國立交通大學

資訊科學研究所

博士論文

利用位能場模型作路徑規劃及物體形狀比對

Potential-Based Path Planning and Shape Matching

研　究　生：林建州
指導教授：莊仁輝　博士

中華民國九十三年九月

利用位能場模型作路徑規劃及物體形狀比對

Potential-Based Path Planning and Shape Matching

研究生：林建州　　Student：Chien-Chou Lin
指導教授：莊仁輝　　Advisor：Jen-Hui Chuang

國立交通大學
電機資訊學院
資訊科學研究所
博士論文

A Thesis
Submitted to Institute of Computer and Information Science
College of Electrical Engineering and Computer Science
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in

Computer and Information Science

September 2004

Hsinchu, Taiwan, Republic of China

中華民國九十三年九月

利用位能場模型作路徑規劃及物體形狀比對

研究生：林建州　　指導教授：莊仁輝 博士

國立交通大學資訊科學研究所

摘要

本博士論文乃利用位能場模型來呈現自由空間的方法，探討位能場於不同問題的應用。我們提出數個以位能場模型為基礎的演算法來協助解決（1）機器人路徑規劃與（2）物體形狀比對的問題。在這些演算法中，其共同的主要概念是將物體及自由空間的邊界帶相同之電性後，靜止的自由空間會有位能場形成，且自由空間對物體產生推斥力，再將作用於物體上的推斥力作為物體移動的推力與轉動的轉矩，使物體降低在自由空間中位能與自由空間達成最佳的形狀比對，物體的形式可以是剛體或是連結物。最佳的形狀比對可藉由調整物體的位置、姿態來降低位能而達成。

在機械手臂路徑規劃研究中，我們探討了利用牛頓位能場應用於二維路徑規劃問題的可行性。在研究中，我們將機械手臂及自由空間的邊界以均勻或不均勻的帶電後，如同於電學中的定義，可計算得知機械手臂在工作空間中所受的斥力與轉矩，藉由這些斥力與轉矩來調整手臂於自由空間之組態使其位能降低。透過尋找這一連串的組態後，一條避碰的路徑即可由這些一系列組態來構成。我們也將上述演算法推廣到三維工作空間，亦即將原來牛頓位能場模型用三維廣義位能場 [1] 來取代。在 [1] 中提到，牛頓位能場模型在三維空間之中會造成物體撞入障礙物中，並不能達到避碰的要求。因而，我們採用廣義位能場模型來達成避碰。相對於機械手臂的基點是固定的，連結型機器人因有可移動的基點所以在路經規劃上有較高的自由度。在本論文中，我們亦發展了相對應之路徑規劃演算法。

另外，透過相同的廣義位能場模型的建立，我們亦提出一三維物體形狀比對的演算法。在 [2] 中，廣義位能場模型被用來做單一剛體在一靜態空間的路徑規劃；

事實上，在路徑規劃的過程中，降低剛體於靜態空間位能的作法可視為某種程度的剛體與自由空間的形狀比對。在路徑規劃演算法中，剛體位置會沿著路徑改變、但大小不變；然而，在形狀比對中，比對物體的位置變化不大，但其大小卻會隨之增長。當比對物體增大後，如能調整其位置及姿態以降低位能，則可獲得一個較好的比對形狀。此外，由於三維物體資料經常為點資訊（如 range data)，資料量大且不易解讀，以位能場為基礎之形狀比對演算法可直接使用物體的點資訊作比對，可省去費時的物體點資訊前處理。由實驗結果得知，所提出的演算法在物體之路徑規劃與形狀比對均有不錯之成果，而後者對於部分遮蔽的物體亦可適用。

關鍵字：廣義位能場，路徑規劃，障礙物避碰，形狀比對

Potential-Based Path Planning and Shape Matching

Student：Chien-Chou Lin　　　Advisor：Dr. Jen-Hui Chuang

Institute of Information and Computer Science
National Chiao Tung University

Abstract

In this thesis, along the general direction of free space modeling using potential models, various applications of potential models are investigated. Variant potential-based algorithms are proposed to solve (1) path planning and (2) shape matching problems. The common idea of these algorithms is to use the repulsion exerted on an object, in forms of repulsive force and torque, from free space boundaries to achieve the best shape match between them. The object can be rigid or articulated, and the best match in shape is accomplished by adjusting object configuration, i.e., location and orientation, to minimize the potential fields among them.

In the path planning algorithm of manipulators, the Newtonian potential is used to represent manipulators and obstacles with charged boundaries in a 2-D workspace. The approach computes, similar to that done in electrostatics, repulsive force and torque between charged objects in the workspace. A collision-free path of a manipulator will then be obtained by locally adjusting the manipulator configuration to search for minimum potential configurations using these force and torque. The proposed approach is efficient because these potential gradients are analytically tractable. The above potential-based path planning approach for manipulators is extended to three dimensions using the generalized potential model [1] instead of Newtonian potential. In [1], it is shown that the Newtonian potential, being harmonic in the 3-D space, can not prevent a charged point object from running into another object whose surface is uniformly charged. While the base of a manipulator is fixed, an articulated robot has higher DOF due to its moving base. A modified path planning algorithm based on the same generalized potential model is also proposed for articulated robots with moving bases in this thesis.

In addition, the repulsion between an input object and a shape template is also utilized in the shape matching approach proposed in this thesis. In [2], it proposed a potential-based path planner for a single rigid robot among stationary and rigid obstacles in 3-D workspace. Indeed, the minimization of potential between robots and obstacles is a shape matching procedure of a robot within a free space in some respects. While a rigid robot moves along a path in a free space without changing its size in path planning, the object stays about the same location inside the shape template with growing size in shape matching. According to the proposed approach, a better match in shape between the template object and the input object can be obtained if the input object translates and reorients itself to reduce the potential while growing in size. Since objects are usually represented by mass and unstructured row data, e.g., range data, existed shape matching algorithms may have a preprocessing procedure to extract features from row data of objects. However, the proposed potential-based algorithm can directly perform the matching with range data of objects without preprocessing procedures. Simulation results show that the proposed algorithms work well for both path planning and shape matching applications. The latter is also practicable to objects with incomplete surface descriptions, e.g., due to a partial view.

Keywords: generalized potential model, path planning, obstacle avoidance, shape matching

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Recently, robots are widely used in factories and in our daily life because researchers are improving robots to do more complex jobs. Path planning is one of the most fundamental problems in robotics. The basic path planning problem of a robot is to find a collision-free path in a static environment being cluttered with obstacles. In the past two decades, a variety of planners [3] [4] [5] [6] [7] have been proposed in the 2-D workspace. Many of the above planner can also be generalized for robots of high degrees of freedom (DOF) and to perform well in 3-D workspace. However, the computation complexity increases dramatically as the DOF of robot is high ($DOF > 3$).

For the application sake, it is necessary to develop a fast planner for robot manipulator with many DOFs. The fast planner should be capable of: (i) producing solutions operating in a static environment and (ii) dealing with a changing environment while providing small reaction time compared with the robot's motion cycle. The most straightforward approaches are geometric algorithms which use spatial occupancy information in the workspace to solve path planning problem directly [1] [2] [8] [9] [10] [11] [12] [13] [14] [15].

In [1], an analytically tractable generalized potential model is proposed for obstacle avoidance, which assumes surfaces of obstacles and robots are uniformly charged. It is shown that the potential and the resulting repulsion (in forms of force and torque) between point charges and polygonal surfaces in the 3-D space regions can be calculated in closed form. A rigid robot can thus be adjusted by the repulsion to move in free space with low potential configurations. In this thesis, such an idea is generalized to multi-link articulated robots in 2-D and 3-D environments, respectively.

On the other hand, the repulsion between an object and a shape template is also utilized in the shape matching approach proposed in this thesis. While a rigid robot moves along a path in a free space without changing its size in path planning, the object stays about the same location inside the shape template with growing size in shape matching.

1.2 Survey of Path Planning in 2-D Environments

Path planning of a manipulator is to determine a collision-free trajectory from its original location and orientation (called starting configuration) to goal configuration [16]. The simplest case involves a planar polygonal robot static polygonal obstacles. There exists a large number of methods for solving the basic path planning problem that can be categorized into two general approaches: configuration space (c-space) based approach, and workspace-based approach. Some planners adopt the c-space-based approach [3] [4] [5] [6] [7], which considers both the manipulator and obstacles at the same time by identifying manipulator configurations intersecting the obstacles. A point in a c-space indicates a configuration of manipulator. A configuration is usually encoded by a set of manipulator's parameters, i.e., angles of links of manipulators. The forbidden regions in the c-space are the points which imply manipulator configurations intersecting the obstacles. Thus, path planning is reduced to the problem of planning a path from a start point to goal in free space.

Unlike c-space based approaches, geometric algorithms, such as cell decomposition and potential field based approaches, directly use spatial occupancy information of the workspace to solve path planning problem [1] [9] [10] [11] [12]. Workspace-based algorithms usually extract the relevant information about the free space and use them together with the manipulator geometry to find a path. Cell decomposition methods [10] [17] consist of decomposing the robot's free space into simple regions, called cells. Thus a path between any two configurations in a cell can be easily generated. A connectivity graph, representing the connectivity between the cells, is then constructed and searched for a path represented by a sequence of free cells. However, the paths generated are not intended to be the safest one. One way to obtain the best match and minimize the risk of collision is to define a repulsive potential field between the robot and obstacles [9] [11] [12].

In general, the potential function used to model the workspace can be a scalar function of the distances between boundary points of the robot and those of obstacles. The gradient of such a scalar function, i.e., the repulsive force between the robot and obstacles, can be used to move the former away from the latter making potential-based methods simple. (For a survey of related works please see also [11] and [16].)

In this thesis, an artificial potential field, whose magnitude is unbounded near the obstacle boundary and decreases with range [18], is applied to model the workspace for the path planning of robot manipulators. The approach computes, similar to that done in electrostatics, repulsive force and torque between objects in the workspace. A collision-free path of a manipulator will then be obtained by locally adjusting the manipulator configuration to search for minimum potential configurations using these force and torque. The proposed approach uses one or more guide lines (GLs) as final or intermediate goals in the workspace. The GLs are line segments among obstacles in the free space, providing the manipulator a general direction to move forward. As a GL is an intermediate goal for a manipulator to reach, it also helps to establish certain motion constraints for adjusting manipulator configuration during path planning.

1.3 Survey of Path Planning in 3-D Environments

The earlier works in path planning have focused on the problem of a rigid robot in 2-D workspace [3] [4] [5] [6] [7] [8] [9] [11] [12]. However, not all of them solve the problem in its full generality. For example, some methods require the workspace to be 2-D. Some approaches may be extended to 3-D workspace, but have exponential complexity and require exact descriptions of both robot and workspace [6] [7] [14] [15].

In this thesis, the potential field model presented in [1] for 3-D space is adopted to model the workspace for the path planning of robot manipulators. The approach computes, similar to that done in [1] for the 2-D case, repulsive force and torque between objects in the workspace. A collision-free path of a manipulator is also obtained by locally adjusting the manipulator configuration to search for minimum potential configurations using the repulsion. The proposed approach uses one or more guide planes (GPs) among obstacles in the 3-D free space as final or intermediate goals in the workspace for the manipulator to reach. These GPs provide the manipulator a general direction to move forward and also help to establish certain motion constraints for adjusting manipulator configuration during path planning.

1.4 Survey of Shape Matching of 3-D Objects

Although the recognition of 3-D objects is of primary interest in computer vision, many 2-D shape matching algorithms are developed for situations which can be regarded as two dimensional. Template matching methods are presented in [19][20] [21] for object recognition. Fourier descriptors [22] [23] transform the coordinates of boundary points into a set of complex numbers for the matching. Moments of 1D functions representing segments of shape contours are used in the matching in [24] [25] [26] [27]. Stochastic models, e.g., autoregressive models, are used for shape classification in [28] [29] [30]. In [31] [32] [33], the boundary of a region is represented by a sequence of numbers and the shape matching is accomplished by string matching. Other shape matching methods involve finding the polar transform of the shape sample [34] or calculating the distances of the feature points from the centroid [35], etc. In [36], the shape matching is accomplished by graph matching for multilevel structural descriptions of shape samples. Multiple 1D matching processes for multiscale curvature descriptions are adopted in [37] in building shape models. Usually, the scaling of the object, the viewscale of the shape contour, and the rotation and the translation of the object need to be determined before a final matching process can take place. The matching process may involve the matching of binary images, discrete 1D data, or extracted shape features arranged into structured data. The recognition of 3-D objects is one of the most challenging problems in computer vision. The recognition is much harder than the 2-D case because of the complexity involved with the extra degree of freedom.

Several recognition systems extract features from 2-D images and match them to corresponding features in a database of 3-D object models. The approaches turn the recognition process into a procedure of verifying each candidate hypothesis and finally ranking the verified hypotheses [38] [39] [40] [41]. Other approaches of recognition of 3-D objects from 2-D images use different image features [42] [43] [44] [45] [46]. Several physically based computer vision methods have been proposed for the representation and recognition of 3-D objects using the finite element method [47] [48] [49].

In [50], a potential-based approach for 2-D shape matching is proposed. The matching process involves the minimization of a scalar function, the potential. The potential model assumes that the border of any 2-D region is uniformly charged. If a shape template is small in size and can be placed inside a region whose shape is to be determined, the template

Table 1.1: A comparison of proposed algorithms for different applications.

Algorithm	Potential Model	Object Size	Object Location	Base Point
2-D Manipulator Path Planning	2-D	no change	along a path	fixed base
3-D Manipulator Path Planning	3-D	no change	along a path	fixed base
3-D Articulated Robot Path Planning	3-D	no change	along a path	moving base
3-D Shape Matching	3-D	growing	about the same place	

will experience repulsive force and torque arising from the potential field. The basic idea of the approach is to achieve a better match in shape between the template and the given region by translating and reorienting the template along the above force and torque directions, respectively, toward the configuration of the minimum potential. The template is then expanded and its configuration re-adjusted until the template almost touches the border of the given region. For a selected group of shape templates, the template with the largest final size is considered as the best match. Such an approach is intrinsically invariant under translation, rotation and size changes of the shape sample. The potential-based shape matching approach is extended to three dimensions using a generalized potential model presented in [1] in this thesis.

1.5 Organization of the Thesis

In this thesis, along the general direction of free space modeling using potential models, various applications of potential models are investigated. The common idea of these algorithms is to use the repulsion exerted on an object, in forms of repulsive force and torque, from free space boundaries to achieve the best shape match between them. The object can be rigid or articulated, and the best match in shape is accomplished by adjusting object configuration, i.e., location and orientation, for minimum potential. Depending on the dimensions of free space, the degrees of freedom of the object, the extent of location change of the object (end-effecter), and the possible size change of the object, algorithms investigated in this thesis are summarized in Table 1.1.

The remainder of this thesis is organized as follows. In Chapter 2, the adopted potential field models presented in [13] [1] are reviewed. In 2-D environment, the free space are modelled by considering the potential due to both uniform and non-uniform source distributions on polygonal region borders of robots and obstacles. In 3-D environment, the free space are modelled by considering the potential due to uniform source distributions on surfaces of robots and obstacles. The repulsion experienced by an object of finite size due to the potential gradient is considered. It is shown that the repulsion between polygonal object and obstacles, in forms of force and torque, can be derived in closed form.

In Chapter 3, applications of the Newton potential model [18] to path planning of a manipulator in 2-D work space are considered. The repulsive force and torque between the robot manipulator and obstacles are used to adjust the position and orientation of the former so as to keep it away from the latter. In Chapter 4, a potential-based path planning algorithm is proposed for 3-D manipulators. The proposed approach uses one or more guide planes (GPs) among obstacles in the free space as final or intermediate goals in the workspace for the articulated object to reach. These GPs provide the articulated object a general direction to move forward and also help to establish certain motion constraints for adjusting robot configuration for minimum potential during path planning. An extension of the path planning approach is proposed in Chapter 5 to derived a collision-free path for articulated robots with moving bases. In Chapter 6, a shape matching approach of 3-D object using the same 3-D potential model is presented which identifies the object shape with respect to a set of shape templates. The approach allows the object to grow in size inside a shape template and adjusts the object configuration according to the repulsive force and torque until a potential minimum is reached. Chapter 7 summarizes this thesis.

Chapter 2

Reviews of Adopted Potential Models

2.1 A Review of the 2-D Potential Model

2.1.1 Repulsion due to Charged Polygonal Region Borders

Polygonal description of obstacles is often used in path planning because of its simplicity. The repulsive force and torque between these polygonal regions can be derived by superposing the repulsion between pairs of border segments, each contains one line segment from the the moving object and the other from one of the obstacles. In general, each pair of the repelling line segments can have arbitrary configuration in the workspace as shown in Fig. 2.1(a). To simplify the expressions of the repulsion between them in the following subsections, a coordinate system is chosen so that the obstacle line segment, $\overline{AB}$, lies on the base line, as shown in Fig. 2.1(b). In the new coordinate system (uv-plane), the coordinates of the endpoints of $\overline{AB}$ are assumed to be $(0, 0)$ and $(d > 0, 0)$, respectively, and the line containing the object line segment , $\overline{CD}$, can be represented as $v = au + b$, $u_1 \leq u \leq u_2$.

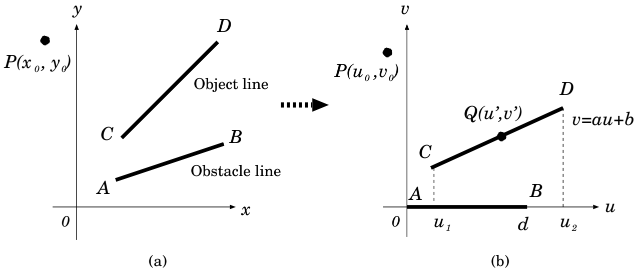


Figure 2.1: Coordinate transformation. (a) The original coordinate system (xy-plane). (b) The new coordinate system (uv-plane) after the transformation.

2.1.2 Integral equations for forces and torques

Consider the electric field at point $Q = (u', v')$ of $\overline{CD}$ due to a point $(u, 0)$ on $\overline{AB}$ shown in Fig. 2.1(b). We have

$$\vec{E} \triangleq (E_u, E_v) = -\nabla \left[\frac{1}{r}\right] = -\frac{1}{r^2}\hat{r} \tag{2.1}$$

where $\vec{r} = (u' - u, v')$, $r = |\vec{r}|$, and $\hat{r} = \vec{r}/r$. Thus, we have the total force at point Q due to $\overline{AB}$, $\vec{F} \triangleq (F_u, F_v)$, with

$$F_u(u', v') = \int_{q_0}^{q_d} E_u dq = \int_0^d \frac{u' - u}{r^3}\rho(u)du \tag{2.2}$$

$$F_v(u', v') = \int_{q_0}^{q_d} E_v dq = \int_0^d \frac{au' + b}{r^3}\rho(u)du \tag{2.3}$$

where $\rho(u)$ is the charge density along $\overline{AB}$.

For the total force on the object line segment $\overline{CD}$, we have[1]

$$F_u = \int_{q_1}^{q_2} F_u(u', v')dq = \int_{s_1}^{s_2} F_u(u', v')\rho(s)ds \tag{2.4}$$

where $s = \sqrt{1 + a^2}u'$, $\rho(s)$ is the charge density along $\overline{CD}$, and $dq = \rho(s)ds$. Thus, the above integral equations can be formulated as

$$F_u = \sqrt{1 + a^2} \int_{u_1}^{u_2} \int_0^d \frac{u' - u}{r^3}\rho(u)\rho(u')dudu' \tag{2.5}$$

Let $P = (u_0, v_0)$ be a reference point, e.g., the rotation center of the object, as shown in Fig. 2.1(b). The torque with respect to P, due to the repulsive force from $\overline{AB}$ on point Q is equal to

$$\begin{aligned}\tau_P(u', v')\vec{i_z} &= \vec{l}(u', v') \times \vec{F}(u', v') \\ &= (u' - u_0, v' - v_0) \times (F_u(u', v'), F_v(u', v'))\end{aligned} \tag{2.6}$$

where $\vec{i_z} = \vec{i_u} \times \vec{i_v}$ and $\vec{l}(u', v') = \vec{PQ}$. Thus, the total torque becomes

$$\begin{aligned}\tau_P \;=\; & \int_{s_1}^{s_2} \tau_P(u', v')\rho(s)ds = \sqrt{1 + a^2} \\ & \left(\int_{u_1}^{u_2} \int_0^d (u' - u_0)\frac{au' + b}{r^3}\rho(u)\rho(u')dudu'\right. \\ - & \left.\int_{u_1}^{u_2} \int_0^d (au' + b - v_0)\frac{u' - u}{r^3}\rho(u)\rho(u')dudu'\right)\end{aligned} \tag{2.7}$$

[1] For simplicity, only the u-component is considered for the rest of this thesis.

It is shown in [13] that with the above integral equations, the repulsive forces and torques between a rigid object and obstacles can be evaluated analytically for linear and quadratic charge distributions along their boundaries, as reviewed next. Thus, for the application of the proposed potential model in achieving collision avoidance in path planning, the optimal object configurations along a path can be found more efficiently.

