un problema "geometrico": si devono trovare le roto-traslazioni che massimizzano la sovrapposizione di due forme bidimensionali. Esso trova applicazione in alcuni degli approcci a localizzazione e SLAM.

La Trasformata di Hough (HT) è uno strumento usato da decenni per la visione artificiale ed è stato già usato in questo contesto per l'individuazione di *feature* - segmenti lineari - dai dati sensoriali.

Presentiamo una generalizzazione della HT e mostriamo come abbia notevole espressività e alcune interessanti proprietà di invarianza. Grazie a queste proprietà possiamo definire uno *spettro* della trasformata invariante a traslazione e rotazione. E' quindi possibile progettare un metodo di scan matching che non sia legato alla presenza di linee nell'ambiente e possa condurre una ricerca globale, multi-modale operando in ambienti destrutturati.

Alcuni dei risultati qui presentati appariranno [1] negli atti della *IEEE International Conference on Robotics and Automation (ICRA'05)*.

Il documento è organizzato nel modo seguente:

- Il capitolo 2 descrive la ricerca corrente nel campo dello scan matching.
- Il capitolo 3 descrive la trasformata di Hough e le proprietà del dominio di Hough.
- Il capitolo 4 descrive in dettaglio il nostro algoritmo.
- Il capitolo 5 descrive gli esperimenti eseguiti per validare il metodo.

Qui di seguito è riportato un sunto dei risultati in italiano.

1.1 Il problema dello scan matching

Il problema dello scan matching bidimensionale può essere espresso come: *dati due insiemi di dati 2D, determinare un insieme di trasformazioni 2D (una traslazione T e una rotazione ϕ) che li facciano sovrapporre.*

L'input tipico per un algoritmo di scan matching è

1. l'output di un sensore di distanza 2D (uno *scan*) che rappresenta il contorno dell'ambiente visibile intorno al robot;
2. dei dati di riferimento che rappresentano l'ambiente in cui si muove il robot e che possono variare a seconda dell'applicazione (una mappa, uno scan precedente).

Lo scan matching è usato, applicato ai dati provenienti da sensori di distanza densi (per esempio, un sensore laser), in due contesti tipici: la localizzazione globale e il tracciamento della posizione. In entrambi i casi lo scan matching ha, oltre alla geometrica, un'interpretazione legata alla formulazione del problema come problema di filtraggio bayesiano.

Scan matching per localizzazione globale Nel caso della localizzazione globale, la mappa a disposizione del robot è tipicamente molto più grande della porzione percepita istantaneamente dai sensori a disposizione. Il problema può essere risolto in modo efficiente [2–4] in due passi: prima si usa un metodo come lo scan matching per ottenere un insieme di pose candidate - cioè un'approssimazione di $p(x_t|y_t, m)$, dove x_t è la posa del robot al tempo t , y_t i dati sensoriali e m la mappa disponibile - quindi si usa un metodo di filtraggio nel tempo per discriminare tra i candidati.

Uno scan matcher per localizzazione globale deve produrre un output multi-modale, essere molto veloce e robusto al rumore del sensore e della mappa.

Scan matching per SLAM Lo scan matching è usato da molti dei metodi di SLAM che usano una rappresentazione “densa” dell'ambiente.

I dati odometrici dei robot mobili sono tipicamente corrotti da rumore sia sistematico che casuale. Lo scan matching può essere usato per eliminare il rumore sistematico o per sostituire del tutto l'odometria. E' quindi possibile generare mappe consistenti almeno localmente (l'errore residuo è comunque non limitato).

Per i metodi che trattano SLAM come un problema di filtraggio [5], lo scan matching può considerarsi come l'approssimazione di una distribuzione di probabilità. Sia u_t la stima odometrica, y_t i dati sensoriali, m_t la mappa stimata al tempot, x_t la stima corrente per la posa. Se si considerano come input due scan successivi, il matcher approssima $p(x_t|x_{t-1}, u_{t-1}, y_t, y_{t-1})$; se uno scan è allineato rispetto a una mappa, approssima $p(x_t|x_{t-1}, u_{t-1}, y_t, m_{t-1})$.

Sarebbe desiderabile avere una approssimazione analitica e multimodale, per esempio come somma di gaussiane; molti metodi forniscono un solo risultato (il picco della distribuzione), altri danno anche una misura della bontà della stima.

1.2 Il nostro approccio

Illustriamo un metodo di scan matching basato sulla trasformata di Hough (HT) chiamato Hough Scan Matching (HSM). HSM è un matcher globale, multimodale, non iterativo che può operare in ambienti non strutturati.

- Non usa la trasformata di Hough nel modo tradizionale [6], cioè estraendo le feature dell'ambiente corrispondenti ai picchi della HT, ma lavora in modo denso su "distribuzioni di feature" usando tutta l'informazione contenuta nella trasformata. E' quindi robusto al rumore non assumendo l'esistenza di singole feature definite.
- Può eseguire una ricerca globale (per esempio per la localizzazione), ma può anche restringersi a ricerche locali, riducendo il tempo necessario (per esempio per il tracciamento della posizione).

- E' possibile dimostrare che l'algoritmo è completo se esiste una soluzione esatta.

L'algoritmo trova prima delle ipotesi per ϕ , rotazione, e quindi per ciascuna di esse delle ipotesi per la traslazione T .

Come mostriamo nel capitolo 3 è possibile definire una funzione "spettro", calcolata a partire dalla HT e unidimensionale, che ha queste proprietà:

1. è invariante alle traslazioni dell'input.
2. trasla se l'input subisce una rotazione

E' quindi possibile ottenere delle ipotesi per la rotazione confrontando gli spettri dei due input: è sufficiente calcolare la correlazione incrociata dei due spettri e prendere i picchi di questa come ipotesi per ϕ .

Se i due input sono uguali (a meno di una roto-traslazione ϕ, T), allora i due spettri saranno uguali (a meno di una traslazione di ϕ). La correlazione avrà quindi un massimo assoluto corrispondente alla rotazione ϕ .

Una volta trovata ϕ , è possibile esprimere dei vincoli lineari per T applicando ancora la correlazione incrociata a colonne della trasformata.

Gli esperimenti eseguiti in simulazione mostrano che in ambienti strutturati il metodo può tollerare ampie differenze nei due input oltre a diversi tipi di rumore sul sensore. In ambienti non strutturati si hanno buoni risultati per input relativamente simili.

Chapter 2

Scan matching

The 2D scan matching problem can be expressed as: *given two sets of 2D data (i.e. a reference scan and a current scan), determine a set of 2D transformations (a translation T and a rotation ϕ) that makes the scan data overlapping the reference data.*

The typical input for a scan matching algorithm is composed by:

1. the output of a 2D range sensor (e.g. a laser range finder) representing the contour of the visible environment around the sensor;
2. a reference scan representing the environment in which the robot moves and that can vary depending on the application (e.g. a pre-built map, a previous scan, etc.).

Scan matching for localization In global localization, the robot must localize itself on a map much larger than the single sensor scan, without an initial guess of the robot position. The problem can be solved using a first step (scan matching) for getting a set of candidate poses (an approximation of $p(x_t|y_t, m)$, where x_t is the robot pose at time t , y_t is the current sensor reading and m the available map), then by employing a filtering method to discriminate between those [2,3].

