March, 1999

LIDS- TH 2453

Research Supported By:

ARO Grant DAAL03-92-G-0115
ARO Grant DAAH04-96-10445

Robust Algorithms For Model-Based Object Recognition and Localization

Louay Mohamad Jamil Bazzi

Robust Algorithms For Model-Based Object Recognition and Localization

by

Louay Mohamad Jamil Bazzi

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1999

Author ....
Department of Electrical Engineering and Computer Science
February 2, 1999

Certified by ....
Sanjoy k. Mitter
Professor of Electrical Engineering
Thesis Supervisor

Accepted by ....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Robust Algorithms For Model-Based Object Recognition and Localization

by

Louay Mohamad Jamil Bazzi

Submitted to the Department of Electrical Engineering and Computer Science
on February 2, 1999, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

We consider the problem of model-based object recognition and localization in the presence of noise, spurious features, and occlusion. We address the case where the model is allowed to be transformed by elements in a given space of allowable transformations. Known algorithms for the problem either treat noise very accurately in an unacceptable worst case running time, or may have unreliable output when noise is allowed. We introduce the idea of tolerance which measures the robustness of a recognition and localization method when noise is allowed. We present a collection of algorithms for the problem, each achieving a different degree of tolerance. The main result is a localization algorithm that achieves any desired tolerance in a relatively low order worst case asymptotic running time. The time constant of the algorithm depends on the ratio of the noise bound over the given tolerance bound. The solution we provide is general enough to handle different cases of allowable transformations, such as planar affine transformations, and scaled rigid motions in arbitrary dimensions.

Thesis Supervisor: Sanjoy k. Mitter
Title: Professor of Electrical Engineering

Acknowledgments

I greatly acknowledge the support and guidance of my thesis supervisor Professor Sanjoy Mitter. I would like to thank him for his support, his encouragement, his easy going supervision style, and his valuable suggestions and ideas. The reported work has been strongly shaped and oriented by his guidance and his continuous stress on simplicity and generality.

I thank Doctor Mohamad Akra for introducing me to this field and his constant encouragement. I am thankful to him for reading an early draft of this thesis and offering many useful comments that reshaped the style and the presentation of the work.

Many thanks go to my friends in the Lebanese Club at MIT for their encouragements.

Finally, I would like to thank my father, mother, brothers, and sisters for being an unlimited source of emotional support, and an infinite supply of motivation. To them I dedicate my thesis.

This work was supported by ARO grant DAAL03-92-G-0115 and DAAH04-96-10445.

Contents

Chapter 1

Introduction

The problem being considered is a key one in model-based object recognition. The ultimate goal is to enable a machine to recognize and locate *models* "in" noisy *observations.* Models and observations are finite sets of *features.* The features may be points in the plane or in a higher dimensional space. The model is allowed to be transformed by elements in a given space of *allowable transformations.* Allowable transformations can be planar rigid motions, planar affine transformations, or - in general - a class of parameterized mappings over the space of features. The "presence" of the model in the observation is manifested by having - possibly many - subsets of the observation that are noisy, transformed, and occluded instances of the model. Transformed instances of the model are *noisy* in the sense that their corresponding features are perturbed by noise vectors. In addition to being noisy, some features of those instances may be missing, a phenomenon called *occlusion.* In addition to those instances (if any), the observation - possibly - contains other features called *spurious features.*

For a given space of allowable transformations, the objective is to develop a recognition and localization algorithm that has the following characteristics. The algorithm should take as input the model, the observation, and a collection of constraints describing the allowable noise and occlusion phenomena. In terms of recognition, the algorithm should be able to decide whether the observation contains noisy, transformed, and occluded instances of the model satisfying the given input constraints.

In terms of localization, the algorithm should be able to report the locations of theses instances.

The problem being considered has applications in Pattern Recognition, Image Understanding, Robotics, and Astronomy. In these fields, the model is a 2D or 3D pattern representing an ideal object. The observation is a processed version of a set of measurements supplied by a sensory device in the machine environment. The features in the observation are usually noisy due to sensing errors. The model is possibly present in the observation among many other objects. Theses other objects are the source of spurious features. Moreover, the model might be partially present in the observation in the sense that some of its parts are hidden due to an overlap with an other object. This leads to the occlusion phenomenon. The space of allowable transformations depends on the specific application. It may consists of translation only, planar rigid motions and scaling, or planar affine transformations. The latter is equivalent to 3D rigid motions composed with weak perspective projections. It is usually used to locate 3D objects in 2D observations.

1.1 Literature Survey

Due to its wide range of applications, the problem has attracted considerable attention. What might be considered as a solved problem is the case where there is no noise (e.g., Huttenlocher & Ullman [1987], Huttenlocher & Ullman [1990]). Allowing for noise makes the problem much more difficult. A large number of related research efforts were reported, each for a specific class of allowable transformations.

Grimson & Lozano-Perez[1984,1987] presented an algorithm based on searching the space of matchings between the model and the observation features. The algorithm is robust and accurate, but it has a worst case exponential running time in terms of the number of features in the model. This issue was treated using several speedup heuristics.

Cass [1990] developed a polynomial algorithm for the case when allowable transformations are planar rigid motions. The algorithm is robust with respect to noise, but suffers from a high (twelfth order) running time. Its worst case running time is $O(m^6 n^6)$, where m and n are the number of points in the model and the observation, respectively.

Many research efforts handle the noise by heuristics. Related efforts treat noise by adding heuristics to hypothesize and test techniques that where originally developed for a noiseless setting (e.g., Baris & Ullman [1988], Huttenlocher & Ullman [1987], Huttenlocher & Ullman [1990], Thompson & Mundy [1987], Costa et al. [1990], Landman et al. [1990], Landman & Wolfson [1988], Van Gool et al. [1991]). All the above techniques are for the case when allowable transformations are affine transformations. Grimson, Huttenlocher, & Jacobs [1994] studied the methods on which such techniques are based. They reported that there is a significant probability of error in the output even with a moderate level of noise, suggesting the importance of good verification and grouping techniques.

Other research effort were reported on restricted versions of the problem, such as recognition - only algorithms that do not handle occlusion (Chew et al. [1997]). Even in this restricted environment, the worst case running time is $O(m^3 n^2 \log mn)$ in the setting of planar rigid motions. Other algorithms work under the assumption that the number of points in the model and the observation are equal (e.g., Alt et al. [1988] and Braid [1985]), thus no spurious features or occlusion are allowed. The corresponding worst case running times are $O(n^6)$ for translation only and $O(n^8)$ for translation, rotation, and reflection.

1.2 Contribution

Known algorithms for the problem either treat noise very accurately in an unacceptable worst case running time, or may have unreliable output when noise is allowed. We introduce the concept of tolerance which measures the robustness of a recognition and localization method, when noise is allowed. We present a collection of localization algorithms for the problem each achieving a different degree of tolerance. We start with an efficient algorithm that achieves a tolerance bound in the order of the noise bound. We build on this algorithm to reach any given tolerance bound at the cost of increasing the algorithm constant. The correctness proof of the algorithms is based on a robust mathematical treatment of the problem that leads to two interesting tight bounds.

Tolerance

We briefly explain the idea of tolerance in the recognition framework. We need first a tentative definition. Assume that m is the number of points in the model. Let $\xi > 0$ be an upper bound on the noise norm and let p be an upper bound on the number of points missing from an instance of the model in the observation. Say that an allowable transformation is (ξ, p)-*feasible* if it maps at least $m - p$ point of the model to within ξ-distance of points in the observation.

Let $\mu > 0$. We say that a decision algorithm is a recognition algorithm that *achieves a tolerance* μ if it satisfies the following two conditions.

1. The input of the algorithm is the model, the observation, an upper bound ξ on the noise norm, and an upper bound p on the number of points missing from an instance of the model in the observation.

2. The output of the algorithm is guaranteed to be YES if there exists an (ξ, p)-feasible transformation and NO if there are no $(\xi + \mu, p)$-feasible transformations.

If, in addition, any positive value of μ can be prespecified in the algorithm input, the algorithm is said to *achieve any given tolerance.*

Thus $[\xi, \xi + \mu)$ is the tolerance region of the algorithm. The algorithm is provably accurate and robust in the region $[0, \xi) \cup [\xi + \mu, \infty)$, but its results are uncertain in the region $[\xi, \xi + \mu)$. Introducing the tolerance region is a relaxation of the problem that dramatically reduces the time complexity. Note that - practically - values of μ such as ξ are sufficient in general.

In terms of localization, an accurate formulation of the idea is lengthy due to the complicated setting of the problem. We will postpone this formulation to the next section.

Known Approaches

In terms of tolerance, known algorithms for the problem can be generally classified into two categories: Algorithms that achieve zero tolerance and algorithms that have unbounded (or unknown) tolerance.

The main disadvantage of the methods that achieve a zero tolerance is their time requirements (e.g., Grimson & Lozano-Perez[1984,1987] is worst case exponential, Cass [1990] is polynomial of twelfth order for planar rigid motions).

Research efforts that handle noise by heuristics have unbounded (or unknown) tolerance. Most related efforts allow for noise by adding heuristics to hypothesize and test techniques that were originally developed for a noiseless setting (e.g., Baris & Ullman [1988], Huttenlocher & Ullman [1987], Huttenlocher & Ullman [1990], Thompson & Mundy [1987], Costa et al. [1990], Landman et al. [1990], Landman & Wolfson [1988], Van Gool et al. [1991]). Grimson, Huttenlocher, & Jacobs [1994] studied the methods on which such techniques are based. They reported that there is a significant probability of error in the output even with a moderate level of noise, suggesting the importance of good verification and grouping techniques.

Contribution

Our contribution is a collection of localization algorithms each achieving a different level of tolerance. We start with a worst case efficient algorithm that achieves a tolerance bound in the order of the noise bound. We build on this algorithm to reach any given tolerance bound at the expense of increasing the algorithm constant. The case when occlusion is allowed is considered independently, and the solution obtained is general enough to handle different cases of allowable transformations. Mainly, we concentrate in this work on two cases : planar affine transformations and scaled rigid motions in arbitrary dimensions. We summarize below the worst case running times in each of the two cases. Note that m and n are - respectively - the number of points in the model and the observation. Note also that in all the bounds below the norm used on the features space is the ∞-norm.

In the planar affine setting, we show first how to achieve an tolerance bounded by 3ξ in $O(m^4 + n^3 m \log n)$ worst case running time when occlusion is not allowed (i.e., $p = 0$) and in $O(n^3 m^4 \log n)$ when occlusion is allowed. Then we show how to reach any given tolerance μ in $O(m^4 + (\frac{3\xi}{\mu})^6 n^3 m \log n)$ worst case running time when occlusion is not allowed and in $O((\frac{3\xi}{\mu})^6 n^3 m^4 \log n)$ when occlusion is allowed. We show also that, under the assumption that the minimum distance between any two distinct points in S is at least 8ξ, zero tolerance is achievable in $O(m^4 + n^3 m \log mn)$ time when occlusion is not allowed. Observe that the $(\frac{3\xi}{\mu})^6$ time constant is very crude in the sense that it is an upper bound on the worst case constant of a heuristic free algorithm.

In the d-dimensional rigid motions and scaling setting, we have similar results. We show first how to achieve a tolerance bounded by $(1 + \sqrt{3(d-1)})\xi$ in $O(m^3 + n^2 m \log^{d-1} n)$ worst case running time when occlusion is not allowed, and in $O(n^2 m^3 \log^{d-1} n)$ when occlusion is allowed. Then we show how to reach any given tolerance μ in $O(m^4 + (\frac{c\xi}{\mu})^{2d} n^2 m \log^{d-1} n)$ worst case running time when occlusion is not allowed, and in $O((\frac{c\xi}{\mu})^{2d} n^2 m^3 \log^{d-1} n)$ when occlusion is allowed, where $c = 1 + \sqrt{3(d-1)}$. We show also that, under the assumption that the minimum distance between any two distinct points in S is at least $2(2 + \sqrt{3(d-1)})\xi$, zero tolerance is achievable in $O(m^3 + n^2 m H_{2d}(m) + n^2 m \log^{d-1} n)$ time when occlusion is not allowed ($H_{2d}(m)$ is the time needed to test the feasibility of a $2d$-dimensional linear program of m constraints).

An other contribution of this work is two tight bounds in the setting of finite sets subject to affine transformations and scaled rigid motions. In each case, we derive a tight bound that illustrates to what degree a finite set of points can be approximated, with respect to the corresponding transformations space, by one of its bounded cardinality subsets. In the planar affine case, we prove that any finite planar set M (not lying on a line) contains a 3-point subset U satisfying: $||t||_M \leq 3||t||_U$, for each affine transformation t, where $||t||_M$ and $||t||_U$ are - respectively - the supnorms of t on M and U. In the scaled rigid motions setting, we prove that any finite d-dimensional set M (consisting of more than one point) contains a 2-point subset U satisfying: $||t||_M \leq \left(1 + \sqrt{3(d-1)}\right) ||t||_U$, for each d-dimensional transformation t

composed of rotation translation and scaling. Both bounds are used in the correctness proof of the algorithms.

Chapter 2

Problem Definition

In this section, we present a robust definition for the recognition and the localization problem and we formalize the idea of tolerance. Due to the complicated nature of the problem, we follow a differential approach. We start with the recognition case when occlusion is not allowed. Then we move to the localization task again when occlusion is not allowed. We formalize this task first without tolerance, then we modify the definition to allow for tolerance. Finally, we generalize the ideas to the setting of occlusion.

Notation:

- Let $\mathcal{F} = \mathbb{R}^d$ be the space of *features*. The *model* is a finite subset of $\mathcal{F}$ denoted by M and the *observation* is a finite subset subset of $\mathcal{F}$ denoted by S. We shall call the elements of $\mathcal{F}$ *points* and any finite subset of $\mathcal{F}$ a *pattern*. So, the model M and the observation S are both patterns.

- The space of *allowable transformations* is a set of parameterized mappings from $\mathcal{F}$ to $\mathcal{F}$ denoted by $\mathcal{A}$.

An example of $\mathcal{F}$ is the plane . Examples of $\mathcal{A}$ are: planar translation, planar affine transformations, a subset of the space of affine transformation defined by some given constraints, or planar rigid motions.

In order to deal with noise, we need a norm to measure its magnitude. Say that $||.||$ is a norm on $\mathcal{F}$.

Notation:

- If x is a point, let $B_\varepsilon(x)$ be the ε-neighborhood of x, i.e., $B_\varepsilon(x) = \{y \in \mathcal{F}; \|x - y\| < \varepsilon\}$.
- If V is a pattern, let $B_\varepsilon(V)$ be the ε-neighborhood of V, i.e., $B_\varepsilon(V) = \bigcup_{\alpha \in V} B_\varepsilon(\alpha)$.

2.1 Tolerance and Recognition

We start with case when occlusion is not allowed.

Definition 2.1.1 *A transformation t in $\mathcal{A}$ is said to be ξ-*feasible *if $t(M) \subset B_\xi(S)$. In other words, an allowable transformation is said to be ξ-feasible if it maps the model inside the ξ-neighborhood of the observation.*

To put it in another way, an allowable transformation is ξ-feasible if it maps the points of the model to within ξ-distance of points in the observation.

The recognition problem is about checking the existence of noisy transformed instances of the model in the observation.

Let $\mu \geq 0$. In the case when occlusion is not allowed, we say that a decision algorithm is a recognition algorithm that achieves a tolerance μ if it satisfies following two conditions.

1. The algorithm input consists of the model M, the observation S, and an upper bound ξ on the noise norm.
2. The algorithm output is YES if there exits ξ-feasible transformation and NO if there are no $(\xi + \mu)$-feasible transformations.

If, in addition, any positive value of μ can be prespecified as an input to the algorithm, we say that the algorithm achieves any given tolerance.

A zero value of μ means that the output is YES if and only if there exists an ξ-feasible transformation. A positive value of μ means that $[\xi, \xi + \mu)$ is a tolerance region of such an algorithm. The algorithm is accurate and robust in the region $[0, \xi) \cup [\xi + \mu, \infty)$, but its output is uncertain in the region $[\xi, \xi + \mu)$. As long as the algorithm achieves a small value of μ or as long as μ is prespecifiable as an input, the relaxed problem is practically as relevant as the original one. Practically, values of μ such as ξ are sufficient in general.

The motivation behind introducing the tolerance idea is that known methods either deal with noise very accurately (have zero tolerance) in a very high order worst case running time or may have unreliable worst case performance in terms of noise (have unbounded tolerance). This motivates introducing a measure of performance in terms of noise and searching for algorithms that can achieve low levels (not necessarily zero) of tolerance in an acceptable time. The reason why a zero tolerance level may be, in general, computationally expensive is that, for high dimensional allowable transformations, the set of ξ-feasible transformations has a very complicated boundary structure that makes testing its emptiness computationally expensive as function of the number of points in the model and the observation. Allowing for occlusion and working in the localization setting makes the computational price even higher. For an example, see Cass [1990].

2.2 Localization

Let us move to the localization problem while still not allowing for occlusion. To simplify the process, we start with a definition that does not include tolerance, then we accommodate the definition to the tolerance setting.

The localization problem is about computing a set of allowable transformations that "represents the locations" of the noisy transformed instances of the model in the observation, if any. So we are interested in the ξ-feasible transformations. First, note that - usually - there are infinitely many ξ-feasible transformations.

Example: Consider the simple case where $\mathcal{F}$ is the plane and $\mathcal{A}$ consists of translation - only. Say that the model consists of a single point α, and the observation S consists of a single point β. In this setting, the ξ-feasible transformations are the translation vectors in ξ-neighborhood of the point $\beta - \alpha$.

This means that we can not ask for an algorithm that computes all the ξ-feasible transformations. The natural alternative is to compute a representative subset of the ξ-feasible transformations. One way to formalize the notion of a representative subset is through sampling. In order to formalize the idea, we need a distance function between allowable transformations, which we define below.

Definition 2.2.1 *Given a pattern V, let $E(V)$ to be the set of all mappings to $\mathcal{F}$ from patterns containing V, and define ρ_V to be the mapping $\rho_V : E(V) \times E(V) \to \mathbb{R}$ given by:*

$$\rho_V(h_1, h_2) \equiv \|h_1 - h_2\|_{\rho_V} \equiv \max_{\alpha \in V} \|h_1(\alpha) - h_2(\alpha)\|.$$

To simplify the notations we denote $\|.\|_{\rho_V}$ by $\|.\|_V$.

Note that ρ_V satisfies the triangle inequality regardless of whether or not the involved mappings have the same domain. Note also that ρ_V is a metric on the set of all the mappings to $\mathcal{F}$ whose domain is equal to V. But it need not be a metric over the set of allowable transformations if V is an arbitrary pattern (Remember that the set $\mathcal{A}$ of allowable transformation is set of parameterized mappings defined on $\mathcal{F}$).

Definition 2.2.2 *We say that a pattern V is* regular *if $\forall t_1, t_2 \in \mathcal{A}$, $\|t_1 - t_2\|_V = 0$ implies that $t_1 = t_2$.*

The regularity of a pattern V guarantees that ρ_V is a metric over $\mathcal{A}$ and in general a norm over the vector space generated by $\mathcal{A}$ (the other defining conditions of a norm are inherited from the norm of $\mathcal{F}$).

Assumption In what follows we assume that *the model M is regular.*

We will see soon that the regularity assumption is far from being restrictive (in the planar affine case, for instance, it is equivalent to the fact that the pattern contains three noncollinear points)

We are now in a position to formalize the sampling idea.

Notation: If X be a subset of $\mathcal{A}$. By the *δ-neighborhood* of X we mean the set of all the transformations in $\mathcal{A}$ that are within δ-distance to elements in X, where the distance is measured with respect to the metric ρ_M.

The localization problem can be simply stated as follows "Compute a set of ξ-feasible transformations whose δ-neighborhood covers (i.e., contains) all the ξ-feasible transformations, for some δ in the order of ξ".

This is equivalent to saying: compute any subset $\mathcal{X}$ of the space of allowable transformations satisfying :

1. *covering condition:* for each ξ-feasible transformation t_1, there exists a transformations t_2 in $\mathcal{X}$ such that $||t_1 - t_2||_M < \delta$.

2. *feasibility condition:* each transformation in $\mathcal{X}$ is ξ-feasible.

for some δ in the order of ξ.

The covering condition means that any ξ-feasible transformations is represented by (i.e., close to) some transformation in the output. In contrast to the value of μ, *the value of δ is not critical*, as long it is not very large compared to ξ. The value of δ represents to what extent the algorithm is able to distinguish between close occurrences of the model in the observation.

A solution of the above localization problem implies a solution of the corresponding recognition problem regardless of the value of δ. If $\mathcal{Q}$ is an algorithm of this localization problem, we can build a recognition algorithm $\mathcal{Q}'$ by answering YES if and only if the output of $\mathcal{Q}$ is not empty. Regardless of the value of δ, it is easy to see that answer of $\mathcal{Q}'$ is YES if and only if there exist an ξ-feasible transformations

Now we accommodate the above definition to allow for tolerance. Tolerance can be introduced by allowing some of the transformations in the output to be $(\xi+\mu)$-feasible and not ξ-feasible.

Let $\mu \geq 0$. Consider the case when occlusion is not allowed. Say that an algorithm is a localization algorithm that achieves a tolerance μ if it satisfies the following two conditions. The algorithm input consists of the model M, the observation S, and an upper bound ξ on the noise norm. The algorithm output is a set of $(\xi + \mu)$-feasible transformations whose δ-neighborhood covers all the ξ-feasible transformations, for some δ in the order of ξ. Here again, if - in addition - any positive value of μ can be prespecified as an input to the algorithm, we say that the algorithm achieves any given tolerance.

