Application of ϵ-testers algorithms under sketch and streaming calculation model in robot navigation

CARLOS RODRIGUEZ LUCATERO
Universidad Autónoma Metropolitana Campus Cuajimalpa
Department of Information Technology
Av. Constituyentes 1054, Col. Lomas Altas, Del. Miguel Hidalgo, Mexico D.F.
MEXICO
crodriguez@correo.cua.uam.mx

Abstract: The goal of this article is the application of efficient approximate testing methods for the robot tracking problem as well as for the building map problem . In the Databases field, one wish to classify uncertain data and to answer queries in an approximated way in the case that data sources are incoherent. Our interest is based on the fundamental problems in the randomized approximate testing algorithms. Concerning robot navigation we would like to apply efficient data sketching algorithms for testing the automata inferred on robot motion tracking problems as well as on maps sensorialy generated by robot exploration and building map process. In the case of the information generated by sensors, given that we have no complete access to the information, and given the inherent limitations on memory space, we will apply streaming algorithms for approximate efficiently the solution of computational problems that take place on robot navigation problems.

Key–Words: Robot navigation, Motion tracking, Exploration, Sketching and streaming algorithms, Testability

1 Introduction

The robot navigation deals with many aspect of the robot motion. The first one is the robot motion planning that has been largerly studied during the last twenty years. This problem can be roughly stated as the calculation of sequence of movements of a point on the free space of the configuration space and it has been proved to be solved in a efficient way for the case of a 2D when there is no uncertainty [2] and the 3D case as well as the 2D with moving obstacles have been shown to be computationally hard in [8]. For special 3D cases it has been shown in [7] that good approximations of the shortest path can be obtained in polynomial time. This methods work well quite well under simulation or well controlled conditions but unfourtunately they have a bad performance when they run in real robots due to the uncertainty caused by the movements and the sensorial information given by the odometers, US proximity sensors and vision cameras. The problem of finding a robust motion strategy with under motion deviations is computational complex [6] as well as under sensorial uncertainty with limited type of sensors [16]. This lend us to the second navigation aspect that has to do with sensorial information [29]. As it is well known the robot wheels slide while the robot moves and it cause to lose information about the real position of the robot. This position uncertainity due to the robot motion is accumulative so many different methods as Kalman filters, landmarks and triangulation methods, or more recently particle filters [3], to retreive his position can be used. So one option is the fusion of the odometric, visual and sonic information to retrieve the real position on the scene and reduce the inherent uncertainty produced by each information source. Another robot navigation problem that has interested many researchers in Computer Science as well as in robotics, for the last fifteen years till our days, is the robot exploration problem. This last problem consist on obtention of an online algorithm that makes the worst case distance-traversal in order to see all the visible points of the environment and to be able to build or learn a map of an unknown environment in a competitive way. Here competitive means that the ratio between the optimal traversal produced by an off-line algorithm and the exploration online algorithm is bounded by a constant value. The problem (the geometric off-line problem is *given a map find the shortest path that verifies it visually*) in his more simple setting is computationally very hard given that the off-line algorithm is related with Euclidean Traveler Salesman Problem that is known to be **NP-hard** so it must be solved approximately. It can be proved the impossibility of learning with a competitive radomized algorithm in the case of general environments but it has been proved in [28] that it can be obtained a competitive strategy in the case of arbitrary

polygonal room with a bounded number of polygonal obstacles. We can imagine how the complexity of the robot exploration problem can change if we add to it the sensorial uncertainties. Nevertheless, if we want to apply this solutions on real robots, we cannot avoid this issue. So we have to device robust solutions for robot navigation problems in presence of uncertainty. Because of that we have to interact with the world by means of sensors and try to minimize their inherent uncertainties by fussion of the sensorial information. Another very interesting problem that have been interested many researchers in the computer science, robot motion planning and multi-robot field is the robot motion tracking problem. Roughly speaking the problem consist on putting in two robots in a scene with obtacles or corridors and one of the robots being the *observer* and the other the *target*. The goal is to find efficient algorithms for track of the *target* robot. Some very interesting methods based on sensorial and geometrical have been proposed in the last twelve years [12] [14]. More recently other algorithms have been proposed for solving the same problem under game theory approach [17] [18] as well as under automata learning approach [19] [20]. As it will be explained in further sections, this methods can be related with the notions of ϵ-testability and streaming algorithms as part of the methods that are proposed in the present research work. The data flow provided by the sensors produce similar problems to those that can be found on the databases. The robot should make the fusion of the information sources to determine his motion strategy. Some sources, called bags, allow the robot to self locate geometrically or in his state graph. While the robot executes his strategy, it is subject to movement uncertainties and then should find robust strategies for such uncertainty source. The goal is to achieve the robustness integrating the data flow of the captors to the strategies. We consider the classical form of simple Markovian strategies. In the simplest version, a Markov chain, MDP, is a graph where all the states are distinguishable and the edges are labeled by actions $L_1, L_1, \ldots, L_p$. If the states are known only by his coloration in k colors $C_1, C_2, \ldots, C_k$ two states having the same coloration are undistinguishable and in this case we are talking about POMDP (Partially Observed Markov Decision Process) see [13]. The robustness problem is fundamental in the information and communication domain given that it introduce a physical element, the lack of precision and the uncertain aspects that impose new scientific problems. The classic computational techniques are often not very robust to data errors and the good notions of approximation are frequently hard to pose. The relation between approximation and computational complexity is a recent area that have allowed to clarify some computer science issues since ten years ago. The error correcting codes as well as information theory fields have been interested since Shannon in studying the lack of precision and the algorithmic techniques that allow to transmit information in a noisy channel. It is only till our days that the fundamental relations between computational complexity and correcting error codes have emerged. The property testing notion ([23] [24]) introduced in [15] [21] is one of the bases for our research. This notion is related with computational learning. The proposal is to study fundamental problems and to show how to apply to the robotics field some results obtained in the domain of databases on the Web.