Scan matching turns the localization problem in a pattern-matching problem with some peculiarities: a range finder scan has particular ge-

ometric properties (radial ordering) and a sensor error model, moreover an underlying world model should be taken into account: the world can have changed since the map was created (dynamic elements such as doors, people, etc. are always present). The output of a scan matching method for this problem must be multi-modal.

Scan matching for SLAM Scan matching is used in most SLAM methods that use a non-feature based, "dense" representation of the environment.

The odometry readings of a robot are typically corrupted by both systematic and random noise. Scan matching can be used to eliminate systematic noise, therefore it is possible to build approximately locally consistent maps (the residual error is still unbounded).

A less naive interpretation: for methods who phrase the SLAM problem as a filtering problem [5], a scan matching procedure can be seen as a computation of a probability distribution. Let u_t be the encoder data, y_t the sensor reading, m_t the map estimated at time t and x_t the pose estimate. If two successive scans are matched, the matcher approximates $p(x_t|x_{t-1}, u_{t-1}, y_t, y_{t-1})$; if a scan is matched against a map, the matcher approximates $p(x_t|x_{t-1}, u_{t-1}, y_t, m_{t-1})$.

An analytical and multi-modal expression of the distribution would be desirable. Most of the scan matching methods provide only one answer (the peak of the distribution), while others provide a measure of the variance. Multi-modality is important because it provides a hook to perform explicit handling of ambiguities or uncertainty in the mapping process; however most of the current methods employ a mono-modal scan matcher and overcome this problem by other means [7].

2.1 Related work

Many techniques have been proposed for scan matching in the past years; however, a definitive solution does not exist because scan matching is

used in a vast range of operative conditions.

Scan matching methods differ largely based on the availability of a guess of the solution with relatively tight bounds (e.g. pose tracking) or the complete lack of a guess (e.g. global localization). In addition to this, not all methods provide a multi-modal solution.

Another distinction can be made with respect to the assumptions about the sensor data: presence of sensor noise and feature-richness of the environment. Feature-based scan matching is computationally more efficient, but suffers from two limitations:

- features must exist in the environment,
- feature extraction produces information loss.

On the other hand, dense scan matching is more robust to noise but can be very costly.

We first list some scan matching approaches that need a guess of the solution to operate.

Methods needing a guess of the solution If the environment is rich in roto-translations-invariant features (corners, straight walls [8]) and these are preserved notwithstanding the sensor noise, then it is possible to do simple filtering on the scan, extract them and find a solution, sometimes in closed form and linear time with respect to the number of points in the scan [9]. This approach has been proved to be efficient and robust when sufficient features are present in the environment.

In unstructured environments with relatively low noise, it is possible to employ algorithms of the ICP family [10] that do not assume the existence of features. These employ a two-step process: first, a series of heuristic correspondences between points in the two scans is established. Then a roto-translation that approximately satisfies the series of

constraints is found. The solution is obtained by iteratively executing the two steps until the error drops below a given threshold. In order to achieve convergence, such an approach requires the two scans to be taken at close enough positions. Modifications of the original algorithm allow for efficient implementations. [11]

If data noise is higher, there exists dense methods that search in the solution space and do not require the establishment of any feature-to-feature or point-to-point correspondence. In [12] the solution is searched by performing gradient descent on a score function. Such a function is built by convolving the reference scan with a Gaussian kernel and then correlating it with the sensor scan. This approach has been subsequently refined in [13] by defining a closed form for a potential function that can be minimized using the Newton algorithm. These approaches require a number of iterations that depends on the input configuration and the entity of the error.

Methods performing a global search In theory any “local” matcher can be (inefficiently) used for a global search. As for the iterative approaches, one can use as initial guesses random points that represent a fine sampling of the solutions space - finer than the realignment margins of the local scan matcher. Analogously, with features based matcher, a combinatorial approach to feature correspondences can be employed. However, methods specifically designed for global search have better performance.

In [10] a technique for dealing with arbitrary orientation errors has been proposed. This is done by computing an histogram of the local best fitting tangent lines in the two scans to align. Assuming a small translational error, the rotational invariance of such a normal histogram allows to determine the heading component of the two scans. The translational pose component is then computed by minimizing the distance among correspondent scan points.

In [14] a correlation based approach that is well suited in polygonal environments is presented. The orientation of each scan point is found

and the circular histogram of these is build. Then the rotation is found by correlating the normal histograms. This can be done since the relative angles of a set of lines does not change when changing the reference system. Once the heading is known, the translational component of the displacement is recovered by determining pairs of non parallel lines in the reference scan, and computing the translational error among the normal directions of those lines. Both these methods need that point orientation be extracted from data therefore are not reliable to sensor noise.

In [15] global localization is performed by using a two-dimensional correlation operator. This method evaluates every point in the search space and is therefore very robust to noise. Nevertheless, it can be implemented efficiently by the use of processor-specific SIMD (single instructions, multiple data) extensions.

2.2 Our approach

We present a scan matching method based on the Hough Transform (HT), called Hough Scan Matching (HSM). With respect to related work, HSM is a global, multi-modal, non-iterative matcher that can work in unstructured environments.

- Despite the use of the HT, it does not extract line features from the scan (as in previous approaches [6]), but, working in the Hough domain, it matches dense data that can be interpreted as “feature distributions” using all the information contained in the transform. It does not rely on specific features, therefore it is robust to sensor noise. The Hough domain enhances the properties of linear segments, resulting in better performance when they are present.
- HSM is able to perform a multi-modal, global search in the reference space, thus it is suitable for processing requiring extensive search (e.g. global localization). However, it can be restricted to perform local searches reducing computational time, when global search is not necessary (e.g. position tracking).

- The method is sound, in the sense that if the scan matching problem has only one exact solution (i.e. there exists only one rigid transformation that allows for perfectly overlapping the two scans), then HSM, without any initial guess, will include that one in the solution set.

Overview of HSM We will define a “spectrum” function, computed from the HT, which has these properties:

1. it is invariant to input translations,
2. it is (circularly) shifted on input rotation.

We are therefore able to get several estimates of the rotation by comparing the spectra of sensor and reference data - this is done via a cross-correlation operator. Compared to the angle-histogram method this first part of the algorithm does not need to know the surface direction therefore it is very robust to sensor noise.

At this point we have a list of hypotheses for orientation. Other existing algorithms could be used from here, to be run in a reduced search space to compute translation. We propose one that exploits the already computed HT and that can produce a list of solutions or a continuous distribution.

HSM will provide a ranking of the solutions that in most cases seems quite right for localization; however in the general case another likelihood should be used (e.g. log likelihood).

Chapter 3

Properties of the Hough Transform

The Hough Transform (HT) has been used in the computer vision community since the sixties (proposed by Hough and generalized in works such as [16]) as a method for detecting geometric curves (lines, circles) or patterns in digital pictures. Nowadays the name *Hough* refers to many methods employing some sort of parameter space [17].

In this section we will introduce a generalization of the HT and describe some of its properties.

Definition of the Hough Transform The HT maps an input $i(\mathbf{s})$, $\mathbf{s} \in \mathcal{S}$ (the *input space*) to a function $\text{HT}\{i\}(\mathbf{p})$, $\mathbf{p} \in \mathcal{P}$ (the *parameter space*).