Let $\mathcal{X}$ be the output of such an algorithm. The above conditions on the output are equivalent to:

1. *covering condition:* for each ξ-feasible transformation t_1, there exists a transformation t_2 in $\mathcal{X}$ such that $||t_1 - t_2||_M < \delta$

2. *relaxed feasibility condition:* each transformation in $\mathcal{X}$ is $(\xi + \mu)$-feasible.

Once again a solution of the above localization problem implies a solution of corresponding recognition problem regardless of the value of δ. Given any such localization algorithm $\mathcal{Q}$. We can build a recognition algorithm $\mathcal{Q}'$ by answering YES if and only if the output of $\mathcal{Q}$ is not empty. Regardless of the value of δ, it is easy to see that answer of $\mathcal{Q}'$ is YES if there exists an ξ-feasible transformation and NO if there are no $(\xi + \mu)$-feasible transformation.

Observe that, in general, tolerance can not be removed from the output by adding a verification stage because discarding those elements of the output that fails the ξ-feasibility test may violate the covering condition for all the δ's.

2.3 Occlusion

Extending the above definitions to allow for occlusion is simple. First we need to extend the feasibility notion to subsets of the model. In what follows p is an upper bound on the number points missing from an instance of the model in the observation.

Definition 2.3.1 *If t is an allowable transformation*

1. *Say that t is ξ-feasible on a pattern V if $t(V) \subset B_\xi(S)$.*

2. *Say that t is (ξ, p)-feasible if it is ξ-feasible on a regular subset of M containing at least $m - p$ points.*

To put it in an other way, an allowable transformation is (ξ, p)-feasible if it maps at least $m - p$ point of the model to within ξ-distance of points in S (with the additional requirement that those points form a regular pattern).

In the occlusion setting, given a bound p on the number of missing points in M as an additional input, we say that the recognition algorithm achieves a μ tolerance if its output $\mathcal{X}$ is a subset of the space of allowable transformations satisfying :

1. *covering condition:* If V is a regular subset of M containing at least $m - p$ points, then for each transformation t_1 that is ξ-feasible on V, there exists a transformation t_2 in $\mathcal{X}$ such that $\|t_1 - t_2\|_V < \delta$

2. *relaxed feasibility condition:* each transformation in $\mathcal{X}$ is $(\xi + \mu, p)$-feasible,

for some δ in the order of ξ.

Chapter 3

Summary of Results

We start in Chapter 4 with the case when $\mathcal{F} = \mathbb{R}^2$ and $\mathcal{A}$ is the space of planar affine transformations. We will be using the ∞-norm[1] on $\mathcal{F}$, i.e., $||(x, y)|| = max\{x, y\}$. We begin with case where occlusion is not allowed.

In **Section 4.2**, we present a localization algorithm $\mathcal{Q}_1$ that has an unbounded tolerance: The algorithm takes as input M, S , ξ and any 3-point regular subset U of M. The algorithm achieves in $O(n^3 m \log n)$ time an $\xi\Delta(U, M)$ tolerance, where

$$\Delta(U, M) = \sup_{\{t \in \mathcal{A}^*; ||t||_U \leq 1\}} ||t||_M,$$

and $\mathcal{A}^*$ is the vector space generated by $\mathcal{A}$. The tolerance of this routine is scaled by $\Delta(U, M)$, a value that may be very large for an arbitrary U. $\mathcal{Q}_1$ works by trying all the mappings form U to S. For each mapping it computes in $O(1)$ time the unique affine transformation that agrees with the mapping on U. Then it tests in $O(m \log n)$ time the $(\xi + \xi\Delta(U, M))$-feasibility of hypothesized transformation before adding it to the output.

In **Section 4.3**, we describe $\mathcal{Q}_2$, a localization algorithm that achieves a tolerance

[1]The motivation behind selecting the ∞-norm is that it will highly simplify the involved computational and analytical problems. However, it should be noted that the main results in this work are not dependent on the usage of the ∞-norm. Essentially, any well behaved norm should work, but it may require more complicated computational and analytical techniques. While presenting the work, we briefly explain how to generalize the results to the 2-norm.

bounded by 3ξ in $O(m^4 + n^3 m \log n)$ time : First, we prove that any regular model M contains a 3-point regular subset U satisfying $\Delta(U, M) \leq 3$. Then we show how to compute in $O(m^4)$ time the 3-point regular subset U^* of M that minimizes $\Delta(U, M)$. This means that $\Delta(U^*, M) \leq 3$. By supplying U^* to $\mathcal{Q}_1$, we end up with a tolerance bounded by 3ξ.

In **Section 4.4**, we describe how any given tolerance μ can be achieved $O(m^4 + (\frac{3\xi}{\mu})^6 n^3 m \log n)$ time. The new tool is an observation processing routine. The routine computes in linear time a new observation whose $\frac{\mu}{\Delta(U^*,M)}$-neighborhood is equal to the ξ-neighborhood of S. It turns out that if we run $\mathcal{Q}_1$ on the modified observation and a new value of ξ equal to $\frac{\mu}{\Delta(U^*,M)}$, the tolerance in its output decreases to μ. Due the the fact that $\Delta(U^*, M) \leq 3$, the size of the modified observation is bounded by $(\frac{3\xi}{\mu})^2 n$.

In **Section 4.5**, we show that zero tolerance can be achieved in $O(m^4+n^3 m \log mn)$ time if the minimum distance between any two distinct points in S is bounded by 8ξ. The idea is a verification stage on top of $\mathcal{Q}_2$ that makes use of the given hypothesis on S.

In **Section 4.6**, we generalize the idea to the occlusion setting. First, we construct $\mathcal{P}_1$, a modified version of $\mathcal{Q}_1$ that partially handles occlusion. The input of $\mathcal{P}_1$ is (M, S, U, ξ, p), where p is the bound on the number of missing points in the model. $\mathcal{P}_1$ partially handles occlusion in the sense that the points of U are not allowed to be missing. The running time of $\mathcal{P}_1$ is $O(n^3 m \log n)$ and the tolerance in its output is $\xi\Delta(U, M)$. Next, we construct $\mathcal{P}_2$, an algorithm that handles occlusion in $O(n^3 m^4 \log n)$ time and achieves a tolerance bonded by 3ξ (Essentially, this is the main result of this section). For each 3-point regular subset U of M, $\mathcal{P}_2$ computes a new model V_U equal to the largest subset V of M satisfying $\Delta(U, V) \leq 3$. Then for each such U it runs $\mathcal{P}_1$ on (V_U, S, U, ξ, p) if $|V_U| \geq m - p$. The output of $\mathcal{P}_2$ is the union of the sets returned by $\mathcal{P}_1$ in each run. The main idea in the correctness proof of $\mathcal{P}_2$ is the fact that any regular model V contains a 3-point regular subset U satisfying $\Delta(U, V) \leq 3$. Finally, we show how to handle occlusion while achieving any given tolerance μ in $O((\frac{3\xi}{\mu})^6 n^3 m^4 \log n)$ time. The idea is pretty much like the one in the case when occlusion is not treated: feed $\mathcal{P}_2$ with a new observation and new value of ξ.

In **Chapter 5**, we describe how to generalize the results to the case when the space of allowable transformations consists of d-dimensional rigid motion and scaling. The main new result needed for the generalization is another tight bound on $\Delta(U, M)$. We show that, in the setting of this new set of allowable transformations, each model M contains a 2-point subset U satisfying $\Delta(U, M) \leq 1 + \sqrt{3(d-1)}$. Using this new bound we generalize all the previous results to the following. In the case when occlusion is not allowed, a tolerance bounded by $1 + \sqrt{3(d-1)}$ can be achieved in $O(m^3 + n^2 m \log^{d-1} n)$ time. Any given tolerance μ can be achieved in $O(m^3 + (\frac{(1+\sqrt{3(d-1)})\xi}{\mu})^{2d} n^2 m \log^{d-1} n)$ time. A zero tolerance can be achieved in $O(m^3 + H(m) n^2 m + n^2 m \log^{d-1} n)$ if the distance between any two distinct points in S is at least $2(2 + \sqrt{3(d-1)})\xi$, where $H_{2d}(m)$ is the time needed to test the feasibility of a linear program of $2d$ variables and m constraints. In the occlusion setting, a tolerance bounded by $1 + \sqrt{3(d-1)}$ can be achieved in $O(n^2 m^3 \log^{d-1} n)$ time and any given tolerance μ can be achieved in $O((\frac{(1+\sqrt{3(d-1)})\xi}{\mu})^{2d} n^2 m^3 \log^{d-1} n)$ time.

Finally, in **Chapter 6** we present a sampling algorithm that can be used to reduce the number of points in the observation while keeping its information content with respect to recognition unaffected. The sampling algorithm computes in $O(n(\log^{d-1} n + h^{d-1} \log h))$ time a locally optimal subset S' of S whose ξ-neighborhood is equal to the ξ-neighborhood of S, where h is the maximum number of points in S lying in a 2ξ-neighborhood.

Chapter 4

Localization in the Setting of Planar Affine Transformation

In this chapter we will be working in the framework of planar affine transformation. Thus the space of features $\mathcal{F} = \mathbb{R}^2$ and and the space of allowable transformation $\mathcal{A}$ is the set of planar affine transformations. We will present the solution in a form that can be easily generalized to other case.

4.1 Regular Patterns

Let $\mathcal{A}^*$ be the vector space generated by $\mathcal{A}$. In the setting of affine transformation it is obvious that $\mathcal{A}^* = \mathcal{A}$.

Note:

- In terms of $\mathcal{A}^*$, the regularity condition can be restated as: A pattern is regular iff $\forall t \in \mathcal{A}^*, t|V = 0$ implies that $t = 0$.

The regularity condition in the case case of affine transformation is equivalent to a simple condition. It is well known that a planar affine transformation is completely determined by its restriction to any three noncollinear points. So, it is not hard to see that: *a pattern is regular iff it contains three noncollinear points.* For completeness, we prove this equivalence in the appendix of this section.

We introduce below a measure between a regular pattern and a pattern.

Definition 4.1.1 *If V is a pattern and U is a regular pattern define $\Delta(U,V)$ by*

$$\Delta(U,V) = \sup_{\{t\in\mathcal{A}^*;||t||_U\leq 1\}} ||t||_V.$$

Note:

- The supremum can be replaced by a maximum because $||.||_V$ is continuous and $\{t \in \mathcal{A}^*; ||t||_U \leq 1\}$ is compact due to the fact the U is regular (if U is not regular, the resulting region become unbounded). So, specifically, due to the regularity of U, $\Delta(U,V)$ is always less than ∞. This will become clear as we proceed.

The motivation behind this definition is that

Lemma 4.1.1 $\forall t \in \mathcal{A}^*, ||t||_V \ \leq \ \Delta(U,V)||t||_U$

proof: If $t=0$, then $||t||_V = ||t||_U = 0$, hence $||t||_V = \Delta(U,M)||t||_U$. On the other hand if $t \neq 0$, then $||t||_U \neq 0$ because U is regular, so

$$\frac{||t||_V}{||t||_U} = ||\frac{t}{||t||_U}||_V \leq \sup_{\{h\in\mathcal{A}^*;||h||_U=||\frac{t}{||t||_U}||_U=1\}} ||h||_V \leq \sup_{\{h\in\mathcal{A}^*;||h||_U\leq 1\}} ||h||_V = \Delta(U,V),$$

where the first equality is meaningful because $\mathcal{A}^*$ is a vector space. □

Notes:

- $||t||_U \leq 1$ is equivalent to $t(U)$ is a subset of the 1-neighborhood of the origin. Moreover, the 1-neighborhood of the origin is a unit square centered at the origin; this the case because we are using the ∞-norm. These observations lead to the following intuitive view of $\Delta(U,V)$: The value of $\Delta(U,V)$ is the radius of the smallest enclosing square of the region spanned by $t(V)$ as t spans the subset of $\mathcal{A}^*$ consisting of the transformations that maps U inside the unit square centered at the origin. See Figures 4-1,4-2, and 4-3 for different examples.

- If $U \subset V$, then $\Delta(U, V) \geq 1$.

Appendix

Proposition 4.1.2 *A pattern is regular iff it contains three noncollinear points*

proof: Assume that V contains 3 noncollinear points, say α_1, α_2 and α_3. In order to show that V is regular we have to demonstrate that $\forall t \in \mathcal{A}^*, t|V = 0$ implies $t = 0$. Consider any $t \in \mathcal{A}^*$ with $t|V = 0$. Specifically we have $t(\alpha_i) = 0$ for $i = 1, 2, 3$. Let $H_{2\times 2}$ and $r_{2\times 1}$ be respectively the linear transformation matrix and the translation vector of t (i.e. $t(x) = Hx + r, \forall x \in \mathbb{R}^{2\times 1}$). So we have $H\alpha_i + r = 0$ for $i = 1, 2, 3$. Noting that $H\alpha_i + r = [H|r] \begin{bmatrix} \alpha_i \\ 1 \end{bmatrix}$, we obtain

$$[H|r] \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ 1 & 1 & 1 \end{bmatrix} = 0.$$

The matrix $A = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ 1 & 1 & 1 \end{bmatrix}$ is invertible because α_1, α_2 and α_3 are noncollinear: If $A\,[p_1 \quad p_2 \quad p_3]^T = 0$, we get $p_1\alpha_1 + p_2\alpha_2 + p_3\alpha_3 = 0$ and $p_1 + p_2 + p_3 = 0$, but $\alpha_1, \alpha_2, \alpha_3$ are affinely independent (the general notion of noncollinearity), so $p_1 = p_2 = p_3 = 0$ by the definition of affine independence. It follows that $[H|r] = 0$ and hence $t = 0$.

To proof the converse, assume that all points in V belongs to the same line. We need to find a nonzero element t of $\mathcal{A}^*$ with $t|V = 0$. Say that $V = \{\alpha_1, \ldots, \alpha_s\}$. To construct t, let β be a planar point not on the line containing V and not at the origin. The points α_1, α_2 and β are noncollinear, so there exists a unique affine transformation t that maps α_1 and α_2 to the origin and β to β. We know that $t \neq 0$ because $t(\beta) = \beta$. We also have $t(\alpha_1) = t(\alpha_2) = 0$. To see why $t(\alpha_i) = 0$, for all $i \geq 3$, let H and r be respectively the linear transformation matrix and the translation vector of t. Because the points of V are on the same line , for each $i \geq 3$, we can find a a real number k_i such that $\alpha_i = k_i(\alpha_2 - \alpha_1) + \alpha_1$. For each $i \geq 3$, we have

$$t(\alpha_i) = H\alpha_i + r = H(k_i(\alpha_2 - \alpha_1) + \alpha_1) + r = k_i((H\alpha_2 + r) - (H\alpha_1 + r)) + (H\alpha_1 + r) = 0.$$

□

4.2 Unbounded Tolerance

We describe a localization algorithm $\mathcal{Q}_1$ that has an unbounded tolerance. The algorithm takes as input M, S , ξ and any 3-point regular subset U of M. We show that the algorithm achieves in $O(n^3 m \log n)$ time an $\xi\Delta(U, M)$ tolerance, a value that may be very large for an arbitrary U.

Note that if U is a 3-point regular subset of M, and f is a mapping from U to S, then there exits a unique affine transformation t that agree with f on U, i.e., $t|U = f$. This is an other way of saying that a planar affine transformation is uniquely specified by it restriction to any three noncollinear points.

Algorithm 4.2.1 *Consider the following algorithm that takes as input: the model M, the observation S, the noise bound ξ, and any 3-point regular subset U of M.*

$\mathcal{Q}_1$ = " *On input* (M, S, U, ξ)

1. compute $\Delta(U, M)$

2. initialize $\mathcal{X}$ *to the empty set*

3. repeat the following for each mapping f *from* U *to* S

4. compute the unique affine transformation that agrees with f *on* U,
i.e, compute the element t *of* $\mathcal{A}$ *satisfying* $f = t|U$

5. check if t *is* $(\xi + \xi\Delta(U, M))$*-feasible*

6. if so add t *to* $\mathcal{X}$

7. return $\mathcal{X}$"

$\mathcal{Q}_1$ works by trying all the mappings form U to S. For each mapping it computes the unique affine transformation that agrees with the mapping on U, then it tests the $(\xi + \xi\Delta(U, M))$-feasibility of the hypothesized transformations before adding them to the output.

Proof of correctness

In order to establish the result, we need first to look at the feasibility condition from a different angle.

Lemma 4.2.2 *A transformation $t \in \mathcal{A}$ is ε-feasible iff there exits a mapping $g : M \to S$ such that $\|g - t\|_M < \xi$.*

proof: If t is ε-feasible , then by definition $t(M) \subset B_\xi(S)$. So for each point α in M there exists a point β in S such that $\alpha \in B_\xi(\beta)$ or $\|\alpha - \beta\| < \xi$. If for each α in M, we let $g(\alpha)$ be any such β, we obtain a mapping $g : M \to S$ that is ξ-close to t with respect to ρ_M.
If on the other hand there exits a mapping $g : M \to S$ s.t. $\|g - t\|_M < \xi$. Then $\forall \alpha \in M, \|t(\alpha) - g(\alpha)\| < \xi$ or $t(\alpha) \subset B_\xi(g(\alpha)) \subset B_\xi(S)$. So, $t(M) \subset S$. □

Theorem 4.2.3 (An $\xi\Delta(U, M)$ tolerance) *The output $\mathcal{X}$ of $\mathcal{Q}_1$ is a set of $(\xi + \xi\Delta(U, M))$-feasible transformations whose $\xi\Delta(U, M)$-neighborhood covers all the ξ-feasible transformations.*

Proof: We have to show that

(i) covering condition: for each ξ-feasible transformations t', there exists a transformation t in $\mathcal{X}$ such that $\|t - t'\|_M < \xi\Delta(U, M)$.

(ii) relaxed feasibility condition: each transformations in $\mathcal{X}$ is $(\xi + \xi\Delta(U, M))$-feasible

First (ii) is true because only those transformations that passed the feasibility test at line 5 where added to the output. We only have to proof (i). To prove (i) consider any ξ-feasible transformations t'. Using Lemma 4.2.2 let $g : M \to S$ be such that $\|t' - g\|_M < \xi$. And finally let t be the unique transformation in $\mathcal{A}$ satisfying $t|U = g|U$. We show first that $\|t - t'\|_M < \xi\Delta(U, M)$, then we conclude that $t \in \mathcal{X}$. This will establish (ii).
We have

$$\begin{aligned} \|t - t'\|_M &\leq \Delta(U, M)\|t - t'\|_U && (4.1) \\ &= \Delta(U, M)\|g - t'\|_U && (4.2) \\ &\leq \Delta(U, M)\|g - t'\|_M && (4.3) \\ &< \xi\Delta(U, M), && (4.4) \end{aligned}$$

where (4.1) follows from Lemma 4.1.1, (4.2) follows from the fact that $t|U = g|U$, and (4.3) from the fact $U \subset M$. Now

$$\|t - g\|_M \leq \|g - t'\|_M + \|t - t'\|_M \leq \|g - t'\|_M + \Delta(U, M)\|g - t'\|_M < \xi + \xi\Delta(U, M),$$

where the second inequality follows from (4.3). To see why $t \in \mathcal{X}$, let $f = g|U$. Note that f is mapping from U to S. Moreover, t is the unique transformation in $\mathcal{A}$ satisfying $t|U = f$. So t was computed at Line 4 when the mapping f was considered in the loop of line 3. Now t must have passed the test at Line 5 because by Lemma 4.2.2 it is $(\xi + \xi\Delta(U, M))$-feasible due to the fact that $\|t - g\|_M < \xi + \xi\Delta(U, M)$. So, it must have been added to $\mathcal{X}$ at Line 6. □

We have shown that the tolerance in the output of $\mathcal{Q}_1$ is $\xi\Delta(U, M)$. The reader might be a bit confused because the value of $\Delta(U, M)$ is computed in $\mathcal{Q}_1$. At the end of this section, we will briefly describe an alternative algorithm $\mathcal{Q}_1'$ that leads to the same worst case tolerance $(\xi\Delta(U, M))$. The alternative algorithm might be more transparent because the value of $\Delta(U, M)$ is not needed in the computational process. The alternative algorithm even leads to lower average tolerance, but it is more complicated than $\mathcal{Q}_1$.

Time analysis

We will elaborate on the computational tools needed to realize $\mathcal{Q}_1$. We will postpone the computation of $\Delta(U, M)$ until the next section; In Proposition 4.3.4 we show how to compute $\Delta(U, M)$ in $O(m)$ time. We need to know how to do the feasibility test in Line 5 and how to compute the affine transformation that agree with the hypothesized mapping on U in Line 4. Both tasks are essentially easy. We first show that the feasibility test can be done in $O(m \log n)$ assuming an $O(n \log n)$ preprocessing time of S. Then we describe how the transformation t in Line 3 can be efficiently computed in constant time. We have $O(|S|^{|U|}) = O(n^3)$ mapping from U to S and each mapping require an $O(m \log n)$ processing time. After adding the $O(n \log n)$ time needed to preprocess S, we end up with an $O(n \log n + n^3 m \log n) = O(n^3 m \log n)$ time needed by $\mathcal{Q}_1$ to halt.