2.1.3 Repulsion due to linear and quadratic charge distributions

Assume the charge density $\rho(u)$ is equal to 1, u, or u^2 for an obstacle line, and $\rho(s) = 1$, s, or s^2 for an object line. Nine different combinations of charge distributions will need to be considered in evaluating the repulsion between the two line segments. For example, from (2.5), the repulsive force along the u-axis for these nine combinations can be obtained from

$$\begin{aligned} F_u^{ij} =& F_u^{ij}(u_2) - F_u^{ij}(u_1) \\ =& (1+a^2)^{\frac{j+1}{2}} \int_{u_1}^{u_2} \int_0^d \frac{u' - u}{r^3} u^i du (u')^j du' \end{aligned} \tag{2.8}$$

where i is equal to the order of the charge density of the obstacle line, and j is equal to that of the object line. It is shown in [13] that analytic expressions exist for all these integral equations. For example, we have

$$F_u^{00}(u') = \log \frac{f_1'(u')/2 + \sqrt{1+a^2} f_1^{1/2}(u')}{f_2'(u')/2 + \sqrt{1+a^2} f_1^{1/2}(u')} \tag{2.9}$$

where $f_1(u') = (au' + b)^2 + (u - d)^2$ and $f_2(u') = (au' + b)^2 + u^2$.

In general, $\rho(u)$ and $\rho(s)$ can be any quadratic functions, or

$$\rho(u) = \alpha_1 u^2 + \beta_1 u + \gamma_1 \tag{2.10}$$

$$\rho(s) = \alpha_2 s^2 + \beta_2 s + \gamma_2 \tag{2.11}$$

where coefficients α_1, β_1, γ_1, α_2, β_2, and γ_2 are some real numbers. The repulsive force along the u-axis can still be evaluated analytically as

$$\begin{aligned} F_u \quad = \quad & \alpha_1\alpha_2 F_u^{22} + \alpha_1\beta_2 F_u^{21} + \alpha_1\gamma_2 F_u^{20} \\ & + \beta_1\alpha_2 F_u^{12} + \beta_1\beta_2 F_u^{11} + \beta_1\gamma_2 F_u^{10} \\ & + \gamma_1\alpha_2 F_u^{02} + \gamma_1\beta_2 F_u^{01} + \gamma_1\gamma_2 F_u^{00} \end{aligned} \tag{2.12}$$

Similar results can be obtained for F_v and τ_P. Thus, for any $\rho(u)$ and $\rho(s)$, we can first evaluate the coefficients of the charge density functions and then use the nine sets of expressions of the repulsion, as in (2.12), to evaluate the total repulsion.

2.2 A Review of the Generalized 3-D Potential Model

In planning a path of a robot, a repulsive potential function is usually used to keep a safe distance between the robot and obstacles. In general, a potential function used to model the workspace can be a scalar function of the distances between the boundary points of the robot and those of obstacles. The gradient of such a scalar function can be used as a repulsive force between the robot and obstacles, making potential-based methods simple. Ideally, as mentioned in [12], a desirable potential function should have the following attributes.

- The magnitude of potential should be unbounded near the obstacle boundaries and should decrease with range. This property captures the basic requirement of collision avoidance.
- The potential should have a spherical symmetry far away from the obstacle.
- The equipotential surface near an obstacle should have a shape similar to that of the obstacle surface.
- The potential, its gradient, and their effects on path must be spatially continuous.

In [1], it is shown that the Newtonian potential, being harmonic in the 3-D space, can not prevent a charged point object from running into another object whose surface is uniformly charged. This is because the value of such a potential function is finite at the continuously charged surface. Subsequently, generalized potential models are proposed to assure collision avoidance between 3-D objects. The generalized potential function is inversely proportional to the distance between two point charges to the power of an integer and, as reviewed next, the potential and thus its gradient due to polyhedral surfaces can be calculated analytically. The path planner proposed in this thesis will use these results to evaluate the repulsion between manipulators and obstacles.

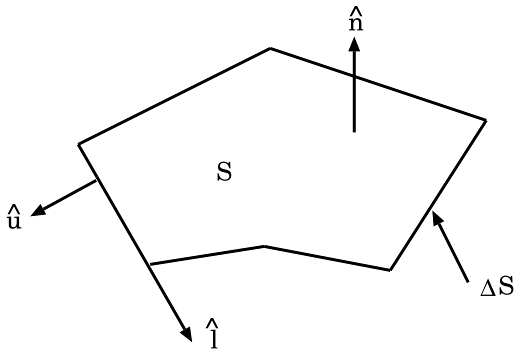


Figure 2.2: A polygonal surface S in the 3-D space.

2.2.1 Integral Equation of 3-D Potential Modeling

Consider a planar surface S in the 3-D space as shown in Figure 2.2. The direction of its boundary, ΔS, is determined with respect to its surface normal, $\hat{\mathbf{n}}$, by the right-hand rule, $\hat{\mathbf{u}}\times\hat{\mathbf{l}} = \hat{\mathbf{n}}$, where $\hat{\mathbf{u}}$ and $\hat{\mathbf{l}}$ are along the (outward) normal and tangential directions of ΔS, respectively. For the generalized potential function, the potential value at point $\mathbf{r}$ is defined as

$$\int_S \frac{dS}{R^m}, \qquad m \geq 2 \tag{2.13}$$

where $R = |\mathbf{r}' - \mathbf{r}|$ for point $\mathbf{r}'$, $\mathbf{r}' \in S$, and integer m is the *order* of the potential function. The basic procedure to evaluate the potential at $\mathbf{r}$ is similar to that outlined in [51] for the evaluation of the Newtonian potential (m=1) and can be summarized as follows:

(i) Write the integrand of the potential integral over S as surface divergence of some vector function.

(ii) Transform the integral into the one over ΔS based on the surface divergence theorem.

(iii) Evaluate the integral as the sum of line integrals over edges of ΔS.

Related geometric quantities associated with an edge C_i of S in the plane containing S, Q, are shown in Figure 2.3 for $\mathbf{r}' \in C_i$. Without loss of generality, it is assumed that

$$d \stackrel{\Delta}{=} \hat{\mathbf{n}}\cdot(\mathbf{r} - \mathbf{r}') > 0 \tag{2.14}$$

which is equal to the distance from $\mathbf{r}$ to Q.

For (i), we have (see [1])

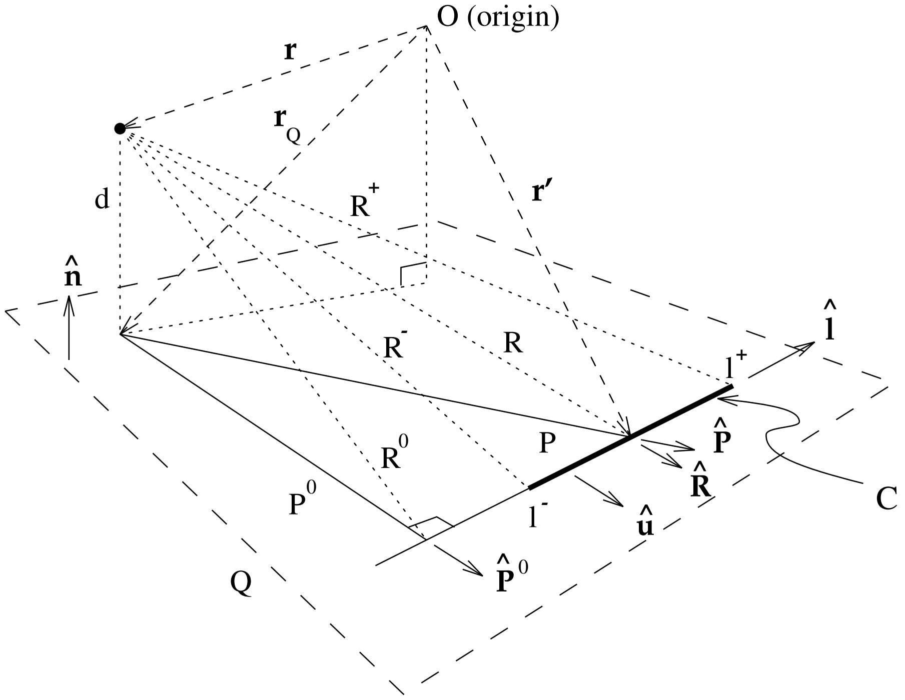


Figure 2.3: Geometric quantities associated with a point, an edge C_i (subscript i is omitted) of S shown in Figure 2.2 and the plane Q containing S.

$$\frac{1}{R^m} = \nabla_S\cdot(f_m(R)\mathbf{P}) \tag{2.15}$$

where $\mathbf{P}$ is the position vector of $\mathbf{r}'$ with respect to the projection of $\mathbf{r}$ on Q, $\mathbf{r}_Q$, and

$$f_m(R)=\begin{cases} \dfrac{\log R}{R^2-d^2}, & m=2 \\ \dfrac{-1}{(m-2)R^{m-2}(R^2-d^2)}. & m\neq 2 \end{cases} \tag{2.16}$$

Note that if $\mathbf{r}_Q$ is inside S, $f_m(R)$ will become singular for some $\mathbf{r}'' = \mathbf{r}_Q$, i.e., $R = d$. Let S_ε denote the intersection of S and a small circular region on Q of radius ε and centered at $\mathbf{r}_Q$, the potential due to S can be evaluated as

$$\begin{aligned} \int_S \frac{dS}{R^m} &= \lim_{\varepsilon\to 0}\left[\int_{S-S_\varepsilon} \nabla_s\cdot(f_m(R)\mathbf{P})dS + \int_{S_\varepsilon}\frac{dS}{R^m}\right] \\ &= \int_{\Delta S} f_m(l)\mathbf{P}\cdot\hat{\mathbf{u}}dl + \lim_{\varepsilon\to 0}\left[\int_0^\alpha\int_0^\varepsilon \frac{pdpd\theta}{(p^2+d^2)^{\frac{m}{2}}}\right] \\ &= \sum_i \mathbf{P_i}^0\cdot\hat{\mathbf{u}}_\mathbf{i}\int_{C_i} f_{m,i}(l_i)dl_i + g_m(\alpha), \end{aligned} \tag{2.17}$$

where

$$f_{m,i}(l_i) = f_m(R = \sqrt{{l_i}^2 + d^2 + ({P_i}^0)^2}), \tag{2.18}$$

$$g_m(\alpha) = \begin{cases} \alpha \log d, & m = 2 \\ \dfrac{\alpha}{(m-2)d^{m-2}}, & m > 2 \end{cases} \tag{2.19}$$

${P_i}^0$ is the distance between $\mathbf{r}_Q$ and C_i, l_i is measured from the projection of $\mathbf{r}$ on C_i along the direction of $\hat{\mathbf{l}}_i$, and α is the angular extent of the circumference of S_ε lying inside S as $\varepsilon \to 0$. For example, $\alpha = 2\pi$ if $\mathbf{r}_Q$ is inside S, $\alpha = \pi$ if $\mathbf{r}_Q$ is on an edge of S and α is equal to the angle between two edges if $\mathbf{r}_Q$ is a vertex of S where the two edges are connected.

Since $f_{m,i}(l_i)$ is a rational function for even m's when $m \neq 2$ and is rationalizable for odd m's (see [52]), the line integrals can always be evaluated in closed form except for $m = 2$. For example, if ${P_i}^0 \neq 0$, we have [2]

$$\int_{C_i} f_{3,i}(l_i)dl = \int_{{l_i}^-}^{{l_i}^+} f_{3,i}(l_i)dl = \frac{1}{{P_i}^0 d}\left[\tan^{-1}\frac{{l_i}^- d}{{P_i}^0 R^-} - \tan^{-1}\frac{{l_i}^+ d}{{P_i}^0 R^+}\right]. \tag{2.20}$$

with R^- and R^+ equal to the distances from $\mathbf{r}$ to the two end points of C_i, respectively. Thus, the repulsive force exerted on a point charge at (x, y, z) due to polygon j, denoted as

$$\Phi(x, y, z) = \sum_i [\phi(x_2^i, y_2^i, z) - \phi(x_1^i, y_1^i, z)] + \frac{\alpha}{z}, \tag{2.21}$$

can be obtained analytically by evaluating the gradient of the potential function for $m = 3$ with

$$\phi(x, y, z) = \frac{1}{z}\tan^{-1}\frac{xz}{y\sqrt{x^2 + y^2 + z^2}} \tag{2.22}$$

where the $x_i - y_i - z$ coordinate system is determined by the right-hand rule for each edge i of the polygon such that z is measured along the normal direction and x_i is measured along edge i of the polygon, respectively.

2.2.2 Integral Equation of Repulsion

Therefore, the repulsive force at point (x, y, z) due to polygon j can be denoted as

$$f^j = (f_x^j, f_y^j, f_z^j) = (\frac{\partial \Phi^j}{\partial x}, \frac{\partial \Phi^j}{\partial y}, \frac{\partial \Phi^j}{\partial z}) \tag{2.23}$$

For the potential-based path planning of manipulators, the evaluation of the repulsion between manipulators and obstacles involves the calculation of the repulsion between pairs

[2] Detailed discussions concerning some special cases for the potential calculation, e.g., for ${P_i}^0 = 0$ or $d = 0$, etc., as well as the associated collision avoidance property can be found in [1]. In this thesis, only $m = 3$ is considered.

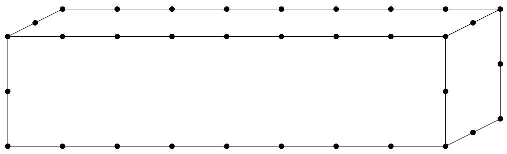


Figure 2.4: The sampling model for a link of a manipulator.

of polygons; each pair has a polygon from the manipulator and the other from obstacles. To simplify the mathematics, links of manipulator are approximately represented by a set of point samples on their surfaces in this thesis. Usually, as shown in Fig. 2.4, the sampling points are located on the vertexes and edges of links and their distribution should be as uniform as possible.

Assume that a link has m point samples and the obstacles have n polygon surfaces. If the repulsive force between point sample p_k and surface S_j is denoted as f_k^j, the repulsive force exerted on p is equal to

$$f_k = (\sum_j^n f_{k_x}^j, \sum_j^n f_{k_y}^j, \sum_j^n f_{k_z}^j) \tag{2.24}$$

according to (11), and the total repulsive force exerted on the link can be evaluated as

$$f = \sum_k^m f_k. \tag{2.25}$$

The repulsion expressed by (2.25) will be used in the potential-based path planning for robot manipulators in 3-D workspace, as discussed next. The total repulsive torque exerted on a link with respect to a point, say a joint, can be derived similarly, which is omitted for brevity.

Chapter 3

Potential-based Path Planning in 2-D Environments

3.1 Overview

Path planning of a manipulator is to determine a collision-free trajectory from its original location and orientation (called starting configuration) to the goal configuration [53]. Some planners adopt the configuration space (c-space) base approach [3] [4] [5] [6] [7] [15]. A point in a c-space indicates a configuration of manipulator which is usually encoded by a set of manipulator's parameters; e.g., joint angles between manipulator links. Then, path planning of a robot is reduced to the problem of planning a path of a point in free space of the c-space. The performance of c-space approaches is usually restricted by the exponential growth of computation with robot degrees of freedom. Unlike c-space based approaches, geometric algorithms directly use spatial occupancy information of the workspace (w-space) to solve path planning problem [8] [9] [1] [10] [11] [12] [14] [13] [2]. Workspace-based algorithms usually extract relevant information about the free space and use them together with the manipulator geometry to find a path. However, not all approaches try to find path with minimum risk of collision. To minimize such a risk, repulsive potential fields between charged manipulator and obstacles are used in [9] [11] [12] and [13] to match their shapes in the path planning.

In this thesis, the potential field model presented in [13], as reviewed in the next section, is adopted to model the workspace for the path planning of robot manipulators. A collision-free path of a manipulator will then be obtained by locally adjusting the manipulator configuration to search for minimum potential configurations using these force and torque. The proposed approach uses one or more guide lines (GLs) among obstacles in the free space as final or intermediate goals in the workspace for a manipulator to reach. These GLs provide the manipulator a general direction to move forward and also help to establish certain motion constraints for adjusting manipulator configuration during path planning, as discussed in Section 3.2. In Section 3.3, the implementation details are given. In Section 3.4, simulation results are presented for path planning performed on SUN workstation for manipulators in 2-D environment. Section 3.5 gives conclusions of this chapter.

3.2 A Potential-Based Path Planning algorithm in 2-D Environments

The application of the potential model reviewed in the previous section for path planning of manipulators will be discussed in this section. For a rough description of object path, the proposed approach uses one or more guide lines (GLs) as final or intermediate goals in the workspace. The GLs are line segments among obstacles in the free space, providing the manipulator a general direction to move forward (see Fig. 3.1). A collision-free traversal of a given sequence of GLs by the end-effector is regarded as a global solution of the path planning problem of a manipulator. Although free space bottlenecks which are defined by minimal distance links (MDLs) among the obstacles are good candidates as GLs for path planning[1], other links among the obstacles can also be used, e.g., GL_3 shown in Fig. 3.1.

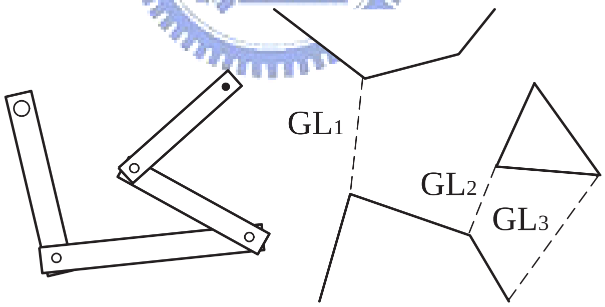


Figure 3.1: A manipulator is moved toward the goal (not shown) by sequentially traversing a sequence of GLs.

3.2.1 Basic procedure of path planning

The proposed path planning approach derives a series of minimum potential configurations along the path of a manipulator by locally adjusting its configuration for minimum potential

[1] For example, MDLs are used to connect (convex) obstacle nodes in the obstacle neighborhood graph in [17].

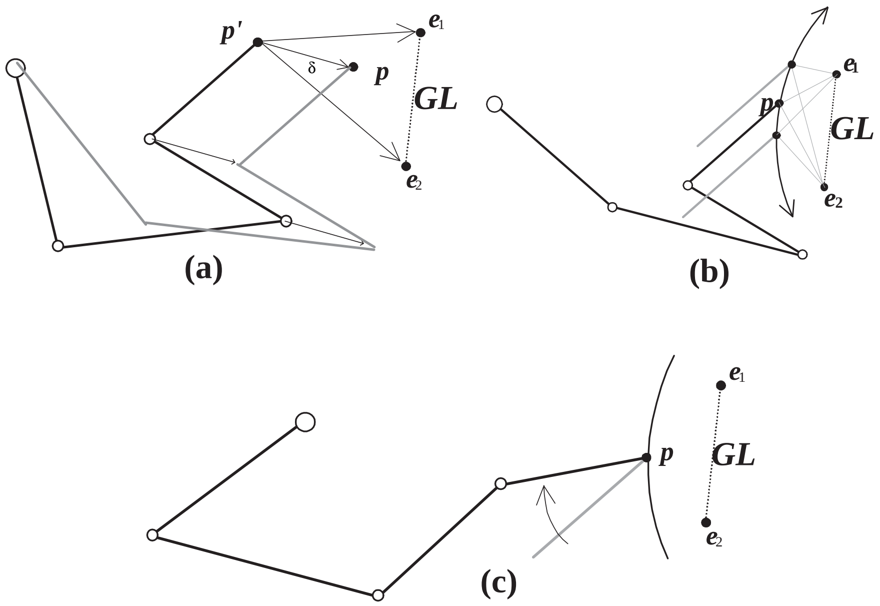


Figure 3.2: Basic planning procedure for a given GL.

using the results given in Sec. 2. Assuming that a guide line $\overline{e_1e_2}$ is given as an intermediate goal, the basic path planning procedure for moving the end-effector p of a manipulator onto $\overline{e_1e_2}$ includes (see Fig. 3.2):

(i) Translate the distal links of the manipulator to move p toward the $\overline{e_1e_2}$. (Fig. 3.2(a))

(ii) For constant $|pe_1| + |pe_2|$, repeatedly execute :

  (a) Search for the minimum potential configuration of the manipulator with the distal link fixed in orientation. (Fig. 3.2(b))

  (b) Search for the minimum potential configuration of the manipulator with p fixed in position. (Fig. 3.2(c))

(iii) Repeat (i) and (ii) until the end-effector reaches the line segment $\overline{e_1e_2}$.

In general, there are different ways to change the manipulator configuration to move p toward $\overline{e_1e_2}$. A simple translation of distal links is adopted in (i). As shown in Fig. 3.2(a), the translation of a predetermined distance δ of the distal link is carried out to move the end-effector from p' to p. The direction of the translation is determined with respect to e_1 and e_2 such that $\overrightarrow{p'p}$ bisects $\angle e_1p'e_2$. If there is a collision, each time the distance of the translation is reduced by 50% until it is collision-free. No configuration improvement using the repulsion experienced by the manipulator is considered at this stage.

In (ii), the associated constrained optimization problem is divided into two iterative univariant optimization procedures, as in (ii-a) and (ii-b). In (ii-a), the distal link is fixed in its orientation (see Fig. 3.2(b)) as p slides along the elliptic trajectory to search for the minimal potential configuration and other distal links are sequentially adjusted in orientation, staring from the link connected to the distal link. In (ii-b), the distal link is adjusted in orientation while fixed in position (see Fig. 3.2(c)) and the procedure for adjusting the rest links is similar to that in (ii-a). Detailed implementation is presented in next section.

For each elliptical trajectory, (ii-a) and (ii-b) are repeatedly performed until negligible changes in the manipulator configuration are obtained. Then another smaller elliptical trajectory is obtained with (i) and the process repeats. The path planning algorithm, as summarized below, ends as the end-effector reaches the given GL.

Algorithm End_Effector_to_GL

Step 0 Initialize $\delta = \delta_0$, where δ_0 is arbitrarily chosen.

Step 1 Translate the manipulator with distance δ along the direction of the bisector vector of $\overrightarrow{p'e_1}$ and $\overrightarrow{p'e_2}$. If collision occurs or if the two base links become unconnected, $\delta \longleftarrow \delta/2$ and go to Step 1.