Let $\mathcal{F}_{\mathcal{P}}$ be a family of sets indexed by \mathbf{p} such that $\mathcal{F}_{\mathbf{p}} \subset \mathcal{S}$. We define the Hough Transform of input $i(\mathbf{s})$, $\mathbf{s} \in \mathcal{S}$ as:

$$\text{HT}[\mathcal{F}, i](\mathbf{p}) = \int_{\mathcal{F}_{\mathbf{p}}} i(\mathbf{s}) ds \quad (3.1)$$

This definition is quite general; the only information is that the HT is linear with respect to $i(s)$.

3.1 Hough Transform for lines

The parameter space that we use here is the one representing straight lines in $\mathcal{S} = \mathbb{R}^2$. As we will show later, this choice does not impose a requirement about the presence of lines in the input; however, with this parametrization we can both exploit some important properties (described later in this Chapter) and take advantage of the presence of lines for increasing performance.

Different parameterizations can be found for representing a line in \mathbb{R}^2 . The common choice is to use the polar representation (θ, ρ) :

$$x \cos \theta + y \sin \theta = \rho \quad (3.2)$$

With this representation ρ is the distance of the line from the origin, and θ the direction of the normal vector (see Fig. 3.1). The family of sets for applying the Hough Transform is therefore the family of lines indexed by ρ and θ :

$$\mathcal{F}_{(\theta, \rho)} = \{(x, y) \mid x \cos \theta + y \sin \theta = \rho\} \quad (3.3)$$

Later in this paper we assume the input space to be a finite set of points $P = \{p_j\}$ (i.e. the output of a range sensor). Thus, we define $\mathbf{s} := (x, y) \in \mathcal{S}$ and

$$i(\mathbf{s}) = \sum_j \delta(\mathbf{s} - \mathbf{p}_j) \quad (3.4)$$

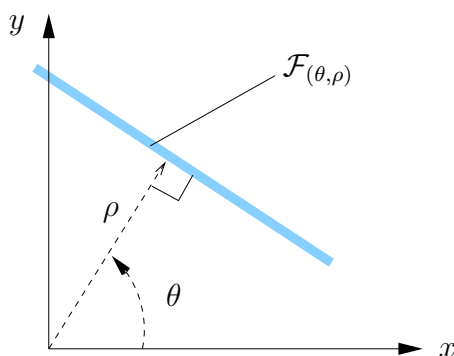


Figure 3.1: Parameterization for the line HT .

where δ is the Dirac impulse distribution. However note that in the following we will not impose any restriction on $i(s)$.

Maxima of (3.1) are set of parameters \mathbf{p}_j for which the sets $\mathcal{F}_{\mathbf{p}_j}$ are best represented in $i(\mathbf{s})$. For the straight lines, maxima of the HT are the parameters (θ, ρ) corresponding to the lines in the picture: this justifies the use of the HT as a feature detector.

3.2 Properties of the parameter space for lines

We describe some aspects of the Hough Transform that will lead to some invariance results.

Redundancy Some intuitive properties:

- the two points in \mathcal{P} : (θ, ρ) and $(\theta + 2\pi, \rho)$ represent the same line in \mathbb{R}^2 (HT(θ, ρ) is 2π -periodic).

$$\text{HT}(\theta, \rho) = \text{HT}(\theta + 2\pi, \rho) \quad (3.5)$$

- the two points in \mathcal{P} : (θ, ρ) and $(\theta + \pi, -\rho)$ represent the same line:

$$\text{HT}(\theta, \rho) = \text{HT}(\theta + \pi, -\rho) \quad (3.6)$$

It follows that for representing the set of lines in \mathbb{R}^2 only a subset of \mathcal{P} is needed: two possible choices are $\mathcal{P} = \Theta \times \mathcal{R} = [0, \pi) \times \mathbb{R}$ and $\Theta \times \mathcal{R} = [0, 2\pi) \times \mathbb{R}^+$; both can be used depending on the application.

Rigid transforms It is interesting to analyze the behavior of the HT with respect to rigid transformations of the input space. Let $i(\mathbf{s})$ and $i'(\mathbf{s})$ be two inputs such that

$$i'(\mathbf{s}) = i(R_\phi \cdot \mathbf{s} + T) \quad (3.7)$$

with R_ϕ and T denoting a rigid 2D transformation, and let $\text{HT}(\theta, \rho)$ and $\text{HT}'(\theta, \rho)$ be their transforms, then, if we remain within the boundaries of \mathcal{P} , we have:

$$\text{HT}'(\theta, \rho) = \text{HT}(\theta + \phi, \rho + (\cos \theta \quad \sin \theta) T) \quad (3.8)$$

Note that:

- If $T = 0$, then $\text{HT}'(\theta, \rho) = \text{HT}(\theta + \phi, \rho)$, that is, the \mathcal{P} space translates in the θ direction (Fig. 3.2).

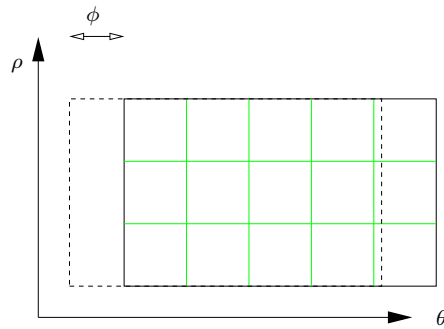


Figure 3.2: Deformation if $T = 0$.

- If $\phi = 0$, then the \mathcal{P} space bends only in the ρ direction, because $(\cos \theta \quad \sin \theta) T$, the displacement for the θ column, does not depend on ρ (Fig. 3.3).

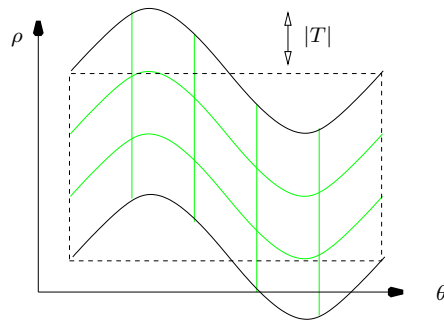


Figure 3.3: Deformation if $\phi = 0$.

3.3 The Discrete Hough Transform

The HT as defined by (3.3) is a mathematical object which will serve us well for proving theoretical results, but the algorithmic implementation of HT uses the *Discrete Hough Transform* (DHT), which is quite easy and efficient to compute.

A precise analytical definition of the DHT can be given using definition (3.1). Given the family of sets for the HT (Fig. 3.4):

$$\mathcal{D}_{(\theta,\rho)} = \{(x, y) \mid \rho \leq x \cos \theta + y \sin \theta < \rho + \Delta\rho\} \quad (3.9)$$

and a sampling $\Delta\rho, \Delta\theta$ of \mathcal{P} , the parameter space is discrete and finite once appropriate bounds are chosen (note that there's no harm in considering \mathcal{P} limited; in fact if the support of $i(\mathbf{s})$ is limited, also the support of $\text{HT}(\theta, \rho)$ is limited). We call $\text{HT}\{\mathcal{D}\}$ the *Discrete Hough Transform* (DHT).

A fast way to compute the DHT [16] is to set up a bi-dimensional array of accumulators, whose size depends on the discretization for ρ and θ , that represents the whole \mathcal{P} space. The accumulators are initialized at 0, then for each p_j in the input space, the curve

$$\rho = (\cos \theta \ \sin \theta) p_j \quad (3.10)$$

is drawn on the buffer, increasing the accumulators by one.