Given a transformation $t \in \mathcal{A}$, t is ε-feasible iff $t(M) \subset B_\varepsilon(S)$. This is equivalent to saying $\forall \alpha \in M$, $B_\varepsilon(t(\alpha)) \subset B_\varepsilon(S)$ or $B_\varepsilon(t(\alpha)) \cap S \neq \emptyset$. To sum up : t is ε-feasible iff $B_\varepsilon(t(\alpha)) \cap S \neq \emptyset, \forall \alpha \in M$. Note that $B_\varepsilon(x)$ is a square centered at x, since we are using the ∞-norm. So, we can test the ε-feasibility of t by passing over each $\alpha \in M$, and checking whether the square $B_\varepsilon(t(\alpha)$ contain points from S. We declare t to be ε-feasible iff for each $\alpha \in M$, the intersection of the corresponding square with S is nonempty. Assuming that S is suitably preprocessed, the number of points in the intersection of S with a query square can be efficiently computed. In fact, assuming that an $O(n \log n)$ preprocessing for a given n-points planar set, the number of points in the intersection of a query rectangular region with the set can be computed in $O(\log n)$ (See [2] page 578). Thus, assuming an $O(n \log n)$ preprocessing time for S the ε-feasibility of t can be tested in $O(m \log n)$.

We are left with the simple task of computing the affine transformation that agrees on U with a given mapping from U to S.

Notation: Each planar affine transformation t is uniquely specified by a 2×2 matrix H and a 2×1 translation vector r. We denote this correspondence by $t \sim (H, r)$. So, $t \sim (H, r)$ means $t(x) = Hx + r, \forall x \in \mathbb{R}^{2\times 1}$.

Say that f is a mapping from U to S. If t is the affine transformation that agrees with f on U, then $t \sim (H, r)$, where

$$[\,H \quad | \quad r\,] = [\,f(\alpha_1) \quad f(\alpha_2) \quad f(\alpha_3)\,] \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ 1 & 1 & 1 \end{bmatrix}^{-1}_{3\times 3}, \tag{4.5}$$

and $\alpha_1, \alpha_2, \alpha_3$ are the (2×1) points of U. The reason is simple, we have $t(\alpha_i) = H\alpha_i + r = f(\alpha_i), i = 1, 2, 3$. Noting that $H\alpha_i + r = [\,H \quad | \quad r\,] \begin{bmatrix} \alpha_i \\ 1 \end{bmatrix}$, we obtain

$$[\,H \quad | \quad r\,] \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ 1 & 1 & 1 \end{bmatrix} = [\,f(\alpha_1) \quad f(\alpha_2) \quad f(\alpha_3)\,].$$

Equation (4.5) follows from the fact that the matrix $A_U = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ 1 & 1 & 1 \end{bmatrix}$ is invertible due to the noncollinearity of $\alpha_1, \alpha_2, \alpha_3$. So, given a mapping f from U to S the (H, r) representation of the affine transformation that agree with f on U can be computed in constant time by multiplying the matrix 2×3 matrix $[\, f(\alpha_1) \quad f(\alpha_2) \quad f(\alpha_3)\,]$ with the inverse of the 3×3 matrix A_U. Note that A_U depends only on U, so the inverse need to be computed only once.

To sum up, we have shown that

Theorem 4.2.4 *$\mathcal{Q}_1$ runs in $O(n^3 m \log n)$ time.*

Notes:

- If we are given some constraints on the $\mathcal{A}$ (for example, on the scaling and slanting parameters) the constraints can be used in the hypothesis generation process to eliminate many mappings from U to S. The mappings that are consistent with constraints can be generated by a recursive process on the points of U. This leads to a large speed up factor.

- If the norm used on $\mathcal{F}$ is the 2-norm, the same tolerance and running time bounds holds. The only difference is in the feasibility test implementation. Here the test can be done by computing the nearest neighbor in of $t(\alpha)$, for each $\alpha \in T$. This requires constructing the Voronoi diagram of S instead of preprocessing it for rectangular range searching purposes.

An equivalent algorithm [1]

We briefly describe an alternative algorithm $\mathcal{Q}_1'$ that leads to the same worst case tolerance $(\xi\Delta(U, M))$ in the same worst case running time. $\mathcal{Q}_1'$ might be more transparent than $\mathcal{Q}_1$ because the value of $\Delta(U, M)$ is not needed in the computational process. It even leads to lower average tolerance, but it is more complicated than $\mathcal{Q}_1$.

Algorithm 4.2.5 *Consider the following algorithm that takes as input: the model M, the observation S, the noise bound ξ, and any 3-point regular subset U of M.*

[1] This part can be skipped without any loss of continuity.

$\mathcal{Q}'_1$ = " *On input* (M, S, U, ξ)

1. *initialize* $\mathcal{X}$ *to the empty set*
2. *repeat the following for each mapping* f *from* U *to* S
3. *compute the unique affine transformation that agrees with* f *on* U, *i.e, compute the element* t *of* $\mathcal{A}$ *satisfying* $f = t|U$
4. *compute the set of transformations* $C = \{t' \in \mathcal{A}; \|t' - f\|_U < \xi\}$
5. *for each* $\alpha \in M$
6. *compute the planar region* $R = C(\alpha) + B_\xi(0)$
7. *check if* $R \cap S \neq \emptyset$
8. *if each point in* M *leads to a nonempty intersection, add* t *to* $\mathcal{X}$
9. *return* $\mathcal{X}$"

Note: $C(\alpha)$ means $\{t'(\alpha); t' \in C\}$ and $C(\alpha) + B_\xi(0)$ means $\{x + y; x \in C(\alpha)$ and $y \in B_\xi(0)\}$, i.e., the Minkowsky sum of $C(\alpha)$ with the ξ-square centered at the origin.

We will show in Appendix A that $\mathcal{Q}'_1$ is equivalent to $\mathcal{Q}_1$ in the sense that the output of $\mathcal{Q}'_1$ satisfies the same conditions of that of $\mathcal{Q}'_1$. In other words, the output of $\mathcal{Q}_1$ is a set of $(\xi + \xi\Delta(U, M))$-feasible transformations whose $\xi\Delta(U, M)$-neighborhood covers all the ξ-feasible transformations.

Regarding the implementation of $\mathcal{Q}'_1$, we shall argue in Appendix A also that $\mathcal{Q}'_1$ runs in $O(n^3 m \log n)$ time. Briefly, each of the C's computed at Line 4 is a $6D$-convex polytope that can be written as the product of two $3D$-convex polytopes C_1 and C_2 each having $O(1)$ extreme points and computable in $O(1)$ time. As for the R's, each R is a planar rectangular region that can be computed from the extreme points of C_1 and C_2 in $O(1)$ time.

The link between $\mathcal{Q}_1$ and $\mathcal{Q}'_1$ is that the diameter of the smallest enclosing square of $C(\alpha)$ can be as high as $\xi\Delta(U, M)$. Thus, the diameter of the smallest enclosing square of R is bounded by $\xi + \xi\Delta(U, M)$. So when the feasibility test in $\mathcal{Q}_1$ is done at $\xi + \xi\Delta(U, M)$, the worst case diameter of R is assumed.

4.3 A Tolerance Bounded by 3ξ

We described in the previous section an algorithm $\mathcal{Q}_1$ that has an $\xi\Delta(U,M)$ tolerance. To minimize tolerance, we have to minimize $\Delta(U,M)$.

Algorithm 4.3.1 *Consider the following algorithm that takes as input: the model M, the observation S, and the noise bound ξ.*

$\mathcal{Q}_2$ = *" On input (M,S,ξ)*

1. Compute U^* the 3-point regular subset of M that minimizes $\Delta(U,M)$, i.e., $\Delta(U^*,M) \leq \Delta(U,M), \forall U$ a 3-point regular subset of M.

2. Run $\mathcal{Q}_1$ on (M,S,U^*,ξ) and return its output. "

Before knowing how to compute U^*, we need to know what can we expect from $\mathcal{Q}_2$, i.e., what values of tolerance we are guarantee to go below?

Theorem 4.3.2 *Any regular pattern M contains a regular subset U consisting of 3 points and satisfying $\Delta(U,M) \leq 3$.*

proof: The norm $||.||$ on the feature space was - by default - the ∞-norm. In this proof we will be dealing with both the ∞-norm and the 2-norm. In order to avoid confusion, we will write $||.||_\infty$ instead of $||.||$. Let $U = \{\alpha,\beta,\gamma\}$, where α and β are the farthest points in M with respect to the 2-norm, and γ is a point farthest from the Line passing through α and β again with respect to the 2-norm. We will show that $\Delta(U,M) \leq 3$. In order to establish this fact, we need the following.

Lemma 4.3.3 *for each point $\theta \in M$, there exist 3 points p_1, p_2, and p_3 in the convex span [2] of U such that $\theta = p_1 + p_2 - p_3$.*

[2] By the convex span of a planar set A, we mean the set of all convex combinations of the points in A. This should not be confused with the convex hull of A which means the minimal subset of A whose convex span is equal to that of A.

proof: Without loss of generality, we can assume that the coordinates of α, β, and γ are respectively $(a, 0), (b, 0)$ and $(0, c)$, where $a, c \geq 0$. So,

$$||\theta_1 - \theta_2||_2 \leq |a - b|, \forall \theta_1, \theta_2 \in M \quad (4.6)$$

$$|\theta_y| \leq c, \forall \theta \in M. \quad (4.7)$$

Note that c should be $\neq 0$ because M is regular. Note also that b should be ≤ 0: First, b can't be in $(0, a]$ because otherwise we get

$$||\gamma - \alpha||_2 = ||(-a, c)||_2 > a > a - b = |a - b|,$$

which contradicts 4.6. Moreover, b can't be larger than a because otherwise we get

$$||\gamma - \beta||_2 = ||(-b, c)||_2 > b \geq b - a = |a - b|,$$

which again contradicts (4.6). Now, consider any $\theta \in M$. We have two cases to consider when $\theta_y \geq 0$ and when $\theta_y < 0$.

Case 1: (when $\theta_y \geq 0$) We can express θ as $\theta = (\theta_x, 0) + (0, \theta_y) - (0, 0)$. We will show that $(\theta_x, 0), (0, \theta_y), (0, 0) \in conv(U)$ (If A is a subset of the plane, by $conv(A)$ we mean the convex span A, i.e., the set of of all convex combinations of the points in A). We start with the origin, $b \leq 0 \leq a$ hence $(0, 0) \in conv\{(a, 0), (b, 0)\} \subset conv(U)$. Now, consider $(\theta_x, 0)$. It is sufficient to demonstrate that $b \leq \theta_x \leq a$, because then $(\theta_x, 0) \in conv\ \{(a, 0), (b, 0)\} \subset conv(U)$. First, θ_x can't be smaller than b, because otherwise we get

$$||\alpha - \theta||_2 = ||(a - \theta_x, -\theta_y)||_2 \geq |a - \theta_x| > |a - b|,$$

which contradicts (4.6). Similarly θ_x can't be larger than a because otherwise we get

$$||\theta - \beta||_2 = ||(\theta_x - b, -\theta_y)||_2 \geq |\theta_x - b| > |a - b|.$$

Finally, consider $(0, \theta_y)$. We are assuming that $\theta_y \geq 0$ and we know from (4.7) that $|\theta_y| \leq c$, so we have $0 \leq \theta_y \leq c$. Therefore $(0, \theta_y) \in conv\{(0,0),(0,c)\} \subset conv(U)$.

Case 2: (when $\theta_y < 0$) We can express θ as $\theta = (0,0) + (\theta_x, 0) - (0, -\theta_y)$. We know form the previous case that $(0,0)$ is in the convex span of U. Using exactly the same argument we conclude also that $(\theta_x, 0) \in conv(U)$. As for $(0, -\theta_y)$, we are assuming that $\theta_y < 0$ and we know from (4.7) that $|\theta_y| \leq c$, so we have $0 \leq -\theta_y \leq c$. It follows that $(0, -\theta_y) \in conv\{(0,0),(0,c)\} \subset conv(U)$. This proves the lemma. $\bigtriangledown$

Now, by definition,

$$\Delta(U, M) = \sup_{\{t \in \mathcal{A}^*; \|t\|_U \leq 1\}} \max_{\theta \in M} \|t(\theta)\|_\infty.$$

The constraint $\|t\|_U \leq 1$ is equivalent to $t(U) \subset B_1(0)$. Because the square $B_1(0)$ is convex, we can write the constraint as $conv(t(U)) \subset B_1(0)$. Now, convexity is invariant under affine transformations, i.e., $conv(t(U)) = t(conv(U))$ (See Webster [1992]). So the constraint is equivalent to $t(conv(U)) \subset B_1(0)$, or in other words, $\|t(x)\|_\infty \leq 1, \forall x \in conv(U)$. Consider any point $\theta \in M$ and any $t \in \mathcal{A}$ satisfying the constraint. Using the previous lemma, write $\theta = p_1 + p_2 - p_3$, for some p_1, p_2, and p_3 in the convex span of U. Say that H and r are the linear transformation matrix and the translation vector representation of t. We have

$$\begin{aligned}
\|t(\theta)\|_\infty &= \|H\theta + r\|_\infty = \|H(p_1 + p_2 - p_3) + r\|_\infty \\
&= \|H(p_1 + p_2 - p_3) + r + r - r\|_\infty \\
&= \|(Hp_1 + r) + (Hp_2 + r) - (Hp_3 + r)\|_\infty \\
&\leq \|Hp_1 + r\|_\infty + \|Hp_2 + r\|_\infty + \|Hp_3 + r\|_\infty \\
&= \|t(p_1)\|_\infty + \|t(p_2)\|_\infty + \|t(p_3)\|_\infty \\
&\leq 3.
\end{aligned}$$

This is true for any $\theta \in M$. It follows that $\Delta(U, M) \leq 3$. $\square$

Notes:

- The bound in the above theorem is tight in the sense that there are M's where the best 3-point regular subset U of M achieves a value of $\Delta(U, M) = 3$. An example is the case when M consists of the four corners of a square (This might be the only set, up to affine transformations, where the bound become tight).

- If the 2-norm is the one under consideration, the same bound holds. In fact the proof of the above theorem did not use any special feature of the ∞-norm.

Returning to Q_2, this means that $\Delta(U^*, M) \leq 3$, and consequently the tolerance achieved by $\mathcal{Q}_2$ is bounded by 3ξ. Now, we are left with task of computing U^*. The following proposition says that if U consists of 3 points, then $\Delta(U, M)$ can be computed in $O(m)$ time. Therefore, U^* can be computed in $O(m^4)$ time because we have $O(m^3)$ 3-point regular U to test before reaching the one that achieves the minimum. We know from the previous section that $\mathcal{Q}_1$ runs in $O(n^3 m \log n)$ time. So $\mathcal{Q}_2$ runs in $O(m^4 + n^3 m \log n)$.

Proposition 4.3.4 *If $U = \{\alpha_1, \ldots, \alpha_k\}$ is a regular subset of M , then*

1.

$$\Delta(U, M) = \max_{\beta \in M} \max_{x \in \mathbb{R}^{1\times 3}; -e_k \leq xA \leq e_k} x \begin{bmatrix} \beta \\ 1 \end{bmatrix}, \tag{4.8}$$

where

$$A = \begin{bmatrix} \alpha_1 & \alpha_2 & \ldots & \alpha_k \\ 1 & 1 & \ldots & 1 \end{bmatrix}_{3\times k} \quad \text{and } e_k = \begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix}_{1\times k}.$$

Moreover, the region defined by the constraint $-e_k \leq xA \leq e_k$ is bounded (this shows that $\Delta(U, M)$ is always less than ∞) .

2. $\Delta(U, M)$ can be computed in $O(mk \log k)$ time.

proof: By definition,

$$\Delta(U, M) = \sup_{\{t \in \mathcal{A}^*; \|t\|_U \leq 1\}} \max_{\beta \in M} \|t(\beta)\| = \max_{\beta \in M} \sup_{\{t \in \mathcal{A}^*; \|t\|_U \leq 1\}} \|t(\beta)\|$$

Using the linear transformation-translation representation of t, we can write

$$\sup_{\{t \in \mathcal{A}^*; \|t\|_U \leq 1\}} \|t(\beta)\|$$

as

$$\sup\left\{\|H\beta + r\|; (H,r) \in \mathbb{R}^{2\times 2} \times \mathbb{R}^{2\times 1} \text{ and } \|H\alpha_i + r\| \leq 1, i = 1, \ldots, k\right\}$$
$$= \sup\left\{\left\|[H|r]\begin{bmatrix}\beta\\1\end{bmatrix}\right\|; [H|r] \in \mathbb{R}^{2\times 3} \text{ and } \left\|[H|r]\begin{bmatrix}\alpha_i\\1\end{bmatrix}\right\| \leq 1, i = 1, \ldots, k\right\}$$
$$= \sup\left\{\left|x\begin{bmatrix}\beta\\1\end{bmatrix}\right|; x \in \mathbb{R}^{1\times 3} \text{ and } \left|x\begin{bmatrix}\alpha_i\\1\end{bmatrix}\right| \leq 1, i = 1, \ldots, k\right\}$$
$$= \sup\left\{\left|x\begin{bmatrix}\beta\\1\end{bmatrix}\right|; x \in \mathbb{R}^{1\times 3}; -e_k \leq xA \leq e_k\right\}.$$

Noting that the region defined by the constraints is symmetric, we obtain

$$\sup_{\{t \in \mathcal{A}^*; \|t\|_U \leq 1\}} \|t(\beta)\| = \sup_{\{x \in \mathbb{R}^{1\times 3}; -e_k \leq xA \leq e_k\}} x\begin{bmatrix}\beta\\1\end{bmatrix}. \tag{4.9}$$

Now we show that the region E =$\{$x $\in \mathbb{R}^{1\times 3}; -e_k \leq xA \leq e_k\}$ is bounded. The pattern U is regular, so it contains 3 noncollinear points. Without loss of generality we can assume that $\alpha_1, \alpha_2, \alpha_3$ are noncollinear. Let $B = \begin{bmatrix}\alpha_1 & \alpha_2 & \alpha_3\\ 1 & 1 & 1\end{bmatrix}$ and $F = \{x \in \mathbb{R}^{1\times 3}; -e_3 \leq xB \leq e_3\}$. $E \subset F$, so it is sufficient to show that F is bounded. If $x \in F$, then $-e_3 \leq xB \leq e_3$. Hence $\|xB\|_\infty \leq 1$. As we argued in the proof of Proposition 4.1.2, the matrix B is invertible because $\alpha_1, \alpha_2, \alpha_3$ are noncollinear. So $\sigma_{\text{min}}(B)$, the smallest singular value of B, is not zero. Therefore

$$\|x\|_\infty \leq \frac{\|xB\|_\infty}{\sigma_{\text{min}}(B)} \leq \frac{1}{\sigma_{\text{min}}(B)} < \infty$$

It follows that F (and hence E) is bounded.

The region E is also closed, so it is compact. Therefore, the continuous function $h_\beta(x) = x\begin{bmatrix}\beta\\1\end{bmatrix}$ achieves a maximum over E. So the supremum in (4.9) can be replaced by a maximum, which leads to Equation 4.8.

Now, to compute $\Delta(U, M)$, we have to solve m linear programs, each for an α in

M, and each of 3 variables and $2k$ constraints. Being 3-dimensional linear programs, each is solvable in $O(k \log k)$ (See Muller & Preparata [1978]), so the overall time needed to compute $\Delta(U, M)$ is $O(mk \log k)$. □

Note:

- The same expression of $\Delta(U, M)$ holds in the 2-norm setting.

To sum up, we have just proved that

Theorem 4.3.5 (a tolerance bounded by 3ξ) $\mathcal{Q}_2$ *computes in* $O(m^4 + n^3 m \log n)$ *time a set of* $(\xi + 3\xi)$*-feasible transformations whose* 3ξ*-neighberhood covers all the* ξ*-feasible transformations.*

Notes:

- Observe that the proof of Theorem 4.3.2 is constructive. In fact, the U suggested by the proof can be computed in $O(m \log m)$: We can compute the two farthest points in M in $O(m \log m)$ time and the point farthest from the line passing through them in $O(m)$ time, which leads to an overall $O(m \log m)$ time need by the algorithm induced from the proof to halt. The reason behind finding U^* by brute force is that the U suggested by that proof does not always minimize $\Delta(U, M)$. The bound 3 is tight in the sense that there are rare M's (probably only one up to affine transformations) where the best U achieves a value of $\Delta(U, M)$ equal to 3. But, on the average, the value of $\Delta(U^*, M)$ is less than this. It was found experimentally that $\Delta(U^*, M)$ is on the average in the order of 1.8. So, finding U^* by brute force leads - on the average - to a lower tolerance. Note that the algorithm induced from the proof of Theorem 4.3.2 can relaxed to give hints to the brute force approach in order to discard many computational branches at earlier stages. Note also that the computation of U^* can be a seen as a model preprocessing stage.

- Figure 4-1 illustrates several values of $\Delta(U^*, M)$ for different M's, M is shown together with the U^* that minimize $\Delta(U, M)$.

- To see to what degree a bad choice of U can lead to a large value of $\Delta(U, M)$, see Figure 4-2. In fact, for any real number N, there exists an M and a 3-point regular subset U of M with $\Delta(U, M) \geq N$.

We know that the lowest value of $\Delta(U, M)$ is 1 because $U \subset M$. It is interesting to note that this value is achieved when U is the convex hull of M.