2 Context and related work

Fifteen years ago some researchers on robot exploration and map building on unknown environments were trying to investigate the exploration and building map problem with continuous relocalization of the robot in an integrated maner. In order to add information to a map in the building map process, it is necessary for the robot to know his real position. In that time a frequently used method for estimating the robot position in absence of a map was the dead reckoning, but given that the more the robot moves the more the robot loses his real position because of the sliding of the wheels, then this method becomes quickly useless for exploration purposes. The heurístic used in some of these investigations consisted in exploring the frontiers or boundaries between open space and uncharted territory for maximizing the gain of new information about the world. The way of representing the space is by means of evidence grids where each cell has a probability of being free or being occupied by an obstacle. This information was updated based on the sensorial information in such a way that a cell can have one of the three following states *free*, *occupied* and *unknown*. The evidence grids were constructed using the lasser sonar information of the robot and the frontiers were detected with edge detection information given by the cameras. Errors on the position of the robot were minimized by the continuous localization process [3]. Later some other researchers developped probabilistic methods for simultaneous localization and mapping methods and proposed a spatial semantic hierarchy (SSH) representation of space to factor spatial uncertainty into distinct components and control each of them with distinct methods: movement uncertainty is controlled by feedback-driven motion control laws and pose uncertainty is controlled in basic SSH by hill-climbing to distinctive states and in the hybrid SSH by incremental localization within

the local perceptual map. In the hybrid SSH mapping method a local perceptual map is used for local motion planning and obstacle avoidance using metrical representation for small-scale and topological representation is used for the large-scale space representation [11]. Conceptually the topological map-builder mantains a tree whose nodes are pairs $\langle M, x\rangle$ where M is a topological map and x is a distinctive state within M that represents the current position of the robot. In the same research vein in [29] the author asociates the topography of the investigated environement with a Riemannian differentiable manifold, related to which the access ways from one point to another have to be chosen among the geodesics obtained frome the geometrical description of the robotic scene. In the same paper the authors divide the geometrical models in two big catergories, one with singularities and the other without them. Additionally they propose some guidelines on the implementation of those models taking into account the operating capacity of the computer used for the task. Concerning the robot motion tracking problem, the agents observe the actions taken by the other agents, and try to predict the behaviour of them and react in the best possible way. The fundamental aspect of the robot tracking problem is that the agents have a bounded computational power which implies that the game can achieve a Nash equilibrium that causes a colaborative behaviour on the agents involved in the game if the *prisoner's dilema repeated game* was played by automatas having les than exponential number of states [22]. Because of that it has been proposed in [18] to pose the robot tracking problem as a repeated game and to model the robots as automata agents. By other side many advances have been recently taking place in the approximation of several classical combinatorial problems on strings in the context of *Property Testing* [25] inspired on the notion of *Self-Testing* [4] [5] [15]. What has been shown in [25] is that, based on a statistical embedding of words, and constructing a tolerant tester for the equality of two words, it is possible to obtain an approximate normalized distance algorithm whose complexity does not depend on the size of the string. In the same paper [25] the embedding is extended to languages and get a geometrical approximate description of regular languages consisting in a finite union of polytopes. As an application of that its is obtained a new tester for regular languages whose complexity does not depend on the automaton. Based on the geometrical description just mentioned it is obtained an deterministic polynomial equivalent-tester for regular languages for a fixed threshold distance. Computing *edit distance* between two words is an important subproblem of many applications like text-processing, genomics and web searching. Another field in Computer Science that where important advances recently have been taking place is that of embeddings of sequences [9]. The sequences are fundamental objects in computer science because they can represent vectors, strings, sets and permutations. For beeing able to measure their similarity a distance among sequences is needed. Sometimes the sequences to be compared are very large so it is convenient to map or embedd them in a different space where the distance in that space is an approximation of the distance in the original space. Many embeddings are computable under the streaming model where the data is too large to store in memory, and has to be processed as and when it arrives piece by piece.

3 Notion definiton and Notation

One fundamental notion introduced in the approximation of combinatorial objects context is the *edit-distance*. This concept can be defined as follows:

Definition 1 *The* **edit distance** *between two words is the minimal number of character substitutions to transform one word into the other. Then two words of size n are ϵ-far if they are at distance greater than ϵn.*

Another very important concept is the *property testing.* The *property testing* notion introduced in the context of program testing is one of the foundations of our research. If **K** is a class of finite structures and *P* is a property over **K**, we wish to find a **Tester**, in other words, given a structure *U* of **K**:

- It can be that *U* satisfy *P*.
- It can be that *U* is ϵ-far from *P*, that means, that the minimal distance between *U* and *U'* that satisfy *P* is greater than .
- The randomized algorithm runs in $O(\epsilon)$ time independently of n, where n represent, the size of the structure *U*.