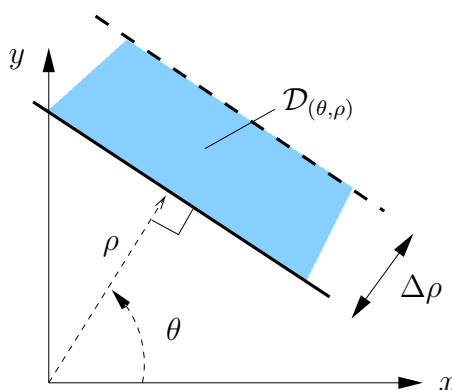


Figure 3.4: Parameterization for the line DHT.

Expressiveness of the transform Fig. 3.5 shows that the loss of information when performing the DHT is small: an anti-transform can recover most of the information contained in the original input. In particular it can represent curved lines.

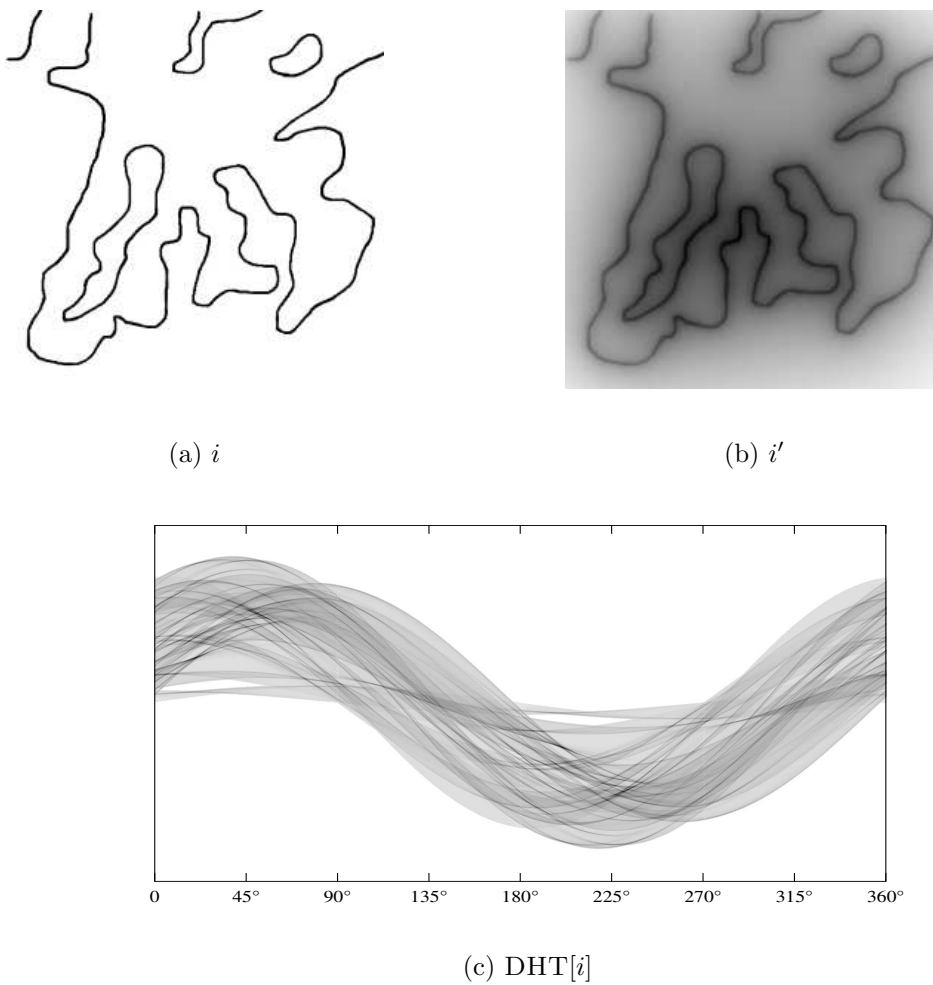


Figure 3.5: DHT introduces small loss of information. Here $i' = \text{DHT}^{-1}[\text{DHT}[i]]$.

3.4 The Hough Spectrum

We introduce an invariance result based on the definition of a "spectrum" function.

Definition of Hough Spectrum 1. Let g be a translation invariant functional (that is, $f'(\tau) = f(\tau + a) \Rightarrow g[f] = g[f']$). We define the Hough Spectrum as

$$\text{HS}_g[i](\theta) := g\left[\text{HT}[i](\theta, \cdot)\right] \quad (3.11)$$

This is the main property of our interest:

Spectrum invariance 2. Let $\mathcal{R} = \mathbb{R}$ and

$$i'(\mathbf{s}) = i(R_\phi \cdot \mathbf{s} + T) \quad (3.12)$$

Then

$$\text{HS}_g[i](\theta) = \text{HS}_g[i'](\theta + \phi) \quad (3.13)$$

Proof. It follows from (3.8). \square

Note that

- If $\phi = 0$, the spectrum does not change, regardless of T .
- if $\phi \neq 0$, the spectrum is (circularly) shifted by ϕ , regardless of T .

Choice of the g functional A broad class of functionals can be chosen as g : a norm, the entropy function, number of peaks, etc. The straightforward choice for a translation invariant functional is the simple integration:

$$g[f] = \int_{\mathbb{R}} f(x) dx \quad (3.14)$$

however one can show that in this case $\text{HS}_g[i](\theta) = \int i(\mathbf{s}) ds = \text{const}$, so it wouldn't be of much use.

The choice of a functional would be influenced by the characteristics of the input data. We have used the energy of the sequence in the discrete domain:

$$g[f] = \sum_i f_i^2 \quad (3.15)$$

Apart from its simplicity, it works well for our sensor data and it has some advantages that we will explain in Chapter 4 (4.2).

Unfortunately, property (3.13) doesn't hold exactly when considering the DHT, because for a single column $\text{DHT}(\theta, \cdot)$ the translational invariance holds exactly only if the quantity $(\cos \theta \quad \sin \theta)T$ is a multiple of $\Delta\rho$; however, the subsequent steps of our algorithm are robust to this small variation of $g[\text{DHT}'(\theta, \cdot)]$.

The following are minor properties of the spectrum function that we will refer to later.

Spectrum period 3. *Let g be translation invariant and mirroring invariant (that is, $f'(\tau) = f(-\tau) \Rightarrow g[f] = g[f']$) and let $\mathcal{R} = \mathbb{R}$. Then $\text{HS}_g[i](\theta)$ is π -periodic.*

Proof. It follows from (3.6). □

Spectrum invariance to small translations 4. *Let $i'(\mathbf{s}) = i(R_\phi \cdot \mathbf{s} + T)$, $\mathcal{R} = \mathbb{R}^+$, g as in 3.14. Then if $|T|$ is small*

$$\text{HS}_g[i](\theta) \simeq \text{HS}_g[i'](\theta + \phi) \quad (3.16)$$

Proof. Not very interesting. The difference between the terms can be bounded by $|T|$ and, as intuitions shows, it goes to 0 when $|T| = 0$. □

Chapter 4

The Hough Scan Matcher

Overview of the algorithm:

1. The DHT and DHS are computed for reference and sensor data.
2. Local maxima of the spectra cross-correlation are used to produce hypotheses on ϕ .
3. For each hypothesis ϕ , linear constraints for T are produced by correlating columns of the DHT.
4. Two alternatives:
 - (a) The linear constraints are combined to produce multiple hypotheses for T and solutions are ordered with a likelihood function.
 - (b) The results of the correlations are accumulated in a buffer producing a dense output.