Theorem 4.3.6 $\Delta(convhull^3(M), M) = 1$.

proof: Let $V = convhull(M)$. We show that $||t||_V \leq 1$ if and only if $||t||_M \leq 1, \forall t \in A$. This lead us to

$$\Delta(V, M) = \sup_{\{t \in \mathcal{A}^*; ||t||_V \leq 1\}} ||t||_M = \sup_{\{t \in \mathcal{A}^*; ||t||_M \leq 1\}} ||t||_M = 1.$$

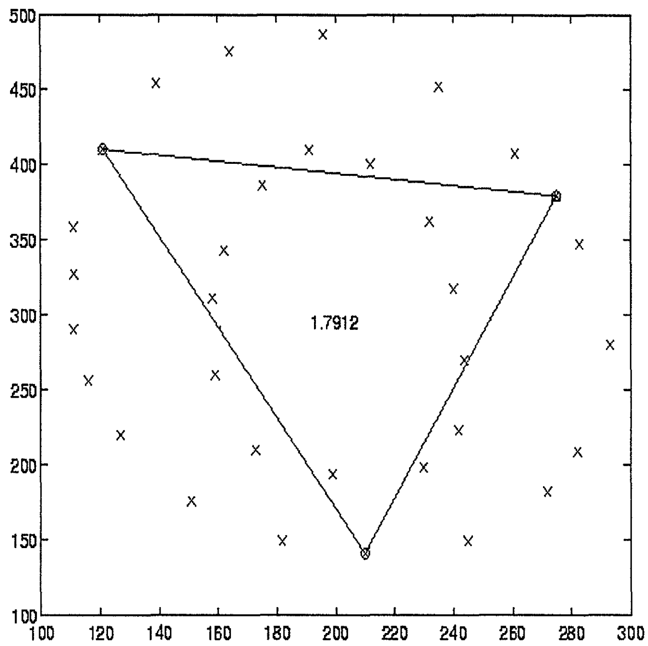
Because $V \subset M$, we have $||t||_V \leq ||t||_M$, hence $||t||_M \leq 1$ implies $||t||_V \leq 1$. To show the other direction, assume that $||t||_V \leq 1$. $||t||_V \leq 1$ is equivalent to $t(V) \subset B_1(0)$. Because the square $B_1(0)$ is convex, we also have $conv(t(V)) \subset B_1(0)$. Convexity is invariant under affine transformations, so $conv(t(V)) = t(conv(V))$. On the other hand, $conv(V) = conv(M)$ because V is the convex hull of M. It follows that $t(conv(M)) \subset B_1(0)$, hence $t(M) \in B_1(0)$, or $||t||_M \leq 1$. □

This makes sense, since - as we argued before - the value of $\Delta(U, M)$ is the radius of the smallest enclosing square of the region spanned by $t(M)$ as t spans the subset of $\mathcal{A}^*$ consisting of the transformations that maps U inside the unit neighborhood centered at the origin. So $\Delta(U, M)$ - in some sense - measures to what degree the convex span of U is a good approximation of the convex span of M. Figure 4-2 shows different values of $\Delta(U, M)$ in the case when U consists of more than 3 points. It is clear from the defintion of $\Delta(U, M)$ and from those figures that $\Delta(U, M)$ decreases as the number of points in U increases (compare Figure 4-1.d and Figure 4-2.a).

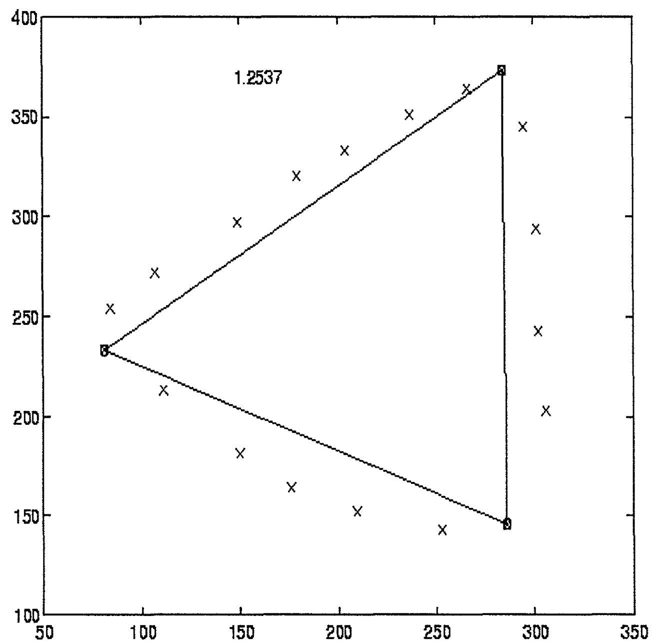
To sum up, in the setting where occlusion is not allowed, we have shown how a tolerance bounded by 3ξ can be achieved in $O(m^4 + n^2 m \log n)$ time. Such a

[3] By the convex hull of a planar set A, we mean the minimal subset of A whose convex span is equal to that of A. This should not be confused with the convex span of A which we denote by $conv(A)$ and by which we mean the set of of all convex combinations of the points in A.

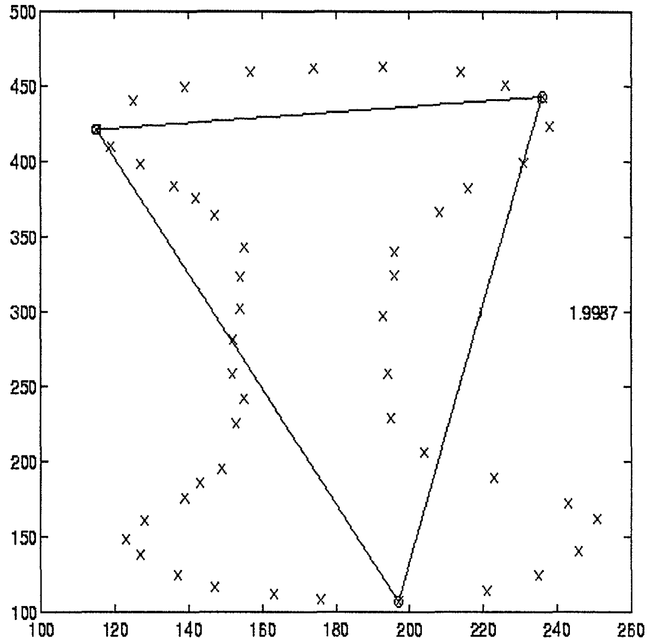
tolerance might be sufficient for practical purposes. In fact the author is not aware of a localization algorithm that achieves a similar or better output quality in a similar or lower worst case running time. But the natural question to ask is: can we do better than this? i.e., can we achieve lower values of tolerance? This is the topic of the next section.



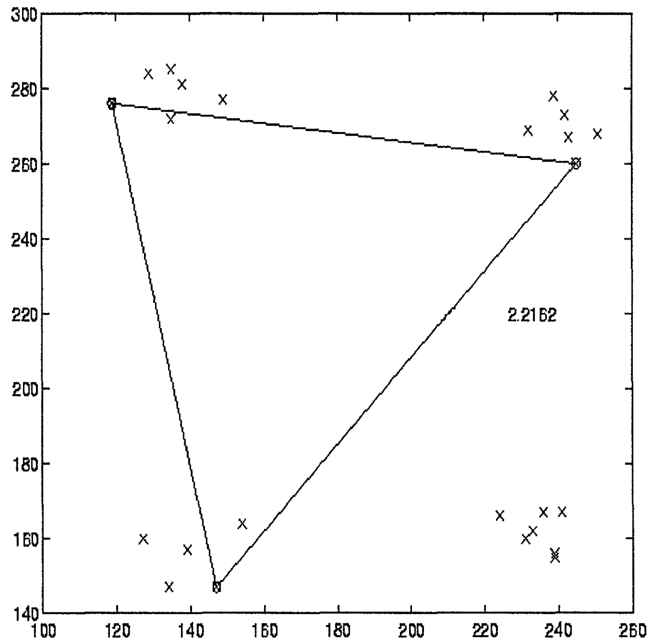
(a) $\Delta(U^*, M) = 1.7912$



(b) $\Delta(U^*, M) = 1.2537$

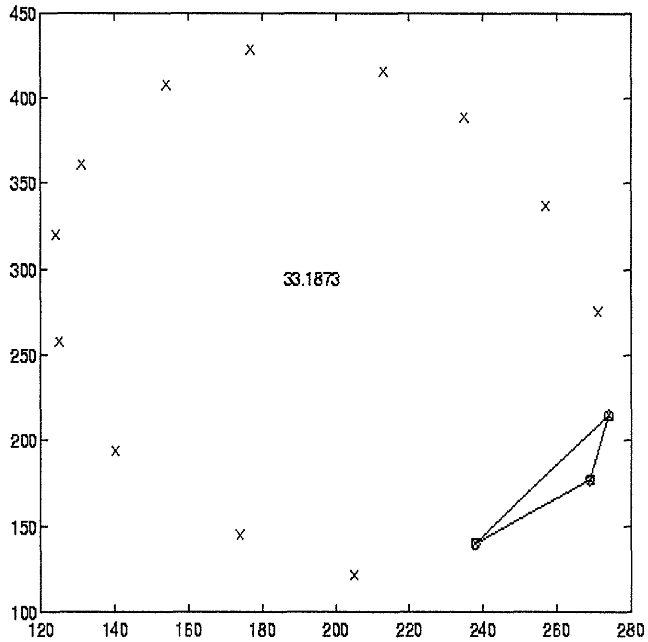


(c) $\Delta(U^*, M) = 1.9987$

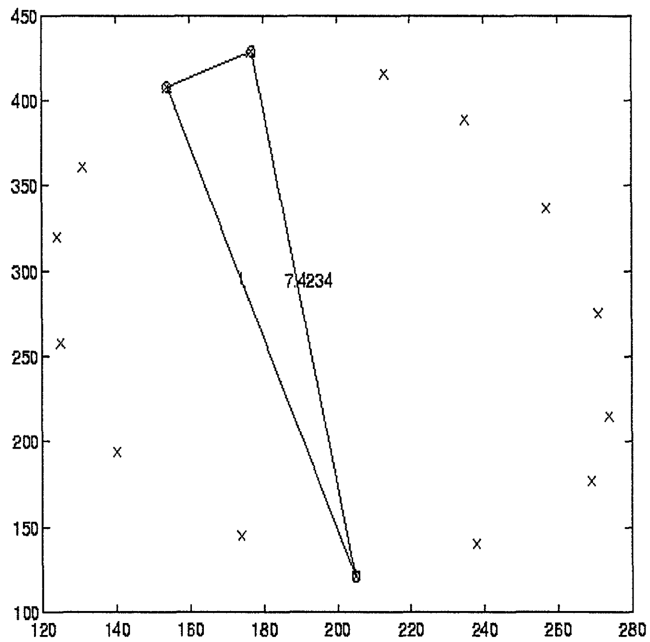


(d) $\Delta(U^*, M) = 2.2162$

Figure 4-1: This figure illustrates four models each with the corresponding 3-point regular subset U^* that minimizes $\Delta(U, M)$. In each case, the points of M are the set of the $\times$'s, the points of U^* are distinguished by the $\circ$'s and the edges of their convex span.

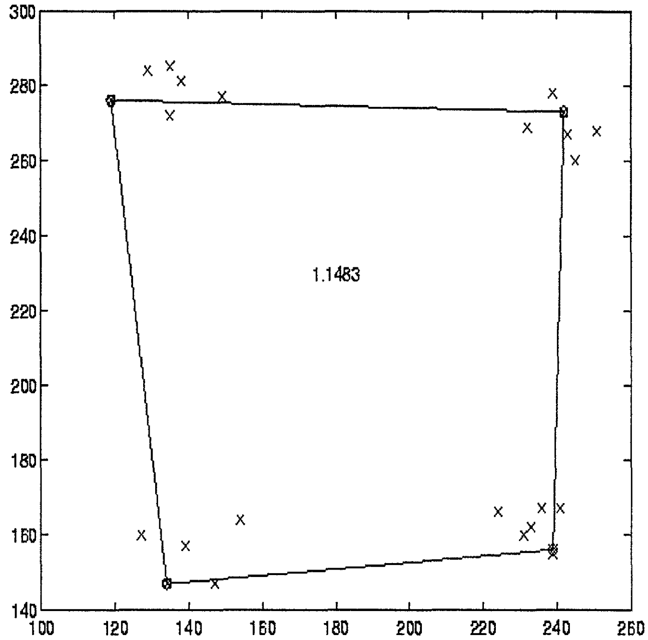


(a) $\Delta(U, M) = 33.1873$

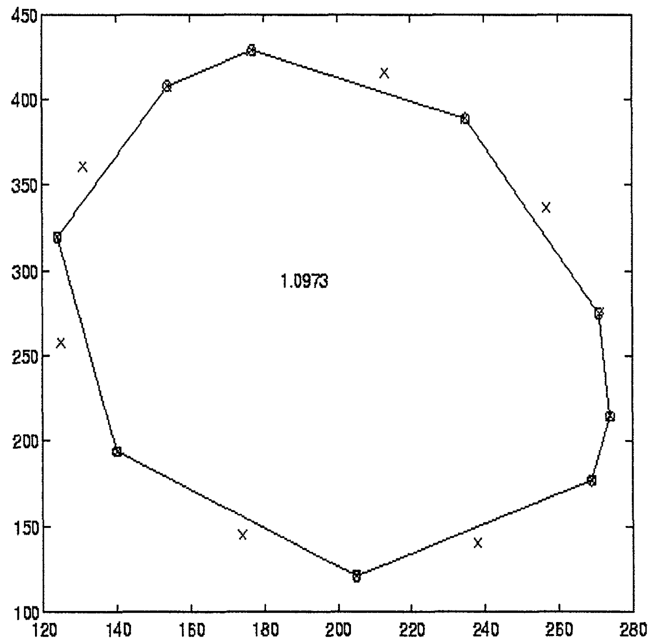


(b) $\Delta(U, M) = 7.4234$

Figure 4-2: This figure illustrates how a bad selection of U can lead to very large value of $\Delta(U, M)$. In each case, the points of M are the set of the $\times$'s, the points of U are distinguished by the $\circ$'s and the edges of their convex span.



(a) $\Delta(U, M) = 1.1483$



(b) $\Delta(U, M) = 1.0973$

Figure 4-3: This figure illustrates values of $\Delta(U, M)$ when U is a subset of M containing more than 3 points. In each case, the points of M are the set of the $\times$'s, the points of U are distinguished by the $\circ$'s and the edges of their convex span.

4.4 Any Given Tolerance

In this section we build on the results of the previous sections to reach any given uncertainly. In Section 4.2, we started with an algorithm called $\mathcal{Q}_1$ that had an $\xi\Delta(U, M)$ tolerance. In the previous section, we decreased the tolerance of $\mathcal{Q}_1$ by minimizing $\Delta(U, M)$ and we proved that the minimum is less than 3. Now, once $\Delta(U, M)$ is minimized, how can we reach lower values of tolerance? we are not allowed to decrease ξ because it is a given input. Is this true? We can: decrease ξ and replace S by a new observation whose neighborhood with respect to the new ξ is equal to the original ξ-neighborhood of S. The point is that the ξ-feasible transformations depends on the ξ-neighborhood of S, not on the points in S.

We will be working with two different observations S and S'. So to avoid any confusion in the definitions of feasible transformations, we need a more accurate notation.

Definition 4.4.1 *Define $\mathcal{C}_\xi(M, S)$ to be the set of ξ-feasible transformations with respect to the model M and the observation S, i.e.,*

$$\mathcal{C}_\xi(M, S) = \{t \in \mathcal{A}; t(M) \subset S\}.$$

Theorem 4.4.1 (Any given tolerance) *Given any tolerance bound μ, we can compute in $O(n^4 + (\frac{3\xi}{\mu})^6 n^3 m \log n)$ time a set of $(\xi + \mu)$-feasible transformations, whose μ-neighborhood covers all the ξ-feasible transformations.*

Proof: Consider the following algorithm that takes as input: The model M, the observation S, the error bound ξ, and a tolerance bound μ.

$\mathcal{Q}_3 =$ " On input (M, S, ξ, μ)

1. Compute U^* the 3-point regular subset of M that minimizes $\Delta(U, M)$
2. Compute a planar set S' satisfying $B_{\frac{\mu}{\Delta(U^*,M)}}(S') = B_\xi(S)$, i.e., whose $\frac{\mu}{\Delta(U^*,M)}$-neighborhood is equal to the ξ-neighberhood of S.
3. Run $\mathcal{Q}_1$ on $(M, S', U^*, \frac{\mu}{\Delta(U^*,T)})$ and return its output. "

Consider the following realization of $\mathcal{Q}_3$. Let $\Delta^* = \Delta(U^*, M)$. We can compute S' as follows: Pick each $\beta \in S$ and cover the square $B_\xi(\beta)$ by a collection Z_β of squares each with a radius $\frac{\mu}{\Delta^*}$ in such a way that the union of those squares is equal to $B_\xi(\beta)$. Then let S' to be the set of the centers of the squares in the union of the Z_β's as β spans S. Each $B_\xi(\beta)$ can be covered by $\lceil \frac{\xi}{\mu/\Delta^*} \rceil^2$ such squares. So a set S' containing $n\lceil \frac{\Delta^* \xi}{\mu} \rceil^2$ points and satisfying $B_{\frac{\mu}{\Delta^*}}(S') = B_\xi(S)$ can be constructed in linear time. We know from Theorem 4.3.2 that $\Delta^* \leq 3$, so the cardinality of S' is at most $n\lceil \frac{3\xi}{\mu} \rceil^2$, i.e., $|S'| = O((\frac{3\xi}{\mu})^2 n)$.

According to Theorem 4.2.4, the time needed by $\mathcal{Q}_1$ to halt is

$$O(|S'|^3 m \log |S'|) = O(((\tfrac{3\xi}{\mu})^2 n)^3 m \log (\tfrac{3\xi}{\mu})^2 n) = O((\tfrac{3\xi}{\mu})^6 n^3 m \log n).$$

We argued in the previous sections that U^* can be computed in $O(m^4)$ time. So, $\mathcal{Q}_3$ runs in $O(m^4 + (\frac{3\xi}{\mu})^6 n^3 m \log n)$ time.

Let $\xi' = \frac{\mu}{\Delta(U^*, T)}$ and let $\mathcal{X}$ be the output of $\mathcal{Q}_1$ and hence that of $\mathcal{Q}_3$. From Theorem 4.2.3, we know that $\mathcal{X}$ is a subset of $\mathcal{C}_{\xi' + \xi'\Delta^*}(M, S')$ whose $\xi'\Delta^*$-neighborhood covers $\mathcal{C}_{\xi'}(M, S')$.

The set S' is constructed in such a way that $B_\xi(S) = B_{\xi'}(S')$. It follows from Definition 4.4.1 that

$$\mathcal{C}_{\xi'}(M, S') = \mathcal{C}_\xi(M, S). \tag{4.10}$$

On the other hand, because $B_\xi(S) = B_{\xi'}(S')$, we also have

$$\begin{aligned}
B_{\xi'+\xi'\Delta^*}(S') &= B_{\xi'+\frac{\mu}{\Delta^*}\Delta^*}(S') = B_{\xi'+\mu}(S') = \bigcup_{\alpha \in S'} B_{\xi'+\mu}(\alpha) = \bigcup_{\alpha \in S'} \bigcup_{x \in B_{\xi'}(\alpha)} B_\mu(x) \\
&= \bigcup_{x \in B_{\xi'}(S')} B_\mu(x) = \bigcup_{x \in B_\xi(S)} B_\mu(x) \\
&= \bigcup_{\alpha \in S} \bigcup_{x \in B_\xi(\alpha)} B_\mu(x) = \bigcup_{\alpha \in S} B_{\xi+\mu}(\alpha) = B_{\xi+\mu}(S),
\end{aligned}$$

thus

$$\mathcal{C}_{\xi'+\xi'\Delta^*}(M, S') = \mathcal{C}_{\xi+\mu}(M, S), \tag{4.11}$$

which follows also from Definition 4.4.1. Noting that $\xi'\Delta^* = \mu$ and using (4.10) and (4.11) we obtain that $\mathcal{X}$ is a subset of $\mathcal{C}_{\xi+\mu}(M,S)$ whose μ-neighborhood covers $\mathcal{C}_{\xi}(M,S)$, which is the notational form of the theorem statement. $\square$

Notes:

- The exact time constant is $(\frac{\Delta(U^*,M)\xi}{\mu})^6$ which is - on the average - less than $(\frac{3\xi}{\mu})^6$ (see Figure 4-1).

- It is important to understand that $\mathcal{Q}_3$ is a simple algorithm presented for the sake of a simple proof of the worst case time bound in the above theorem. Rather than dealing with $\mathcal{Q}_1$ as a black box as is the case with $\mathcal{Q}_3$, $\mathcal{Q}_1$ can be opened and adapted to the setting by many techniques that leads to a better average running time. One such technique, is a multiresolution approach that gradually decrease the tolerance. The idea is that by verifying the hypothesized transformations at earlier stages, many computational branches can be discarded if the transformations fail the feasibility test.

- Even in $\mathcal{Q}_3$ the technique we used to construct S' contains only what is needed to prove the worst case bound. By adding another stage to the process of construction, we might be able to remove many points from S' while keeping $B_{\frac{\mu}{\Delta^*}}(S') = B_{\xi}(S)$. In a rather general setting, we will present in Chapter 6 a sampling algorithm that takes as input a pattern V and a real number $\varepsilon > 0$. The output of the sampling algorithm is a locally optimal sampling of of V, i.e., a subset V' of V satisfying: $B_{\varepsilon}(V') = B_{\varepsilon}(V)$ and $\forall \alpha \in V', B_{\varepsilon}(V' - \{\alpha\}) \neq B_{\varepsilon}(V)$. As far as the construction of S' is concerned, after constructing S' as explained in the proof, we can run the sampling algorithm on $(S', \frac{\mu}{\Delta^*})$ and assign its output to S'. This technique is supposed to dramatically reduce the number of points in S' when S is dense with respect to ξ.