Formally an ϵ-tester can be defined as follows.

Definition 2 *An ϵ-tester for a class $K_0 \subseteq K$ is randomized algorithm which takes a structure U_n of size n as input and decides if $U_n \in K_0$ or if the probability that U_n is ϵ-far from K_0 is large. A class K_0 is testable if for every sufficiently small ϵ there exists an ϵ-tester for K_0 whose time complexity is in $O(f(\epsilon))$, i.e. independent of n*

For instance, if **K** is the class of graphs and *P* is a property of being colorable, it is wanted to decide if a graph *U* of size n is 3-colorable or ϵ-far of being 3-colorable, i.e. the Hamming distance between

U and U' is greater than $\epsilon \cdot n^2$. If **K** is the class of binary strings and *P* is a regular property (defined by an automata), it is wished to decide if a word U of size n is accepted by the automata or it is ϵ-far from being accepted, i.e., the *Edition distance* between U and U' is greater than $\epsilon \cdot n$. In both cases, it exists a **tester**, that is, an algorithm in that case take constant time, that depends only on and that decide the proximity of these properties. In the same way it can be obtained a **corrector** that in the case that U does not satisfy *P* and that U is not ϵ-far, finds a structure U' that satisfy *P*. The existence of **testers** allow us to approximate efficiently a big number of combinatorial problems for some privileged distances. As an example, we can estimate the distance of two words of size n by means of the *Edition distance* with shift, we mean, when it is authorized the shift of a sub-word of arbitrary size in one step. To obtain the distance it is enough to randomly sample the sub-words between two words, to observe the statistics of the random sub-words and to compare with the L_1 norm. In a general setting, it is possible to define distances between automata and to quickly test if two automata are near knowing that the exact problem is NEXPTIME hard. By other side, an important concept that is very important in the context of sequence embeddings is the notion of *sketch*. A *sketch* algorithm for *edit distance* consit of two compression procedures, that produce a finger print or *sketch* from each input string, and a reconstrucction procedure that uses the sketches for approximating the *edit distance* between the to strings. A sketch model of computation can be described informally as a model where given an object x a shorter *sketch* x can be made so that compairing to *sketches* allow a function of the original objects to be approximated. Normally the function to be approximated is the distance. This allow efficient solutions of the next problems:

- Fast computation of short *sketches* in a variety of computing models, wich allow sequences to be comapred in constant time and spaces non depending on the size of the original sequences.
- Approximate nearest neighbor and clustering problems faster than the exact solutions.
- Algorithms to find approximate occurrences of pattern sequences in long text sequences in linear time.
- Efficient communication schemes to approximate the distance between, and exchange, sequences in close to the optimal amount of communication.

Definition 3 *A distance sketch function $sk(a, r)$ with parameters ϵ, δ has the property that for a distance $d(\cdot, \cdot)$, a specified deterministic function f outputs a random variable $f(sk(a, r), sk(b, r))$ so that*

$$\begin{aligned}(1-\epsilon)d(f(sk(a,r),sk(b,r)))\\ \leq f(sk(a,r),sk(b,r))\\ \leq (1+\epsilon)d(f(sk(a,r),sk(b,r)))\end{aligned}$$

for any pairs of points a and b with probability $1-\delta$ taken over small choices of a small seed r chosen uniformly at random

The sketching model assumes complete access to a part of the input. An alternate model is the streaming model, in which the computation has limited access to the whole data. In that model the data arrive as a stream but the space for storage for keeping the information is limited.

4 Applications to robot navigation problems

As I mentioned in 2 one automata model based aproach for solving the robot motion tracking has been proposed in [19]. The problem consited in building a model in each robot of the navigation behaviour of the other robot under the assumption that both robots, target and observer, were following an automata behaviour. Once the approximate behaviour automata has been obtained the question that arises is, *how can be measured the compliance of this automata with automata followed by the target robot ?*. Stated otherwise *How can be tested the automata of the target robot es equivalent to the one obtained by the observer robot ?* Is exactly in that context that the property testing algorithms can be applied for testing the equivalence automata in a computationally efficient way. It is well known that the problem of determining equivalence between automatas is hard computationally as was mentioned in 3. The map learning can be as well formulated as an automata with stochastic output inferring problem [10]. It can be usefull to compare the real automata describing the map of the environment and the information inferred by the sensorial information. This can be reduced to the equivalence automata problem, and as a consequence, an approximate property testing algorithm can be applied. In [21] can be found non obvious relations between property testing and learnability. As can be noted testing mimimics the standar frameworks of learning theory. In both cases one given access to an unknown *target* function. However there are important differences between testing and learning. In the case of a learning algorithm the goal is to find an approximation of a function $f \in K_0$, whereas in the case of testing the goal is to test that $f \in K_0$. Apparently it is

harder to learn a property than to test it. [21] it shown that there are some functions class which are harder to test than to learn provided that $NP \not\subset BPP$. In [21] when they speak about the complexity of random testing algorithms they are talking about query complexity (number test over the input) as well as time complexity (number of steps) and hey show there that both types of complexities depend polynomially only on ϵ not on n for some properties on graphs as colorability, clique, cut and bisection. Their definition of property testing is inspired on the PAC-learning model [27], so there it is considered de case of testers that take randomly chosen instances with arbitrarly distribution instead of querying. Taking into account the progress on property testing mentioned in 3, the results that will be defined further can be applied to the problem of testing how well the inferred automata of the *robot observer* in the robot motion tracking problem solved in [19], fits the behaviour automata followed by the *target robot*. The same results can be applied to measure how much the automata obtained by explorations fits the automata that describes the space explored by the robot. Roughly speaking, the equivalence ϵ-tester for regular languages obtained in [26], makes a statistical embedding of regular languages to a vectorial statistical space which is an approximate geometrical description of regular languages as a finite union of polytopes. That embedding enables to approximate the *edit distance* of the original space by the ϵ-tester under a *sketch* calculation model. The automata is only required in a preprocessing step, so the ϵ-tester does not depend on the number of states of the automata. Before stating the results some specific notions must be defined