Step 2 Translate the distal link to move p along the elliptical trajectory, $|pe_1| + |pe_2| = constant$, to minimize the potential.

Step 3 Joint angle adjustment for the minimum potential configuration with p fixed in position.

Step 4 Go to Step 2 if the translation in Step 2 or the joint angle adjustment in Step 3 is not negligible.

Step 5 If p reaches GL, the planning is completed. Otherwise, go to Step 1 with $\delta = \delta_0$.

For path planning involving multiple GLs, the above algorithm will be executed for each of them sequentially. It is assumed that the planning for a GL starts as the planning of the previous GL is accomplished. The path planning ends as the end-effector reach the goal, which is usually a (goal) GL in the path planning problems considered in this thesis.

3.3 Implementation details

3.3.1 Step 1 of *END_EFFECTOR_TO_GL*

Consider the manipulator shown in Fig. 3.2(a). Its end-effector is initially located at point p' and is moved toward an intermediate goal GL, such that it is moved with distance δ along the direction of the bisector vector of $\overrightarrow{p'e_1}$ and $\overrightarrow{p'e_2}$, from p' to p. Such a movement is achieved in the implementation by translating every link except the two base links. It is not hard to see that the two base links have at most two possible configurations; i.e., the two base links together may have two, one, or no feasible configurations depending on the amount of translations of the distal links. In the computer implement, if there are two feasible configurations for the two base links, the one requiring less adjustments in joint angles of the two links is selected.

3.3.2 Implementation of Step 2

In Step 2, since the minimization is constrained by $|pe_1|+|pe_2| = constant$, only the resultant force experienced by the distal link along the tangential direction of the ellipse is taken into account. Consider the forces exerted on the distal link lnk_n, as shown in more detail in Fig. 3.3. Let f_1 be the repulsive force exerted on lnk_n due to the repulsion between lnk_n and the obstacles, and f_2 be the force exerted on J_{n-1} due to the repulsive torque between other manipulator links and obstacles. For a univariant minimization approach, only one variable is adjusted at a time. To determine the minimum potential location of lnk_n under the elliptical constraint, all of the joint angles of the manipulator, except the base joint J_0, are assumed to be fixed. Therefore, the above repulsive torque, denoted as τ_0, is calculated for a single rigid composite link formed by all the other manipulator links with respect to J_0. Thus, we have

$$f_2 = \frac{\tau_0}{l_0} \tag{3.1}$$

where l_0 is the length of $\overline{J_0 J_{n-1}}$.

To determine the direction in which p should slide along the ellipse E_1, and thus lnk_n should translate, to reduce the repulsive potential, the direction of the projection of the resultant force exerted on lnk_n along the tangent of E_1 at p, L,

$$f_L = f_{1L} + f_{2L} \tag{3.2}$$

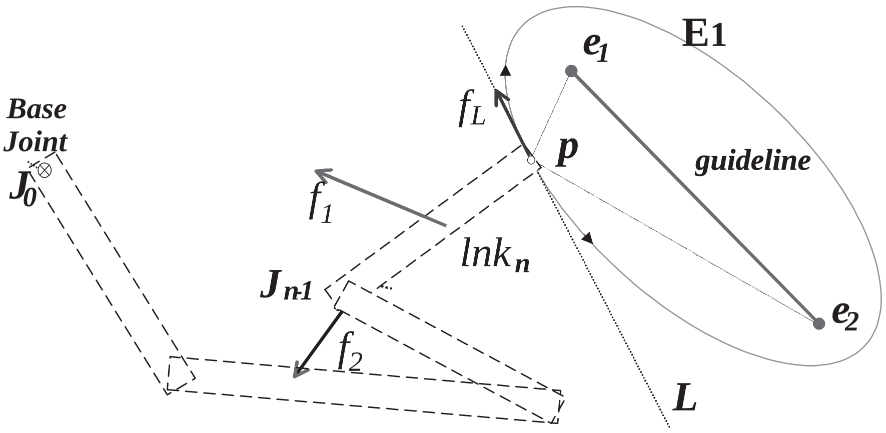


Figure 3.3: Sliding p along ellipse E_1, by translating lnk_n to reduce the repulsive potential.

is calculated. A gradient-based binary search for the minimum potential location of p along E_1 can thus be performed using (14). The search ends when the movement of lnk_n or the magnitude of f_L is negligible.

Each time the position p, and lnk_n, is changed in the above search process, the orientation of rest links, i.e., joint angles at J_0 through J_{n-1}, need to be adjusted for connectivity and for minimum potential of the manipulator. Such a procedure is very similar to the joint angle adjustment performed in Step 3, as discussed next.

3.3.3 Adjusting joint angles in Step 3

Once the minimum potential position of distal link is determined with Step 2, another univariant procedure, which allows the distal link to adjust its orientation but not its location, is performed to reduce the potential further, as shown in Fig. 3.2(c). Under the constraint that end-effector is fixed in location p, the distal link can rotate with respect to p to reduce the repulsive potential. The direction in which the distal link should rotate is determined by the repulsive torque experienced by the distal link with respect to p.

Consider the repulsion experienced by the distal link lnk_n with end-effector p fixed in position. Let τ_n be the repulsive torque experienced by lnk_n with respect to p due to the repulsion between lnk_n and let f_2, as described in the previous subsection, be the force exerted on J_{n-1} due to the repulsive torque between other manipulator links and obstacles. The resultant torque experienced by lnk_n with respect to p is equal to

$$\tau_n{}^* = \tau_n + f_{2\perp} \cdot l_n \tag{3.3}$$

where l_n is the length of lnk_n and $f_{2\perp}$ is the projection of f_2 along the direction perpendicular to lnk_n. A gradient-based binary search for the minimum potential orientation of lnk_n for p fixed in position can thus be performed using the direction of $\tau_n{}^*$. Each time the orientation of lnk_n is changed , the orientation of the rest links are adjusted iteratively for connectivity and for minimum potential using $\tau_{n-1}{}^*$, $\tau_{n-2}{}^*$, ..., etc. The search ends when the movement of lnk_n or the torque $\tau_n{}^*$ is negligible.

3.4 Simulation Results

In this section, simulation results are presented for path planning performed on SUN Ultra$-$1 for manipulators in 2-D environment. In order to show the path of a manipulator more clearly, every configuration of a path obtained by *End_Effector_to_GL* is shown in different gray level, i.e., the initial configuration is shown in black and final configuration is shown in white. Moreover, all manipulator and obstacle boundaries are assumed to be uniformly charged, except for those marked out explicitly for non-uniform charge distributions. The precisions for optimal configurations obtained in Steps 2 and 3 are equal to 0.3 units and 0.625^o, respectively, for link locations and orientations.

Fig. 3.4 shows the path planning result similar to an example shown in [54]. In our simulation, there are two GLs predetermined in the workspace of size 30 by 20 units. The initial distance of translation, δ_0, is arbitrarily chosen as 2.2 units. The simulation takes a total of 5.66 seconds to plan the 11-configuration collision-free path and spends most CPU times for the first and seventh configurations. The planned manipulator path is observed to achieve better results in terms of collision avoidance compared with the path obtained in [54]. Fig. 3.5 shows the path planning result similar to an example shown in [55]. In our simulation, there are four GLs predetermined in the workspace of size 30 by 30 units. The initial distance of translation, δ_0, is chosen as 4.0 units. While it takes five seconds for a SUN SPARC to generate a 4-configuration path in [55], the 8-configuration path is obtained by the proposed algorithm in 0.87 seconds for a SUN Ultra$-$1.

In general, the number of configuration and the computation time depend on the size of the initial distance of translation, δ_0. For larger δ_0, less configurations will be generated along the path if there is no collision. On the other hand, if δ_0 is too large, the computation time may increase because collisions may occur frequently. Fig. 3.6 shows the path planning result

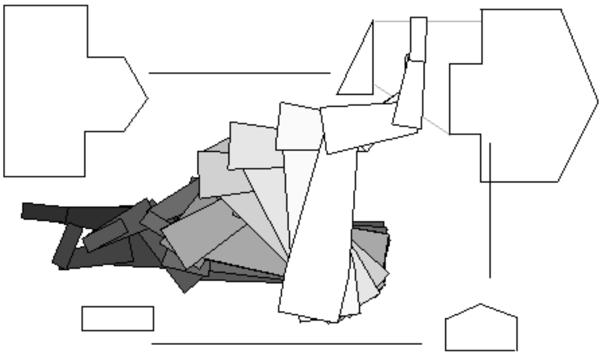


Figure 3.4: A path planning example derived from [54].

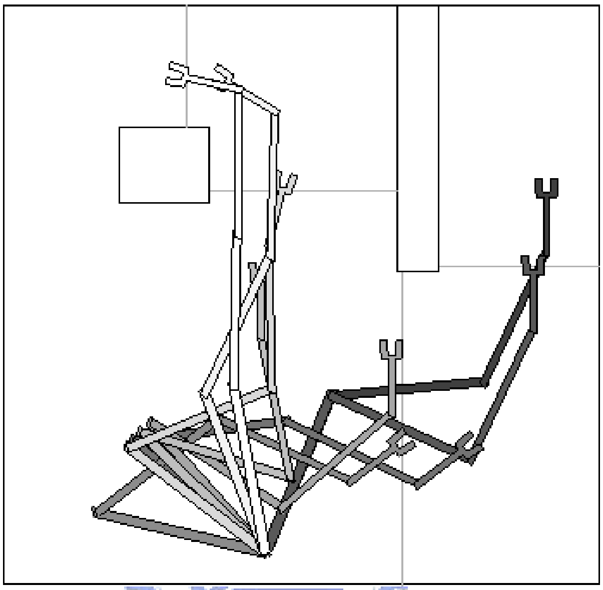


Figure 3.5: A path planning example derived from [55].

similar to an example shown in [7]. In our simulation, there are seven GLs predetermined in the workspace of size 30 by 20 units. The initial distance of translation, δ_0, is chosen as 2.2 units. The simulation takes a total of 5.5934 seconds to plan the 16-configuration collision-free path. Different values of δ_0, from 0.5 to 2.4 units, are also simulated, and the simulations take 6.26 to 16.48 seconds. For the PRM presented in [7] using a DEC Alpha, a computation time of 5 seconds corresponds to success rates of 50%. For a success rate of more than 97%, the total computation time will exceed 15 seconds.

The proposed algorithm is also performed well for high DOF manipulators. Fig. 3.7(a) shows a high DOF example in which a 9-link manipulator need to snake into the cave with a collision-free path. There are four GLs predetermined in the workspace of size 30 by 30 units. The initial distance of translation, δ_0, is chosen as 2.2 units. The computation time

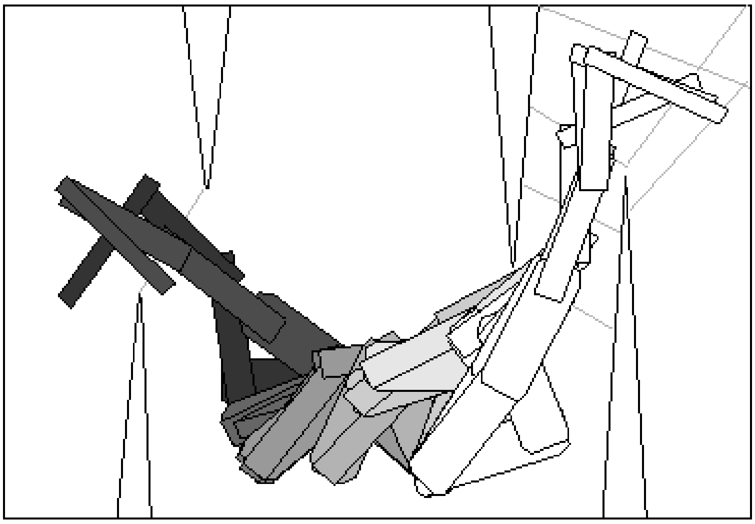
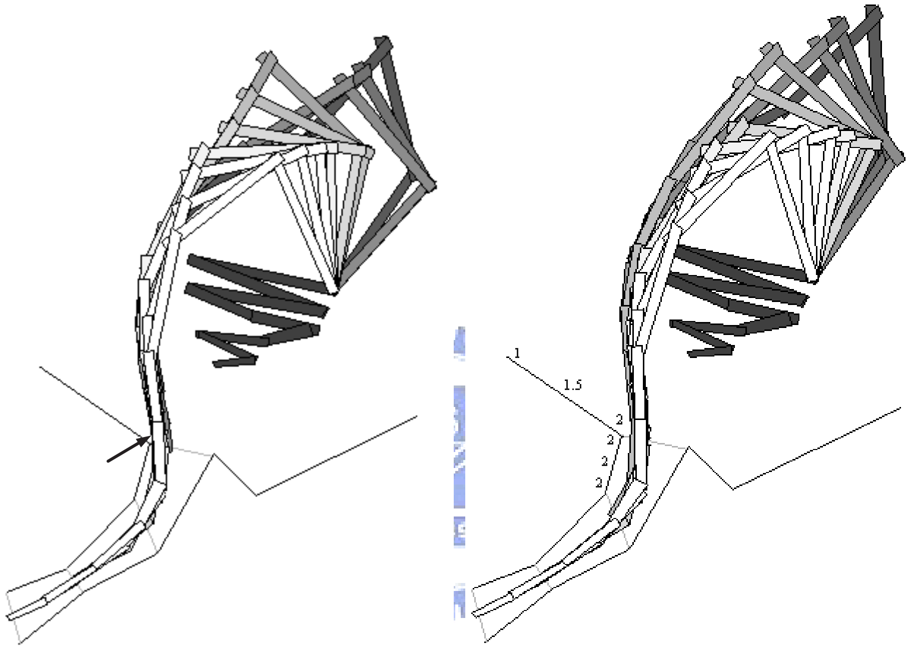


Figure 3.6: A various δ example derived from [5].

of the 11-configuration path takes a total of 9.26 seconds. In Fig. 3.7(a), the manipulator may moves very close to an obstacle's corner (as indicated with a pointer). In order to keep a safe distance away from the corner, the charge distributions of the left side of the cave is modified to be non-uniform and a safer manipulator path is obtained, as shown in Fig. 3.7(b). In Fig. 3.7(b), the charge density is doubled for the segment marked with three 2's and is increased linearly from 1 to 2 for another marked segment. The computation time is equal to 12.4 seconds for the path shown in Fig. 3.7(b).

3.5 Summary

In this thesis, a potential-based algorithm for the path planning of a robot manipulator is proposed. The proposed algorithm uses an artificial potential field to model the workspace wherein object boundaries are assumed to be charged with various source distributions. Simulation results show that a path thus derived is always spatially smooth while its safety level can be adjusted easily by selecting proper charge distributions for the potential-based workspace model. Since the proposed approach uses workspace information directly, it is readily applicable to manipulators of high DOFs. In our experiments, manipulators with DOFs up to 9 have been tested with satisfactory results.



(a) uniform charge

(b) non-uniform charge

Figure 3.7: A 9-link manipulator example.

Chapter 4

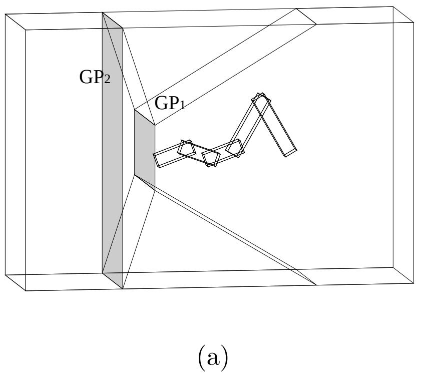
Potential-based Path Planning of Manipulator in 3-D Environments

4.1 Overview

In this chapter, the potential field model presented in [1] is adopted to model the workspace for the path planning of robot manipulators. The approach is similar to that done in 2-D manipulator mentioned in Chapter 3, except the DOFs of joints and adopted potential model. In a 3-D environment, the proposed approach uses one or more guide planes (GPs) instead of guide lines (GLs) among obstacles in the free space as final or intermediate goals in the workspace for the manipulator to reach. These GPs provide the manipulator a general direction to move forward and also help to establish certain motion constraints for adjusting manipulator configuration during path planning, as discussed in Section 4.2. The algorithm proposed in Section 4.2 has similar steps as the one in Section 3.2.1. The implementation details are given in Section 4.3. In Section 4.4, simulation results are presented for path planning performed for manipulators in different 3-D environments. Section 4.5 gives a conclusion of this work.

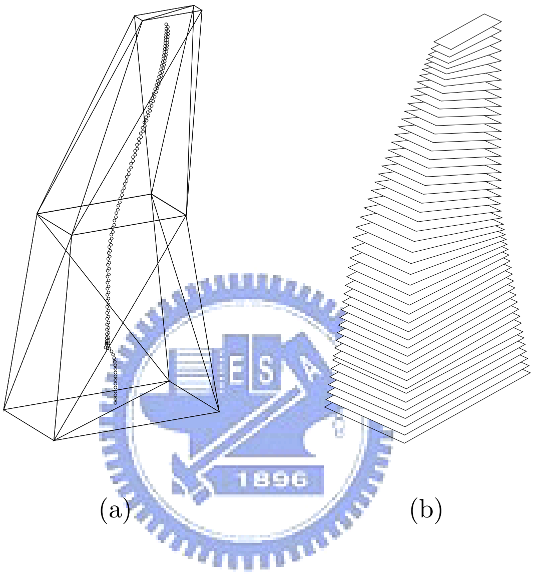
4.2 The Proposed Path Planning Algorithm

The application of the potential model reviewed in the previous section for path planning of manipulators will be discussed in this section. Unlike some c-space based approaches, which often require expensive preprocessing to construct the c-space, the proposed approach uses the workspace information directly. The approach computes repulsive force and torque experienced by each rigid component, e.g., a link, of a manipulator. A collision-free path of the manipulator can then be obtained by locally adjusting its configuration along the path



(a) (b)

Figure 4.1: A manipulator is moved toward the goal (not shown) by sequentially traversing a sequence of GPs.



(a) (b)

Figure 4.2: The generalized cylinder presentation of a passage.

for minimum potential using these force and torque. In this thesis, the spherical joint is adopted to connect links of a manipulator since its high DOFs can take full advantage of the proposed potential minimization algorithm.

For a rough description of manipulator path, the proposed approach uses one or more guide planes (GPs) as final or intermediate goals in the 3-D workspace. The GPs are polygons among obstacles in the free space, providing the manipulator a general direction to move forward (see Fig. 4.1). A collision-free traversal of a given sequence of GPs by the end-effector is regarded as a global solution of the path planning problem of a manipulator.

4.2.1 Generation and Selection of initial GPs

In the proposed algorithm, the GPs provide the articulated robot a general direction to move forward. The selection of the initial GPs may base on (i) their density along the passage, (ii) the visibility between two adjacent GPs, or (iii) the angular variation of two adjacent GPs. For the examples considered in this chapter, the initial GPs are selected arbitrarily and the algorithm seems to work reasonably well in terms of the sensitivity of the planned path to the selection of initial GPs. Often, these initial GPs can also be obtained from the *Generalized Cylinder* (GC) [56] representation as cross-sections perpendicular to the GC axis. Figs. 4.2(a) shows a passage which has approximately rectangular cross-sections, and an axis of its GC representation. In general, there is no limit on the number of cross-sections and their shapes are not explicitly specified in advance. More details about generalized cylinder representation can be found in Appendix A.

In the proposed approach, the shape of the GPs is usually not critical as long as the GPs are confined in the free space. In face, only the normal direction of a GP, e.g., the direction of the GC axis, is important. Such an axial representation can also be derived from (i) the global navigation function as in [57][5][58], as well as (ii) the tree structure representation of free space obtained from the wavefront expansion presented in [59]. While the gradient of (i) leads a point object to the goal, global connectively of free space is available in (ii) and is used in [59] to solve a planning problem in low dimensions.

4.2.2 Basic Procedure of Path Planning

In this chapter, the proposed path planning approach derives a series of minimum potential configurations along the path of a manipulator by locally adjusting its configuration for minimum potential using the results given in Sec. 2.2. Assuming that a guide plane GP_1 is given as an intermediate goal, the basic path planning procedure for moving the end-effector p' of a manipulator onto GP_1 include (see Fig. 4.3):

(i) Translate the distal links[1] of the manipulator to move its end-effector p' toward the GP_1. If p' can not reach GP_1 directly, e.g., due to collision, a virtual intermediate plane GP_1'

[1] In step (i), an intermediate simple solution of the inverse kinematics problem is obtained by translating of all manipulator links except for the two base links, the base link and the link connected to it. For each translation, the two base links, together, can have at most three DOFs. In step (ii), the problem is solved by finding a sequence of sub-optimal solutions with monotonically decreasing potential. Finally, the minimal potential solution is found in (iii).