- The error bound ξ is a number used to create a mathematical *model* for the noise. Unless the value assigned to ξ was a magic number, going for a very small μ compared to ξ is practically meaningless.

- In the 2-norm setting the same result holds asymptotically. The idea behind $\mathcal{Q}_3$ is highly dependent on the usage of the ∞-norm. The reason is that not any neighborhood is a space filler like the square. Nevertheless, the asymptotic time bound is generalizable to the 2-norm. In this setting, it is clear that we can not cover a ball by a collection of smaller balls, each lying inside the larger ball. What we can do is cover the ball by a collection of smaller balls, the center of each lying inside the larger ball. This construction will enlarge the ξ-neighborhood of S, but it is not hard to show that it can decrease the tolerance to any desired value. The difference here is in the cardinality of the needed smaller balls. Although the lowest cardinality is hard to find, asymptotically, an $O((\frac{\xi}{\mu})^2)$ cardinality is achievable in linear time.

So far we have seen how to achieve any given tolerance when occlusion is not allowed and when the space of allowable transformations consists of the planar affine transformations. In the next section, we will see how to reach zero tolerance under a density assumption on the observations.

4.5 Zero Tolerance and Sparse Observations

If the observation is not dense with respect to ξ, a zero uncertainly can be achieved in a time bound similar to the 3ξ tolerance bound. The new component is a verification stage added on top of $\mathcal{Q}_2$. Recall that $\mathcal{Q}_2$ computes in $O(m^4 + n^3 m \log n)$ time a set of 4ξ-feasible transformations whose 3ξ-neighborhood covers all the ξ-feasible transformations.

Theorem 4.5.1 *If $\forall \alpha, \beta \in S$ with $\alpha \neq \beta$, we have $\|\alpha - \beta\| \geq 8\xi$. Then a set of ξ-feasible transformations whose 2ξ-neighborhoods cover all the ξ-feasible transformations can be computed in $O(m^4 + n^3 m \log mn)$ time.*

proof: Consider the following algorithm

$\mathcal{Q}_4$ = " On input (M, S, ξ)

1. initialize $\hat{\mathcal{X}}$ to the empty set
2. run $\mathcal{Q}_2$ on (M, S, ξ) and let $\mathcal{X}$ be its output
3. for each $t \in \mathcal{X}$
4. compute the closest mapping $\hat{g} : M \to S$ to t, i.e., $\hat{g} = \arg\min_{g:M\to S} \|t - g\|_M$
5. test the emptiness of the set $\hat{N} = \{t' \in \mathcal{A}; \|\hat{g} - t'\|_M < \xi\}$
6. if $\hat{N} \neq \emptyset$, pick any element $\hat{t}$ of $\hat{N}$ and add it to $\hat{\mathcal{X}}$
7. return $\hat{\mathcal{X}}$."

Let $\mathcal{C}_\xi$ be the set of ξ-feasible transformations. Using Lemma 4.2.2, we express $\mathcal{C}_\xi$ as as follows

$$\mathcal{C}_\xi = \{t \in \mathcal{A}; \exists g : M \to S \text{ s.t. } \|t - g\|_M < \xi\} \tag{4.12}$$

We know from section 4.3 that the output $\mathcal{X}$ of $\mathcal{Q}_2$ satisfies:

(a) *relaxed feasibility condition:* $\mathcal{X} \subset C_{4\xi}$.

(b) *covering condition:* $\forall t' \in C_\xi, \exists t \in \mathcal{X}$ s.t. $\|t - t'\|_M < 3\xi$.

And we want to show that

(i) *feasibility condition:* $\hat{\mathcal{X}} \subset C_\xi$

(ii) *covering condition:* $\forall t' \in \mathcal{C}_\xi, \exists t \in \hat{\mathcal{X}}$ s.t. $\|t - t'\|_M < 2\xi$.

First, (i) follows from (4.12) because any $\hat{t}$ added to $\hat{X}$ at Line 6 is ξ-close to the corresponding $\hat{g}$.

Rather than proving (ii), we prove something stronger. Define $\mathcal{G}_\xi$ to be the set of ξ-feasible mappings, i.e.,

$$\mathcal{G}_\xi = \{g : M \to S; \exists t \in \mathcal{A} \text{ s.t. } \|t - g\|_M < \xi\}.$$

We show that $\mathcal{G}_\xi$ is equal to the set $\hat{G}$ containing each mapping $\hat{g}$ that was computed at line 4 and that lead to a nonempty $\hat{N}$ at line 5. This will establish (ii) because if $t' \in \mathcal{C}_\xi$, then $\|t' - g\|_M < \xi$ for some $g \in \mathcal{G}_\xi$, but this means that $g \in \hat{G}$ and thus g is ξ-close to a transformation $t \in \hat{\mathcal{X}}$, which proves (ii) since

$$\|t - t'\|_M < \|t - g\|_M + \|g - t'\|_M < \xi + \xi = 2\xi.$$

$\hat{G}$ is clearly a subset of $\mathcal{G}_\xi$ because it only contains mappings that are ξ-close to allowable transformations, so we only need to show that $\mathcal{G}_\xi \subset \hat{G}$.

Consider any $g \in \mathcal{G}_\xi$. Using (b), let $t \in \mathcal{X}$ s.t. $\|t - g\|_M < 4\xi$. Let $\hat{g}$ be the closest mapping to t. We will show that $g = \hat{g}$.

We have

$$\|g - \hat{g}\|_M \leq \|t - g\|_M + \|t - \hat{g}\|_M \leq 2\|t - g\|_M < 8\xi,$$

where the second inequality follows form the fact that $\hat{g}$ is the closest mapping to t. Now if $g \neq \hat{g}$, then g(α) $\neq \hat{g}(\alpha)$, for at least one $\alpha \in M$. This means that $g(\alpha)$ and $\hat{g}(\alpha)$ are two distinct points of S with $\|g(\alpha) - \hat{g}(\alpha)\|_M \leq \|g - \hat{g}\|_M < 8\xi$, which contradicts the given hypothesis on S. Therefore, $g = \hat{g}$.

When $\mathcal{Q}_4$ passed over t in the loop at Line 3, the mapping $\hat{g}$ was computed at Line 5. We know that $\hat{g} = g \in \mathcal{G}_\xi$, hence it follows form the definition of $\mathcal{G}_\xi$ that the corresponding $\hat{N}$ is not empty. Therefore, an element $\hat{t}$ of $\hat{N}$ must have been added to $\hat{\mathcal{X}}$ at Line 6. This means that $g \in \hat{G}$.

Since this is true for any $g \in \mathcal{G}_\xi$, it follows that $\mathcal{G}_\xi \subset \hat{G}$, which completes the proof of (ii). Note that due to the test done at Line 6 we did not need the feasibility condition on $\mathcal{X}$ in the proof.

We still have to find the running time of $\mathcal{Q}_4$. We know from Section 4.3 that $\mathcal{Q}_2$ runs in $O(n^3 m \log n)$ time. We also know that the size of the output $\mathcal{X}$ of $\mathcal{Q}_2$ is $O(n^3)$. We argue below that each $\hat{g}$ can be computed in in $O(m \log n)$ given an $O(n \log n)$ preprocessing time for S. We also argue that the emptiness of each $\hat{N}$ can be tested in $O(m \log m)$ time, and an element of $\hat{N}$ can be computed in $O(m \log m)$ time if it is not empty. Thus the overall time needed by $\mathcal{Q}_4$ to halt is $O(n \log n + n^3 m \log n + n^3(m \log n + m \log m)) = O(n^3 m \log mn)$.

Consider any $t \in \mathcal{X}$. In order to construct the closest mapping $\hat{g}$ to t, we need to compute the nearest neighbor in S of $t(\alpha)$, for each $\alpha \in M$. Due to the special setting of the problem, this can be done by computing the intersection of $B_{4\xi}(t(\alpha))$ with S. The distance assumption on S implies that $B_{4\xi}(t(\alpha)) \cap S$ can not contain more than one point. And the 4ξ-feasibility of t (from (a)) implies that $B_{4\xi}(t(\alpha)) \cap S$ is not empty. So $B_{4\xi}(t(\alpha)) \cap S$ contains one and only one point. It follows that the nearest neighbor of $t(\alpha)$ is $B_{4\xi}(t(\alpha)) \cap S$. We can preprocess S in $O(n \log n)$ time so that the intersection of a query rectangular region R with S can be computed in $O(|R \cap S| + \log n)$ (See [2] page 578). This leads to an $O(m \log n)$ overall time needed to compute $\hat{g}$.

Now consider any $\hat{g}$, by construction

$$\hat{N} = \{t' \in \mathcal{A}; ||t' - \hat{g}||_M < \xi\}$$

We argue below that the emptiness of $\hat{N}$ is reducible to solving two $3D$-linear programs each of $2m$ constraints, and thus computable in $O(m \log m)$ time. Let $\alpha_1, \ldots, \alpha_m$ be the points of M, and let $\beta_i = \hat{g}(\alpha_i)$ for $i = 1, \ldots, m$. If we denote by $H_{2\times 2}$ and $r_{2\times 1}$ - respectively - the linear transformations matrix and the translation vector of t', we can write the constraint $\|t' - \hat{g}\|_M < \xi$ as

$$\left\| [H|r] \begin{bmatrix} \alpha_i \\ 1 \end{bmatrix} - \beta_i \right\| < \xi, i = 1, \ldots, m.$$

If we let $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = [H|r]$ where $x_1, x_2 \in \mathbb{R}^{3\times1}$, we can reduce the constraint to

$$|x_1 \begin{bmatrix} \alpha_i \\ 1 \end{bmatrix} - \beta_{ix}| \quad < \xi \quad , i = 1, \ldots, m, \tag{4.13}$$

$$|x_2 \begin{bmatrix} \alpha_i \\ 1 \end{bmatrix} - \beta_{iy}| \quad < \xi \quad , i = 1, \ldots, m. \tag{4.14}$$

So $\hat{N} \neq \emptyset$ if and only if each of the $2m$ constraints $3D$-linear programs given by (4.13) and (4.14) are feasible. The feasibility of each of the linear programs can be tested in $O(m \log m)$ time using a half space intersection algorithm due to Muller & Preparata [1978]. If they are both feasible, this algorithm leads to the corresponding $3D$-convex polytopes (they are bounded because M is regular). So, to compute an element of $\hat{N}$, we only need to pick an element of each of the resulting convex polytopes. □

Note:

- The theorem still holds if the minimum distance between any two distinct points in S is at least $2(1 + \Delta(U^*, M))\xi$.

4.6 Handling Occlusion

In this section, see show how to handle occlusion. In $\mathcal{Q}_1$, we were hypothesizing mappings from U to S and testing the $(\xi\Delta(U, M) + \xi)$-feasibility of the affine transformations that agreed with those mappings. The output of $\mathcal{Q}_1$ consisted of those affine transformations that passed the $(\xi + \xi\Delta(U, M))$-feasibility test. $\mathcal{Q}_1$ can be simply modified to handle weak occlusion, by which we mean the case when the points in U are not allowed to be missing. This can be done by simply testing the $(\xi\Delta(U, M) + \xi, p)$-feasibility of the hypothesized transformation rather than their $(\xi\Delta(U, M) + \xi)$-feasibility.

Algorithm 4.6.1 *Consider the following algorithm that takes as input: the model M, the observation S, the noise bound ξ, a regular subset U of M consisting of* 3 *noncollinear points, and a bound p on the number of points missing form the instances of M in S.*

$\mathcal{P}_1$ = *" On input (M, S, U, ξ, p)*

1. compute $\Delta(U, M)$

2. initialize $\mathcal{X}$ to the empty set

3. repeat the following for each mapping f from U to S

4. compute the unique transformation $t \in \mathcal{A}$ satisfying $f = t|U$

5. check if t is $(\xi + \xi\Delta(U, M), p)$-feasible

6. if so add t to $\mathcal{X}$

7. return $\mathcal{X}$"

We remind the reader that

Defintion 2.3.1 *If t is an allowable transformation*

1. *t is said to be ξ-feasible on a pattern V if $t(V) \subset B_\xi(S)$.*

2. *t is said to be (ξ, p)-feasible if it is ξ-feasible on a regular subset of M containing at least $m - p$ points.*

To put it in an other way, t is (ξ, p)-feasible if it maps at least $m - p$ point of the model to within ξ-distance of points in S (with the additional requirement that those points form a regular pattern).

Using exactly the same argument of Lemma 4.2.2, we get

Lemma 4.6.2 *A transformation t is ξ-feasible on a pattern V iff there exits a mapping $g : V \to S$ such that $\|g - t\|_V < \xi$.*

The proof that $\mathcal{P}_1$ partially handles occlusion is essentially very similar to the correctness proof of $\mathcal{Q}_1$.

Theorem 4.6.3 (weak occlusion and $\xi\Delta(U, M)$ tolerance) *$\mathcal{P}_1$ computes in $O(n^3 m \log n)$ time a set $\mathcal{X}$ of allowable transformations satisfying:*

(i) relaxed feasibility condition: each transformation in $\mathcal{X}$ is $(\xi + \xi\Delta(U, M), p)$-feasible.

(ii) **weak** *covering condition: If V is a regular subset of M* **containing U** *and at least $m - p$ points, then for each transformation t', that is ξ-feasible on V, there exists a transformation t in $\mathcal{X}$ such that $\|t - t'\|_V < \xi\Delta(U, M)$.*

proof: First (i) is true because only those transformations that passed the feasibility test in Line 5 were added to the output. To prove (ii) consider any regular subset V of M containing U and at least $m-p$ points. Let t be an ξ-feasible transformation on V. Using Lemma 4.6.2, let $g : V \to S$ such that $\|g-t\|_V < \xi$. Finally, let t be the unique transformation in $\mathcal{A}$ satisfying $t|U = g|U$. We show first that $\|t - t'\|_V < \xi\Delta(U, M)$, then we conclude that $t \in \mathcal{X}$. We have

$$\begin{aligned}
\|t - t'\|_V &\leq \Delta(U, V)\|t - t'\|_U &(4.15)\\
&\leq \Delta(U, M)\|t - t'\|_U &(4.16)\\
&= \Delta(U, M)\|g - t'\|_U &(4.17)\\
&\leq \Delta(U, M)\|g - t'\|_V &(4.18)\\
&< \xi\Delta(U, M), &(4.19)
\end{aligned}$$

where (4.15) follows from Lemma 4.1.1, (4.16) follows from the fact that $V \subset M$, (4.17) from the fact that $t|U = g|U$, and (4.18) from the fact $U \subset V$. Now,

$$\|t - g\|_V \leq \|g - t'\|_V + \|t - t'\|_V \leq \xi + \xi\Delta(U, M)$$

To see why $t \in \mathcal{X}$, let $f = g|U$. Note that f is mapping from U to S. Moreover, t is the unique transformation in $\mathcal{A}$ satisfying $t|U = f$. So t was computed at Line 4 when the mapping f was considered in the loop of Line 3. Now t must have passed the test at Line 5 because it is $(\xi + \xi\Delta(U, M), p)$-feasible due to the fact $\|t - g\|_V < \xi + \xi\Delta(U, M)$ and $|V| \geq m - p$ (This follows from Lemma 4.6.2). So it must have been added to $\mathcal{X}$ at Line 6.

We still have to find the running time of $\mathcal{P}_1$. First, we need to know how to do the feasibility test in Line 5. Let $\varepsilon = \xi + \xi\Delta(U, M)$. We want a routine that checks whether an element t of $\mathcal{A}$ is (ε, p)-feasible. The story is similar to that of the ε-feasibility test: Assuming an $O(n \log n)$ preprocessing time for S, this can be done in $O(m \log n)$ time by passing over each element α of M and computing the number of points in the intersection of the square $B_\varepsilon(\alpha)$ with S. t is declared to be (ε, p)-feasible iff at least $m - p$ of the α's lead to a nonempty intersection.

We have $O(n^3)$ mappings to work with, and each mapping needs an $O(m \log n)$ processing time. After adding the $O(n \log n)$ preprocessing time of S, we end up with an $O(n \log n + n^3 m \log n) = O(n^3 m \log n)$ total time needed by $\mathcal{P}_1$ to halt. □

The tolerance of $\mathcal{P}_1$ is $\xi\Delta(U, M)$. It can be decreased to a value bounded by 3ξ by supplying $\mathcal{Q}_1$ with the U^* that minimize $\Delta(U, M)$. But this technique still suffers from the fact that the points of U^* are not allowed to be missing. Of course we can not run $\mathcal{P}_1$ on each 3-point regular subset U of M and take the union of the outputs. This new technique does not require any special points in M to be nonmissing, but it suffers from an unbounded tolerance. The tolerance of this technique can be as large as possible because for each real number N, there exits a model M and a three points regular subset U of M with $\Delta(U, M) \geq N$ (see Figure 4-2). What is the solution ?

Algorithm 4.6.4 *Consider the following algorithm that takes as input: the model*

M, the observation S, the error bound ξ, and a bound p on the number of points missing form the instances of M in S.

$\mathcal{P}_2$ = *" On input (M, S, ξ, p)*

1. Initialize $\mathcal{X}$ to the empty set

2. for each regular subset U of M consisting of 3 noncollinear points

3. Compute $V_U = \{\alpha \in M; \Delta(U, \{\alpha\}) \leq 3\}$

4. If $|V_U| \geq m - p$

5. run $\mathcal{P}_1$ on (V_U, S, U, ξ, p) and add its output to $\mathcal{X}$

6.return $\mathcal{X}$. "

Remark: the number m used in $\mathcal{P}_1$ is not the number of points in the first input of $\mathcal{P}_1$ (i.e. V_U), it is the same number used in $\mathcal{P}_2$, i.e, the number of points in M.

$\mathcal{P}_2$ considers each 3-point regular U of M, it computes a new model V_U equal the maximal subset of M with $\Delta(U, V_U) \leq 3$, then if V_U is large enough, $\mathcal{P}_2$ runs $\mathcal{P}_1$ on the input (V_U, S, U, ξ, p). The output of $\mathcal{P}_2$ is the union of the sets returned by $\mathcal{P}_1$ in each run. Why is $\mathcal{P}_2$ a solution? The main reason is Theorem 4.3.2, which can be written as

Theorem 4.3.2 *Any regular pattern V contains a 3-point regular subset U with $\Delta(U, V) \leq 3$.*

Informally, the idea is the following. $\mathcal{Q}_2$ will work fine under the assumption that points in U are guaranteed to be nonmissing. What $\mathcal{Q}_3$ does is that it guarantees that this is always the case. The point is is that if some points in an instance of M in S are missing then the subset V of M corresponding to this occluded instance should contain some U_0 with $\Delta(U_0, V) \leq 3$. $\mathcal{Q}_3$ is considering this U_0 and it is working with a new model V_{U_0} that provably contains V because V_{U_0} is the maximal subset V of M with $\Delta(U, V) \leq 3$. We will present below a formal version of this argument.

First, note that given any 3-point regular subset U , the maximal subset V of M with $\Delta(U, V) \leq 3$ exists because

Lemma 4.6.5 *If U is a regular pattern and V is a pattern then*

$$\Delta(U,V) = \max_{\alpha \in V} \Delta(U,\{\alpha\})$$

proof:

$$\begin{aligned} \Delta(U,V) &= \sup_{\{t \in \mathcal{A}^*; \|t\|_U \le 1\}} \|t\|_V = \sup_{\{t \in \mathcal{A}^*; \|t\|_U \le 1\}} \max_{\alpha \in V} \|t(\alpha)\| = \max_{\alpha \in V} \sup_{\{t \in \mathcal{A}^*; \|t\|_U \le 1\}} \|t\|_{\{\alpha\}} \\ &= \max_{\alpha \in V} \Delta(U,\{\alpha\}). \end{aligned}$$

□

Another way of saying this is $\Delta(U, V_1 \cup V_2) = \max\{\Delta(U,V_1), \Delta(U,V_2)\}$. Now, we are dealing with many models and soon we will be dealing with more than one observation. In order to avoid any confusion in the notions of ξ-feasible transformations, we need a more accurate notation.

Definition 4.6.1 *Define $\mathcal{C}^p_\xi(M,S)$ to be the set of (ξ,p)-feasible transformations with respect to the model M and the observation S, i.e.,*

$$\mathcal{C}^p_\xi(M,S) = \bigcup_{V \in Reg^p(M)} \mathcal{C}_\xi(V,S),$$

where $Reg^p(M)$ is set of regular subsets of M containing at least $m-p$ points, and

$$\mathcal{C}_\xi(V,S) = \{t \in \mathcal{A}; t(V) \subset B_\xi(S)\}.$$

The following theorem shows that $\mathcal{P}_2$ handles occlusion and achieves a 3ξ tolerance in $O(n^3 m^4 \log n)$ time.

Theorem 4.6.6 (Occlusion and a tolerance bounded by 3ξ) *$\mathcal{P}_2$ computes in $O(n^3 m^4 \log n)$ time a set of allowable transformations satisfying:*

(i) relaxed feasibility condition: $\mathcal{X} \subset \mathcal{C}^p_{\xi+3\xi}(M,S)$

(ii) covering condition: if $V \in Reg^p(M)$, then $\forall t' \in C_\xi(V,S), \exists t \in \mathcal{X}$, s.t. $\|t-t'\|_V < 3\xi$.

proof: If U is a 3-point regular subset of M with $|V_U| \geq m-p$, let $\mathcal{X}_U$ be the output of $\mathcal{P}_1$ on the input (V_U, S, U, ξ, p). If we let $\mathcal{U} = \{U; U \text{ is a 3-point regular subset of } M\}$, we can write the output of $\mathcal{P}_2$ as

$$\mathcal{X} = \bigcup_{U \in \mathcal{U}; |V_U| \geq m-p} \mathcal{X}_U. \tag{4.20}$$

From Theorem 4.6.3 we know that for $\forall U \in \mathcal{U}$,

(a) relaxed feasibility condition: $\mathcal{X}_U \subset \mathcal{C}^p_{\xi+\xi\Delta(U,V_U)}(V_U, S)$

(b) weak covering condition: if $V \in Reg^p(V_U)$ and $U \subset V_U$, then $\forall t' \in C_\xi(V, S), \exists t \in \mathcal{X}_U$, s.t. $\|t - t'\|_V < \xi\Delta(U, V_U)$.