4.1 Spectra correlation and ϕ estimation

Because of (3.13), we can estimate ϕ by a cross correlation of $\text{DHS}^S(\theta)$ and $\text{DHS}^R(\theta)$. Local maxima of the correlation are found and ordered, then for each maximum, beginning with the greatest and up to an arbitrary number, a new hypothesis for ϕ is generated.

Let $\text{DHS}^S(\theta)$ and $\text{DHS}^R(\theta)$ be the discrete spectra of sensor and reference data. Let Φ be the search domain for ϕ ; if one has an estimate ϕ_U of an upper bound for ϕ (for example from odometry data), then $\Phi = [-\phi_U, \phi_U]$ (local search), whereas $\Phi = [-\pi, \pi]$ for the uninformed case (global search). The correlation of the two spectra is defined by:

$$\text{corr}_{\text{DHS}}(\phi) = \sum_{\theta \in \Theta} \text{DHS}^S(\theta) \cdot \text{DHS}^R(\theta - \phi) \quad (4.1)$$

where $\phi \in \Phi$. The hypotheses for ϕ are given by $\{\phi_1, \dots, \phi_k\} = \text{localMaxOf}\{\text{corr}_{\text{DHS}}(\phi)\}^1$. An example of spectra correlation is shown in Fig. 4.1.

Note that if $\mathcal{R} = \mathbb{R}$, (4.1) is π -periodic (thm. 3 on page 27) therefore can be computed for half of the domain; on the other hand we get information only about the direction and not the heading of the robot.

If $|T|$ is small it is useful to use $\mathcal{R} = \mathbb{R}^+$ for which (4.1) is only 2π -periodic (thm. 4 on page 27), therefore giving directly heading information. This is useful for pose tracking.

ϕ hypotheses can be created directly regardless of T , but for large environments this would produce too many hypotheses; a large map should be divided in smaller patches and the matching process be carried out separately for each patch.

If an exact solution exists, that is, $i(s) = i'(R_\phi \cdot s + T)$ holds exactly for a certain ϕ, T , the correlation will have a global maximum for ϕ . Therefore the algorithm is complete: if an exact solution exists, it will be found. As the experiments show, it works even if the sensor data is but a little part of the reference data.

¹Here we denote by `localMaxOf` the extraction of the local maxima - an operation that in practice needs carefully chosen thresholds or smoothing.

4.2 T estimation

At this point we have solved half of the original problem; several 2D estimation algorithms could be plugged here after an hypothesis for ϕ exists, however we will propose one that exploits the already computed DHT. For the sake of notation clarity and without loss of generality, we will assume that $\phi = 0$ (this is equivalent to shifting the columns of either DHT^S or DHT^R by the previous computed ϕ).

Choose an arbitrary $\theta = \hat{\theta}$ and consider the columns $\text{DHT}(\hat{\theta}, \rho)$ for both the reference and sensor reading. Because of (3.8) and $\phi = 0$:

$$\text{DHT}^R(\hat{\theta}, \rho) = \text{DHT}^S(\hat{\theta}, \rho + (\cos \hat{\theta} \quad \sin \hat{\theta})T) \quad (4.2)$$

Again, by correlation of columns of DHT^R and DHT^S we can estimate $(\cos \hat{\theta} \quad \sin \hat{\theta})T = d(\hat{\theta})$, that is the projection of T in the $\hat{\theta}$ direction. If one has an estimate $|T|_U$ of the upper bound for $|T|$, then the search domain for $d(\hat{\theta})$ is $[-|T|_U, |T|_U]$.

By considering two different values for θ and finding the maximum of the correlation, we can build a linear system to determine T . Different pairs (θ_1, θ_2) will give different results for T . One could use the resulting T of each pair as a different hypothesis or combine more than two directions to get an over-constrained system to be resolved by least squares estimation:

$$i = 1 \dots n_c : \quad (\cos \theta_i \quad \sin \theta_i)T = d(\theta_i) \quad (4.3)$$

Choice of alignment directions It is still an open question how to choose the n_c directions for linear constraints.

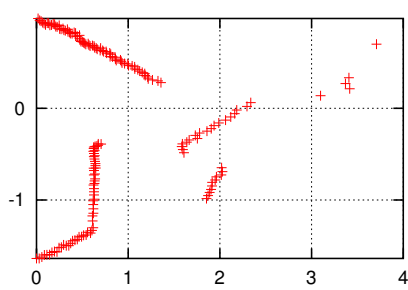
Of course not all directions can be chosen for performance reasons (at least in an on line setting). Even in the $n_c = 2$ case, there are different needs: to keep as independent as possible the observations $d(\theta_1)$ and $d(\theta_2)$ (perpendicular) and choose directions along which are well defined the local maxima of the correlation.

One solution is to choose the local maxima of the sensor DHS: $\{\theta_1, \dots, \theta_n\} = \text{localMaxOf}\{\text{DHS}^S(\phi)\}$. Because of our previous choice of the energy functional for g , we are choosing columns whose correlation will have an high expected energy therefore with clearly definite peaks.

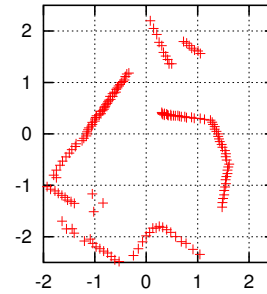
Dense approach Alternatively one can accumulate the results of each correlation and produce a dense output: see Fig. 4.2(b)-4.2(c). One could use up to every column, and approximate complex distributions (see Fig. 4.2(c)). In polygonal environments 2 to 4 directions are usually enough.

The under-constrained case There are situations when the matching problem is ill-posed, for example a robot traveling in a long corridor with limited horizon has essentially only one direction for aligning, and any other choice of alignment direction will cause erroneous results that will deteriorate the estimate of T .

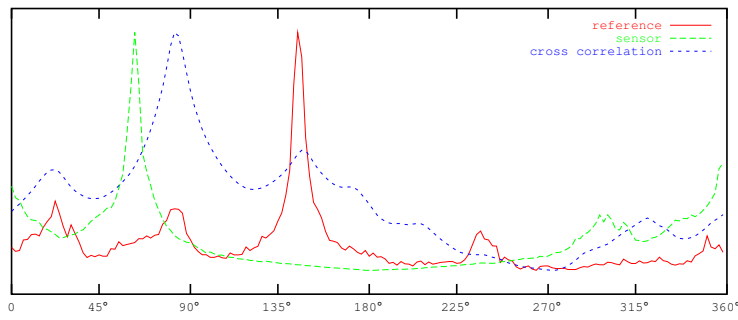
One solution for detecting such under-constrained cases is to linearly normalize the DHS in the $[0, 1]$ range and accept as alignment directions only those values over a certain thresholds, say over 0.5 the global maximum. If only one direction is used then an under determined system is produced and the conservative choice for pose tracking is to assume for T the minimum norm solution: $T = (\cos \theta_1 \quad \sin \theta_1)^t d(\theta_1)$.