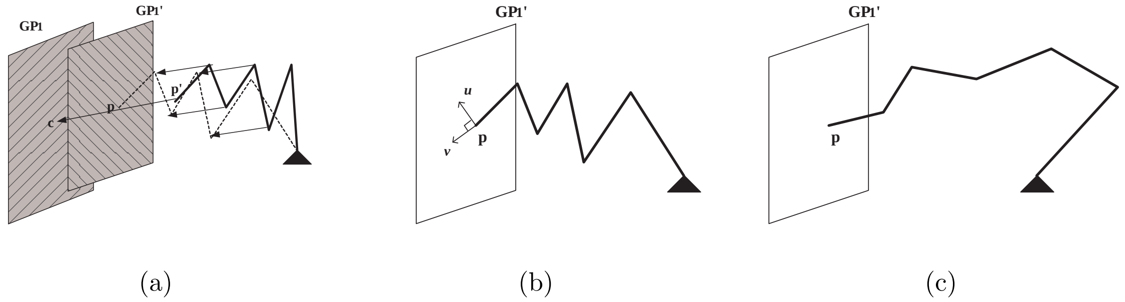


Figure 4.3: Basic path planning procedure for a given GP (see text).

is inserted. (Fig. 4.3(a))

(ii) Search for the minimum potential configuration of the manipulator for $p \in GP'_1$ by repeatedly executing :

(a) Search for the minimum potential configuration of the manipulator with the distal link fixed in orientation. The minimization is performed by sliding p along two orthogonal directions on GP'_1, e.g., $\overrightarrow{u}$ and $\overrightarrow{v}$. (Fig. 4.3(b))

(b) Search for the minimum potential configuration of the manipulator with the end-effector p fixed in position. The minimization is performed by changing the orientation of the distal link with p as its rotation center. (Fig. 4.3(c))

(iii) Repeat (i) and (ii) until p reaches GP_1.

In general, there are different ways to change the manipulator configuration to move p toward GP_1. A simple translation of distal links is adopted in (i) as a preliminary implementation of our algorithm. As shown in Fig. 4.3(a), the translation of the distal links is carried out to move the end-effector from p' to c, where c is the centroid of GP_1. If collision occurs or if the connectivity of manipulator can not be maintained, the distance of the translation is reduced until the translation is collision-free while the manipulator remain connected. Accordingly, a new GP is inserted, e.g., GP'_1. No configuration improvement to reduce the repulsive potential is considered at this stage.

As for the search for the minimum potential configuration of the manipulator in (ii), links of the manipulator are adjusted from the distal link to the base link using the repulsion experienced by the manipulator. The distal link has five DOFs, i.e., two for its location for $p \in GP'_1$ and three for its orientation. While each of other distal links has three DOFs for its orientation, the two base links, together, have at most three DOFs with their connecting joint being constrained to lie on a circle.

In (ii), the associated constrained optimization problem is divided into two iterative univariant optimization procedures, as in (ii-a) and (ii-b). In (ii-a), the distal link is fixed in its orientation (see Fig. 4.3(b)) as p slides on GP_1' to search for the minimal potential configuration and other distal links are sequentially adjusted in orientation, staring from the link connected to the distal link. In (ii-b), the distal link is adjusted in orientation while fixed in position (see Fig. 4.3(c)) and the procedure for adjusting the rest links is similar to that in (ii-a). For a particular GP, say GP_1' in Fig. 4.3, (ii-a) and (ii-b) are repeatedly performed until negligible changes in the manipulator configuration is obtained. Then, another intermediate GP which is closer to GP_1 is obtained with (i) and the process repeats. The path planning algorithm, as summarized below, ends as the end-effector reaches the given GP_1 or exist abnormally for an infeasible problem. Detailed implementation of the algorithm is presented in the next section.

Algorithm End_Effector_to_GP

Step 0 Initialize $\delta = \delta_0 = p'c$ and $GP \leftarrow GP_1$.

Step 1 Translate the distal links of the manipulator with distance δ to move p' to p along the direction of $\overrightarrow{p'c}$. Find the smallest $n \geq 0$ such that $\delta \longleftarrow \delta/2^n$ corresponds to a feasible and collision-free translation.

Step 2 Translate the distal link by sliding p on GP to minimize the potential.

Step 3 Adjust joint angle of the manipulator for the minimum potential configuration with p fixed in position.

Step 4 Go to Step 2 if the translation in Step 2 or the joint angle adjustment in Step 3 is not negligible.

Step 5 If p reaches GP_1, the planning is completed. Otherwise, $p' \leftarrow p$ and go to Step 1 with $\delta = \delta_0$.

For path planning involving multiple GPs, the above algorithm will be executed for each of them sequentially. It is assumed that the planning for a GP starts as the planning of the previous GP is accomplished. The path planning ends as the end-effector reach the goal, which is usually a (goal) GP in the path planning problems considered in this chapter.

4.3 Implementation Details

The implementation details of the algorithm steps are identical to the ones mentioned in 3.3, except additional DOFs. Due to the additional DOFs of robots and environments, the location of end-effector is constrained on a plane not an arc during minimization procedure.

4.3.1 Step 1 of Algorithm *End_effector_to_GP*

In Step 1 of *End_effector_to_GP*, the translation of distal links is achieved in the implementation by translating every link except the two base links. The Step is similar to that described in Sec. 3.3.1for the 2D case, but the adjustment of the two base links is different due to additional DOFs. The configuration of the two base links is determined simply by considering a planar motion on the plane formed by J_0 (the base joint), the old position of J_1 and J_2 such that an additional rotation will move J_2 to its new location. On the other hand, there are different ways to move p' toward GP_1. For example, p' can be moved toward GP_1 in the direction perpendicular to GP_1. In our simulation, for simplicity, the end-effector is translated from p' to p by distance $\delta = max(p'c/2^n), n \geq 0$, along the direction of $\overrightarrow{p'c}$. If GP_1 is directly reachable without any collision and disconnection of the manipulator, p and c are identical. A threshold, i.e., $\delta_{min} = 1\%$ of workspace size, is established to set a lower bound of the magnitude of allowed translation. A translation which requires a smaller movement, $\delta < \delta_{min}$, indicates an infeasible path planning problem.

While no configuration improvement to reduce the repulsive potential is considered in the above translation procedure, Steps 2 and 3 minimize the potential through constrained optimization procedures. To that end, an intermediate GP, GP_1', is added along the path to serve as a geometric constraint for successive adjustments of manipulator configuration. Again, as long as $p \in GP_1'$, the choice of the orientation of GP_1' is not unique. For simplicity, $GP_1' \parallel GP_1$ is adopted in our simulation.

4.3.2 Adjusting end-effector position in Step 2

After the end-effector reaches p, as shown in Fig. 4.3(b), one of the univariant procedures, which allows the distal link to adjust its location, but not its orientation, in minimizing the repulsive potential, is performed in Step 2 of *End_effector_to_GP*. This step is similar to that described in Sec. 3.3.2 except for the constraints for potential minimization: while the GP is used in the former, the ellipse is used in latter.

Consider the forces exerted on the distal link lnk_n, as shown in more detail in Fig. 4.4. Let f_1 be the repulsive force exerted on lnk_n due to the repulsion between lnk_n and the obstacles, and f_2 be the force exerted on J_{n-1} due to the repulsive torque, denoted as τ_0, between other manipulator links (lnk_1 through lnk_{n-1}) and obstacles. For a univariant minimization approach, only one variable is adjusted at a time. To determine the minimum potential location of lnk_n as p sliding on GP, all of the joint angles of the manipulator, except the base joint J_0, are assumed to be fixed. Therefore, τ_0 can be calculated by considering a single rigid composite link formed by lnk_1 through lnk_{n-1} with respect to J_0. Thus, we have

$$f_2 = \frac{\tau_0}{l_0} \tag{4.1}$$

where l_0 is the length of $\overline{J_0 J_{n-1}}$.

To determine the direction in which p should slide on GP, and thus lnk_n should translate, to reduce the repulsive potential, the projection of the resultant force exerted on lnk_n along an arbitrary $\overrightarrow{u} \parallel GP$

$$f_u = f_{1u} + f_{2u} \tag{4.2}$$

is calculated, where subscript u denotes the projection along $\overrightarrow{u}$[2]. A gradient-based binary search for the minimum potential location of p along $\overrightarrow{u}$ can then be performed using (4.2). The initial step of sliding is arbitrarily chosen as 10% of the workspace size. If the movement of lnk_n along $\overrightarrow{u}$ or f_u is negligible, e.g., the movement is smaller than 1% of workspace size, another minimizing scheme along $\overrightarrow{v} \in GP$, which is orthogonal to $\overrightarrow{u}$, is performed to minimize f_v. Step 2 ends when two consecutive binary searches along the two orthogonal directions result in negligible movement of lnk_n.

In the above search processes, each time the position of p is changed, the orientation of rest links, i.e., joint angles at J_0 through J_{n-1}, need to be adjusted for connectivity and for minimum potential of the manipulator. Such a procedure is very similar to the joint angle adjustment performed in Step 3 of *End_effector_to_GP*, as discussed next.

4.3.3 Adjusting joint angle in Step 3

Once the minimum potential position of lnk_n is determined with Step 2 of *End_effector_to_GP*, another univariant procedure, which allows the distal link to adjust its orientation with

[2] To move $link_n$ such that the potential will have maximum rate of reduction, $\overrightarrow{u}$ can be chosen as the projection of $\overrightarrow{f_1} + \overrightarrow{f_2}$ on GP.

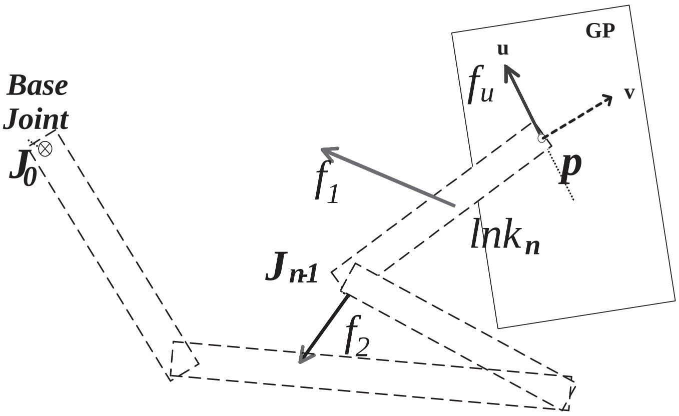


Figure 4.4: Sliding p on $GP_{1'}$, by translating lnk_n to reduce the repulsive potential.

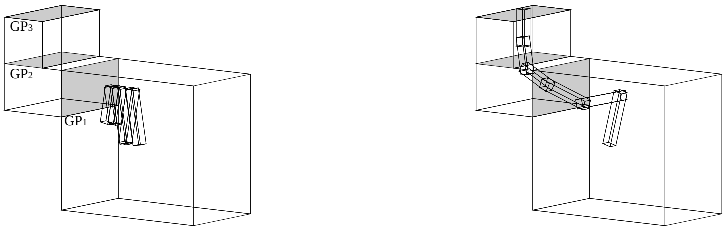
end-effector p fixed in position is performed to reduce the potential further, as shown in Fig. 4.3(c). Under such a constraint, the distal link can rotate with respect to p to reduce the repulsive potential. The direction in which the distal link should rotate is determined by the repulsive torque experienced by the distal link with respect to p. Let τ_n be the repulsive torque experienced by lnk_n with respect to p due to the repulsion between lnk_n and let f_2, as described in the previous subsection, be the force exerted on J_{n-1} due to the repulsive torque between other manipulator links and obstacles. The resultant torque experienced by lnk_n with respect to p is equal to

$$\tau_n{}^* = \tau_n + f_{2\perp} \cdot l_n \tag{4.3}$$

where l_n is the length of lnk_n and $f_{2\perp}$ is the projection of f_2 onto a plane perpendicular to lnk_n.

To find the minimum potential orientation of lnk_n for p fixed in position, gradient-based binary searches are performed using the projection of $\tau_n{}^*$ along three orthogonal axes of rotation, one axis at a time. For each binary search, the initial rotating angle is arbitrarily chosen as 5^o, while the accuracy in identifying the 1-D potential is chosen as 0.5^o. Each time the orientation of lnk_n is changed, the orientation of the rest links are adjusted recursively for connectivity and for minimum potential using $\tau_{n-1}{}^*$, $\tau_{n-2}{}^*$...., etc. Step 3 ends if a binary search using $\tau_n{}^*$ results in a negligible change in the orientation of lnk_n, i.e., 0.5^o.

As for the computation complexity, if every link needs k binary searches on the average to find a best orientation, the total number of binary searches needed for the derivation of



(a) (b)

Figure 4.5: A path planning example for a 6-link manipulator in a 3-GP workspace. (a) The initial configuration. (b) The final configuration.

the minimum potential configuration of an n-link manipulator is equal to

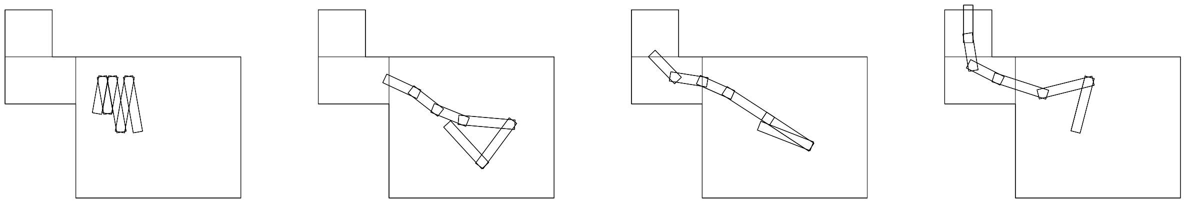
$$k^1 + k^2 + \cdots + k^{n-2} = (k^{n-1} - k)/(k - 1) \tag{4.4}$$

which appears to have a fairly large value. Fortunately, much lower computation complexity is often observed, as in the examples considered in next section.

4.4 Simulation Results

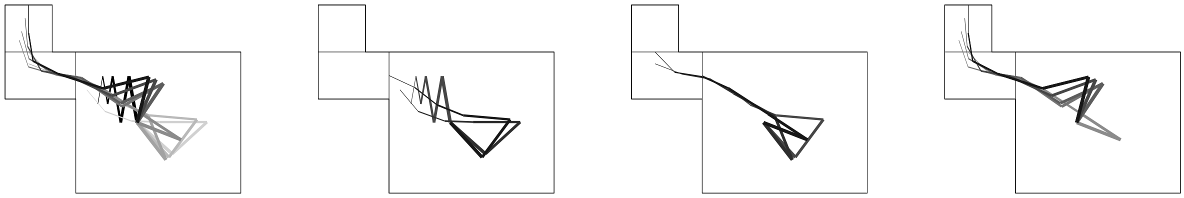
In this section, simulation results are presented for path planning performed on Pentium III (1GHz) for manipulators in 3-D environment. Figs. 4.5(a)(b) show the initial and final configurations, respectively, of the path obtained for a 6-link manipulator in a 3-GP workspace wherein the GPs are shown as gray polygons. While Fig. 4.5(a) shows the crooked initial manipulator configuration, Fig. 4.5(b) shows that the end-effector reaches the final GP safely. The simulation takes a total of 2.924 seconds to plan an 8-configuration collision-free path. Figs. 4.6(a)-(d) show side views of the initial configuration of the manipulator, as well as intermediate configurations as its end-effector reaches each of the three GPs. Links of the manipulator basically lies in the middle of the workspace due to the repulsive potential model. Fig. 4.7 shows complete as well as partial manipulator trajectories. In order to show the manipulator trajectory more clearly, configurations obtained by *End_Effector_to_GP* are shown in different gray levels, i.e., the initial configuration is shown in white while final configuration is shown in black. It is readily observable that these trajectories are safe and smooth.

Figs. 4.8(a)-(c) show the initial and final configurations of the path planning for another 6-link manipulator stretching into a tapered passage with 4 GPs. The passage makes up,



(a) (b) (c) (d)

Figure 4.6: Side views of the initial configuration of manipulator, as well as intermediate configurations as its end-effector reaches each of the three GPs in Fig. 4.5.



(a) (b) (c) (d)

Figure 4.7: (a) The complete manipulator trajectory. (b) The partial trajectory between Figs. 4.6(a) and (b). (c) The partial trajectory between Figs. 4.6(b) and (c). (d) The partial trajectory between Figs. 4.6(c) and (d).

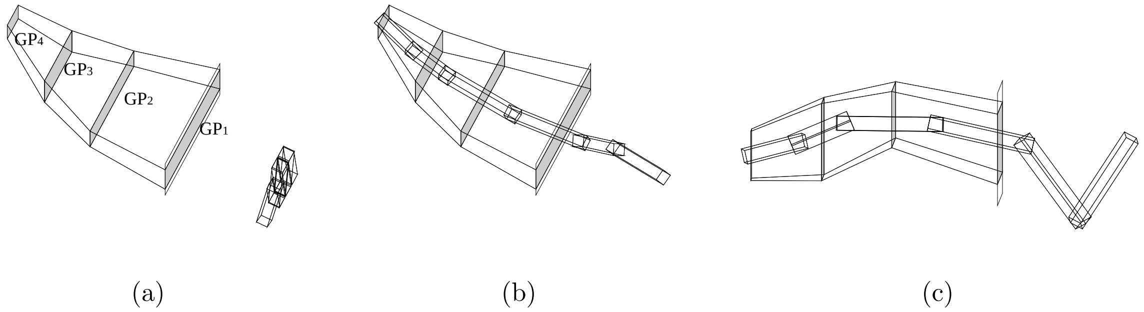
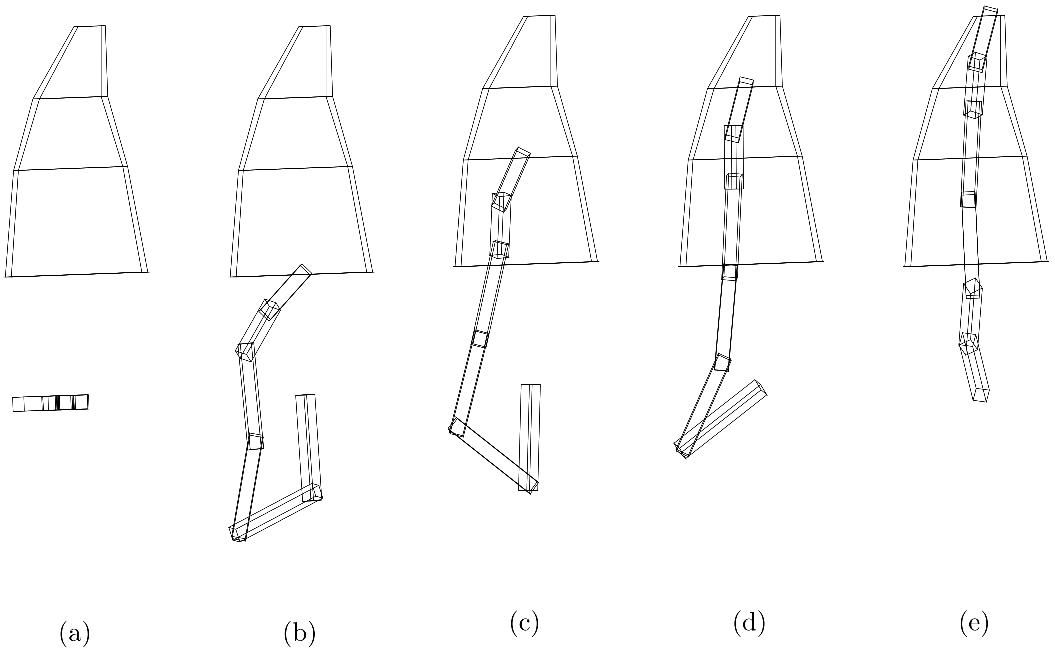


Figure 4.8: A path planning example for a 6-link manipulator in a 4-GP workspace. (a) The initial configuration. (b)-(c) Two different views of the final configuration.

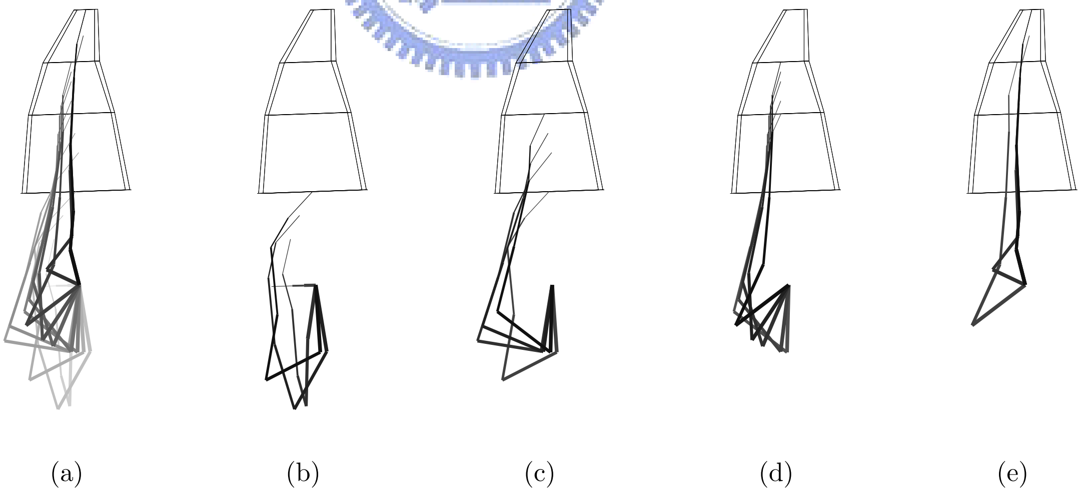
down and right turns at GP_1, GP_2 and GP_3, respectively. Fig. 4.8(a) shows the initial configuration of the manipulator which lies in a plane parallel to GP_1. Figs. 4.8(b)-(c) show top and side views, respectively, of the final configuration indicating that the end-effector of the manipulator reaches the final GP safely. Figs. 4.9(a)-(e) show the top views of the initial configuration of the manipulator, as well as intermediate configurations as its end-effector reaches each of the four GPs. Figs. 4.10(a)-(e) shows complete as well as partial manipulator trajectories. The planned manipulator path is also observed to be a safe and smooth. The simulation takes a total of 26.648 seconds to plan the 12-configuration collision-free path and the computation time for each configuration is shown in Table. 4.1. While the difference in the amount of CPU time required to compute different manipulator configurations are mostly insignificant, it takes much more time to compute the first configuration. This is because the first configuration, which can be seen clearly in Fig. 4.10(b), requires the most adjustments in manipulator joint angles from the previous (initial) configuration.