Using (4.20) and (a) we obtain

$$\mathcal{X} \subset \bigcup_{U \in \mathcal{U}; |V_U| \geq m-p} \mathcal{C}^p_{\xi+\xi\Delta(U,V_U)}(V_U, S) = \bigcup_{U \in \mathcal{U}} \bigcup_{V \in Reg^p(V_U)} \mathcal{C}_{\xi+\xi\Delta(U,V_U)}(V, S).$$

From Lemma 4.6.5 we know that $\forall U \in \mathcal{U}$,

$$\Delta(U, V_U) = \max_{\alpha \in V_U} \Delta(U, \{\alpha\}) \leq 3.$$

It follows that $\xi + \xi_{\Delta(U,V_U)} \leq 4\xi$, hence

$$\mathcal{X} \subset \bigcup_{U \in \mathcal{U}} \bigcup_{V \in Reg^p(V_U)} \mathcal{C}_{4\xi}(V, S) \subset \bigcup_{V \in Reg^p(M)} \mathcal{C}_{4\xi}(V, S) = \mathcal{C}^p_{4\xi}(M, S).$$

This proves (i).

To prove (ii) , consider any regular subset V of M containing at least $m-p$ points and let $t' \in C_\xi(V, S)$. By defintion V is regular. Therefore, according to Theorem 4.3.2, V contains a 3-point regular subset U_0 with $\Delta(U_0, V) \leq 3$. So $\forall \alpha \in V$,

$$\Delta(U_0, \{\alpha\}) \leq \Delta(U_0, V) \leq 3.$$

Hence, $V \subset V_{U_0}$. To sum up, U_0 is an element of $\mathcal{U}$ and V is a regular subset of V_{U_0}

containing U_0 and at least $m-p$, moreover $t' \in C_\xi(V,S)$, it follows from (b) that there exists $t \in \mathcal{X}_{U_0}$ (hence $t \in \mathcal{X}$) s.t.

$$\|t-t'\|_V < \xi + \xi\Delta(U_0, V_{U_0}) \leq \xi + \xi \max_{\alpha \in V_{U_0}} \Delta(U_0, \{\alpha\}) \leq \xi + 3\xi = 4\xi.$$

Summing it up again, for any regular subset V of M containing at least $m-p$ points, and for each $t' \in C_\xi(V,S)$, we can find a $t \in \mathcal{X}$, such that $\|t-t'\|_V < 3\xi$. This proves (ii).

We still have to find the running time of $\mathcal{P}_2$. We have $O(m^3)$ 3-point regular subset U to work with. The time needed by $\mathcal{P}_1$ on each U is according to Theorem 4.6.3 is $O(n^3|V_U|\log n) = O(n^3 m \log n)$. We show below how V_U can be computed in $O(m)$ time, this leads to an $O(m^3(m + n^3 m \log n)) = O(n^3 m^4 \log n)$ time needed by $\mathcal{P}_2$ to halt. We can compute V_U by computing $\Delta(U, \{\alpha\})$ for each $\alpha \in M$ and returning the α's satisfying $\Delta(U,\{\alpha\}) \leq 3$. In Proposition 4.3.4, the regularity of the model and the fact that U is a subset of the model was not used in the proof, the result still holds even if the model consists of a single point, i.e., $\Delta(U,\{\alpha\})$ can be computed in $O(|\{\alpha\}|.|U|) = O(1)$ time. Note that in this case

$$\Delta(U,\{\alpha\}) = \max_{x \in \mathbb{R}^{1\times 3}; -e_3 \leq x A_U \leq e_3} x \begin{bmatrix} \alpha \\ 1 \end{bmatrix}, \text{ where } A_U = \begin{bmatrix} \beta_1 & \beta_2 & \beta_3 \\ 1 & 1 & 1 \end{bmatrix},$$

and $e_3 = [1 \quad 1 \quad 1]$. This leads to an $O(m)$ overall time needed to compute V_U. □

It is important to note that if p is small compared to m, the number of the submodels that will passe the cardinality test at Line 4 (i.e. $|V_U| \geq m-p$) is much less than m^3. For practical purposes, it might be wiser to preprocess M by computing first all the V_U s and storing those that passes the cardinality test, this leads to a time constant much less than 1.

To achieve lower values of tolerance we proceed in exactly the same mentality of Section 4.4.

Theorem 4.6.7 (Occlusion and any given tolerance) *Given any tolerance bound $\mu > 0$, we can compute in $O((\frac{3\xi}{\mu})^6 n^3 m^4 \log n)$ time a set of allowable transformations* *satisfying:*

(i) relaxed feasibility condition: $\mathcal{X} \subset \mathcal{C}^p_{\xi+\mu}(M,S)$

(ii) covering condition: if $V \in Reg^p(M)$, *then* $\forall t' \in C_\xi(V,S); \exists t \in \mathcal{X}$ *s.t.* $\|t-t'\|_V < \mu$.

proof: Consider the following algorithm

$\mathcal{P}_3 =$ " On input (M, S, ξ, μ, p)

1. Compute a planar set S' satisfying $B_{\frac{\mu}{3}}(S') = B_\xi(S)$.
3. Run $\mathcal{P}_2$ on $(M, S', \frac{\mu}{3}, p)$ and return its output. "

We can compute S' as in the nonocclusion case: We can pick each $\beta \in S$, and cover the square $B_\xi(\beta)$ by a collection Z_β of squares the radius of each is $\frac{\mu}{3}$ in such a way that the union of these squares is equal to $B_\xi(\beta)$. Then we can set S' to be the set of centers of the squares in the union of the Z_β's. Each $B_\xi(\beta)$ can be covered by $\lceil\frac{\xi}{\mu/3}\rceil^2$ such squares. So a set S' containing $n\lceil\frac{3\xi}{\mu}\rceil^2$ point and satisfying $B_{\frac{\mu}{3}}(S') = B_\xi(S)$ can be constructed in linear time. The cardinality of S' is at most $n(3\frac{\xi}{\mu}+1)^2$. i.e., $|S'| = O((\frac{3\xi}{\mu})^2 n)$. According to Theorem 4.6.7, the time needed by $\mathcal{P}_2$ (hence $\mathcal{P}_3$) to halt is $O(|S'|^3 m^4 \log|S'|) = O(((\frac{3\xi}{\mu})^2 n)^3 m^4 \log(\frac{3\xi}{\mu})^2 n)$. So, $\mathcal{P}_3$ runs in $O((\frac{3\xi}{\mu})^6 n^3 m^4 \log n)$ time.

let $\mathcal{X}$ be the output of $\mathcal{P}_3$. From Theorem 4.2.3, we know that

(a) relaxed feasibility condition: $\mathcal{X} \subset \mathcal{C}^p_{4\frac{\mu}{3}}(M,S')$

(b) covering condition: if $V \in Reg^p(M)$, then $\forall t' \in C_{\frac{\mu}{3}}(V,S'), \exists t \in \mathcal{X}$, s.t. $\|t-t'\|_V < 3\frac{\mu}{3}$.

It is sufficient to show that $C_\xi(V,S) = C_{\frac{\mu}{3}}(V,S'), \forall V \in Reg^p(M)$ and $\mathcal{C}^p_{\xi+\mu}(M,S) = \mathcal{C}^p_{4\frac{\mu}{3}}(M,S')$. The first follows from the fact that $B_{\frac{\mu}{3}}(S') = B_\xi(S)$ and the second from the fact that

$$
\begin{aligned}
B_{4\frac{\mu}{3}}(S') &= \bigcup_{\alpha\in S'} B_{\frac{\mu}{3}+\mu}(\alpha) = \bigcup_{\alpha\in S'} \bigcup_{x\in B_{\frac{\mu}{3}}(\alpha)} B_\mu(x) = \bigcup_{x\in B_{\frac{\mu}{3}}(S')} B_\mu(x) \\
&= \bigcup_{x\in B_\xi(S)} B_\mu(x) = \bigcup_{\alpha\in S} \bigcup_{x\in B_\xi(\alpha)} B_\mu(x) = \bigcup_{\alpha\in S} B_{\xi+\mu}(\alpha) = B_{\xi+\mu}(S). \qquad \square
\end{aligned}
$$

Chapter 5

Extending the Results to Rigid Motions and Scaling in Arbitrary Dimensions

In this section, we generalize all the previous results to the case when $\mathcal{A}$ consists of d-dimensional rigid motions and scaling. Thus, $\mathcal{F} = \mathbb{R}^d$ and $\mathcal{A}$ is the set of all the mappings from $\mathcal{F}$ to $\mathcal{F}$ that can be written as

$$t(x) = sGx + r, \forall x \in \mathbb{R}^{d \times 1},$$

for some real scalar s, $d \times d$ real unitary matrix G, and a $d \times 1$ translation vector r. We denote this correspondence by $t \sim (s, G, r)$. Here again the vector space $\mathcal{A}^*$ generated by $\mathcal{A}$ is equal to $\mathcal{A}$. As for the regularity condition it is equivalent to the fact the the pattern contains more than one point (the proof is trivial).

As in the affine case, if U is a regular pattern and V is a pattern, define $\Delta(U, V)$ by

$$\Delta(U, V) = \sup_{\{t \in \mathcal{A}^*; ||t||_U \leq 1\}} ||t||_V.$$

In order to generalize the previous results, essentially, we only need a bound on $\Delta(U, M)$ similar to that in Theorem 4.3.2. We also need to know how to compute

$\Delta(U, M)$. The other results can be essentially adapted by replacing 3 by 2 in all the places where the cardinality of U was involved. The reason is that we only need two distinct points to specify an element of $\mathcal{A}$. There are also other minor technical points that needs elaboration. We will elaborate on those points after establishing the following bound on $\Delta(U, M)$, which is basically the main new result of this section.

Theorem 5.0.8 *Any regular pattern M contains a regular subset U consisting of* 2 *points and satisfying* $\Delta(U, M) \leq 1 + \sqrt{3(d-1)}$.

proof: The norm $||.||$ on the feature space was by default the ∞-norm. In this proof we will be dealing with both the ∞-norm and the 2-norm. In order to avoid confusion, we will write $||.||_\infty$ instead of $||.||$. Let $U = \{\alpha, \beta\}$, where α and β are the farthest points in M with respect to the 2-norm, i.e.,

$$||\alpha - \beta||_2 \geq ||p_1 - p_2||_2, \forall p_1, p_2 \in M. \tag{5.1}$$

By definition,

$$\Delta(U, M) = \sup_{\{t \in \mathcal{A}^*; ||t||_U \leq 1\}} ||t||_M = \sup_{\{t \in \mathcal{A}^*; ||t(\alpha)||_\infty \leq 1, ||t(\beta)||_\infty \leq 1\}} \max_{\phi \in M} ||t(\phi)||_\infty.$$

Consider any $t \in \mathcal{A}^*$ satisfying

$$||t(\alpha)||_\infty \leq 1 \text{ and } ||t(\beta)||_\infty \leq 1. \tag{5.2}$$

We need to show that $||t(\phi)||_\infty \leq 1 + \sqrt{3(d-1)}, \forall \phi \in M$.

Consider any $\phi \in M$. We know form (5.1) that $||\phi - \alpha||_2 \leq ||\alpha - \beta||_2$. The transformation t consists only of translation, rotation, and scaling, so we also have

$$||t(\phi) - t(\alpha)||_2 \leq ||t(\alpha) - t(\beta)||_2). \tag{5.3}$$

More precisely, say that $t \sim (s, G, r)$, using the fact that G is orthogonal, we obtain

$$||t(\phi) - t(\alpha)||_2 = ||sG\phi + r - (sG\alpha - r)||_2 = ||sG(\phi - \alpha)||_2 = s||\phi - \alpha||_2$$

$$
\begin{aligned}
&\leq s\|\alpha-\beta\|_2 = \|sG(\alpha-\beta)\|_2 = \|sG\alpha + r - (sG\beta - r)\|_2 \\
&= \|t(\alpha) - t(\beta)\|_2 \quad .
\end{aligned}
$$

Similarly

$$
\|t(\phi) - t(\alpha)\|_2 \leq \|t(\alpha) - t(\beta)\|_2 \tag{5.4}
$$

because from (5.1) we know that $\|\phi - \beta\|_2 \leq \|\alpha - \beta\|_2$. By squaring both sides of each of (5.3) and (5.4) and adding them we get

$$
\begin{aligned}
2\|t(\alpha) &- t(\beta)\|_2^2 \geq \|t(\phi) - t(\alpha)\|_2^2 + \|t(\phi) - t(\beta)\|_2^2 \\
&= \|(t(\phi) - \frac{t(\alpha)+t(\beta)}{2}) - \frac{t(\alpha)-t(\beta)}{2}\|_2^2 \\
&\quad + \|(t(\phi) - \frac{t(\alpha)+t(\beta)}{2}) + \frac{t(\alpha)-t(\beta)}{2}\|_2^2 \\
&= 2\|t(\phi) - \frac{t(\alpha)+t(\beta)}{2}\|_2^2 + 2\|\frac{t(\alpha)-t(\beta)}{2}\|_2^2.
\end{aligned}
$$

After rearranging the terms we obtain

$$
\|t(\phi) - \frac{t(\alpha)+t(\beta)}{2}\|_2^2 \leq 3\|\frac{t(\alpha)-t(\beta)}{2}\|_2^2. \tag{5.5}
$$

We will bound $\|t(\alpha)\|_\infty$ by $1 + \sqrt{3(d-1)}$ subject to the constraints in (5.5) and (5.2). Say that $t(\alpha) = (a_1, \ldots, a_d)$, $t(\beta) = (b_1, \ldots, b_d)$, and $t(\phi) = (x_1, \ldots, x_d)$. The constraints in (5.5) and (5.2) become

$$
\sum_{i=1}^{d}(x_i - \frac{a_i+b_i}{2})^2 \leq 3\sum_{i=1}^{d}(\frac{a_i-b_i}{2})^2 \tag{5.6}
$$

$$
|a_i| \leq 1, |b_i| \leq 1, \forall i \in \{1, \ldots, d\}. \tag{5.7}
$$

For any $k \in \{1, \ldots, d\}$, after rearranging the terms of (5.6), we obtain

$$
\begin{aligned}
(x_k - \frac{a_k+b_k}{2})^2 - 3(\frac{a_k-b_k}{2})^2 &\leq \sum_{i=1, i\neq k}^{d} (\frac{a_i-b_i}{2})^2 - (x_i - \frac{a_i+b_i}{2})^2 \\
&\leq \sum_{i=1, i\neq k}^{d} 3(\frac{a_i-b_i}{2})^2
\end{aligned}
$$

$$\leq \quad (d-1)3\max_{i=1}^{d}(\frac{a_i-b_i}{2})^2$$

We know from (5.7) that $\max_{i=1}^{d}(\frac{a_i-b_i}{2})^2 \leq 1$. It follows that

$$(x_k - \frac{a_k+b_k}{2})^2 - (\frac{a_k-b_k}{2})^2 \leq 3(d-1)$$

i.e,

$$x_k \quad \leq \quad \sqrt{3(d-1)+3\left(\frac{a_k-b_k}{2}\right)^2} + \frac{a_k+b_k}{2}$$

Using the constraints in (5.7), we get

$$x_k \leq \max_{(a,b)\in\mathbb{R}^2;|a|\leq 1;|b|\leq 1} h(a,b)$$

where $h(a,b) = \sqrt{3(d-1)+3\left(\frac{a-b}{2}\right)^2} + \frac{a+b}{2}$. The function h is convex because $\forall, \lambda, \lambda' \geq 0$ with $\lambda+\lambda'=1$, we have

$$\begin{aligned}
&h(\lambda a+\lambda' a', \lambda b+\lambda' b') = \sqrt{3(d-1)+3\left(\frac{\lambda(a-b)+\lambda'(a'-b')}{2}\right)^2} + \lambda\frac{a+b}{2} + \lambda'\frac{a'+b'}{2} \\
&\leq \lambda\sqrt{3(d-1)+3\left(\frac{a-b}{2}\right)^2} + \lambda'\sqrt{3(d-1)+3\left(\frac{a'-b'}{2}\right)^2} + \lambda\frac{a+b}{2} + \lambda'\frac{a'+b'}{2} \\
&= \lambda h(a,b) + \lambda' h(a',b'),
\end{aligned}$$

where the inequality follows from the fact that the function $g(s) = \sqrt{3(d-1)+3\left(\frac{s}{2}\right)^2}$ is convex because

$$\frac{d^2g(s)}{d^2s} = \frac{3}{4}\sqrt{3(d-1)+\frac{3}{4}s^2}\left(1-\frac{\frac{3}{4}s^2}{3(d-1)+\frac{3}{4}s^2}\right) \geq 0, \forall s \in \mathbb{R}.$$

Therefore, the maximum of h over the convex region $E = \{a,b) \in \mathbb{R}^2; |a| \leq 1; |b| \leq 1\}$ is attained at one of the extreme points of E. E has four extreme points $(1,1), (-1,1)$, $(1,-1)$ and $(-1,-1)$. We have $h(1,1) = h(-1,-1) = \sqrt{3(d-1)}+1$ and $h(1,-1) =$

$h(-1,1) = \sqrt{3d}$. So,

$$\max_{(a,b)\in E} h(a,b) = \max\{\sqrt{3(d-1)}+1, \sqrt{3d}\}.$$

Now $\left(\sqrt{3(d-1)}+1\right)^2 - \left(\sqrt{3d}\right)^2 = 2\sqrt{3(d-1)} - 2 > 0, \forall d \geq 2$. Hence $\max_{(a,b)\in E} h(a,b) = \sqrt{3(d-1)}+1$. Therefore, $|x_k| \leq 1+\sqrt{3(d-1)}$ and this is true for any $k \in \{1, \ldots d\}$. It follows that $\|t(\phi)\|_\infty \leq 1 + \sqrt{3(d-1)}$, which completes the proof. □

Notes:

- This bound is tight in the sense that there are rare M's where the best 2-point U achieves a value of $\Delta(U, M) = 1 + \sqrt{3(d-1)}$. One such M consists of the three corners of an equiliteral triangle (This is probably the only one up to scaling and rigid motions).
- If the 2-norm is the one under consideration it can be shown that the bound is 2 if $d \geq 2$. This is in contrast to the affine case, where the bound 3 is also tight in the 2-norm setting.
- Here again note that the U suggested by the proof can be competed in $O(m \log m)$ time, but it does not always lead to the U with the lowest value of $\Delta(U, M)$.

As for the computation of $\Delta(U, M)$, we show below how this can be done in the two dimensional case only and regardless of the number of points in U.

Proposition 5.0.9 *If $d = 2$ and $U = \{\alpha_1, \ldots, \alpha_k\}$ is a regular subset of M , then*

1.

$$\Delta(U, M) = \max_{\beta \in M} \max_{x\in\mathbb{R}^{1\times 4}; -e_k \leq x[A|GA] \leq e_k} \max\{x \begin{bmatrix} \beta \\ 1 \\ 0 \end{bmatrix}, xG \begin{bmatrix} \beta \\ 1 \\ 0 \end{bmatrix}\} \tag{5.9}$$

where

$$A = \begin{bmatrix} \alpha_1 & \alpha_2 & \ldots & \alpha_k \\ 1 & 1 & \ldots & 1 \\ 0 & 0 & \ldots & 0 \end{bmatrix}_{4\times k}, G = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}_{4\times 4},$$

and $e_k = [1 \;\; 1 \;\; \ldots \;\; 1]_{1\times k}$. *Moreover, the region defined by the constraint* $-e_k \leq x[A|GA] \leq e_k$ *is bounded.*

2. $\Delta(U, M)$ *can be computed in* $O(mH_4(4k))$ *time, where* $H_4(4k)$ *is the time needed to solve a linear program of* 4 *variables and* $4k$ *constraints.*

proof: By definition,

$$\Delta(U,M) = \sup_{\{t\in\mathcal{A};\|t\|_U\leq 1\}} \max_{\beta\in M} \|t(\beta)\| = \max_{\beta\in M} \sup_{t\in\mathcal{A};\|t\|_U\leq 1} \|t(\beta)\|.$$

We can write $\sup_{\{t\in\mathcal{A};\|t\|_U\leq 1\}} \|t(\beta)\|$ as the supremum of $\left\| \begin{bmatrix} x_1 & x_2 \\ -x_2 & x_1 \end{bmatrix} \beta + \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} \right\|_\infty$ subject to the constraint

$$[x_1 \; x_2 \; x_3 \; x_4] \in \mathbb{R}^4; \left\| \begin{bmatrix} x_1 & x_2 \\ -x_2 & x_1 \end{bmatrix} \alpha_i + \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} \right\|_\infty \leq 1, i = 1, \ldots, k$$

Or in other words,

$$\begin{array}{ll} \sup & \max\{|[x_1 \; x_2]\beta + x_3|, |[-x_2 \; x_1]\beta + x_4|\}, \\ [x_1 \; x_2 \; x_3 \; x_4] \in \mathbb{R}^4; \text{ for } i = 1, \ldots, k & \\ \quad |[x_1 \; x_2]\alpha_i + x_3| \leq 1 & \\ \quad |[-x_2 \; x_1]\alpha_i + x_4| \leq 1 & \end{array}$$

which leads to the superimum version of Equation (5.9) after some algebraic manipulation. Note that the absolute value in the objective function can be removed because the region defined by the constraints is symmetric.