Definition 4 Block statistics. *Let w and w' two word in Σ each one of length n such that k dived n. Let $\epsilon = \frac{1}{k}$. The statitistics of block letters of w denoted as $b - stat(w)$ is a vector of dimension $|\Sigma|^k$ such that its u coordinate for $u \in \Sigma^k$ (Σ^k is called the block alphabet and its elements are the block letters) satisfies $b - stat(w)[u] \stackrel{def}{=} Pr_{j=1,\ldots,n/k}[w[j]_b = u]$ Then $b - sta(w)$ is called the block statistics of w*

An convenient way to define blck statistics is to use the underlying distribution of word over Σ of size k that is on block letter on Σ^k. Then a uniform distribution on block letters $w[1]_b, w[2]_b, \ldots, w[\frac{n}{k}]_b$ of is the block distribution of w. Let X be a random vector of size $|\Sigma|^k$ where all the coordinates are 0 except its u-coordinate which is 1, where u is the index of the random word of size k that was chosen according to the block distribution of w. Then the expectation of X satisfies $E(X) = b - stat(w)$. The *edit distance* with moves between two word $w, w' \in \Sigma$ denoted as $dist(w, w')$ is the mininimal number of elementary operations on w to obtain w'. A class $K_0 \in K$ is testable if for every $\epsilon > 0$, there exists an ϵ-tester whose time complexity depends only on ϵ.

Definition 5 *Let $\epsilon \geq 0$. Let $K_1, K_2 \subseteq K$ two classes. K_1 is ϵ-contained in K_2 if every but finitely many structures of K_1 are ϵ-close to K_2. K_1 is ϵ-equivalent to K_2 if K_1 is ϵ-contained in K_2 and K_2 is ϵ-contained in K_1*

The following results that are the ones that we are going to apply in the robotics field in the present paper, are stated without demonstration but they can be consulted in [26].

Lemma 6 .-

$$dist(w, w') \leq \left(\frac{1}{2}\left|b - stat(w) - bstat(w')\right| + \epsilon\right) \times n$$

So we can embed a word w into its block statistics $b - stat(w) \in \Re^{|\Sigma|^{1/\epsilon}}$

Theorem 7 *For every real $\epsilon > 0$ and regular language L over a finite alphabet Σ there exists an ϵ-tester for L whose query complexity is $O(\frac{\lg|\Sigma|}{\epsilon^4})$ and time complexity $2^{|\Sigma|^{O(1/\epsilon)}}$*

Theorem 8 *There exists a deterministic algorithm T such that given two autimata A and B over a finite alphabet Σ with at most m states and a real $\epsilon > 0$, $T(A, B, \epsilon)$*

1. *accepts if A and B recognize the same language*
2. *rejects if A and B recognize languages that are not ϵ-equivalent. Moreover the time complexity of T is in $m^{|\Sigma|^{O(1/\epsilon)}}$*

Now based on 8 our main result can be stated as a theorem.

Theorem 9 *The level of approximability of the inferred behaviour automata of a* **target robot** *by an* **observer robot** *with respect to the real automata followed by the* **target robot** *in the motion tracking problem can be tested efficiently.*

Theorem 10 *The level of approximability of the sensorialy inferred automata of the* **environment** *by an* **explorator robot** *with respect to the real environment automata can be tested efficiently.*

5 Application of streaming algorithms on robot navigation problems

The starting premise of the sketching model is that we have complete access to one part of the input data. That is not the case when a robot is trying to build a map of the environemet based on the information gathered by their sensors. An alternative calculation model is the streaming model. Under this model the data arrives as a stream or predetermined sequence and the information can be stored in a limited amount of memory. Additionally we cannot backtrack over the information stream, but instead, each item must be processed in turn. Thus a stream is a sequence of n data items $z = (s_1, s_2, \ldots, s_n)$ that arrive sequentially and in that order. Sometimes, the nomber n of items is known in advance and some other times the last item s_{n+1} is used as an ending mark of the data stream. Many computer applications work with streams of data, as is the case of the video flow analysis in QoS integration and wireless sensor networks [30]. Data streams are fundamental to many other data processing applications as can be the atmospheric forecasting measurement, telecommunication network elements operation recording, stock market information updates, or emerging sensor networks as highway traffic conditions. Frequently the data streams are generated by geografically distributed information sources. Despite the increasing capacity of storage devices, it is not a good idea to store the data streams because even a simple prcessing operation, as can be to sort the incoming data, becomes very expensive in time terms. Then, the data streams are normally processed *on the fly* as they are produced. The stream model can be subdivided in various categories depending on the arrival order of the attributes and if they are aggregated or not. We assume that each element in the stream will be a pair $\langle i, j\rangle$ that indicates for a sequence a we have $a[i] = j$.