Figs. 4.11(a)-(c) show the initial (from two different views) and final configurations, respectively, of the path obtained for a 6-link manipulator working in a blocked workspace. Figs. 4.12(a)-(c) show side views of the intermediate configurations as its end-effector reaches each of the three GPs. Figs. 4.13(a)-(c) show top views of the same manipulator configurations. It can be seen clearly from Figs. 4.12 and 4.13 that the end-effector of the manipulator reaches the three GPs safely. Figs. 4.14 and 4.15 show side views and top views, respectively, of some partial trajectories of the manipulator. The safety and smoothness of the manipulator trajectory is also observed. The simulation takes a total of 34.199 seconds to plan a 11-configuration collision-free path. Because the workspace is more complex, the path planning is more time-consuming than other examples.



(a) (b) (c) (d) (e)

Figure 4.9: Top views of the initial configuration of the manipulator, as well as intermediate configurations as its end-effector reaches each of the four GPs in Fig. 4.8.

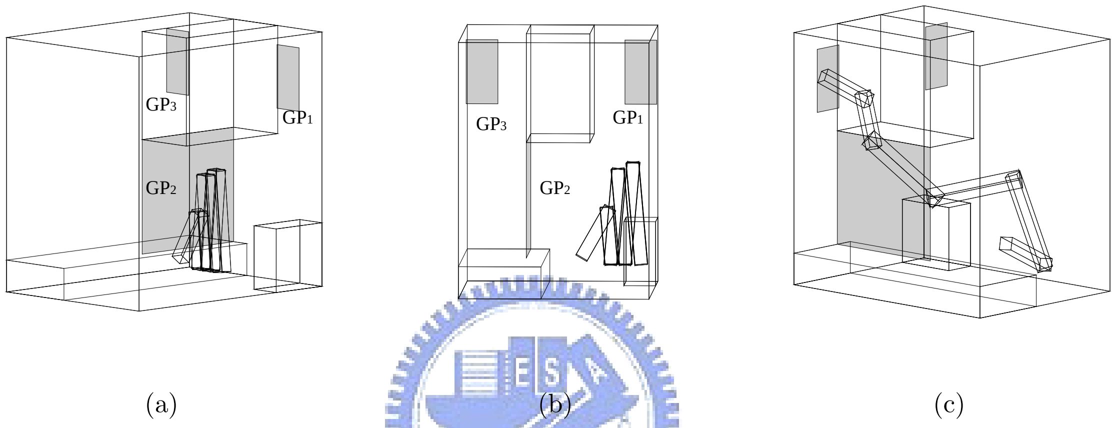


(a) (b) (c) (d) (e)

Figure 4.10: (a) The complete manipulator trajectory. (b) The partial trajectory between Figs. 4.9(a) and (b). (c) The partial trajectory between Figs. 4.9(b) and (c). (d) The partial trajectory between Figs. 4.9(c) and (d). (e) The partial trajectory between Figs. 4.9(d) and (e).

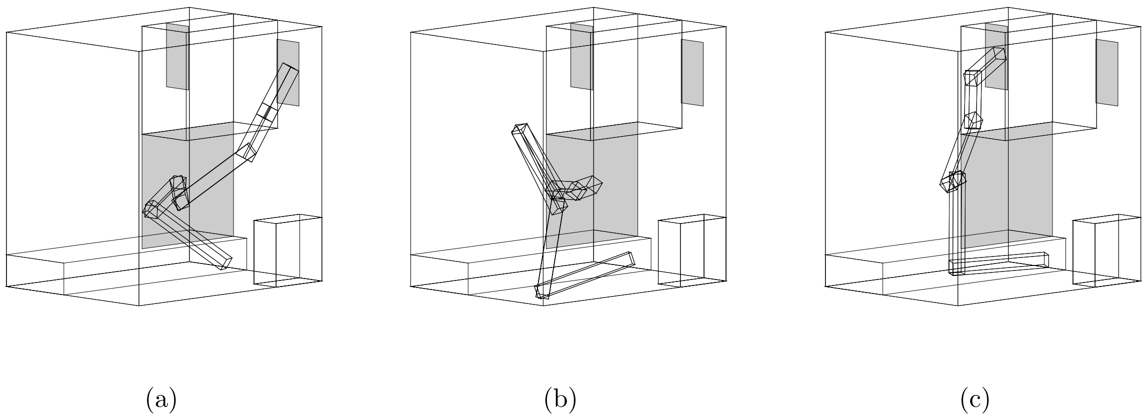
Table 4.1: Computation time for manipulator configurations shown in Fig. 11.

computation times (sec)	
config.1: 5.988	config.7: 1.963
config.2: 3.245	config.8: 1.612
config.3: 1.612	config.9: 1.472
config.4: 2.184	config.10: 1.502
config.5: 1.592	config.11: 2.374
config.6: 1.492	config.12: 1.612
total time: 26.648	



(a) (b) (c)

Figure 4.11: A path planning example for a 6-link manipulator in a 3-GP blocked workspace. (a)-(b) Two different views of the initial configuration. (c) The final configuration.



(a) (b) (c)

Figure 4.12: Side views of the intermediate configurations as its end-effector reaches each of the three GPs in Fig. 4.11.

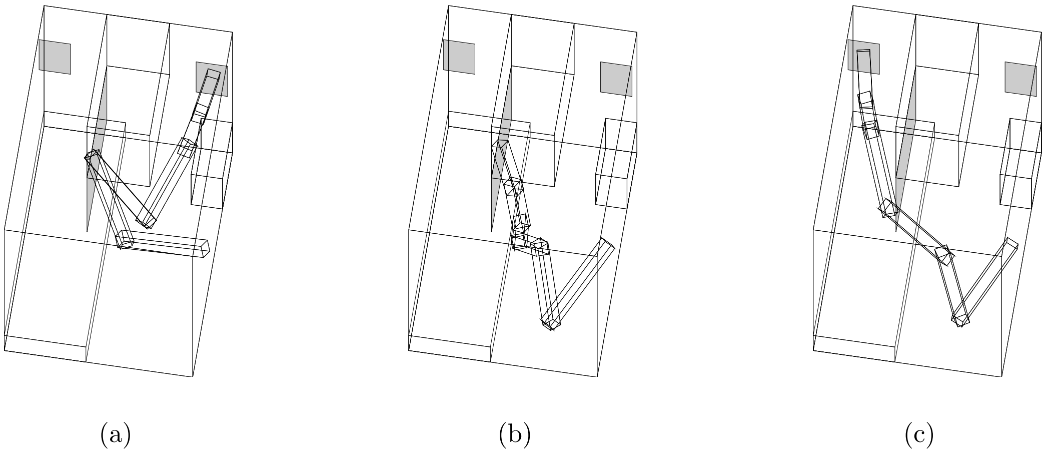

(a) (b) (c)

Figure 4.13: Manipulator configurations shown in Fig. 4.12 observed from a different viewing angle.

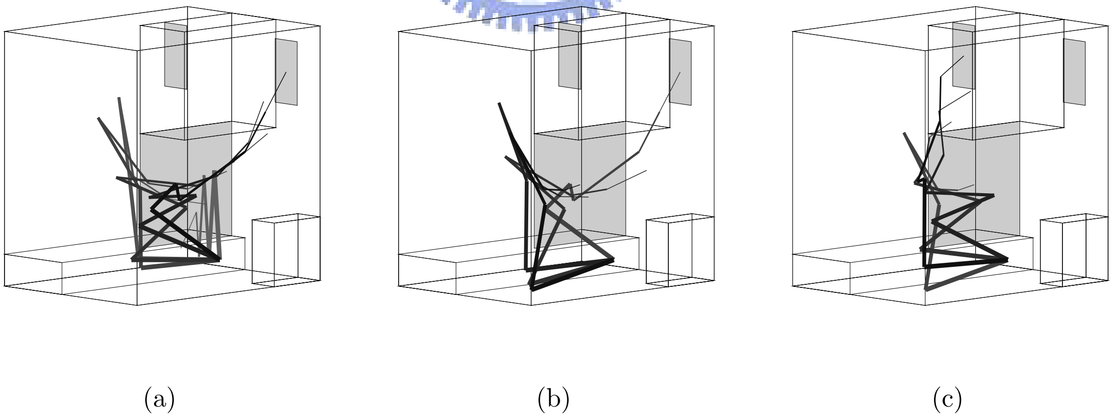
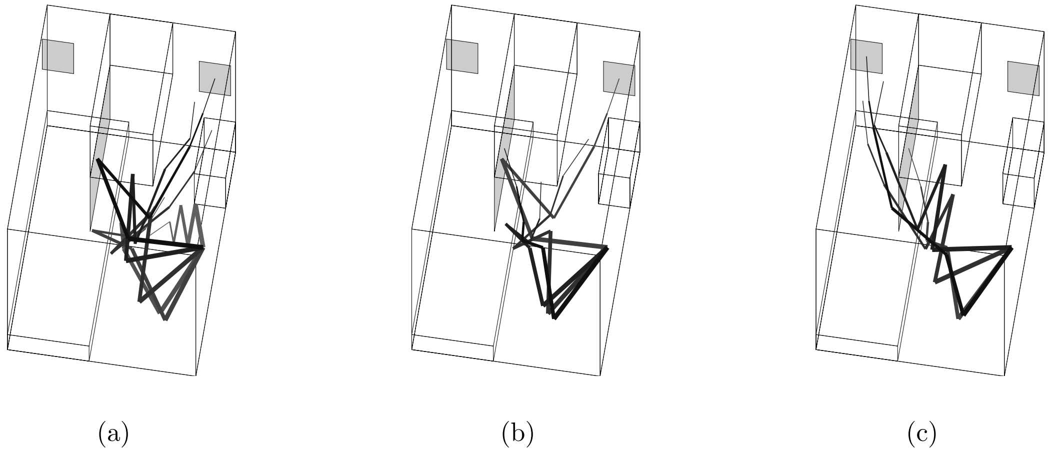

(a) (b) (c)

Figure 4.14: (a) The partial trajectory between Figs. 4.11(a) and 4.12(a). (b) The partial trajectory between Figs. 4.12(a) and (b). (c) The partial trajectory between Figs. 4.12(b) and (c).



(a)

(b)

(c)

Figure 4.15: Partial manipulator trajectories shown in Fig. 4.14 observed from a different viewing angle.

4.5 Summary

In this chapter, a potential-based algorithm for a robot manipulator is proposed to solve path planning problems in variant 3-D workspace. The proposed algorithm uses an artificial potential field to model the workspace wherein obstacles surfaces are assumed to be charged uniformly and the manipulator is represented as a set of charged sampling points. The repulsive force and torque between manipulator and obstacles thus modeled are analytically tractable, which makes the algorithm efficient. To give a general direction of the path, a sequence of GPs to be reached by the manipulator are assumed to be given in advance in the workspace. As a GP is an intermediate or final goal for the end-effector of a manipulator to reach, it also helps to establish certain motion constraints for adjusting manipulator configuration during path planning. According to such constraints, the proposed approach derives the path for a manipulator by adjusting its configurations at different locations along the path to minimize the potential using the above force and torque. Simulations show that a path thus derived is always spatially smooth and effective. Since the proposed approach uses workspace information directly, it is readily applicable to manipulators of high DOFs. While only the static environments are considered in this chapter, the proposed approach can be extended to workspace of moving obstacles with essentially no change in the path planning algorithm.

Chapter 5

Potential-based Path Planning of Articulated Robots with Moving Bases in 3-D Environments

5.1 Overview

In this chapter, the algorithm of path planning of articulated robots by using potential field model presented in [1] to model the workspace is proposed. The proposed approach is similar to that done in 3-D manipulator mentioned in Chapter 4. In this algorithm, a sequence of predetermined GPs are also used to provide the robot a general direction to move forward and also help to establish certain motion constraints for adjusting robot configuration during path planning. A collision-free path of a robot will then be obtained by locally adjusting the robot configuration to search for minimum potential configurations using the force and torque. While the base of a manipulator is fixed, the base of an articulated robot is movable. Since the base of an articulated robot is not fixed, the force exerted on a link due to the torque of other links does not longer exist. Thus, a sequential planning strategy [60] is adopted to speedup the computation.

The algorithm for articulated robot is proposed in Sec. 5.2. An implementation of the proposed approach based on a sequential planning strategy is presented in Sec. 5.3. In Sec. 5.4, simulation results are presented for path planning performed for robots in different 3-D environments. Sec. 5.5 has a discussion about GPs. Sec. 5.6 gives conclusions of this work.

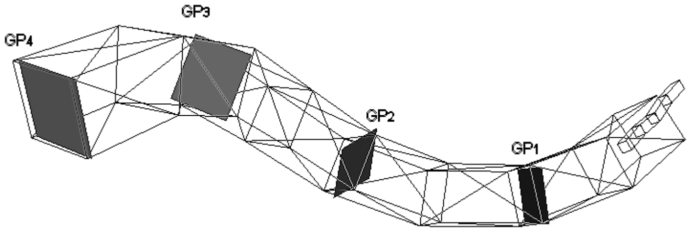


Figure 5.1: A 3-D articulated robot is to move toward the goal by sequentially traversing a sequence of GPs.

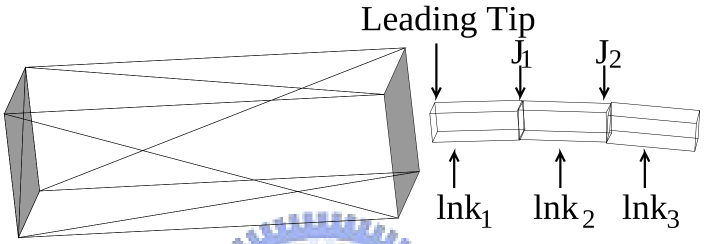


Figure 5.2: The leading tip, joints and links of a 3-D articulated robot with a moving base moved toward the goal.

5.2 The Proposed Path Planning Algorithm

In this section, the application of the potential model reviewed in the previous section to path planning of 3-D articulated robots with moving bases will be discussed. Unlike some c-space based approaches, which often require expensive preprocessing to construct the c-space, the proposed approach uses the workspace information directly. The approach computes repulsive force and torque experienced by each rigid component, e.g., a link, of a 3-D articulated robot. A collision-free path of the articulated robot can then be obtained by locally adjusting its configuration along the path for minimum potential using these force and torque. In this thesis, it is assume that links of the articulated robot are connected with spherical joints since the high DOF of such joints can take full advantage of the proposed potential minimization algorithm.

For a rough description of the path of an articulated robot, the proposed approach identifies one or more guide planes (GPs) as its final or intermediate goals in the 3-D workspace.

The GPs are polygons among obstacles in the free space, providing the articulated robot a general direction to move forward. In this thesis, it is assumed the sequence of GPs are given in advance. Usually, for an elongated free space passage, like the one shown in Fig. 5.1, the GPs can be obtained as a subset of the cross-sections associated with the *generalized cylinder* [56] representation of the passage (see Fig. 5.1). As for a description of the configuration of the articulated robot, an n-link articulated robot is represented by its leading tip and the $n-1$ joints, as shown in Fig. 5.2. A sequential and collision-free traversal of a given sequence of GPs by these control points (and the links) is regarded as a global solution of the path planning problem of the articulated robot. In the planning procedure discussed next, intermediate GPs will also be added along the path each time when a given GP is not directly reachable.

5.2.1 Basic Procedure of Path Planning

In this thesis, the proposed path planning approach derives a series of minimum potential configurations along the path of an articulated robot by locally adjusting its configuration for minimum potential using the results reviewed in Sec. 2.2. Assuming that a guide plane GP_1 is given as an intermediate goal, the basic path planning procedure for moving the leading tip, p', of an articulated robot onto GP_1 include (see Fig. 5.3):

(i) Translate links of the articulated robot to move p' toward the GP_1. If p' can not reach GP_1 directly, e.g., due to collision, an intermediate plane GP_1' is inserted. (Fig. 5.3(a))

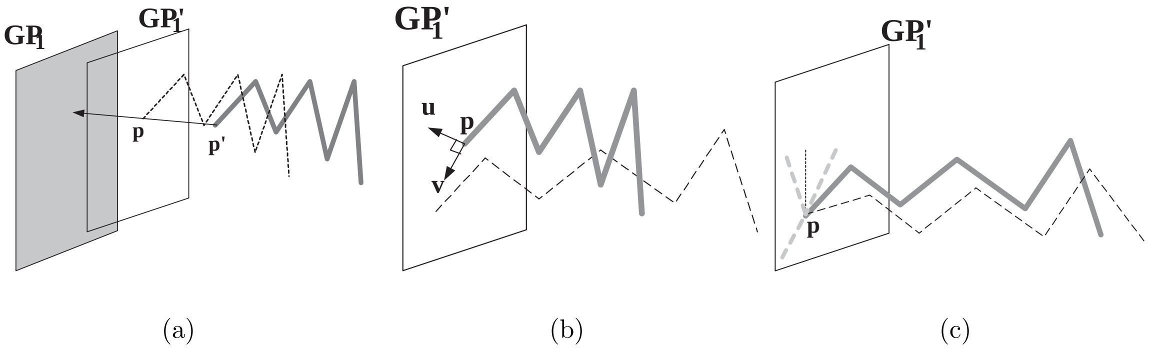
(ii) Search for the minimum potential configuration of the articulated robot for $p \in GP_1'$ by repeatedly executing :

   (a) Search for the minimum potential configuration of the articulated robot with lnk_1 fixed in orientation by sliding p on GP_1' (see Fig. 5.3(b)).

   (b) Search for the minimum potential configuration of the articulated robot with p fixed in position (see Fig. 5.3(c)).

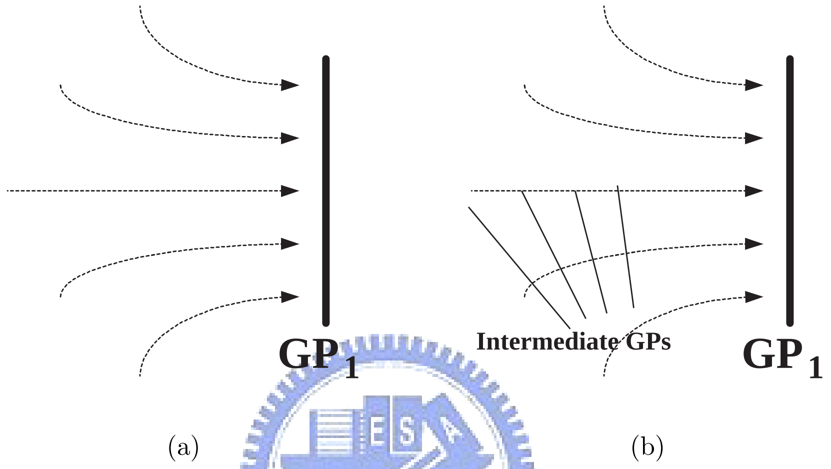
(iii) Repeat (i) and (ii) until p reaches GP_1.

In general, there are different ways to change the configuration of the articulated robot to move p' toward GP_1. A simple translation of all links is adopted in (i) as a preliminary



(a) (b) (c)

Figure 5.3: Basic path planning procedure for a given GP (see text).



(a) (b)

Figure 5.4: The moving direction of articulated robots and intermediate GPs (see text).

implementation of our algorithm, as shown in Fig. 5.3(a). The direction of the translation is determined such that $\overrightarrow{p'p}$ is in the direction of the attractive force $\overrightarrow{F_{p'}}$ experience by p' due to GP_1. A collision check is performed for such a translation. If collision occurs, the distance of the translation is halved until the translation is collision-free. Accordingly, a new GP is inserted, e.g., GP_1'. No configuration improvement to reduce the repulsive potential is considered at this stage.

As for the search for the minimum potential configuration of the articulated robot in (ii), since lnk_1 has five DOFs, i.e., two for its location for $p \in GP_1'$ and three for its orientation, the associated constrained optimization problem is divided into two iterative univariant optimization procedures, as in (ii-a) and (ii-b). In (ii-a), lnk_1 is fixed in its orientation (see Fig. 5.3(b)) as p slides on GP_1' to search for the minimal potential configuration and other links are sequentially adjusted in orientation, staring from lnk_2. In (ii-b), lnk_1 is adjusted in orientation while fixed in position (see Fig. 5.3(c)) and the procedure for adjusting the rest links is similar to that in (ii-a). For a particular GP, say GP'_1 in Fig. 5.3, (ii-a) and (ii-b) are repeatedly performed until negligible changes in the configuration of the articulated robot is obtained. Then, another intermediate GP which is closer to GP_1 is obtained with (i) and the process repeats. The path planning algorithm, as summarized below, ends as the leading tip reaches the given GP_1 or exist abnormally for an infeasible problem. Detailed implementation of the algorithm is presented in the next subsection.

Algorithm Articulated_Robot_ to_GP

Step 0 Initialize $\delta = \delta_0$ and $GP \leftarrow GP_1$.

Step 1 Translate all links of the articulated robot with distance δ to move p' to p along $\overrightarrow{F_{p'}}$. Find the smallest $n \geq 0$ such that $\delta \longleftarrow \delta/2^n$ corresponds to a feasible and collision-free translation. If $\delta < \delta_{min}$, go to Step 6; otherwise, construct GP'_1 with $GP'_1 \perp \overrightarrow{F_p}$ and $p \in GP'_1$, and let $GP \leftarrow GP'_1$.

Step 2 Translate lnk_1, by sliding p on GP, and adjust the orientations of the rest links to minimize the potential.

Step 3 With p fixed in position, adjust the orientations of all links to minimize the potential.

Step 4 Go to Step 2 if the translation in Step 2 or the joint angle adjustment in Step 3 is not negligible.