Now we show that the region $E = \{x \in \mathbb{R}^{1\times 4}; -e_k \leq x[A|GA] \leq e_k\}$ is bounded. The pattern U is regular, so it contains more than one point, i.e, $k \geq 2$. So let $B = \begin{bmatrix} \alpha_1 & \alpha_2 \\ 1 & 1 \\ 0 & 0 \end{bmatrix}$ and $F = \{x \in \mathbb{R}^{1\times 4}; -e_3 \leq x[B|GB] \leq e_3\}$. $E \subset F$, so it is sufficient to show that F is bounded.

If $x \in F$, then $-e_3 \leq x[B|GB] \leq e_3$. Hence, $\|x[B|GB]\|_\infty \leq 1$. It is easy to check that $det[B|GB] = \|\alpha_1 - \alpha_2\|_2^2$, so $[B|GB]$ is invertible, and hence $\sigma_{\min}([B|GB])$, the smallest singular value of $[B|GB]$, is not zero. It follows that

$$\|x\|_\infty \leq \frac{\|x[B|GB]\|_\infty}{\sigma_{\min}([B|GB])} \leq \frac{1}{\sigma_{\min}([B|GB])} < \infty.$$

Therefore F (and hence E) is bounded.

The region E is also closed, so it is compact. Therefore, the continuous function

$$h_\beta(x) = \max\{x \begin{bmatrix} \beta \\ 1 \\ 0 \end{bmatrix}, xG \begin{bmatrix} \beta \\ 1 \\ 0 \end{bmatrix}\}$$

achieves a maximum over E. So the superimum can be replaced by a maximum, which leads to Equation 5.9.

Now, to compute $\Delta(U, M)$, we have to solve $2m$ linear programs, two for each α in M, and each of 4 variables and $4k$ constraints. Thus, the overall time needed to compute $\Delta(U, M)$ is $O(mH_4(4k))$. □

A similar result can be derived for an arbitrary d, but the equations are messy. We leave it to the interested reader.

Note:

- The above expression of $\Delta(U, M)$ does not hold for the 2-norm. In the 2-norm setting and when U consists of two points only, $\Delta(U, M)$ has a simple closed form expression that does not require solving a linear program.

Now, we start the adaptation process. $\mathcal{Q}_1$ can be simply modified by replacing the 3-point regular U with a 2-point U, and dealing with rigid motions and scaling instead of affine transformations. The correctness proof of the algorithm still holds. The only difference in the realization of $\mathcal{Q}_1$ is in the feasibility test. In this setting, rather than taking $O(m \log n)$ time, the test takes $O(m \log^{d-1} n)$ time because the cardinality of the intersection of a query d-dimensional rectangular region with an n-points d-dimensional set needs $O(\log^{d-1} n)$ computation time. As for the hypothesized transformations, each can be computed in constant time by solving a linear system of $O(1)$ equations. So the overall running time of $\mathcal{Q}_1$ is $O(n^2 m \log^{d-1} n)$ and it computes a set of $(\xi+\xi\Delta(U,M))$-feasible transformations whose $\xi\Delta(U,M)$-neighberhood covers all the ξ-feasible transformations. In $\mathcal{Q}_2$ we have to replace 3 by 2 whenever the number of points in U or U^* is involved. Using the bound that we have just established in Theorem 5.0.8, we obtain

Theorem 5.0.10 (a tolerance bounded by $(1+\sqrt{3(d-1)})\xi$) *In $O(m^3 + n^2 m \log n)$ time we can compute a set of $(\xi + (1+\sqrt{3(d-1)})\xi)$-feasible transformations whose $(1+\sqrt{3(d-1)})\xi$-neighborhood covers all the ξ-feasible transformations .*

In $\mathcal{Q}_3$ the only difference is that when S' is computed the number of the $\frac{\mu}{\Delta^*}$-hypercubes needed to cover an ξ-hypercube is now $\lceil \frac{\xi}{\mu/\Delta^*} \rceil^{2d}$. This leads to the following

Theorem 5.0.11 (Any given tolerance) *Given any tolerance bound μ, we can compute in $O(n^3 + (\frac{(1+\sqrt{3(d-1)})\xi}{\mu})^{2d} n^2 m \log n)$ time set of $(\xi+\mu)$-feasible transformations whose μ-neighborhood covers all the ξ-feasible transformations.*

In the affine case with sparse observations, we were lucky enough to be able to solve a 6-dimensional linear program of m constraints in $O(m \log m)$ time by separating it into two 3-dimensional linear programs. Here, the story is different. We have an inseparable linear program in $\mathbb{R}^{2d}$.

Theorem 5.0.12 (zero tolerance and sparse observations) *If $\forall \alpha, \beta \in S$ with $\alpha \neq \beta$, we have $\|\alpha - \beta\| \geq 2(2+\sqrt{3(d-1)})\xi$. Then a set of ξ-feasible transformations whose 2ξ-neighborhood covers all the ξ-feasible transformations can be computed in $O(m^3 + n^2 m H_{2d}(m) + n^2 m \log^{d-1} n)$ time, where $H_{2d}(m)$ is the time needed to solve a $2d$-dimensional linear program of m constraints.*

Note that even when $d = 2$, $H_{2d}(m)$ is expensive. This illustrates the price of zero tolerance, even when occlusion is not allowed and when the observation is sparse enough with respect to ξ.

In the occlusion case, we first remind the reader of the following

Definition 5.0.2 *Define $\mathcal{C}^p_\xi(M,S)$ to be the set of (ξ,p)-feasible transformations with respect to the model M and the observation S, i.e.,*

$$\mathcal{C}^p_\xi(M,S) = \bigcup_{V \in Reg^p(M)} \mathcal{C}_\xi(V,S),$$

where $Reg^p(M)$ is set of regular subsets of M containing at least $m-p$ points, and

$$\mathcal{C}_\xi(V,S) = \{t \in \mathcal{A}; t(V) \subset (S)\}.$$

By applying all the above changes to the occlusion case we obtain

Theorem 5.0.13 (Occlusion and a tolerance bounded by $(1+\sqrt{3(d-1)})\xi$) *In $O(n^2m^3\log n)$ time we can compute a set of allowable transformations satisfying:*

(i) relaxed feasibility condition: $\mathcal{X} \subset \mathcal{C}^p_{\xi+(1+\sqrt{3(d-1)})\xi}(M,S)$

(ii) covering condition: if $V \in Reg^p(M)$, then $\forall t' \in C_\xi(V,S), \exists t \in \mathcal{X}$, s.t. $\|t-t'\|_V < (1+\sqrt{3(d-1)})\xi$.

Theorem 5.0.14 (Occlusion and any given tolerance) *Given any tolerance bound $\mu > 0$, we can compute in $O((\frac{(1+\sqrt{3(d-1)})\xi}{\mu})^{2d}n^2m^3\log n)$ time a set of allowable transformations satisfying:*

(i) relaxed feasibility condition: $\mathcal{X} \subset \mathcal{C}^p_{\xi+\mu}(M,S)$

(ii) covering condition: if $V \in Reg^p(M)$, then $\forall t' \in C_\xi(V,S); \exists t \in \mathcal{X}$ s.t. $\|t-t'\|_V < \mu$.

Notes:

- In all the above, $2+\sqrt{3(d-1)}$ is a bound on $\Delta(U^*,M)$ or $\Delta(U,V_U)$ which is on the average less than this (See Figure 5-1 and note that when d is equal to 2, $1+\sqrt{3(d-1)}$ is equal to 2.7321). This means a lower tolerance in Theorem 5.0.10 & 5.0.13, a lower time constant in Theorem 5.0.11 & 5.0.14, and a lower distance lower bound on the points of S in Theorem 5.0.12.

- See Figure 5-2 to see the result of a bad selection of U.

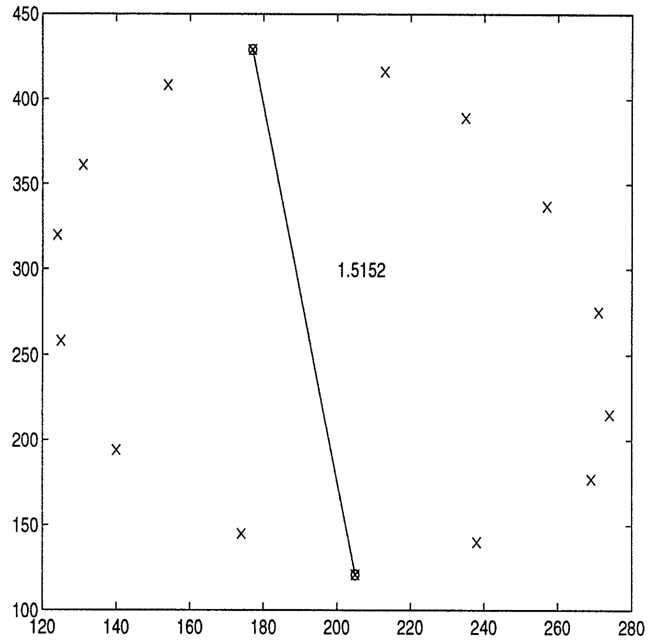
- See Figure 5-3 for values of $\Delta(U, M)$ when U contains more than 2 points.

- A sampling stage and and a multiresolution technique can be embedded in the prespecifiable tolerance algorithm for a better average time constant.

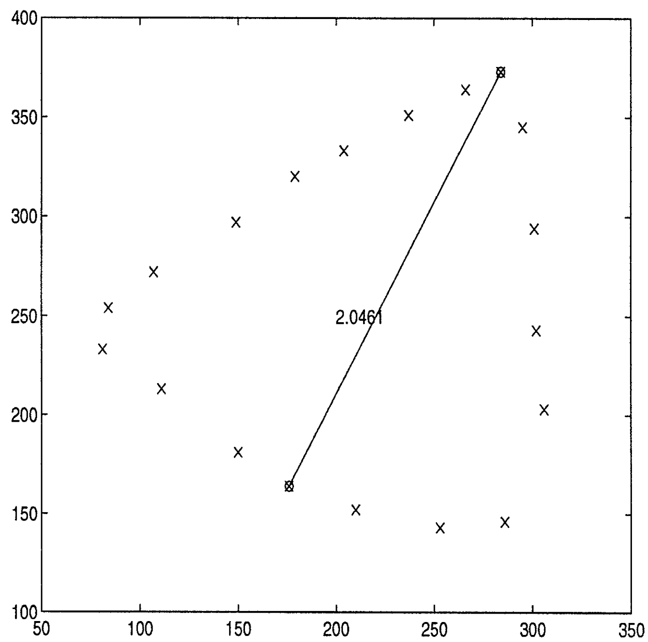
- If we are given some constraints on the $\mathcal{A}$ (for example on the scaling parameter) the constraints can be used in the hypothesis generation process. This leads to a large speed up factor.

- In the occlusion setting, when p is small compared to m and during the submodel generation process, it is wiser to preprocess the model by storing those submodels that passed the cardinality test because most of them will fail the test. This leads to a better time.

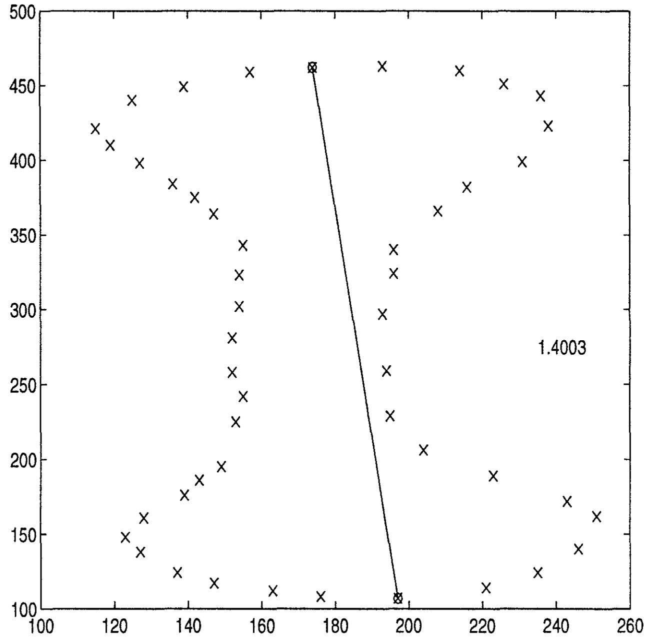
- Except for the zero tolerance result, all the results asymptotically hold for the 2-norms (In the zero tolerance setting, the resulting program become nonlinear).



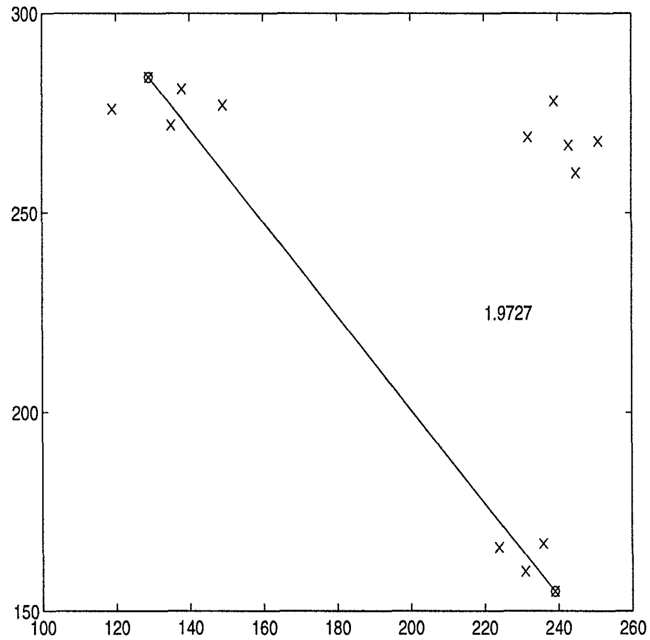
(a) $\Delta(U^*, M) = 1.5152$



(b) $\Delta(U^*, M) = 2.0461$

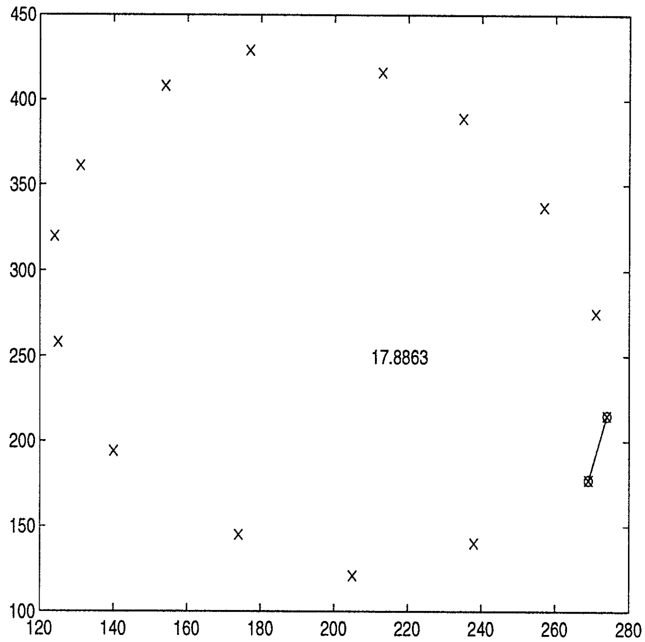


(c) $\Delta(U^*, M) = 1.4003$

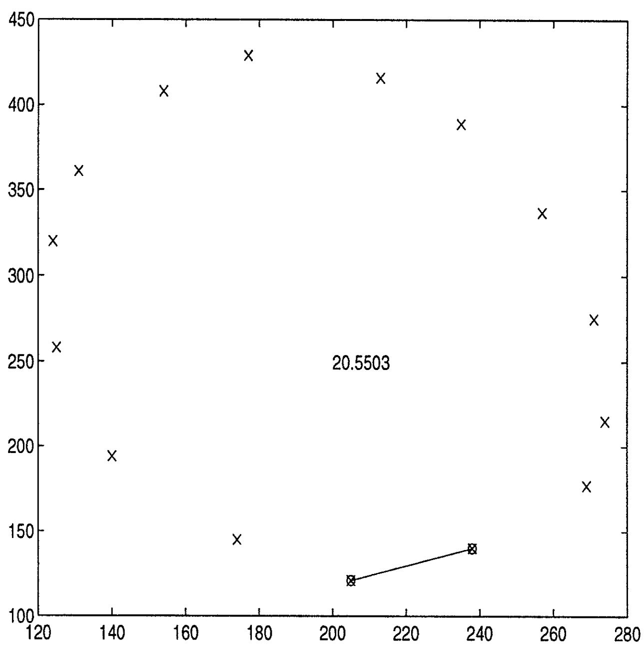


(d) $\Delta(U^*, M) = 1.9727$

Figure 5-1: This figure illustrates four models each with the corresponding 2-point regular subset U^* that minimizes $\Delta(U, M)$. In each case, the points of M are the set of the $\times$'s, the points of U^* are distinguished by the $\circ$'s and the line connecting them.

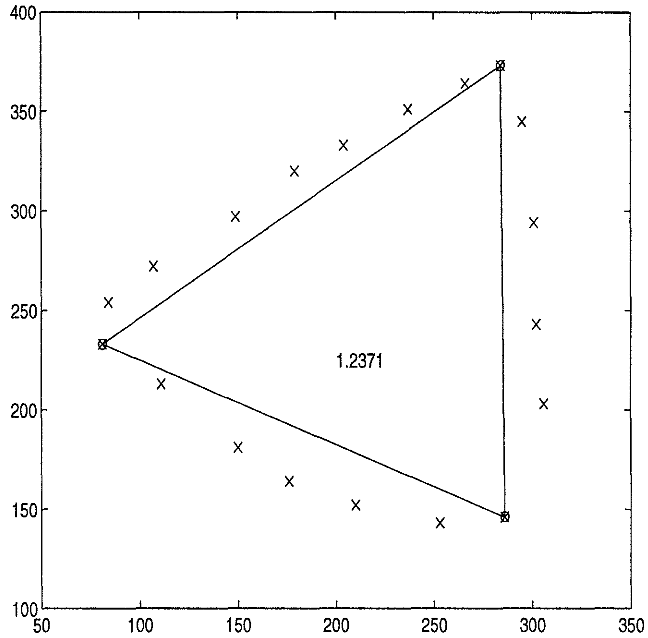


(a) $\Delta(U, M) = 17.8863$

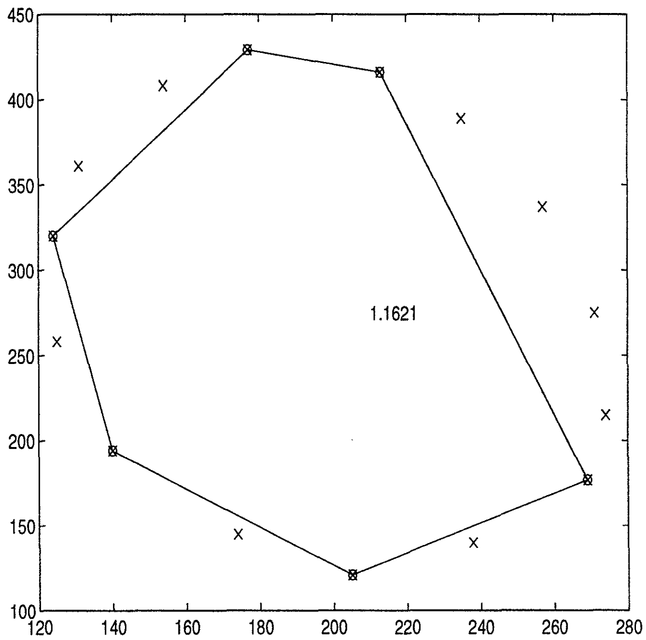


(b) $\Delta(U, M) = 20.5503$

Figure 5-2: This figure illustrates how a bad selection of U can lead to very large value of $\Delta(U, M)$. In each case, the points of M are the set of the $\times$'s, the points of U are distinguished by the $\circ$'s and the line connecting them.



(a) $\Delta(U, M) = 1.2371$



(b) $\Delta(U, M) = 1.1621$

Figure 5-3: This figure illustrates values of $\Delta(U, M)$ when U is a subset of M containing more than 2 points. In each case, the points of M are the set of the $\times$'s, the points of U are distinguished by the $\circ$'s and the edges of their convex span.