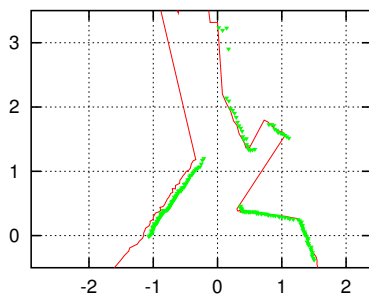
(a) Sensor data



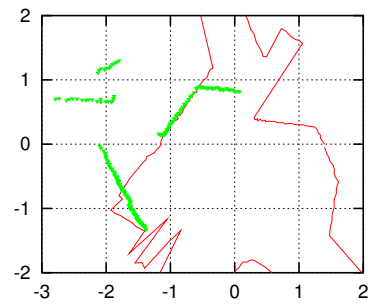
(b) Reference data



(c) Spectra comparison ($\mathcal{R} = \mathbb{R}^+$)



(d) First solution



(e) Second solution

Figure 4.1: Example of real-world input data. Final results: Fig. 4.1(d) is the “exact” match, apart from map inconsistencies, however also Fig. 4.1(e) is a plausible solution.

4.3 Complexity

Let

- T_{\max} and ϕ_{\max} be bounds for $|T|$ and $|\phi|$
- σ_ϕ and σ_T the required angular/linear discretization (i.e. resolution of the solution)
- S, M the number of points in sensor/reference
- r_S, r_M the minimum radii containing all points for sensor/reference (i.e. range of the scans)
- n_ϕ the number of ϕ hypotheses tracked
- n_c the number of linear correlations.

In this case the DHT buffer dimensions are $|\Theta| \propto 2\pi\sigma_\phi$, $|\mathcal{R}| = r_M\sigma_T$ therefore space occupation is $O(\sigma_\phi r_M \sigma_T)$.

Building the HT and HS takes $O(S\sigma_\phi)$ and $O(M\sigma_\phi)$ and finding ϕ costs $O(\phi_{\max}\sigma_\phi \cdot \sigma_\phi)$. The column correlation is done $n_\phi n_c$ times and each costs $(T_{\max}\sigma_T)(\sigma_T r_S)$. Total cost is:

$$O(S\sigma_\phi + M\sigma_\phi + \phi_{\max}\sigma_\phi^2 + n_\phi n_c T_{\max} r_S \sigma_T^2) \quad (4.4)$$

When doing position tracking we are interested in increasing the resolution of the sensor and of the solution. In this case the cost grows as $S\sigma_\phi + \sigma_\phi^2 + \sigma_T^2$, that is, linear in the number of data points and quadratic on each of the discretizations. Note that in this case the search space grows as $\sigma_\phi \sigma_T^2$.

For global localization we are interested in increasing the environment size. In this case the cost of our algorithm grows as $n_\phi n_c T_{\max}$, while the search space grows as T_{\max}^2 . However, note that in this case also n_ϕ grows and as mentioned before one should divide the map in patches before applying HSM.

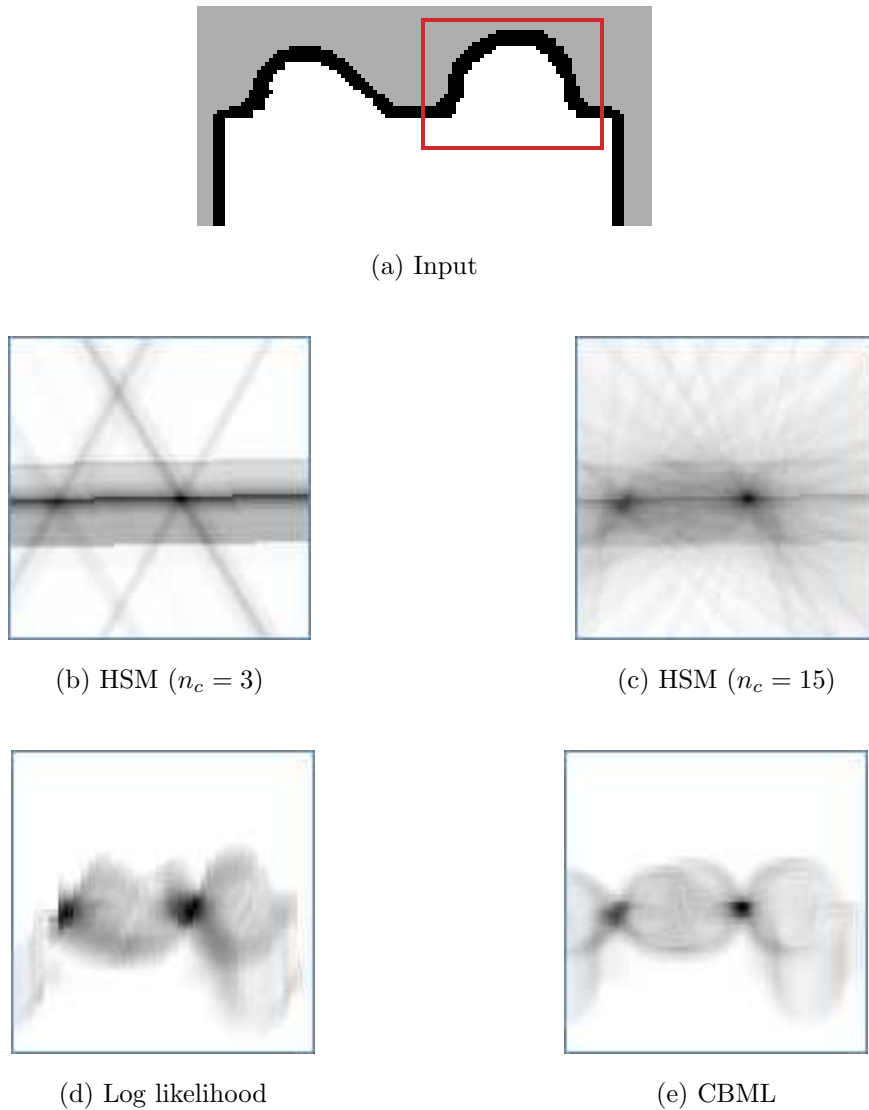


Figure 4.2: Fig. 4.2(a) shows the input data: the red mark encloses the part that it is used as sensor data. These pictures show the second part of the algorithm, T estimation with dense buffer, after that HSM has (correctly) found as a solution $\phi = 0$. Fig. 4.2(b) shows the distribution resulting from correlating 3 pairs of columns of the HT (corresponding to the three peaks of the HS), while 4.2(c) comes from uniformly sampling 15 columns from the HT. Note that the 3 directions used for the correlation contain the most information, enough to discriminate the two distribution peaks. For comparison, Fig. 4.2(d) shows the output of the log likelihood for $\phi = 0$, likewise fig. 4.2(e) shows the output of CBML [15].

Chapter 5

Experiments

We present here a simulated example that shows how the method performs for data completely without line features.

Then we present the results from systematic simulated experiments performed in realistic environments.

5.1 A somewhat limit case

We simulated the output of a 360° range finder with 2° resolution and high noise (this could be the output of an edge-extracting pipeline of an omnidirectional camera) placed in an ellipsoidal environment (eccentricity = 1.25). Two scans that differ both for orientation and displacement are taken (Fig. 5.1).

As Fig. 5.2(c) shows, our method succeeds in aligning the two scans: the two symmetric hypotheses for ϕ are clearly showed in the correlation. In this case it is evident that the method works by matching *distribution of features* instead of single features.