Definition 11 *A streaming algorithm accepts a data stream z and outpus a random variable $str(z, r)$ to approximate a function g so that*

$$(1-\epsilon)g(z) \leq str(z,r) \leq (1-\epsilon)g(z)$$

with probability $1-\delta$ over all choices of the random seed r, for parameters ϵ and δ

The streaming models can be adapted for some distances functions. Let suppose tha z consists of two interleaved sequences, a and b, and that $g(z) = d(a, b)$. Then the streaming algorithm to solve this proble approximates the distance between a and b. It is possible that the algorithm can can work in the sketching model as well as in the streaming model. Very frequently a streaming algorithm can be initially conceived as a sketching one, if it is supposed that the sketch is the contents of the storage memory for the streaming algorithm. However, a sketch algorithm is not necesarilly a streaming algorithm, and a streaming algorithm is not always a sketching algorithm. So, the goal of the use of this kind of algorithms, is to test equality between two object, approximately and in an efficient way. A very clear example is given in [9] concerning hash functions and equality testing between two sequences. In the example just mentioned it is proposed that two strings a and b are equal ($a = b$) if $hash(a) = hash(b)$. Ideally $|hash(a)| << |a|$. It turns out that this goal is impossible if counting arguments are taken into account. However, if it is accepted some very small error probability, then probabilistic methods can be proposed such that $hash(a) \neq hash(b)$ if $a \neq b$. Of this relies too, on the assumption that the hash function is picked at random from a family of hash functions with some probability distribution. Naturally, if $a = b$ then $hash(a) = hash(b)$, because the hash functions are deterministic functions. These hash functions are called *fingerprints* and the basic idea is that if two *fingerprints* match it is very likely that they were produced by the same individual. This last afirmation can be stated as a lemma as follows.

Lemma 12 .- *There exists fingerprints functions for vectors which represents vectors of size n bits using $O(\log n)$ bits*

The *fingerprints* have the nice property that the *fingerprint* of a vector formed as the concatenation of two vectors equals the composition of the *fingerprint* of each vector. Let $a\|b$ be the concatenation of two vectors a and b such that $|a + b| = |a| + |b|$. This last *fingerprint* nice property can be stated formally as follows:

Lemma 13

$$hash(a\|b) = (hash(a) + 2^{|a|}hash(b)) \ mod \ p$$

Proof:

$$\begin{aligned} hash(a\|b) &= \textstyle\sum_{i=1}^{|a|+|b|}(a\|b)_i 2^{i-1} \ mod \ p \\ &= (\textstyle\sum_{i=1}^{|a|} a_i 2^{i-1} + \sum_{i=1}^{|b|} b_i 2^{|a|+i-1}) \ mod \ p \\ &= ((\textstyle\sum_{i=1}^{|a|} a_i 2^{i-1} \ mod \ p) \\ &\quad +2^{|a|}(\textstyle\sum_{i=1}^{|b|} b_i 2^{i-1} \ mod \ p)) \ mod \ p \\ &= (hash(a) + 2^{|a|}hash(b)) \ mod \ p \end{aligned}$$

It should be noted that the *fingerprints* can be computed under an unordered and unaggregated streameng model when values of the vector arrive as a stream in arbitrary, interleaved order. That is, $hash(a)$ can be computed when a is presented in the form of a stream of tuples (i, c), meaning that $a_i = c$. It implies that the space required is the same as for the sketch model, $O(\log n)$. Another advantage of using *fingerprints* is that they are integers that can be represented in $O(\log n)$ bits. In the commonly used RAM calculation model it is assumed that this kind of quantities can be worked with in $O(1)$ time. This quantities can be used for building has tables allowing fast access to them without the use of special complex data structures or sorting preprocessing. Approximation of L_p distances can be considered that fit well with sketching model as well as with the streaming model. Initially it can be suppossed that the vectors are formed of positive integers bounded by a constant, but it can be extended the results to the case of rational entries. An important property possesed by the sketches of vectors is the *composability*, that can be defined as follows:

Definition 14 *A sketch function is said to be composable if for any pair of sketches $sk(a, r)$ and $sk(b, r)$ we have that $sk(a + b, r) = sk(a, r) + sk(b, r)$*

One theoretical justification that enables us to embed an Euclidean vector space in a much smaller space with a small loss in accuracy is the Johnson-Lindenstrauss lema [31] that can be stated as follows:

Lemma 15 *.- Let a, b be vectors of length n. Let v be a set of k different random vectors of length n. Each component $v_{i,j}$ is picked independently from de Gaussian distribution $N(0, 1)$, then each vector v_i is normalised under the L_2 norm so that the magnitude of v_i is 1. Define the sketch of a to be a vector $sk(a, r)$ of length k so that $sk(a, r)_i = \sum_{j=1}^{n} v_{i,j} a_j = v_i \cdot a$. Given parameters δ and ϵ, we have with probability $1 - \delta$*

$$\frac{(1-\epsilon)\|a-b\|_2^2}{n} \leq \frac{\|sk(a,r)-sk(b,r)\|_2^2}{k} \leq \frac{(1+\epsilon)\|a-b\|_2^2}{n}$$