Chapter 6

Pattern Sampling

In this chapter, we present a technique that can be used to remove points from the observation while keeping its information content in terms of recognition and localization unaffected. The idea is simple, the set of ε-feasible transformation is given by

$$C_\varepsilon(T,S) = \{t \in \mathcal{A}; t(M) \subset B_\varepsilon(S)\},$$

and the set of (ε, p)-feasible transformation is given by

$$\mathcal{C}_\varepsilon^p(T,S) = \bigcup_{V \in Reg^p(M)} \{t \in \mathcal{A}; t(V) \subset B_\varepsilon(S)\}.$$

So they both depend on the ε-neighberhood of S rather than the points of S. This means that if we can replace S by a new observation S' whose ξ-neighborhood is equal to the ξ-neighborhood of S, i.e,

$$B_\xi(S) = B_\xi(S'), \tag{6.1}$$

we get

$$C_\xi(T,S) = C_\xi(T,S') \tag{6.2}$$

$$C_\xi^p(T,S) = C_\xi^p(T,S') \tag{6.3}$$

$$C_{\xi+\mu}(T,S) = C_{\xi+\mu}(T,S') \tag{6.4}$$

$$C^p_{\xi+\mu}(T,S) \;\; = \;\; C^p_{\xi+\mu}(T,S'), \tag{6.5}$$

where (6.4) and (6.5) follow from the fact that

$$\begin{aligned} B_{\xi+\mu}(S') &= \bigcup_{\alpha\in S'} B_{\xi+\mu}(\alpha) = \bigcup_{\alpha\in S'} \bigcup_{x\in B_\xi(\alpha)} B_\mu(x) = \bigcup_{x\in B_\xi(S')} B_\mu(x) \\ &= \bigcup_{x\in B_\xi(S)} B_\mu(x) = \bigcup_{\alpha\in S} \bigcup_{x\in B_\xi(\alpha)} B_\mu(x) = \bigcup_{\alpha\in S} B_{\xi+\mu}(\alpha) = B_{\xi+\mu}(S). \end{aligned}$$

Therefore, we can run all the previously described algorithms on S' instead of than S, while knowing that the output will be satisfying the same conditions. The motivation is that when S is dense with respect to ξ a large number of points can be removed from S while keeping its ξ-neighborhood unaffected, this leads to a lower running time. It is important to understand that there are cases where we can't remove any point from S (for example when the distance between any two points in S is at least 2ξ). So in contrast to the other algorithms in this work, the algorithm of this section does not make a worst case asymptotic difference. But, it reduces the observation to an irreducible form.

Rather than asking for computing the minimal subset S' of S satisfying (6.1), we will be looking for a locally optimal S', i.e., a subset S' of S that we can't remove any point from without violating (6.1). The motivation behind this tactic is to reduce computational complexity.

Proposition 6.0.15 *Given an n-points subset S of $\mathbb{R}^d$, a subset S' of S satisfying*

(i) $B_\xi(S') = B_\xi(S)$

(ii) $\forall\alpha \in S', B_\xi(S' - \{\alpha\}) \neq B_\xi(S)$

can be computed in $O(n(\log^{d-1} n + h^{d-1}\log h))$ *time, where h is the maximum number of points in S lying in a* 2ξ*-neighborhood.*

proof: Consider the following greedy-like algorithm that takes as input a pattern S and $\xi > 0$.

$SAMPLE =$ " On input (S, ξ)

1. $S' \leftarrow S$
2. for each point α in S
3.   if $B_\xi(\alpha) \subset B_\xi(S' - \{\alpha\})$
4. $S' \leftarrow S' - \{\alpha\}$
5. return S'. "

Let $\alpha_1, \alpha_2, \ldots, \alpha_n$ be the elements of S in the order considered by the loop in Line 2. Let S_i be the value of S' in Line 3 for $\alpha = \alpha_i$, and let $S_{n+1} = S'$. Note that

$$S' = S_{n+1} \subset S_n \subset \ldots \subset S_i \subset \ldots \subset S_1 = S.$$

We use induction to prove that $B_\xi(S_i) = B_\xi(S)$, for $i = 1 \ldots n+1$. Initially, $B_\xi(S_1) = B_\xi(S)$. Now assume that $B_\xi(S_i) = B_\xi(S)$. If $B_\xi(\alpha_i) \not\subset B_\xi(S_i - \{\alpha_i\})$, then $S_{i+1} = S_i$ and consequently $B_\xi(S_{i+1}) = B_\xi(S_i) = B_\xi(S)$. Else if $B_\xi(\alpha_i) \subset B_\xi(S_i - \{\alpha_i\})$, then $S_{i+1} = S_i - \{\alpha_i\}$, and consequently

$$B_\xi(S_{i+1}) = B_\xi(S_i - \{\alpha_i\}) = B_\xi(S_i - \{\alpha_i\}) \cup B_\xi(\alpha_i) = B_\xi(S_i) = B_\xi(S)$$

where the third equality follows from the fact that $B_\xi(\alpha_i) \subset B_\xi(S_i - \{\alpha_i\})$. This completes the induction. For $i = n + 1$, we obtain $B_\xi(S') = B_\xi(S)$, which proves (i). To prove (ii), assume that $B_\xi(S' - \{\alpha_i\}) = B_\xi(S')$ for some $\alpha_i \in S'$. We have

$$B_\xi(\alpha_i) \subset B_\xi(S') = B_\xi(S' - \alpha_i) \subset B_\xi(S_i - \{\alpha_i\}).$$

So α_i must have been removed from S_i at Line 4, which contradicts the fact that it is still in S'.

In order to realize *SAMPLE*, we need to know how to do the test at Line 3. We will see that given an $O(n \log^{d-1} n)$ preprocessing time for S, this can be done in $O(\log^{d-1} n + h^{d-1} \log h)$ time. This leads to an overall $O(n(\log^{d-1} n + h^{d-1} \log h))$ time needed by SAMPLE to halt. Checking whether $B_\xi(\alpha) \subset B_\xi(S' - \{\alpha\})$ can be done as follows:

1. compute $H = B_{2\xi}(\alpha) \cap S$
2. compute $G = H \cap S' - \{\alpha\}$
3. compute the collection of hyperrectangular regions $\Omega = \{B_\xi(\alpha) \cap B_\xi(\beta); \beta \in G\}$
4. check if the volume of the union of the elements of Ω is equal to $(2\xi)^d$

Given an $O(n \log^{d-1} n)$ preprocessing time of S, the intersection of the query square $B_{2\xi}(\alpha)$ with S can be computed in $O(\log^{d-1} n + |H|)$ time. Assuming that we are marking each point of S removed from S', we compute G in $O(|H|)$ time by passing over each point in H and testing whether it is marked in $O(1)$ time. The volume of the union of a collection of N hyperrectangular regions in $d \geq 2$ can be computed in $O(N^{d-1} \log N)$ time. This leads to an $O(\log^{d-1} n + |H| + |\Omega|^{d-1} \log |\Omega|) = O(\log^{d-1} n + h^{d-1} \log h)$ query time needed to test whether $B_\xi(\alpha) \subset B_\xi(S' - \{\alpha\})$. □

Chapter 7

Open Questions

We conclude by a series of open questions in random order:

- Consider the following in the case of planar affine transformations. Let M_k be a planar set consisting of k points uniformly distributed on a circle, U_k be a $(k-1)$-cardinality subset of M_k, and $D_k = \Delta(M_k, U_k)$ (note that D_k is a well defined number). Show that: $\forall k \geq 4$, any finite set M contains a k-points regular subset U with $\Delta(U, M) \leq D_k$ (note that we established this result for $k = 3$). Informally, the idea is that as the number of points in M increases and as M becomes less symmetric, approximating the convex span of M by the convex span of one of its bounded cardinality subsets becomes easier. A similar bound sounds to be true for other cases of allowable transformations.

- Under a distance assumption on the observation, generalize the zero tolerance result to the occlusion setting.

- Define discrete connectivity as follows: Say that a finite set (planar for instance) is r-connected if the r-neighborhood of this set is connected. The derived upper bounds on $\Delta(U, M)$ as M goes for the worst and U for the best does not make use of any assumption on M. On the other hand, $\Delta(U^*, M)$ approaches the derived bounds as the model become more and more disconnected. Under a connectivity assumption on M find a lower upper bound on $\Delta(U, M)$ (as a function of r for instance). The idea is that usually the model is a connected object both ideally and in the observation, and a lower upper bound on $\Delta(U, M)$ means: A lower upper bound on the initial tolerance, a lower upper bound on time constant when the tolerance is given as an input, a lower distance assumptions on the observation in the zero tolerance setting, and a lower number of the derived submodels in the the occlusion setting.

Appendix A

An Equivalent Startup Algorithm

We describe $\mathcal{Q}_1'$, an algorithm equivalent to $\mathcal{Q}_1$ (of section 4.2). $\mathcal{Q}_1'$ leads to the same worst case tolerance ($\xi\Delta(U,M)$) in the same worst case running time ($O(n^2 m \log n)$). $\mathcal{Q}_1'$ might be more transparent than $\mathcal{Q}_1$ because the value of $\Delta(U,M)$ is not needed in the computational process. $\mathcal{Q}_1'$ even leads to a lower average tolerance, but it is more complicated than $\mathcal{Q}_1$.

Algorithm A.0.16 *Consider the following algorithm that takes as input: the model M, the observation S, the noise bound ξ, and any 3-point regular subset U of M.*

$\mathcal{Q}_1'$ = " *On input* (M, S, U, ξ)

1. *initialize* $\mathcal{X}$ *to the empty set*
2. *repeat the following for each mapping* f *from* U *to* S
3. *compute the unique affine transformation that agrees with* f *on* U, *i.e, compute the element* t *of* $\mathcal{A}$ *satisfying* $f = t|U$
4. *compute the set of transformations* $C = \{t' \in \mathcal{A}; \|t' - f\|_U < \xi\}$
5. *for each* $\alpha \in M$
6. *compute the planar region* $R = C(\alpha) + B_\xi(0)$
7. *check if* $R \cap S \neq \emptyset$
8. *if each point in* M *leads to a nonempty intersection, add* t *to* $\mathcal{X}$
9. *return* $\mathcal{X}$"

Remark: $C(\alpha)$ means $\{t'(\alpha); t' \in C\}$ and $C(\alpha) + B_\xi(0)$ means $\{x + y; x \in C(\alpha)$ and $y \in B_\xi(0)\}$, i.e., the Minkowsky addition of $C(\alpha)$ with the ξ-square centered at the origin.

Theorem A.0.17 (An $\xi\Delta(U, M)$ tolerance) *The output $\mathcal{X}$ of $\mathcal{Q}'_1$ is a set of $(\xi + \xi\Delta(U, M))$-feasible transformations whose $\xi\Delta(U, M)$-neighborhood covers all the ξ-feasible transformations.*

Proof: We have to show that

(i) covering condition: for each ξ-feasible transformations t', there exists a transformation t in $\mathcal{X}$ such that $\|t - t'\|_M < \xi\Delta(U, M)$.

(ii) relaxed feasibility condition: each transformations in $\mathcal{X}$ is $(\xi + \xi\Delta(U, M))$-feasible.

We prove *(ii)* first. Consider any t in $\mathcal{X}$. Because t was added to $\mathcal{X}$, we must have

$$(C(\alpha) + B_\xi(0)) \cap S \neq \emptyset, \forall \alpha \in M,$$

where C is the set of transformations computed at Line 4 when the mapping $f = t|U$ was considered. In other words, $C = \{t' \in \mathcal{A}; \|t' - t\|_U < \xi\}$.

Construct the mapping $g : M \to S$ by picking, for each $\alpha \in M$, $g(\alpha)$ form the nonempty set $(C(\alpha) + B_\xi(0)) \cap S$. We argue below that $\|t - g\|_M < \xi + \xi\Delta(U, M)$, which by Lemma 4.2.2 implies that t is $(\xi + \xi\Delta(U, M))$-feasible. This will establish *(ii)*.

Consider any point α of M. We have $g(\alpha) \in C(\alpha) + B_\xi(0)$, or in other words, $\|t'(\beta) - g(\alpha)\| < \xi$, for some $t' \in C$. It follows that

$$\begin{aligned}
\|t(\alpha) - g(\alpha)\| &\leq \|t(\alpha) - t'(\alpha)\| + \|t'(\alpha) - g(\alpha)\| && \text{(A.1)}\\
&< \sup_{t'' \in C} \|t(\alpha) - t''(\alpha)\| + \xi && \text{(A.2)}\\
&\leq \sup_{t'' \in C} \|t - t''\|_M + \xi && \text{(A.3)}\\
&\leq \sup_{t'' \in C} \Delta(U, M)\|t - t''\|_U + \xi && \text{(A.4)}\\
&\leq \Delta(U, M)\xi + \xi, && \text{(A.5)}
\end{aligned}$$

where (A.4) follows from Lemma 4.1.1 and (A.5) from the definition of C.

This is true for any $\alpha \in M$, therefore $||t-g||_M < \xi + \xi\Delta(U, M)$, which by Lemma 4.2.2 implies that t is $(\xi + \xi\Delta(U, M))$-feasible, and hence the correctness of (ii).

To prove (i) consider any ξ-feasible transformations t'. Using Lemma 4.2.2 let $g : M \to S$ s.t. $||t' - g||_M < \xi$. And finally let t be the unique transformation in $\mathcal{A}$ satisfying $t|U = g|U$. We will shows that $||t - t'||_M < \xi\Delta(U, M)$ and $t \in \mathcal{X}$. This will establish (i). We have

$$\begin{aligned} ||t - t'||_M &\leq \Delta(U, M)||t - t'||_U && \text{(A.6)} \\ &= \Delta(U, M)||g - t'||_U && \text{(A.7)} \\ &\leq \Delta(U, M)||g - t'||_M && \text{(A.8)} \\ &< \xi\Delta(U, M), && \text{(A.9)} \end{aligned}$$

where (A.6) follows from Lemma 4.1.1, (A.7) follows from the fact that $t|U = g|U$, and (A.8) from the fact $U \subset M$.

The transformation t agrees on U with a mapping from U to S. So to demonstrate that $t \in \mathcal{X}$, we only have to show that

$$(C(\alpha) + B_\xi(0)) \cap S \neq \emptyset, \forall \alpha \in M, \tag{A.10}$$

where $C = \{t'' \in \mathcal{A}; ||t'' - t||_U < \xi\}$. Consider any $\alpha \in M$, we have $||g(\alpha) - t(\alpha)|| \leq ||g - t||_M < \xi$, or $g(\alpha) \in t(\alpha) + B_\xi(0)$. So $g(\alpha) \in C(\alpha) + B_\xi(0)$ because $t \in C$. Noting that $g(\alpha) \in S$ also, we obtain $(C(\alpha) + B_\xi(0)) \cap S \neq \emptyset$. This is true for any $\alpha \in M$, which proves (A.10) and hence (i). □

Regarding the realization of $\mathcal{Q}'_1$, we argue below that each of the C's computed at Line 4 is a $6D$-convex polytope that can be written as the product of two $3D$-convex polytopes C_1 and C_2, each having $O(1)$ extreme points, and each computable in $O(1)$ time. We argue also that each of the R's is a planar rectangular region that can be computed from the extreme points of C_1 and C_2 in $O(1)$ time. Given an $O(n \log n)$ preprocessing time for S the number of points in the intersection of a query rectangular region with S can be computed in $O(\log n)$ time. So, the feasibility test performed on t in Lines $5-7$ can be done in $O(m \log n)$ time. This leads to an overall $O(n \log n + n^2 m \log n) = O(n^2 m \log n)$ time needed by $\mathcal{Q}_1'$ to halt.

Consider any such C.

$$C = \{t' \in \mathcal{A}; \|t' - f\|_U < \xi\}.$$

Let $\alpha_1, \alpha_2, \alpha_3$ be the points of U and let $\beta_i = f(\alpha_i)$ for $i = 1, 2, 3$. If we denote by $H_{2\times 2}$ and $r_{2\times 1}$ - respectively - the linear transformations matrix and the translation vector of t', we can write the constraint $\|t' - f\|_U < \xi$ as

$$\left\| [H|r] \begin{bmatrix} \alpha_i \\ 1 \end{bmatrix} - \beta_i \right\| < \xi, i = 1, 2, 3.$$

If we let $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = [H|r]$ where $x_1, x_2 \in \mathbb{R}^{3\times 1}$, we can reduce the constraint to

$$|x_1 \begin{bmatrix} \alpha_i \\ 1 \end{bmatrix} - \beta_{ix}| \quad < \xi \quad , i = 1, 2, 3, \tag{A.11}$$

$$|x_2 \begin{bmatrix} \alpha_i \\ 1 \end{bmatrix} - \beta_{iy}| \quad < \xi \quad , i = 1, 2, 3. \tag{A.12}$$

Due to the regularity of U, each of the constraints defines a bounded region in $\mathbb{R}^3$ (This can be established using an argument similar to the one we used in the proof of Proposition 4.3.4). So C is the product of two (open) $3D$-convex polytopes C_1 and C_2, given respectively by (A.11) and (A.12). Each of the polytopes is determined by 6 halfspaces, so it has $O(1)$ extreme points and it is computable in $O(1)$ time.

Now, consider any such R.

$$R = C(\alpha) + B_\xi(0).$$

If t is a planar affine transformation given by the linear transformation H and the translation vector r, we can represent t by x_1 and x_2, the two 1×3 vectors corresponding to the first and second row of $[H|r]_{2\times 3}$. Note that with this representation

$$t(\alpha) = [H|r]\begin{bmatrix}\alpha\\1\end{bmatrix} = \begin{bmatrix}x_1\\x_2\end{bmatrix}\begin{bmatrix}\alpha\\1\end{bmatrix} = \begin{bmatrix}x_1\begin{bmatrix}\alpha\\1\end{bmatrix}\\x_1\begin{bmatrix}\alpha\\1\end{bmatrix}\end{bmatrix}.$$

So we can express $C(\alpha)$ as

$$C(\alpha) = (C_1 \times C_2)(\alpha) = \{t(\alpha); t \in C_1 \times C_2\} = \left\{\begin{bmatrix}x_1\begin{bmatrix}\alpha\\1\end{bmatrix}\\x_1\begin{bmatrix}\alpha\\1\end{bmatrix}\end{bmatrix}; x_1 \in C_1, x_2 \in C_2\right\},$$

i.e.,

$$C(\alpha) = \begin{bmatrix}C_1L_\alpha\\C_1L_\alpha\end{bmatrix}, \text{ where } L_\alpha = \begin{bmatrix}\alpha\\1\end{bmatrix}.$$

In other words, $C(\alpha)$ is the product of C_1L_α and C_2L_α, the images of the convex polytopes C_1 and C_2 by the linear transformation L_α. Because L_α is linear and C_1 is a convex polytope, $C_1\alpha$ is also a convex polytope whose set of extreme points is the convex hull of the image of the set of extreme points of C_1 by L_α. But C_1L_α is one dimensional, so it is an (open) interval whose two extreme points can be computed by maximizing and minimizing the linear transformation L_α over the set of extreme points of C_1. Similarly, C_2L_α is an (open) interval that can be computed from the extreme points of C_2. It follows that $C(\alpha)$ is an (open) rectangular region that can be computed in $O(1)$ time given the extreme points of C_1 and C_2. Once $C(\alpha)$ is computed, R can be simply computed by dilating the rectangular region $C(\alpha)$ with the square $B_\xi(0)$.

Bibliography

[1] Herbert Edlesbrunner, *Algorithms in Combinatorial Geometry* , Springer-Verglas, Berline Heidelberg, 1987.

[2] J. Goodman & J. Rourke, *Handbook of Discrete and Computational Geometry,* CRC Press, Florida, 1997.

[3] F. Preparata and M. Shamos, *Computational Geometry: An Introduction,* Springer-Verlag, New York, 1985.

[4] Roger Webester, *Convexity,* Oxford University Press, New York, 1994.

[5] H. Alt, K. Mehlhorn, H. Wagner, & E. Welzl, *Congruence, similarity, and symmetry of geometric objects,* Discrete and Computational Geometry, 3, pp. 237–256, 1988.

[6] R. Baris & S. Ullman, *The alignment of objects with smooth surfaces,* Second Int. Conf. Comp. Vision, pp. 482-488, 1988.

[7] Henry Braid, *Model-based image matching using location,* MIT Press, Cambridge MA, 1985.

[8] Todd Cass, *Feature matching for object localization in the presence of uncertainty,* Proc. of the IEEE 3rd Conf. Comp. Vision, Osaka, Japan, pp. 360-364, 1990.

[9] M. Costa & R. Haralick L. Shapiro, *Optimal affine invariant point matching,* Proc. 6th Israel Conf. on AI, pp. 35-61, 1990.

[10] L. Chew, M. Goodrich, D. Huttenlocher, K Kedem, J. Kleinberg,& D. Kravets, *Geometric pattern matching under euclidean motion*, Computational Geometry Theory and Application, 7, pp. 113-124, 1997.

[11] D. Huttenlocher & S. Ullman, *Object recognition using alignment*, Proc. First Int. Conf. Comp. Vision, pp. 102-111, 1987.

[12] D. Huttenlocher & S. Ullman, *Recognizing solid objects by alignment with an image*, Inter. Journ. Comp. Vision 5(2):195-212, 1990.

[13] E. Grimson & T. Lozano-Perez, *Model-based recognition and localization from sparse range or tactile data*, Int. J. Robotics Res., 3(3):3-35, 1984.

[14] E. Grimson & T. Lozano-Perez, *Localizing overlapping parts by searching the interpretation tree* , IEEE trans. Patt. Anal. & Mach. Intell., 9(4):469-482, 1987.

[15] Eric Grimson, *Object Recognition by computer: The role of geometric constraints*, MIT Press, Cambridge MA, 1990.

[16] E. Grimson D. Huttenlocher D. Jacobs, *A study of affine matching with bounded sensor error*, Int. Journ. Comp. Vision 13(1):7-32, 1994.

[17] Y. Landman H. Wolfson, *Geometric hashing: a general and efficient model-based recognition scheme*, Second Int. Conf. Comp. Vision pp. 238-249, 1988.

[18] Y. Landman J. Schwartz H. Wolfson, *Affine invariant model based-object recognition*, IEEE trans. Rob. Aut. vol. 6 pp. 578-589, 1990.

[19] D. Thompson J. Muddy, *Three dimensional model matching from unconstrained viewpoint*, Proc. IEEE Conf. Comp. Vis. and Patt. Recog. pp.454-460, 1987.

[20] L. Van Gool P. Kempenaers & A. Oosterlinck, *Recognition and semi-differential invariants*, IEEE Conf. Comp. Vis. and Patt. Recog. pp.473-478, 1991.