5.2 Systematic experiments

To gather significant statistical data the experiments have been conducted with a simulator using scan data available through the Inter-

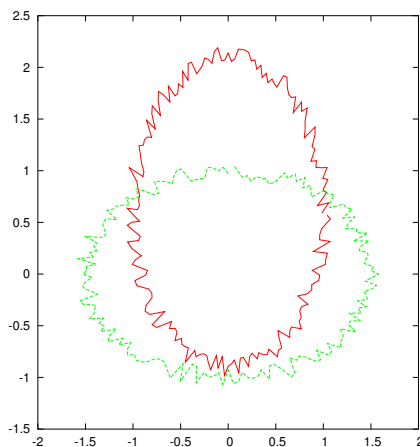
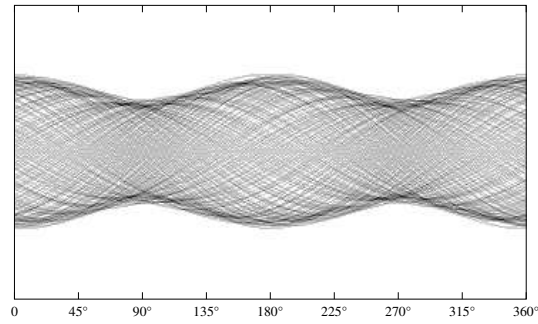


Figure 5.1: Two simulated scans from a noisy range finder in an ellipsoidal environment.

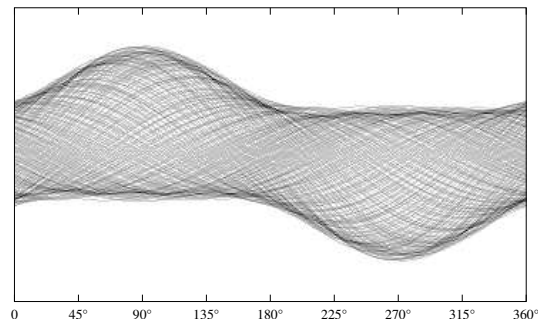
net [18]. Given a map, a reference position and a scan position are randomly sampled from the environment with a fixed displacement. The raw ray-tracing result from the first position is assumed as the reference scan. The sensor scan is simulated using four different sensor models (see Fig. 5.5):

1. *Ideal-180*, a range finder with 180° range and 1° resolution and low noise (similar to *Sick LMS*);
2. *Disc.Noise-180*, a range finder with high quantization noise (similar to *Sick PLS*);
3. *Gaus.Noise-180*, a range finder affected by high Gaussian noise with variance growing with distance (the *Hokuyo PB9-01* as described in [19]);
4. *Syst.Noise-360*, a range finder subject to systematic error but with large field of view (e.g., the edge-extracting pipeline of an omnidirectional camera).

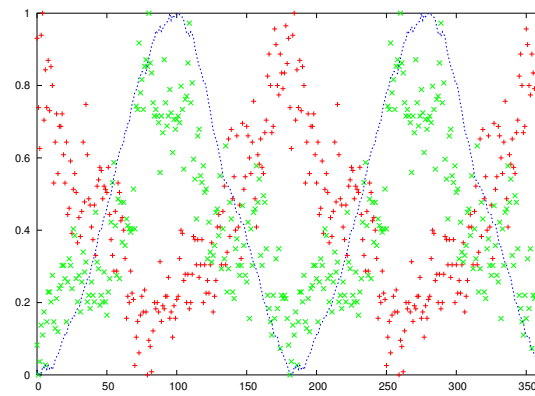
We report the results obtained with two exemplar environments: an office-like environment (a large hospital) and an unstructured environ-



(a) HT for the centered scan



(b) HT for the displaced scan.



(c) Spectra comparison

Figure 5.2: In the last picture, the normalized spectra (red, green) are shown along with their cross correlation (blue). The peaks of the correlation are the two hypotheses for ϕ ($\pm 90^\circ$).

ment (hand-drawn map of a cave)¹.

In these experiments we are mainly interested in evaluating the ability of the system to deal with large uncertainty in the prior pose. Therefore, we evaluate different situations in which the sensor scan is displaced by the reference position by 0, 0.5, and 1 m, respectively. Note that the *average beam reading distance* in the data sets considered is about 1.6 m, therefore the value 1 m is quite high. Furthermore, we assumed no prior information about scan orientation (in fact, performing a 360° matching).

Because this method performs a global search over orientation ϕ , the error distribution $|e(\phi)|$ is multi-modal (due to the presence of orthogonal lines in the hospital environment), and the error distribution $|e(T)|$ is composed of a single definite peak with super-imposed noise. Two typical error distributions are shown in Fig. 5.4. In other words, error distributions are formed by a Normal distribution and additional noise and in order to make a more precise error analysis we have taken into consideration the following parameters for each error distribution: 1) the amount of samples in the principal mode of the distribution, that is a measure of the reliability of the method; 2) the mean of the principal mode, that shows its precision.

Results for the two data sets are reported in Table 5.6. We computed the above mentioned measures for both orientation ϕ and translation T for three scan-reference displacements (0, 0.5, 1 m) and for each of the sensor model considered. Error analysis shows that in a polygonal environment high precision and reliability can be obtained for ϕ and T , and that the method is able to tolerate both various sensor noises and large input differences. In an unstructured environment, we still have good results for limited scan displacements, while performance degrade with increasing distance between current and reference scans.

¹Data sets obtained from the Robotics Data Set Repository (Radish) (radish.sourceforge.net - thanks to Richard Vaughan). Map of the Victoria fossil cave in Naracoorte (Australia), courtesy of the Cave Exploration Group of South Australia (www.cegsa.org.au).



Figure 5.3: Part of the cave map used in the experiments.

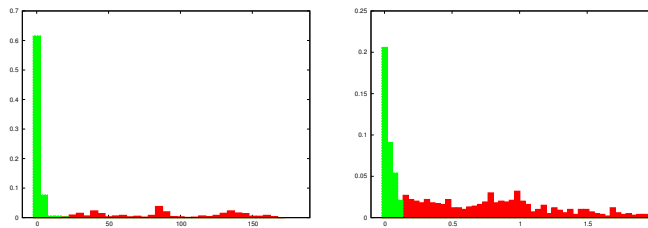


Figure 5.4: Shape of error distributions in worst-case result (Gaus.Noise-160, cave, $d = 1m$) with principal mode highlighted. The $|e(\phi)|$ distribution is multi-modal (left), the $|e(T)|$ distribution has a peak and uniform noise (right).

It is also interesting to notice that, with increasing of input differences, a sensor with large field of view and higher noise has better performance than one with smaller field of view and accurate measures.

Sensor	FOV	res.	quant.	dist.	σ
Ideal-180	180°	1°	0.01m	1	$0.01m$
Disc.Noise-180	180°	1°	0.07m	1	$0.03m$
Gaus.Noise-160	160°	1.78°	0.005m	1	$0.01 \cdot d^2 - 0.0017 \cdot d + 0.0075m$
Syst.Noise-360	300°	4°	0.01m	1.15	$0.01 \cdot d$

Figure 5.5: Parameters for sensor models. If d is the "real" distance to the obstacle then the perceived distance \hat{d} is given by $\hat{d} = \mathcal{N}(\text{dist} \cdot d, \sigma^2)$ with σ computed as in the table.