where k is $O(1/\epsilon^2 \log 1/\delta)$

This lemma means that we can make a sketch of dimension smaller that $O(1/\epsilon^2 \log 1/\delta)$, from the convolution of each vector with a set of randomly created vectors drawn from the Normal distribution. So, this lemma enable us to map m vectors into a reduced dimension space. The sketching procedure cannot be assimilated directly to a streaming procedure, but it has been shown recently how to extend the sketching approach to the streaming environement for L_1 and L_2 distances. Concerning streaming algorithms, some of the first have been published in [1] for calculating the frequency moments. In this case, we have an unordered and unaggregated stream of n integers in the range of $1, \ldots, M$, such that $z = (s_1, s_2, \ldots, s_n)$ for integers s_j. So, in [1] the authors focus on calculating the frequency moments F_k of the stream. Let it be, from the stream, $m_i = |\{j|s_j = i\}|$, the number of the occurrences of the integer i in the stream. So the frequency moments on the stream can be calculated as $F_k = \sum_{i+1}^{M} (m_i)^k$. Then F_0 is the number of different elements in the sequence, F_1 is the length of the sequence n, and F_2 is the repeat rate of the sequence. So, F_2 can be related with the distance L_2. Let us suppose that we build a vector v of length M with entries chosen at random, we process the stream $s_1, s_2, \ldots, s_n$ entry by entry, and initialise a variable $Z = 0$. So, after whole stream has been processed we have $Z = \sum_{i=1}^{M} v_i m_i$. Then F_2 can be estimated as

$$\begin{aligned} Z^2 &= \sum_{i=1}^{M} v_i^2 m_i^2 \\ &+ \sum_{i=1}^{M} \sum_{j \neq i} v_i m_i v_j m_j \\ &= \sum_{i=1}^{M} m_i^2 \\ &+ \sum_{i=1}^{M} \sum_{j \neq i} m_i m_j v_i v_j \end{aligned}$$

So, if the entries of the vector v are pairwise independent, then the expectation of the cross-terms $v_i v_j$ is zero and $\sum_{i=1}^{M} m_i^2 = F_2$. If this calculation is repeated $O(1/\epsilon^2)$ times, with a different random v each time, and the average is taken, then the calculation can be guaranteed to be an $(1 \pm \epsilon)$ approximation with a constant probability, and if additionallly, by finding the median of $O(1/\delta)$ averages, this constant probability can be amplified to $1 - \delta$. It has been observed in [32] that the calculation for F_2 can be adapted to find the L_2 distance between two interleaved, unaggregated streams a and b. Let us suppose that the stream arrives as triples $s_j = (a_i, i, +1)$ if the element is from a and $s_j = (b_i, i, -1)$ if the item is from stream b. The goal is to find the square of the L_2 distance between a and b, $\sum_i (a_i - b_i)^2$. We initialise $Z = 0$. When a triple $(a_i, i, +1)$ arrives we add $a_i v_i$ to Z and when a triple $(b_i, i, -1)$ arrives we subtract $a_i v_i$ from Z. After the whole stream has been processed $Z = \sum (a_i v_i - b_i vi) = \sum_i (a_i - b_i) vi$. Again the expectation of the cross-terms is zero and, then the expectation of Z^2 is L_2 difference of a and b. THe procedure for L_2 has the nice property of being able to cope with case of unaggregated streams

containing multiple triples of the form $(a_i, i, +1)$ with the same i due to the linearity of the addition. This streaming algorithm translates to the sketch model: given a vector a the values of Z can be computed. The sketch of a is then these values of Z formed into a vector $z(a)$. Then $z(a)_i = \sum_j (a_j - b_j)v_{i,j}$. This sketch vector has $O(1/\epsilon^2 \log 1/\delta)$ entries, requiring $O(\log Mn)$ bits each one. Two such sketches can be combined, due to the composability property of the sketches, for obtaining the sketch of the difference of the vectors $z(a-b) = (z(a) - z(b))$. The space of the streamin algorithm, and then the size of the sketch is a vector of length $O(1/\epsilon^2 \log 1/\delta)$ with entries of size $O(\log Mn)$. A natural question can be if it is possible to translate sketching algorithms to streaming ones for distances different from L_2 or L_1 and objects other than vectors of integers. In [9] it shown that it is possible the this translation for the Hamming distance. This can be found in the next theorem of [9].

Theorem 16 *The sketch for the Symmetric Difference (Hamming distance) between sets can be computed in the unordered, aggregated streaming model. Pairs of sketches can be used to make* $1 \pm \epsilon$ *approximmations of the Hamming distance between their sequences, which succeed with probability* $1 - \delta$. *The sketch is a vector of dimension* $O(1/\epsilon^2 \log 1/\delta)$ *and each entry is an integer in the range* $[-n \ldots n]$.

Given that, under some circumstances, streaming algorithms can be translated to sketch algorithms, then the theorem 9 can be applied for the robot motion tracking problem, under the streaming model as well.

In general, the mobile robotics framework is more complex because we should process data flows provided by the captors under a dynamic situation, where the robot is moving, taking into account two kind of uncertainty:

- The sensors have low precision
- The robot movements are subject to deviations as any mechanical object.