Step 5 If p reaches GP_1, the planning is completed. Otherwise, $p' \leftarrow p$ and go to Step 1 with $\delta = \delta_0$.

Step 6 Exit and report that GP_1 is unreachable.

For path planning involving multiple GPs, the above algorithm will be executed for each of them sequentially. It is assumed that the planning for a GP starts as the planning of the previous GP is accomplished. The path planning ends as the leading tip of the articulated robot reaches the goal, which is usually a (goal) GP in the path planning problems considered in this thesis.

5.3 Implementation Details

5.3.1 Initialization and Step 1 of Algorithm *Articulated_Robot_to_GP*

In the initialization of *Articulated_Robot_to_GP*, the initial step size, δ_0, is arbitrarily chosen, i.e., $\delta_0 = 10\%$ of workspace size. Generally, the number of robot configurations, and thus the computation time, depend on δ_0. For larger δ_0, less configurations will need to be calculated along the path if there is no collision. On the other hand, if δ_0 is too large, the computation time may actually increase because collisions may occur frequently. A threshold, i.e., $\delta_{min} = 1\%$ of workspace size, is established to set a lower bound of the magnitude of allowed translation. A translation which requires a smaller movement, $\delta < \delta_{min}$, indicates an infeasible path planning problem.

In Step 1 of *Articulated_Robot_to_GP*, there are different ways to move p' toward GP_1. For example, p' can be moved toward GP_1 in the normal direction of GP_1. In our simulation, the articulated robot is translated, as a rigid object, to move the leading tip from p' to p by distance δ along the direction of attraction, $\overrightarrow{F_{p'}}$, as shown in Fig. 5.4(a). In the far field, $\overrightarrow{F_{p'}}$ has a near spherical symmetry and p' is attracted toward GP_1 as if GP_1 is a point attractor. In the near field, $\overrightarrow{F_{p'}}$ will lead p' toward GP_1 approximately in the normal direction of GP_1. In practice, since δ_0 is finite instead of infinitely small, p' may reach GP_1 at different locations from various directions.

While no configuration improvement to reduce the repulsive potential is considered in the above translation procedure, Steps 2 and 3 minimize the potential through constrained optimization procedures. To that end, an intermediate GP, GP_1', is added along the path to serve as a geometric constraint for successive adjustments of the configuration of lnk_1. Again, as long as $p \in GP_1'$, the choice of the orientation of GP_1' is not unique. For example, GP_1' can be oriented such that it is parallel with GP_1. A more reasonable approach is adopted in our simulation wherein GP_1' is chosen to be perpendicular to $\overrightarrow{F_p}$, as shown in Fig. 5.4(b). Thus, the direction of translation to minimize the potential in Step 2, as discussed next, is perpendicular to that of the translation in the next execution, if necessary, of Step 1.

5.3.2 Adjusting the leading tip position in Step 2

After the leading tip reaches p, as shown in Fig. 5.3(b), one of the univariant procedures to minimize the repulsive potential, which allows lnk_n to adjust its location but not its

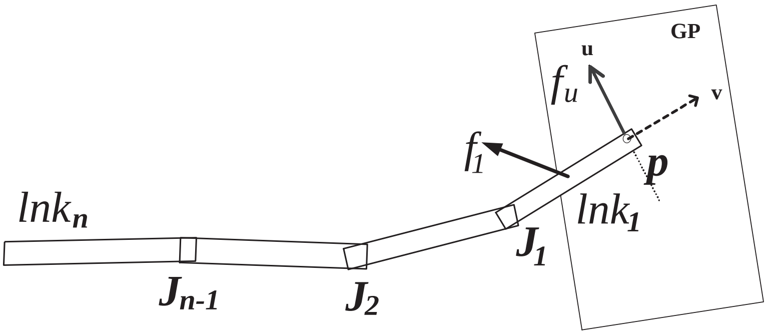


Figure 5.5: Translating lnk_1 to slide p on GP_1' to reduce the repulsive potential.

orientation, is performed in Step 2 of *Articulated_Robot_to_GP*. In order to speed up the computation, a sequential planning strategy [60] which plan the motion of links sequentially is adopted in Step 2. Accordingly, the procedures of Step 2 first minimize the potential of lnk_1 whose translation is constrained in two dimensions by GP. As for the motion of lnk_i $(i > 1)$, as in [60], it is assumed that all links up to lnk_{i-1} has been planned, i.e., one end of lnk_i is fixed in position. Therefore, only the orientation of lnk_i, which has three degrees of freedom, need to be adjusted for minimum potential.

Consider the forces exerted on lnk_1, as shown in more detail in Fig. 5.5. Since the minimization is constrained by $p \in GP$, only the projection of the resultant force experienced by lnk_1 on GP is taken into account. Let f_1 be the repulsive force exerted on lnk_1 due to the repulsion between lnk_1 and the obstacles. For a univariant minimization approach, only one variable is adjusted at a time. To determine the direction in which lnk_1 should translate to slide p on GP to reduce the repulsive potential, the projection of the resultant force exerted on lnk_1 along an arbitrary $\overrightarrow{u} \parallel GP$, $f_{1,u}$ is calculated.A gradient-based binary search for the minimum potential location of p along $\overrightarrow{u}$ can then be performed. The initial step of sliding is arbitrarily chosen as 10% of the workspace size. If the movement of lnk_1 along $\overrightarrow{u}$ or $f_{1,u}$ is negligible, e.g., the movement is smaller than 1% of workspace size, another minimization of potential along $\overrightarrow{v} \parallel GP$, which is orthogonal to $\overrightarrow{u}$, is performed to minimize $f_{1,v}$. Step 2 ends when two consecutive binary searches along the two orthogonal directions result in negligible movement of lnk_1. Once the optimal configuration of lnk_1 is determined, the procedure for adjusting the rest links is similar to that in Step 3, as discussed next.

5.3.3 Adjusting joint angle in Step 3

Once the minimum potential position of lnk_n is determined with Step 2 of *Articulated_Robot_to_GP*, another univariant procedure, which allows lnk_1 to adjust its orientation with leading tip p fixed in position is performed to reduce the potential further, as shown in Fig. 5.3(c). Under such a constraint, lnk_1 can rotate with respect to p to reduce the repulsive potential. The direction in which lnk_1 should rotate is determined by the repulsive torque experienced by lnk_1 with respect to p.

Let τ_1 be the repulsive torque experienced by lnk_1 with respect to p due to the repulsion between lnk_1 and obstacles. To find the minimum potential orientation of lnk_1 for p fixed in position, gradient-based binary searches are performed repeatedly using the projection of τ_1 along three orthogonal axes of rotation, e.g., $\tau_{1,u}$, $\tau_{1,v}$ and $\tau_{1,w}$, respectively. For each binary search, the initial rotating angle is arbitrarily chosen as 5^o, while the accuracy in identifying the 1-D potential minimum is chosen as 0.5^o. Step 3 ends if a binary search results in a negligible change in the orientation of lnk_1, i.e., less than 0.5^o.

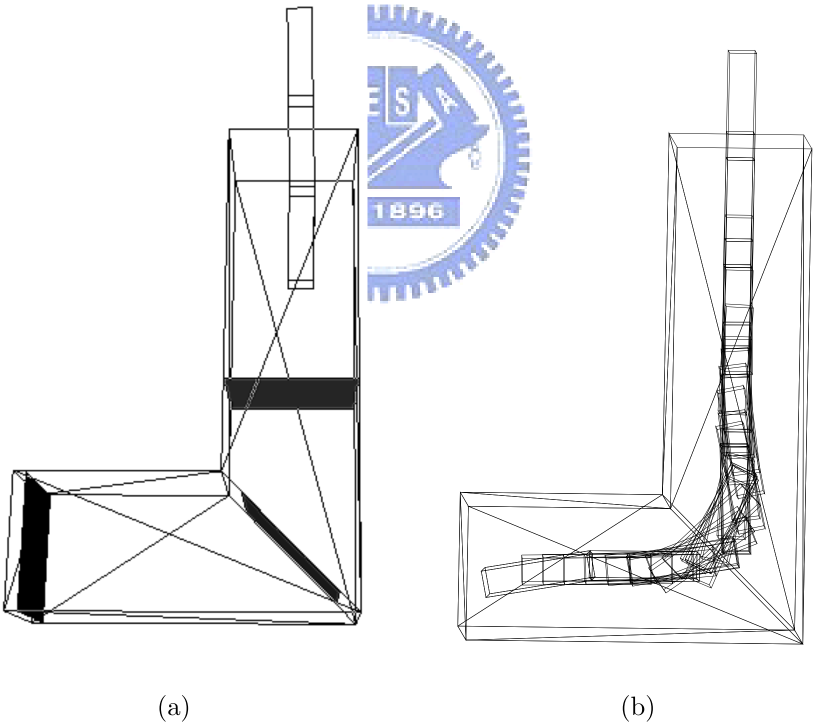
5.4 Simulation Results

In this section, simulation results are presented for path planning performed on Pentium III (500MHz) personal computer for articulated robots in 3-D environments. Fig. 5.6(a) shows the initial configuration of the path obtained for a 3-link articulated robot in a 3-GP workspace wherein the GPs are shown as black polygons. Fig. 5.6(b) shows the complete trajectory of the articulated robot which reaches the final GP safely. Since collisions occur frequently near the 90^o turn of the passage for the δ_0 chosen, more configurations of the articulated robot are planned near the turn than those near start and goal. Due to the repulsive potential model, the trajectory of the articulated robot is smooth and lies near the middle of the workspace. The simulation takes a total of 18.266 seconds to plan the 12-configuration collision-free path.

Figs. 5.7(a)(b) show the initial configuration and the trajectory, respectively, of the path obtained for another 3-link articulated robot in a 2-GP workspace. The simulation takes 26.638 seconds to planning the 8-configuration path within a workspace with 48 triangles of obstacles. While the length of the passage shown in Fig. 5.6 is shorter than that in Fig. 5.7, less robot configurations is generated for the path of the letter because the passage is spatially more smooth.

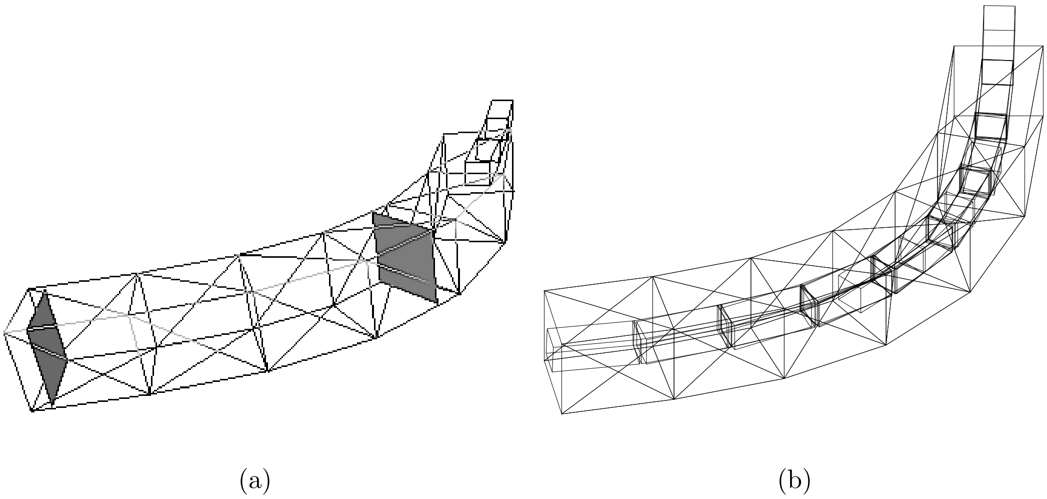
Figs. 5.8(a)(b) show the initial configuration and final trajectory for the path planning of a 4-link articulated robot moving into a winding passage with 4 GPs. Fig. 5.8(a) shows the initial configuration of the articulated robot which lies at the entrance of the passage. The trajectory shown in Fig. 5.8(b) indicates that the articulated robot reaches the final GP safely. While the robot motions in Figs. 5.6 and 5.7 are essentially two-dimensional, the robot trajectory planned in Fig. 5.8 requires three-dimensional maneuvering. The simulation takes a total of 68.999 seconds to plan the 11-configuration collision-free path.

Figs. 5.9(a)(b) show the initial configuration and final trajectory of the path obtained for another 4-link articulated robot moving in a turnaround passage. It can be seen clearly from Fig. 5.9(b) that the articulated robot traverses the five GPs safely and smoothly. The simulation takes a total of 151.518 seconds to plan the 24-configuration collision-free path. Since the passage of this example is more crooked, more intermediate GPs are added into the path, which in turn increases the computation time.



(a) (b)

Figure 5.6: A path planning example for a 3-link articulated robot in a 3-GP workspace. (a) The initial configuration. (b) The trajectory.



(a)

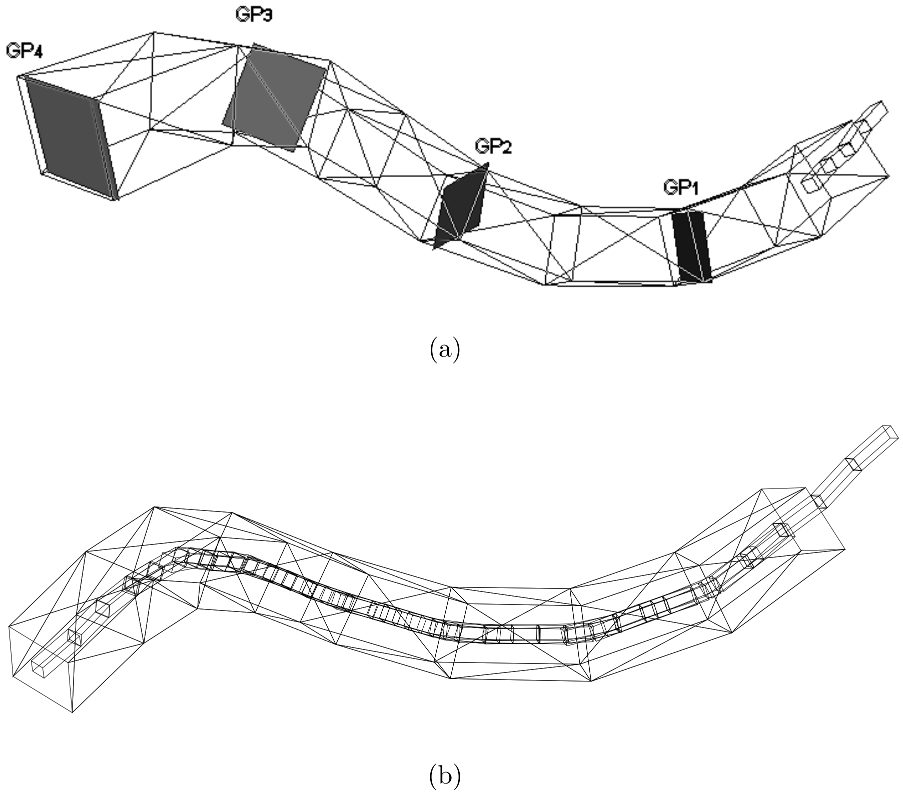
(b)

Figure 5.7: A path planning example for a 3-link articulated robot in a 2-GP workspace. (a) The initial configuration. (b) The final trajectory.(c) The 248-triangle tunnel.

5.5 Discussion

5.5.1 Computational Complexity

In the proposed path planning approach, since robots are represented by point charges and obstacles are represented by charged triangles, the evaluation of repulsion between them involves the calculation of the repulsion between points-triangle pairs. In general, for path planning problems having similar shapes of robot, obstacles, and robot trajectories, the computation time will be proportional to (i) the number of point samples of the robot, (ii) the number of triangles of obstacles, and (iii) the number of robot configurations calculated. For example, the path planning within a workspace similar to Fig. 5.7 but with 248-triangle obstacles as shown in Fig. 5.10 takes 133.311 seconds. The computation time for similar tunnels with different numbers of obstacle triangles is shown in Figs. 5.11(a). One can see clearly that the computation time is close a linear function of the number of triangles. Figs. 5.11(b) shows the computation time for similar tunnels but with different numbers of robot links with about the same length as that in Fig. 5.7 when fully stretched. For identical number of point samples taken into account for each robot link, the computation time is approximately proportional to the number of robot links.



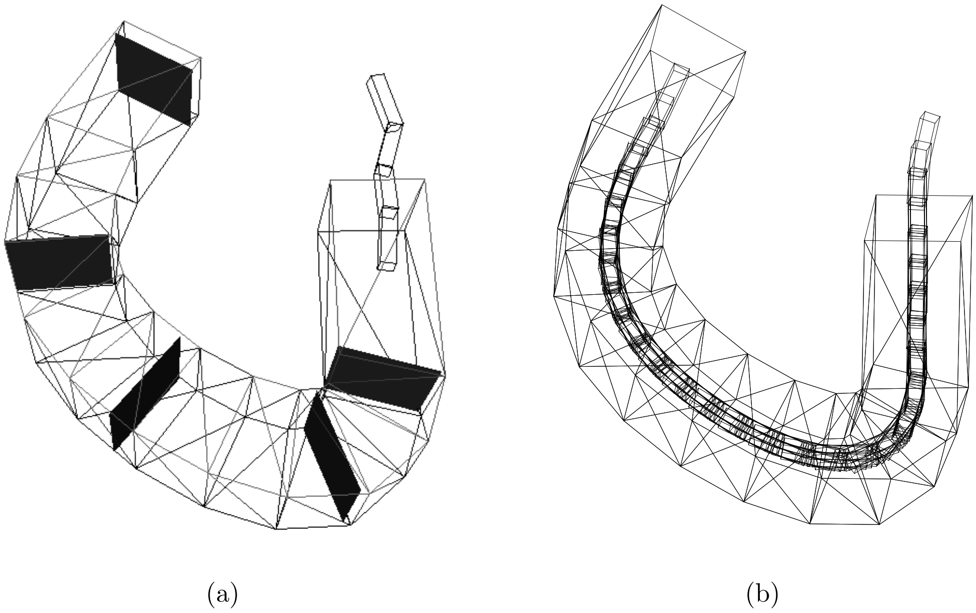
(a)

(b)

Figure 5.8: A path planning example for a 4-link articulated robot in a 4-GP passage. (a) The initial configuration. (b) The final trajectory.

5.5.2 More Details About Generation and Selection of GPs

As mentioned in Section 4.2, the GPs provide the articulated robot a general direction to move forward. In some very rare occasions, Step 1 of *Articulated_Robot_to_GP* may fail to generate intermediate GPs needed in the path planning. For example, Fig. 5.12(a) shows a passage with 3 GPs identified. Since $GP_1 \parallel GP_2$, $\overrightarrow{F_{p'}}$ will always be in the same direction which causes collision. Fortunately, such a problem can be identified and resolve easily. For the triangular-mesh model of the passage show in Fig. 5.12, there are about 20 quadrilateral cross-sections between GP_1 and GP_2, which are possible candidates of a denser set of GPs. Figs. 5.13(b) and (c) show the initial conditions when one and two GPs, respectively, are added as initial GPs between GP_1 and GP_2 in Figs. 5.12(a). One can see easily that the two paths obtained with the proposed approach, as show in Figs. 5.13(a) and (b), are very similar. For the examples considered in this thesis, the initial GPs are selected arbitrarily and the algorithm seems to work reasonably well in terms of the sensitivity of the planned path to the selection of initial GPs.



(a) (b)

Figure 5.9: A path planning example for a 4-link articulated robot in a 5-GP workspace. (a) The initial configuration. (b) The final trajectory.

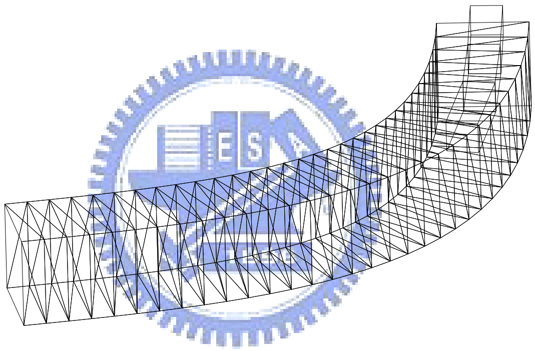
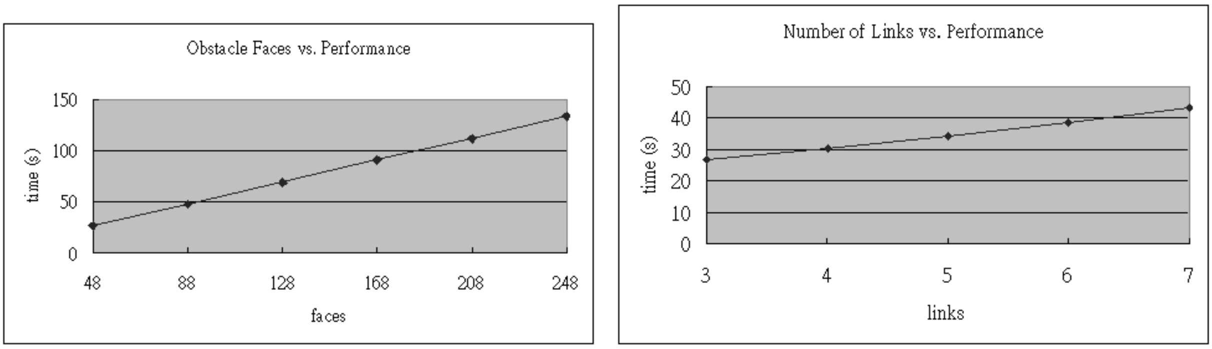


Figure 5.10: A path planning example like Fig. 5.7 with 248 triangles of obstacles.