	$ \phi $ peak mass	$ \phi $ peak avg.	$ T $ peak mass	$ T $ peak avg.	$ \phi $ peak mass	$ \phi $ peak avg.	$ T $ peak mass	$ T $ peak avg.
Ideal-180	98%	$< 1^\circ$	97%	$< 1cm$	90%	$< 1^\circ$	89%	$< 1cm$
Disc.Noise-180	97%	$< 1^\circ$	93%	$4cm$	72%	4°	71%	$4cm$
Gaus.Noise-160	94%	$< 1^\circ$	82%	$2cm$	80%	2°	77%	$2cm$
Syst.Noise-360	99%	$< 1^\circ$	98%	$5cm$	97%	1°	97%	$3cm$

(a) Hospital, $d = 0$ (b) Cave, $d = 0$

Ideal-180	96%	$< 1^\circ$	86%	$1cm$	82%	$< 1^\circ$	70%	$18cm$
Disc.Noise-180	96%	$< 1^\circ$	88%	$5cm$	67%	4°	57%	$8cm$
Gaus.Noise-160	95%	$< 1^\circ$	86%	$3cm$	77%	2°	74%	$10cm$
Syst.Noise-360	98%	$< 1^\circ$	96%	$8cm$	87%	2°	84%	$7cm$

(c) Hospital, $d = 0.5$ (d) Cave, $d = 0.5$

Ideal-180	91%	$< 1^\circ$	72%	$2cm$	74%	$< 1^\circ$	28%	$8cm$
Disc.Noise-180	91%	$< 1^\circ$	71%	$6cm$	58%	4°	40%	$11cm$
Gaus.Noise-160	89%	$< 1^\circ$	68%	$3cm$	70%	2°	47%	$9cm$
Syst.Noise-360	95%	$< 1^\circ$	77%	$10cm$	72%	2°	54%	$10cm$

(e) Hospital, $d = 1$ (f) Cave, $d = 1$

Figure 5.6: Tables show statistical information about the distribution of errors on heading ($|e(\phi)|$) and position ($|e(T)|$) for the two environments (office-like and cave-like), four sensor models (see fig. 5.5) and three values for displacement of sensor and reference scan ($d = 0, 0.5, 1$ m), with unknown orientation (1000 iterations performed for each combination). The faster discrete-constraint HSM has been used with these parameters: $\Delta\theta = 0.5^\circ$, $\Delta\rho = 4cm$ for Gaus.Noise-160, $\Delta\theta = 0.5^\circ$, $\Delta\rho = 2cm$ for the others. Shown in the columns: i) ϕ error samples in the first mode; ii) average of these samples iii) T error samples in the first mode; iv) average of these samples.

Chapter 6

Conclusions and future work

In this paper we have proposed a scan matching algorithm (HSM) that works in the Hough domain. The key concept of HSM is to transfer the matching problem to another domain through the use of an information-preserving transformation. The invariance properties of the Hough domain allow to decouple the problem in 1) finding the orientation error, and 2) finding the translation error once the orientation error is known. HSM does not rely on features extraction, but it matches dense data (that can be interpreted as "features distributions") in a different convenient parameter space; this allows to match non-linear surfaces and makes the process robust to noise.

Derivation of the algorithm was done in a continuous input space $i(s)$. This could allow in the future to relate in an analytical way the matching error to the sensor model, that is to express the uncertainty of the matching to the noise parameters of the sensor's model.

The employment of HSM in large scale localization must be investigated further. CBML [15] has complexity $O(S \cdot \sigma_T^2 T_{\max}^2 \cdot \sigma_\phi \phi_{\max})$ therefore neither HSM nor CBML dominates the other. Moreover both methods are based on a simple operation (bi-dimensional correlation for CBML, unidimensional correlation for HSM) and the use of processor-specific extensions can considerably change the running time. A proper comparison should employ some kind of metrics on the resulting distribution, and

would probably result in a trade-off of speed (HSM) vs. thoroughness of the search (CBML).

In this paper we have dealt with the 2D scan matching, where the input space is bi-dimensional and the solution space is three-dimensional (ϕ, T_x, T_y) , effectively subdividing the original problem in simple uni-dimensional sub-problems (finding maxima of a cross-correlation). 2D scan matching is a low-dimensional problem when compared with the vast range of complex sensors and robot mobilities available nowadays: for example scan matching for a snake-like robot in an unstructured environment with a stereo camera would be a problem with 3D input space and 6 degrees of freedom solution. In the near future we intend to investigate an extension of HSM to higher dimensions, attempting to exploit and refine the ability of subdividing the search problem into low-dimensional problems.

Bibliography

- [1] A. Censi, L. Iocchi, and G. Grisetti. Scan matching in the hough domain. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'05)*, 2005.
- [2] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte carlo localization: Efficient position estimation for mobile robots. In *Proc. of the 16th National Conference on Artificial Intelligence (AAAI99)*, 1999.
- [3] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11:391–427, 1999.
- [4] J.S. Gutmann and K Konolige. Incremental mapping of large cyclic environments. In *Proceedings of the IEEE Interational Symposium on Computational Intelligenece in Robotics and Automation*, 2000.
- [5] D. Haehnel, W. Burgard, D. Fox, and S. Thrun. An highly efficient algorithm for generating maps of large scale cyclic environments from raw laser range measurements, 2003.
- [6] G. Grisetti, L. Iocchi, and D. Nardi. Global hough localization for mobile robots in polygonal environments. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA'2002)*, 2002.
- [7] G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and

- selective resampling. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2005.
- [8] J. S. Gutmann and C. Schlegel. AMOS: Comparison of scan matching approaches for self-localization in indoor environments. In *1st Euromicro Workshop on Advanced Mobile Robots (EUROBOT)*, 1996.
- [9] Kai Lingemann, Hartmut Surmann, Andreas Nüchter, and Joachim Hertzberg. Indoor and outdoor localization for fast mobile robots. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'04)*, pages 2185–2190, 2004.
- [10] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2d range scans. In *CVPR94*, pages 935–938, 1994.
- [11] M.A Greenspan and M Yurick. An approximate k-d tree search for efficient icp. In *Proceedings of the 4th International Conference on 3-D Digital Imaging and Modeling (3DIM03)*, 2003.
- [12] D. Hähnel, D. Schulz, and W. Burgard. Map building with mobile robots in populated environments. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2002.
- [13] P. Biber and W. Strasser. The normal distributions transform: a new approach to laser scan matching. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2003.
- [14] G. Weiß, C. Wetzler, and E. von Puttkamer. Keeping track of position and orientation of moving indoor systems by correlation of range-finder scans. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 1994.
- [15] K. Konolige and K. Chou. Markov localization using correlation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'99)*, 1999.

-
- [16] R. Duda and P. Hart. Use of the hough transformation to detect lines and curves in the pictures. *Communications of the ACM*, 15(1):11–15, 1972.
- [17] A. Kimura and T. Watanabe. An extension of the generalized hough transform to realize affine-invariant two-dimensional (2d) shape detection. In *Proceedings of the 16th International Conference on Pattern Recognition*, pages 65 – 69 vol.1, 2002.
- [18] Andrew Howard and Nicholas Roy. The robotics data set repository (radish) - <http://radish.sourceforge.net/>, 2003.
- [19] Heon-Hui Kim, Yun-Su Ha, and Gan-Gyoo Jin. A study on the environmental map building for a mobile robot using infrared range-finder sensors. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'03)*, pages 711–716, 2003.