The data flow provided by the captors produce similar problems to those that can be found on the databases. The robot should make the fusion of the information sources to determine his motion strategy. Some sources, called bags, allow the robot to self locate geometrically or in his state graph. While the robot executes his strategy, it is subject to movement uncertainties and then should find robust strategies for such uncertainty source. The goal is to achieve the robustness integrating the data flow of the captors to the strategies. We consider the classical form of simple Markovian strategies. In the simplest version, a Markov chain, MDP, is a graph where all the states are distinguishable and the edges are labeled by actions $L_1, L_2, \ldots, L_p$. If the states are known only by his coloration in k colors $C_1, C_2, \ldots, C_k$. Two states having the same coloration are undistinguishable and in this case we are talking about POMDP (Partially Observed Markov Decision Process). A simple strategy σ is a function that associates an action simplex to a color among the possible actions. It is a probabilistic algorithm that allows to move inside the state graph with some probabilities. With the help of the strategies we look for reaching a given node of the graph from the starting node (the initial state) or to satisfy temporal properties, expressed in LTL formalism. For instance, the property C_1 Until C_2 that express the fact that we can reach a node with label C_2 preceded only by the node C_1. Given a property θ and a strategy σ, let $Prob_\sigma(\theta)$ be the probability that θ is true over the probability space associated to σ. Given a POMDP M two strategies σ and π can be compared by means of ther probabilities, that is, $Prob_\sigma(\theta) > Prob_\pi(\theta)$. If $Prob_\sigma(\theta) > b$, it is frequent to test such a property while b is not very small with the aid of the path sampling according to the distribution of the POMDP. In the case that $b < Prob_\sigma(\theta) < b - \epsilon$ it can be searched a corrector for σ, it means, a procedure that lightly modify σ in such a way that $Prob_\sigma(\theta) > b$. It can be modified too the graph associated and in that case, we look for comparing two POMDPs. Let be M_1 and M_2 two POMDPs, we want to compare this POMDPs provided with strategies σ and π in the same way as are compared two automata in the sense that they are approximately equivalent (refer to the section concerning distance between DTDs). How can we decide if they are approximately equivalent for a property class? Such a procedure is the base of the strategy learning. It starts with a low performance strategy that is modified in each step for improvement. The tester, corrector and learning algorithms notions find a natural application in this context. One of the specificities of mobile robotics is to conceive robust strategies for the movements of a robot. As every mechanical object, the robot deviates of any previewed trajectory and then it must recalculate his location. At the execution of an action L_i commanded by the robot, the realization will follow L_i with probability p, an action L_{i-1} with probability $(1-p)/2$ and an action L_{i+1} with probability $(1-p)/2$. This new probabilistic space induce robustness qualities for each strategy, in other words, the $Prob_\sigma(\theta)$ depends on the structure of the POMDP and on the error model. Then the same questions posed before can be formulated: how to evaluate the quality of the strategies, how to test properties of strategies, how to fix the strategies

such that we can learn robust strategies. We can consider that the robots are playing a game against nature that is similar to a Bayesian game. The criteria of robust strategy are similar to those of the direct approach. Another problem that arise in robot motion is the relocalization of a robot in a map. As we mentioned in the section 1, one method that has been used frequently in robot exploration for reducing the uncertainty in the position of robot was the use of landmarks and triangulation. The search of a landmark in an unknown environment can be similar to searching a pattern in a sequence of characters or a string. In the prsent work we applied sketching and streaming algorithms for obtaining distance approximations between objects as vectors in a dimensional reduced, and in some sense, deformated space. If we want to apply sketching or streaming for serching patterns as landmarks in a scene we have to deal with distance between permutations.

6 Conclusion and future work

In this paper we applied some property testing algorithms under the sketch and streaming calculation model for measuring the level of approximation of inferred automata with respect to the true automata in the case of robot motion tracking problem as well as the map construction problem in robot navigation context. The use of sketch algorithms allowed us to approximate the distance between objects by the manipulation of sketches that are significantly smaller than the original objects. I applied streaming algorithms to robot tracking profiting of the possibility of translation of sketch algorithms to streaming ones. Another problem that arise in robot motion is the relocalization of a robot in a map. As we mentioned in the section 1, one method that has been frequently used in robot exploration for reducing the uncertainty in the position of robot was the use of landmarks and triangulation. The search of a landmark in an unknown environment can be similar to searching a pattern in a large sequence of characters or a big string. In the present work we applied sketching and streaming algorithms for obtaining distance approximations between objects as vectors in a dimensional reduced, and in some sense, deformated space. If we want to apply sketching or streaming for searching patterns as can be the landmarks in a scene we have to deal with distance between permutations. In the future I want to apply property testing algorithms under the sketching as well as streaming calculation model for the problem of sensorial fusion as well as for the target localization problem in robot navigation problems.

Acknowledgements: The research was supported by the Universidad Autónoma Metropolitana Campus Cuajimalpa.

References:

[1] Alon N., Matias Y., and Szegedy M.. The space complexity of approximating the frequency moments. *JCSS: Journal of Computer and System Sciences*, 58:137-147,1999

[2] J. Barraquand and J.C. Latombe. Robot motion planning: A distributed representation approach. Technical Report STAN-CS-89-1257, Department of Computer Science, Stanford University, 1989.

[3] Beesson Patrick Foil. Creating and Utilizing Symbolic Representations of Spatial Knowledge using Mobile Robots *PhD Thesis University of Texas at Austin Supervised by Benjamin J. Kuipers*, 2008

[4] Blum, M. and Kannan S.. Designing programs that check their work. *Journal of the ACM*, 42(1):269-291,1995

[5] Blum, M., Luby M. , and Rubinfeld R. Self-Testing/Correcting with application to numerical problems. *Journal of Computer and System Science*, 47(3):549-595,1993

[6] D. Burago , M. de Rougemont, A. Slissenko. The complexity of motion planning under uncertain deviations, Theoretical Computer Science 1996.