5.5.3 Articulated Robots with 2-DOF Joints

In [61], a potential-based path planning algorithm was proposed for articulated robots with 3-DOF joints, spherical joints, as shown in Fig. 5.14(a). Because the joint has highest DOFs, the robot can take full advantage of the proposed potential minimization algorithm in reducing the potential. In this thesis, an improved algorithm of [61] is proposed for a robot with 2-DOF joints. While a robot with active 3-DOF joints is not trivial in practice, a robot with 2-DOF joints is usually adopted. The Hooke joint, which has two orthogonal rotation axes as shown in Fig. 5.14(b), is adopted in this thesis to establish the connection between two links.

Consider joint J_i which connects two links, lnk_i and lnk_{i-1}, as shown in Fig. 5.14(b).



(a)

(b)

Figure 5.11: The computation time will be proportional to the number of triangles of obstacles, and the number of robot configurations calculated. (a) Computation time vs. Number of triangles of obstacles. (b)Computation time vs. Number of links.

Assume R_{i-1} represents the rotation axis of lnk_{i-1} after a potential minimization procedure while R_i^A is the rotation axis of lnk_i (which is to be connected to lnk_{i-1} such that a potential minimization for lnk_i can take place), as shown in Fig.5.15. Because of the constraint of the 2-DOF joint ($R_i \perp R_{i-1}$), lnk_i can not be connected directly to lnk_{i-1} through pure translation if the orientation of lnk_{i-1} has been changed due to potential minimization. Let S_i be the skeleton axis of lnk_i. It is obviously that two rotation axes are not always orthogonal if two links are connected arbitrarily. Such a configuration of these two links is illegal for a robot with 2-DOF joints. Therefore, R_i should be moved from R_i^A to R_i^B and then orthogonal to R_{i-1}. R_i^B can be determined as following. Assume all possible positions of R_i are lying on Plant P if lnk_i rotates about the axis S_i. $R_{i-1}^{\perp}$ is the projection of R_{i-1} on P. If a vector R_i^B of P is orthogonal to $R_{i-1}^{\perp}$, it will be orthogonal to R_{i-1} too. Thus, lnk_i should rotate about S_i by θ to let R_i moves to R_i^B. After the previous adjusting lnk_i orientation, the two links are connected legally with a 2 DOF joint. And, then, the further potential minimization can take place.

A Comparison Between 2-DOF Joints and 3-DOF Joints

In this section, the simulation results of the 2-DOF-joint algorithm is compared with the ones of 3-DOF-joints algorithm. In order to show the difference of algorithms clearly, the third configurations of robots with 2-DOF joints and 3-DOF joints of a simple example are shown in Figs. 5.16(a)(b), respectively. In Fig. 5.16(b), the second link is twisted to minimize the configuration potential due to the additional DOF, rotating with respect to the

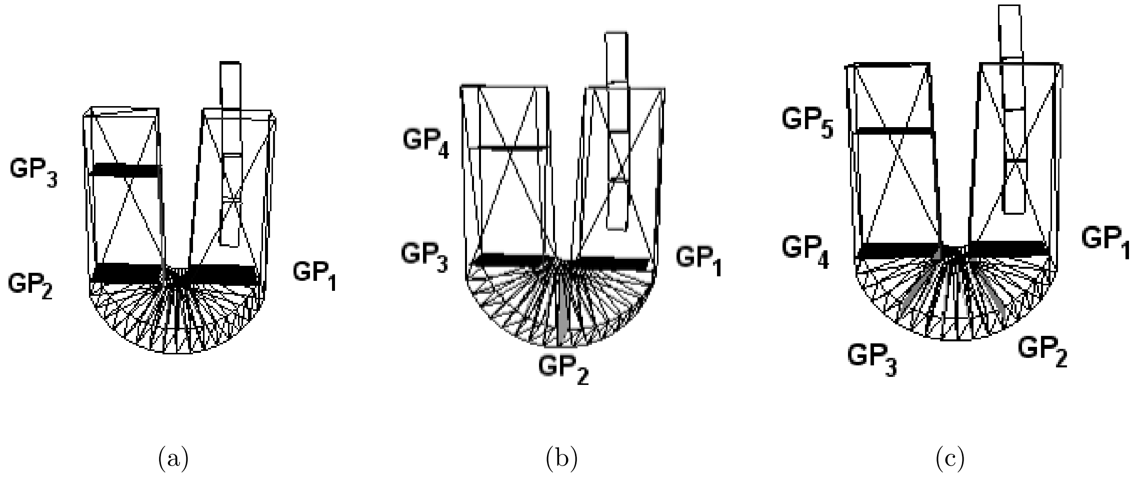


Figure 5.12: Path planning examples for a 4-link articulated robot in an U-shaped workspace with (a) 3 (b) 4 and (c) 5 initial GPs.

skeleton. In general, the algorithm with 2-DOF joints takes more time than a 3-DOF one to plan a collision-free path due to less DOFs to minimize the potential. Figs. 5.17(a)(b) show trajectories of a complex example of algorithms with 2-DOF joints and 3-DOF joints, respectively. While Fig. 5.17(a) takes 91.34 seconds to planning the 24-configuration path, Fig. 5.17(b) taking 88.16 seconds to planning the 18-configuration path. Due to the less freedom, the 2-DOF-joint algorithm needs more time to plan more intermediate configurations. Since the 3-DOF-joint robot has additional DOFs to minimize potential than the former, the planned trajectory is more smoothly. Though there are different between two trajectories of Figs. 5.17, the difference is not so significant. Thus, the proposed algorithm with 2-DOF joints is work well as the algorithm with 3-DOF joints.

5.6 Summary

In this thesis, a potential-based path planning of articulated robots with moving bases is proposed. According to the adopted potential model, surfaces of workspace obstacles are assumed to be uniformly charged and the links of the articulated robot are represented as a set of charged sampling points. The repulsive force and torque between links and obstacles thus modeled are analytically tractable. It is assumed that a sequence of GPs to be traversed by an articulated robot are given in advance in the workspace providing the general direction of the path. According to the motion constraints established by the

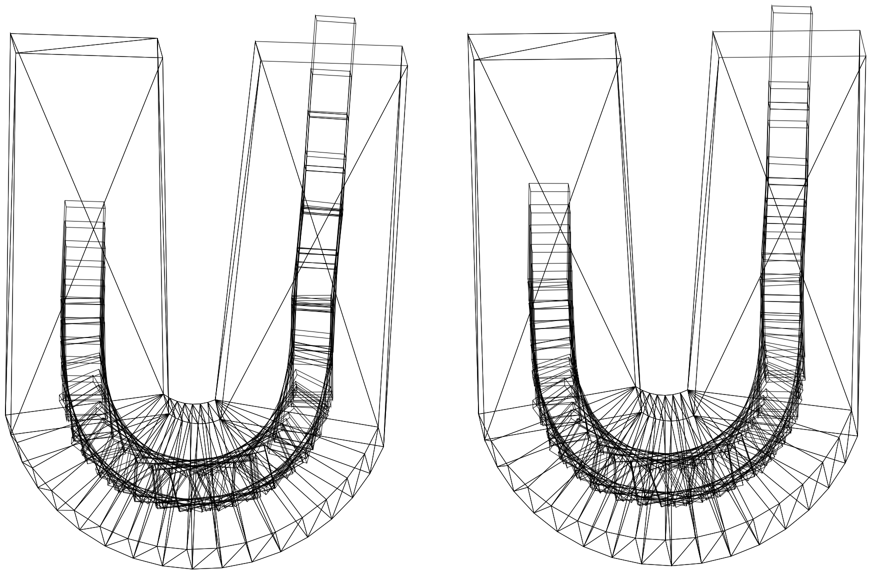


Figure 5.13: Successful paths of Figs. 5.12(b)(c), respectively.

GPs, the proposed approach derives the path for the articulated robot by adjusting its configurations at different locations along the path to minimize the potential using the above force and torque. Simulation results show that a path thus derived always stays away from obstacles and is spatially smooth.

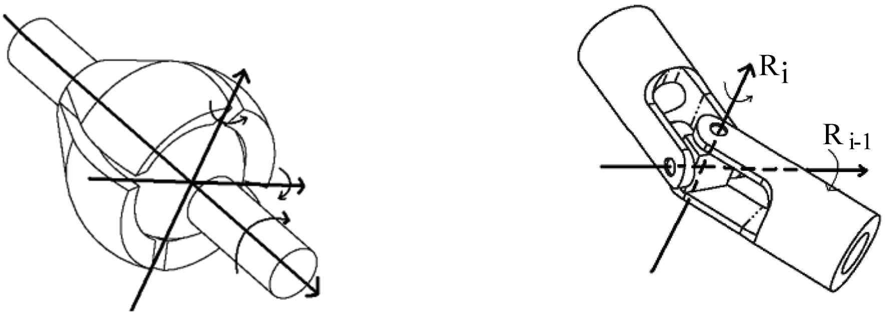


Figure 5.14: A path planning example for a 3-link articulated robot in a 3-GP workspace. (a) The initial configuration. (b) The trajectory.

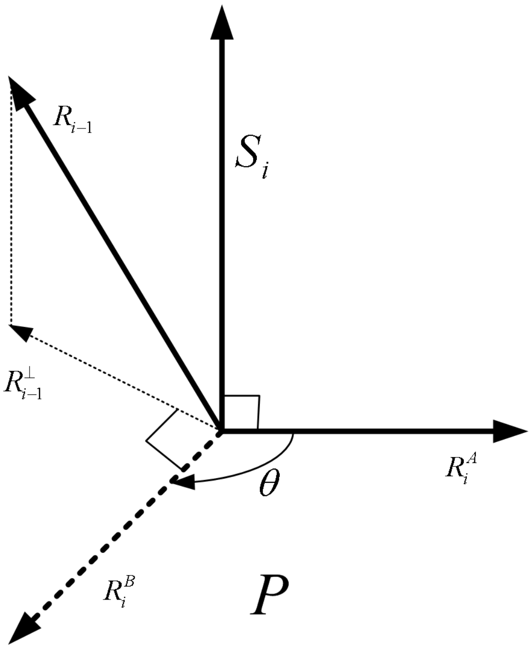


Figure 5.15: Since T_{i-1} is perpendicular with Ti, the lnk_i should rotate with respect to S_i to let T_i' moves to T_i.

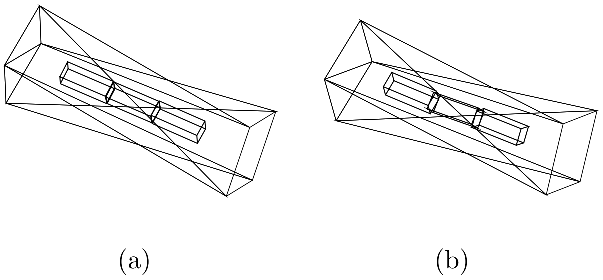


Figure 5.16: Two planned paths for a 3-link articulated robots with 2-DOF and 3-DOF Joints. (a) The third configuration of the robot with 2-DOF joints. (b) The third configuration of the robot with 3-DOF joints.

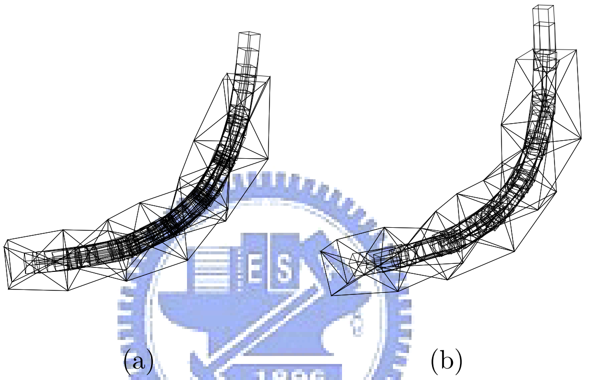


Figure 5.17: Two planned paths for a 3-link articulated robots with 2-DOF and 3-DOF Joints in the same environment.(a) A 24-configuration path of the robot with 2-DOF joints. (b) A 24-configuration path of the robot with 3-DOF joints.

Chapter 6

Potential-based Shape Matching and Recognition of 3-D objects

6.1 Overview

In this chapter, the potential-based shape matching approach presented in [50] is extended to three dimensions using a generalized potential model presented in [1]. The goal is to develop a potential-based 3-D shape matching algorithm which, as its 2-D counterpart, can correctly and efficiently perform the matching without knowing the exact information about the location, orientation and size of the input object. Furthermore, the proposed shape matching scheme is not based on any hypothesis of feature correspondence. Therefore, feature extraction of an input object which is required for a structured object representation is not needed. According to such a model, the repulsion between two 3-D objects, one with polyhedral description and the other represented by point samples on its surface, can be evaluated analytically.

In 6.2, a potential-based shape matching algorithm using these analytic results is developed. Some computer implementation details of the matching algorithm are also presented. Simulation results for the shape matching of some 3-D objects are presented in 6.3. 6.4 presents some concluding remarks.

6.2 The Potential-Based Shape Matching

In this thesis, a potential-based approach to 3-D shape matching is proposed. The matching process is a direct extension of the 2-D approach presented in [50]. While the matching procedure for 2-D shapes starts with a shape template placed inside an input object, the 3-D matching considered in this thesis is performed by first placing input data (which may

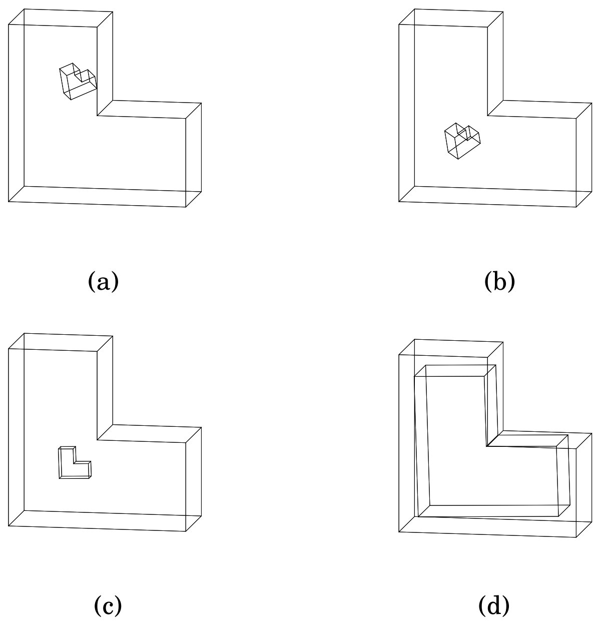


Figure 6.1: The basic shape matching procedure: (a) place a (size-reduced) input object inside a template object, (b) translate and (c) rotate the input object to reduce the potential, then (d) increase the size of the input object.

be obtained from a partial object surface) inside a shape template.

6.2.1 The Shape Matching Process

For the shape matching process, the following operations are performed with an input object placed inside a template object (see Figure 6.1). Since the potential function, and its gradient, will increase indefinitely as a point of the input object approaches the boundary of the template object, the input object will be confined inside the template object throughout the matching process if sufficient samples are taken from the surface of the former.

Step 1. Potential minimization through translations

(a) Compute the total force between the template object and the input object. Find the minimum potential position of the latter along the above force direction.

(b) Repeat (a) until two consecutive executions of (a) result in negligible difference in the two minimum potential positions found.

Step 2. Potential minimization through rotations

(a) With the centroid of the input object chosen as the rotation center, compute the total repulsive torque between the template object and the input object. Find the angular position of minimum potential for the input object by rotating it with the rotation axis aligned with the above torque direction.

(b) Repeat (a) until an execution of it results in negligible difference in the angular position of the input object.

Step 3. Scaling

With the centroid of the input object fixed in space, and with the constraint that the object remains inside the template object,find the maximum size of the input object.

Step 4. End the matching process if the execution of Step 3 results in negligible size change of the input object; otherwise, go to Step 1.

6.2.2 Implementation of the Algorithm

Implementation of Step 1

Since the input object is located inside the template object and a contact between them will result in an infinite potential value, the minimum potential position always exists along the initial force direction found in Step 1(a). (Situations involving multiple local minima in potential will be discussed later in this section.) For the computer implementation, the minimal potential position is identified efficiently by performing a gradient-based binary search using the projection of the repulsive force along the initial force direction. The precision in specifying the location of the input object is chosen to be 0.1% of the length of the template object. [1]

Although a minimal potential position along the initial force direction can be found with an execution of Step 1(a), it is possible to reduce the potential further by searching for minimum potential configuration of the input object in a different direction (ideally orthogonal to the previous search direction) in the 3-D space. In Step 1(b) this is done by recalculating the total repulsive force then repeating Step 1(a). The search process ends when the object translation due to two consecutive executions of Step 1(a) is negligible, i.e., less than 0.1% of the length of the template object.

[1] The precision in specifying the object position and orientation are arbitrarily chosen such that the configuration change due to further minimization of the potential is not noticeable.

Implementation of Steps 2 and 3

While the repulsive force is used to translate the input object to a position of minimum potential in Step 1, if the input object is allowed to rotate, it is possible to reduce the potential further. (The choice of the object centroid as the rotation center is mainly for the convenience of the computer implementation and will have little effect on the shape matching results.) Therefore, the minimum potential object orientation is determined in Step 2 of the proposed shape matching algorithm. Because the degree of freedom of object rotation in searching for such an orientation is comparable to that discussed in the previous subsection for object translation, the minimization procedure is similar to that of Step 1. The precision in specifying the 1D potential minimum in Step 2(a) is chosen to be within 0.5 degrees of the object angular position.

The process of maximizing the size of the input object in Step 3 is similar to the processes discussed above for finding the object configuration of minimum potential. The center of expansion of the object is conveniently chosen as its centroid. The associated binary search is carried out with a finite number of iterations determined by the closest distance allowed between the input object and the template object. For the simulation results presented in this thesis, the closest distance is chosen to be 0.1% of the length of the template object.

Initial Object Configurations

For the process of finding the minimum potential configuration of an input object, the result corresponds to a local minimum of the potential function. In general, depending on the initial object configuration, such a result may not correspond to the global minimum. For example, consider the two rectangles shown in Figure 6.2. As the inner rectangle rotates with respect to its centroid, the potential will have local minima at four different angular positions with two of them do not correspond to the global minimum in the potential value. In order to resolve such a problem, multiple initial object configurations are tried out for each of the shape matching problems considered in this thesis.

For the implementation of the proposed approach, it is assumed that the input object is initially placed near the centroid of a template object. Subsequently, the object is rotated into different orientations each used in a separate shape matching process. For simplicity, the centroid of the input object is chosen as the rotation center and the line containing the centroid and parallel to the z-axis is chosen as the axis of the object (see Figure 6.3). With

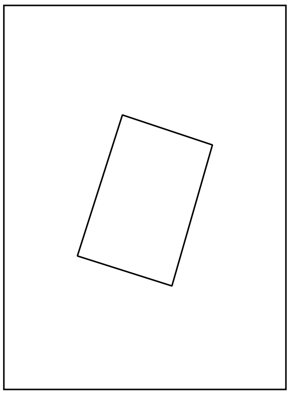


Figure 6.2: With its centroid chosen as the rotation center, the inner rectangular will experience local minima in the potential value at four angular positions.

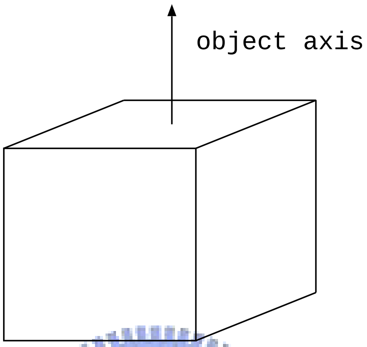


Figure 6.3: The arbitrarily chosen object axis is passing through the object centroid and is parallel to the z-axis.

the axis initially pointed to vertex A of the tetrahedron shown in Figure 6.4, the object is rotated such that the axis is pointed to three other vertices B, C, and D. For each of the four axis directions, the object is rotated by 0°, 120°, and 240° with respect to the object axis, respectively, to generate three initial object orientations. Therefore, there are twelve different initial orientations totally. Initial object configurations thus generated seems to work well for the shape matching problems considered in this thesis[2] , as presented next.

6.3 Simulation Results

We now present some experimental results. The potential-based shape matching system was tested using databases of 3-D object models obtained from:

- Washington State University, NETLIB scientific data repository.

[2] To improve the efficiency of the proposed approach, it will be more desirable if less initial configurations can be used while the matching result will not be affected. Such an important issue is currently under investigation.

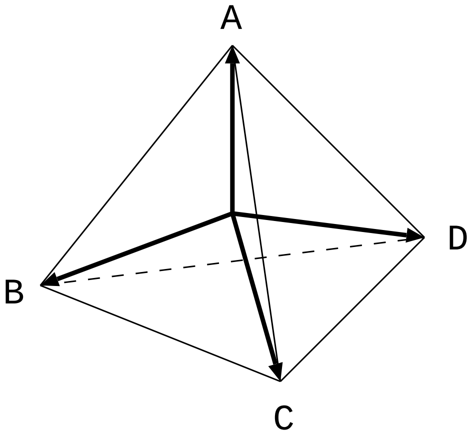


Figure 6.4: Four directions of object axis used to generate twelve initial object orientations.

- University of South Florida's vision research group.
- Michigan State University's PRIP Lab.

The object models comprise not only simple polyhedra but also curved objects and objects with complex geometry. Fig. 6.5 shows images of some of the object models. A text file which contains lists of vertices, polygons, and surfaces of the associated polyhedral representation can be obtained for each object model.

In the following experiments, shape templates are represented by complete polyhedral surfaces given in the databases. Simulation will first be performed for "ideal" input objects, i.e., point samples of an input object are obtained by directly using the above list of vertices. Non-ideal cases will then be considered which include missing vertices due to a partial view of an object and the perturbation in vertex locations. Finally, simulation will be performed to show the effect of sampling.