[7] Burago D., Gregoriev D. and Slissenko A. Approximating shortest path for the skew lines problem in time doubly ogarithmic in $1/\epsilon$ *Peprint submited to Elsevier Science*,8 september 2003

[8] Canny, J. and Reif, J.H. New Lower-Bound Techniques for Robot Motion Planning Problems *Proc. 28st. FOCS*, pages 49-60, 1987

[9] Cormode Graham, Sequence Distance Embeddings, PhD Thesis in CS University of Warmick, 2003.

[10] Dean T., Angluin D., Basye K., Engelson S., Kaelbling L., Kokkevis E. and Maron O., Inferring Finite Automata with Stochastic Output Functions and an Application to Map Learning *Machine Learning*, , 18(1):81-108, 1995.

[11] Kuipers B., Modayil J., Beeson P., MacMahon M. and Savelli F. Local Metrical and Global Topological Maps in the Hybrid Spatial Semantic Hierarchy *IEEE International Conference on Robotics and Automation (ICRA-04)*, 2004

[12] S.M. La Valle, David Lin, Leonidas J. Guibas, J.C. Latombe & Rajeev Motwani, Finding an Unpredictable Target in a Workspace with Obstacles, IEEE International Conference on Robotics and Automation, 1997.

[13] Marion, J.L. and Rodriguez, C. and de Rougemont, M. The evaluation of strategies in motion planning with uncertainty *Proc. IEEE Robotics & Automation ICRA-94*,1994

[14] R. Murrieta-Cid, H.H. Gonzlez-Baos , B. Tovar, A Reactive Motion Planner to Maintain Visibility of Unpredictable Targets, IEEE International Conference on Robotics and Automation, 2002..

[15] R. Rubinfeld and M. Sudan, Robust characterizations of polynomials with applications to program testing, SIAM Journal on Computing, 1996.

[16] Rodríguez-Lucatero, C. Evaluation, Existence and Optimization of reactive strategies in motion with uncertainty *Journal : Expert Systems with Applications, An International Journal, Pergamon, Elsevier Ltd.*,, Vol. 12, Number 1, 1997

[17] C. Rodríguez Lucatero, A. de Albornoz Bueno, R. L. Espinosa, A game theory approach to the robot tracking problem, *WSEAS ISPRA04* Salzburg Austria.

[18] C. Rodríguez Lucatero, A. de Albornoz Bueno and R. Lozano Espinosa, A game theory approach to the robot tracking problem, *Journal WSEAS Transactions on Computers*, Issue 4 Vol. 3, ISSN 1109-2750, pags 862-869.

[19] C. Rodríguez Lucatero and R. Lozano Espinosa, APPLICATION OF AUTOMATA LEARNING ALGORITHMS TO ROBOT MOTION TRACKING, *Journal WSEAS Transactions on Systems*, Issue 1 Vol. 4, ISSN 1109-2777, pags 122-127, 2005

[20] C. Rodríguez Lucatero, THE PROBLEM OF ROBOT RANDOM MOTION TRACKING LEARNING ALGORITHMS, *WSEAS ISPRA2007*, 2007.

[21] O. Goldreich and S.Goldwasser and D.Ron.Property, Testing and Its Connection to Learning and Approximation, IEEE Symposium on Foundations of Computer Science, 1996.

[22] C. Papadimitriou and Mihalis Yannakakis. On Bounded Rationality and Computational Complexity. *Technical Report Indiana University*, 1994.

[23] J. Kleinberg and C. Papadimitriou and P. Raghavan, On the value of private information, Tark conference, 2001.

[24] B. Gueye, P. Rigaux, N. Spyratos, Annotation automatique de documents XML, Actes des journes 'Extraction et gestion des connaissances' (EGC'04), Clermont-Ferrand, France,January,2004.

[25] Magniez F. , de Rougemont, M. Property testing of regular tree languages, ICALP 2004.

[26] Fisher E., Magniez F. , de Rougemont, M. Property and equivalence testing on strings, ECCC 2004.

[27] Valiant L.G. A theory of the learnable,CACM, 27(11):1134-1142,1984

[28] Xiaotie Deng, Tiko Kameda and C. Papadimitriou. How to learn an unknown environment (Extended Abstract). *FOCS 1991*, 298-303 , 1991.

[29] Ovidiu Ilie Sandru. Mathematical models that coordinate the movement through obstacles of the dynamic systems endowed with artificial sight. *WSEAS Transactions on Mathematics*, 482-491, 2008.

[30] Weilian Su and Bassam Almaharmeh. QoS integration of the Internet and Wireless Sensor Networks. *WSEAS Transactions on Computers*, 253-258, Issue 4, Vol. 7, April 2008.

[31] W.B. Johnson and J. Lindenstrauss. Extensions of Lipshitz mapping into Hilbert space. *Contemporary Mathematics*, 26:189-206, 1984.

[32] Feigenbaum J., Kannan S., Strauss M., and Viswanathan M. An approximmate L_1 difference algorithm for massive data streams. *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, 501-511, 1999.