6.3.1 "Ideal" case

In this subsection, we first choose nine simple polyhedra from the databases as shape templates and pick four of them as input objects. Figs. 6.6(a) - 6.6(d) show the shape matching results for the four input objects, respectively. In each of the figures, the matching results shown for each template object include (1) one of the initial input object configurations, (2) the final (size maximized) input object configuration and (3) the template object whose configuration remains unchanged throughout the matching process. The best match which correctly identifies a shape template for one of the input objects is marked with a "V". The

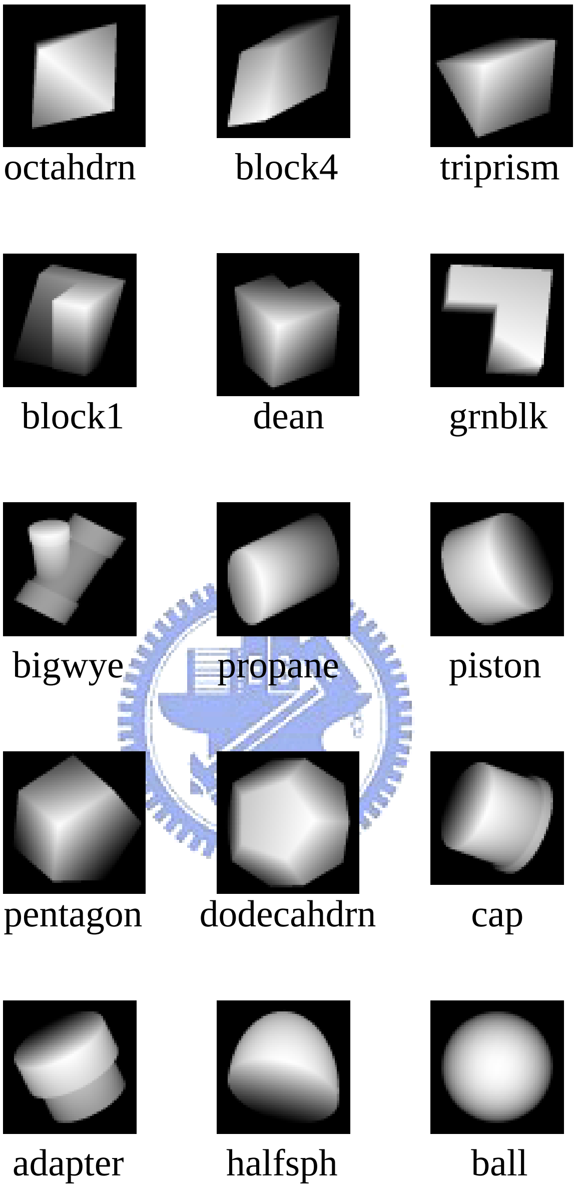


Figure 6.5: Images of some object models.

CPU times spent on a Sun4u Sparc Ultra-1 workstation for the best matches shown in Figs. 6.6(a) - 6.6(d) are equal to 7.6, 32.0, 38.2, and 34.7 seconds, respectively. (For these relatively simple shape templates and input data, an execution of the matching process presented in Sec. 3.2 takes less than 45 seconds.) Note that for each pair of shape template and input object, the time spent in the matching is determined not only by the number of primitives representing the two but also by their shapes. For example, while more primitives are used in obtaining the best match in Fig. 6.6(b) than that in Fig. 6.6(c), the former in fact takes less time to calculate.

The proposed shape matching method is also tested for more complex situations involving curved objects. Fig. 6.7 shows the matching results for *propane* and *piston*, respectively. Because of the similarity in their shapes, the same number of configuration and size adjustments described in Sec. 3.2 are carried out in deriving the best matchs. The CPU time spent in calculating the two best matches are equal to 30.1 and 513.5 seconds, respectively. The ratio of them ($513.5/30.1 = 17.1$) is very close to the ratio of the products of the number of the vertices and the number of polygons of the two objects ($\frac{386\times384}{128\times66} = 17.5$). To further demonstrate the dependency of the computation time on the amount of input data, Fig. 6.8 shows the CPU time spent in the shape matching if different numbers of vertices of *propane* are used. It is readily observable that the former is a linear function of the latter.

In general, the input data may not be acquired with perfect conditions. For example, the data may be obtained with some noise contaminations, or part of the data may not be available due to a partial view of the object. Simulations of some of these non-ideal cases are given next.

6.3.2 Shape matching results for some non-ideal cases

Shape matching based on a partial view

To simulate the shape matching for input objects with incomplete surface descriptions, e.g., due to a partial view, it is assumed that only the vertices of a portion of the polygons of a shape model are available as input data. Fig. 6.9 shows partial object surfaces of some shape models and the corresponding shape matching results. Similar results for objects with more complex boundary descriptions are shown in Fig. 6.10. Table 6.1 shows the computation

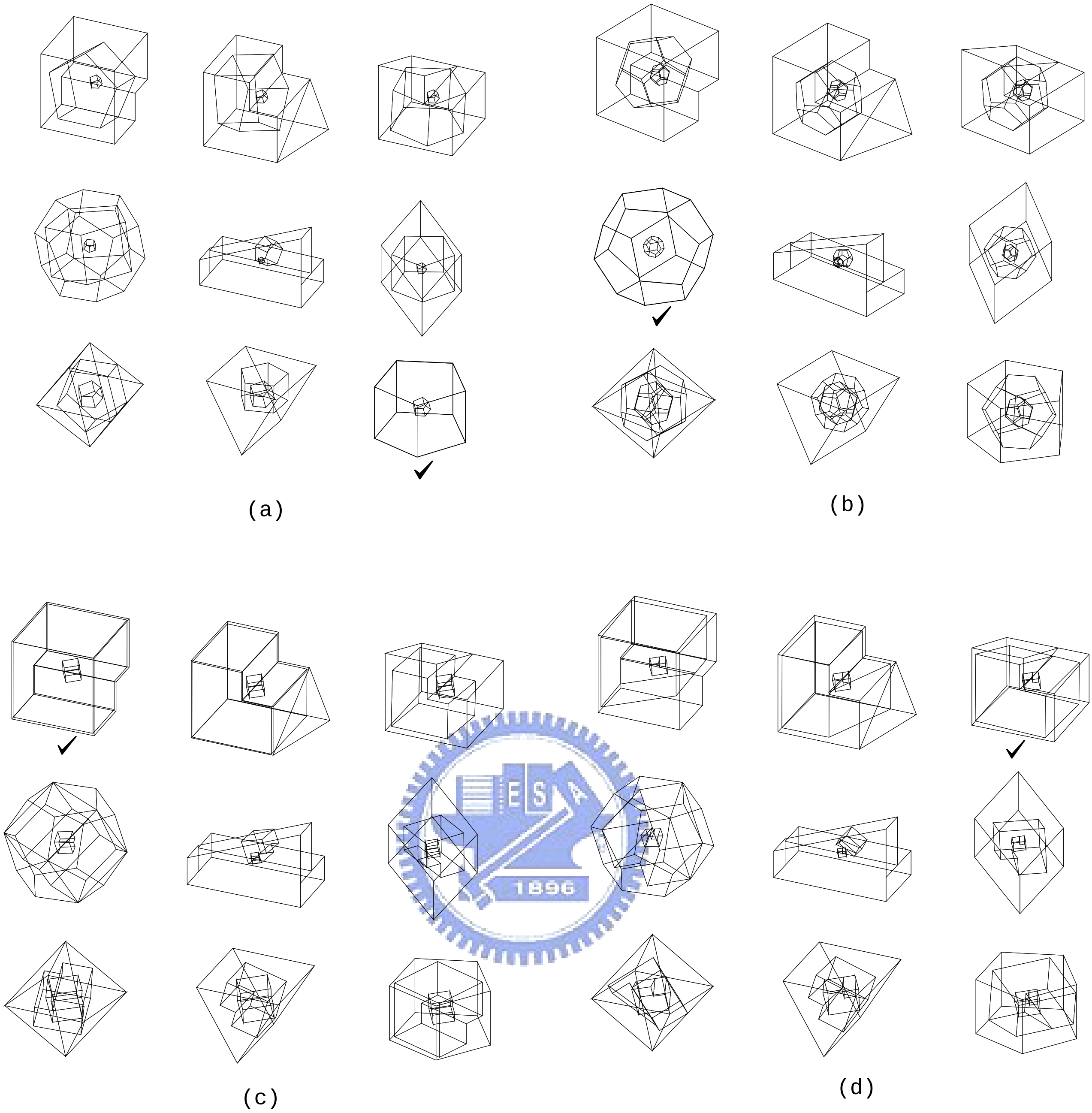


Figure 6.6: Shape matching results obtained for four input objects.

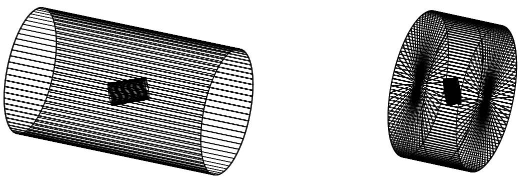


Figure 6.7: Shape matching results for *propane* (left) and *piston* (right).

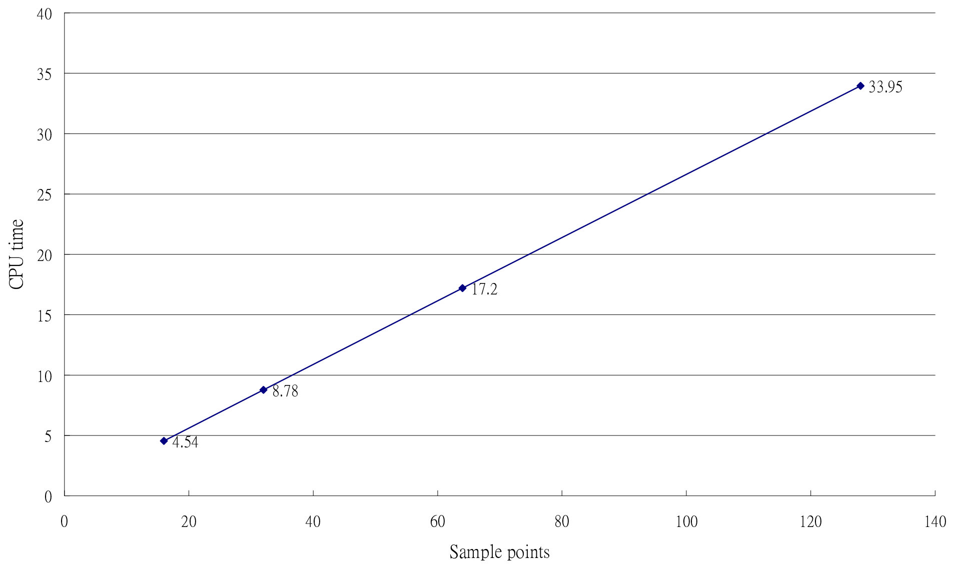


Figure 6.8: The CPU time spent in the shape matching for *propane* as a function of the amount of input data.

times as well as the amounts of different boundary elements, [3] and the percentages of these numbers with respect to those of the complete object boundary, of the input objects shown in Figs. 6.9 and 6.10.

For each of the input objects which has been considered earlier in the previous subsection for shape matching, there is a reduction in the CPU time in this case due to a reduction of the amount of boundary elements (vertices) used. However, the former is not proportional to the latter in general since the effective shape of the input object is changed.

Input data with noise contamination

To simulate the shape matching for input data with data acquisition errors, noises are added to the locations of point samples of the input object. To just demonstrate the basic idea, the noise condition is simplified by considering only the perturbation in one dimension. For each of the two cylindrical objects shown in Fig. 6.7, perturbations are introduced by adding noises uniformly distributed between ±5% and ±10%, respectively, of the coordinates of object vertices measured from its centroid along the direction of the cylinder's axis. Fig.

[3] Although vertices, not polygons, of an input object are used in shape matching, the percentage of the latter will be a good estimate of the former for an evenly and densely sampled object surface.

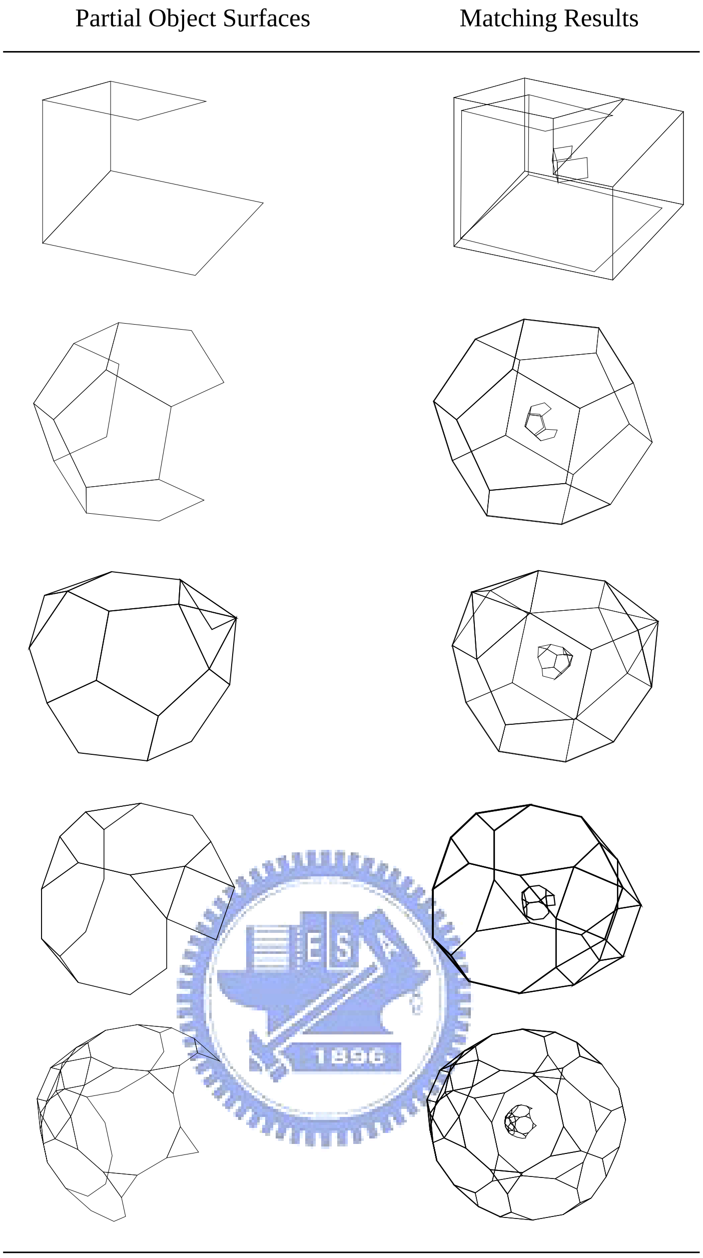


Figure 6.9: Matching results using partial views of, from top to bottom, *dean*, *dodecahdrn*, $M-104$, $M-110$, and $M-112$.

Table 6.1: Boundary elements and CPU times for examples shown in Figs. 12 & 13

	vertices/polygons(percentages)	CPU time (sec)
dean	8/3(73%/36%)	2.66
dodecahdrn	16/6(80%/50%)	2.38
M-104	17/11(77%/55%)	5.24
M-110	20/9(71%/41%)	4.91
M-112	47/26(72%/62%)	29.88
propane	97/33(76%/50%)	26.62
piston	259/192(67%/50%)	381.76

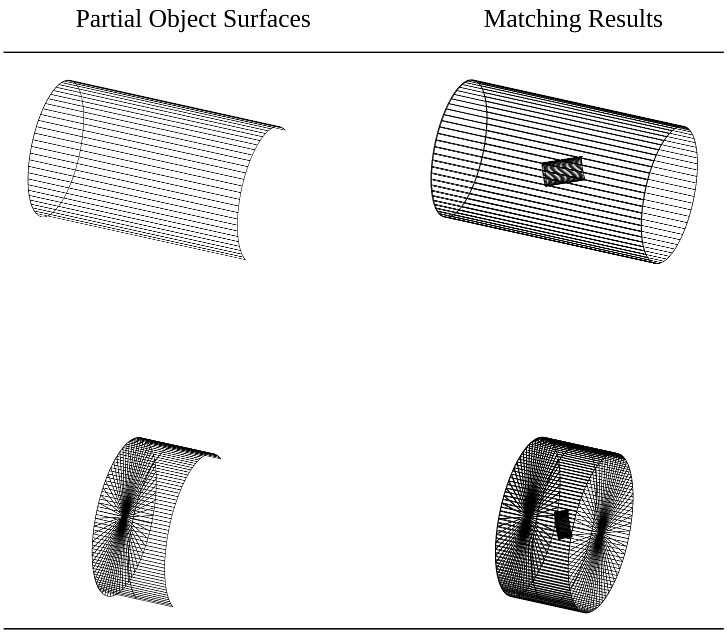


Figure 6.10: Matching results using partial views of *propane* and *piston*.

6.11 shows the shape matching results with the above additive noises. While these results are consistent to those obtained for the noise-free case, the noises result in a small drop in the final size of the input object.

Effects of sampling

Intuitively, in order to properly represent an object with point samples on its surface, it would be desirable that these samples are dense enough and have a near uniform distribution. Nevertheless, for the shape matching examples considered in the previous subsections, the vertices of the corresponding shape models, or a subset of such vertices, are used directly as simulated input data (point samples) of an object. Despite the satisfactory shape matching results presented so far, there are certain occasions when using such data will cause a problem.

Fig. 6.12(a) shows an unsuccessful shape matching example for a cap-shaped object whose boundary consists of two cylindrical surfaces of different cross-sections, the rim connecting them, and two circular bases. While there are 128 vertices along each circumference of the two circular bases, there are 383 vertices located at the rim. Due to such a highly uneven distribution of point samples, the matching process yields a final, minimal potential, configuration of the input object which moves the above (heavily sampled) rim near the central region of the shape template, and in turn pushes one of the bases against the template's

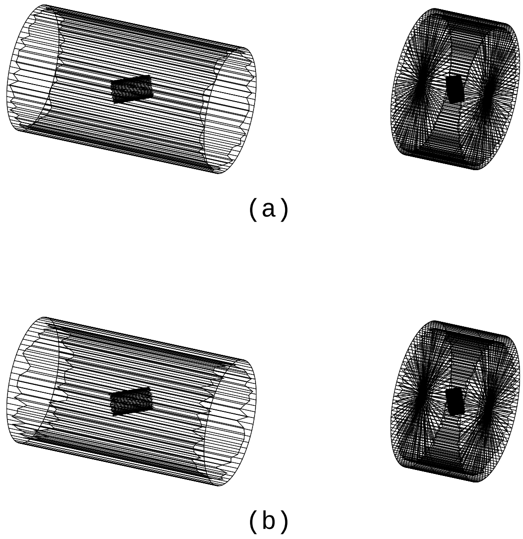


Figure 6.11: Matching results with (a) 5% and (b) 10% noise contamination in the input data.

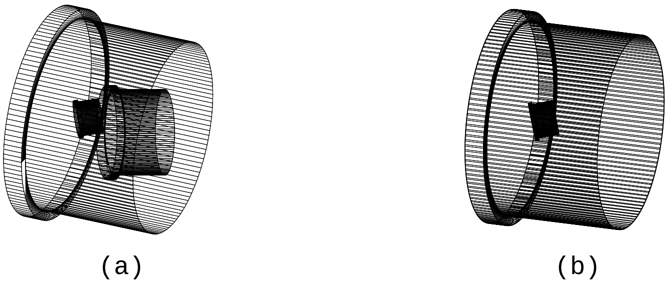


Figure 6.12: (a) An unsuccessful shape matching result due to a highly uneven distribution of input data. (b) The result obtained by using a subset of the input data.

boundary. [4]

To examine the effect of sampling further on an empirical basis, the following simple adjustment to the above problem is considered: only a subset of the vertices, i.e., the 256 vertices on the circumferences of the two bases, are used as input data. Fig. 6.12(b) shows the shape matching result thus obtained. One can see easily that the aforementioned problem is not presented here.

Discussions

In the previous subsections, the proposed shape matching method is shown to work for certain non-ideal cases. However, it is easy to see that, like other shape matching approaches, such an approach will fail to work at some point as the situation gets worse. For example, depending on the viewing angle or the severity of an occlusion, input data obtained from partial object surfaces may not convey enough information regarding the object shape that the shape matching process will generate an erroneous result. Similar arguments also apply to the noise condition.

On the other hand, it is possible for the proposed approach to use raw data of an input object, in form of object points, without first establishing a structural relationship among the data points if the points have a, dense or sparse, fairly uniform distribution and cover enough shape features of the object.

For input data not possessing such desirable attributes, the processes of (i) fitting (polyhedral) surfaces to the raw data, and (ii) obtaining near uniformly distributed point samples from these surfaces may be necessary before desirable object samples can be obtained.[5]

Such issues have been considered by several researchers, e.g., those in [48].

Fig. 6.13(a) shows a 240 × 240 range image of *adapter*. Since very dense (about 40,000) point samples are obtained for the visible surfaces of the object, the above (i) and (ii) are not performed. Instead, in order to reduce the computation complexity, the set of samples are downsized by randomly selecting about 300 of them. Figs. 6.13(b) and (c) illustrate the top and side views of the shape matching result, respectively, using these subset of samples directly. Fig. 6.13(d) shows the matching result using the range image of the object.

[4] While the vertices are uniformly distributed along each of the above circumferences, this is not true for the 383 vertices at the rim. Thus, a tilt can be observed from the shape matching result.

[5] Ideally, the process should perform a ”coarse feature extraction” to get ”meaningful” samples. minimization of the potential is not noticeable.

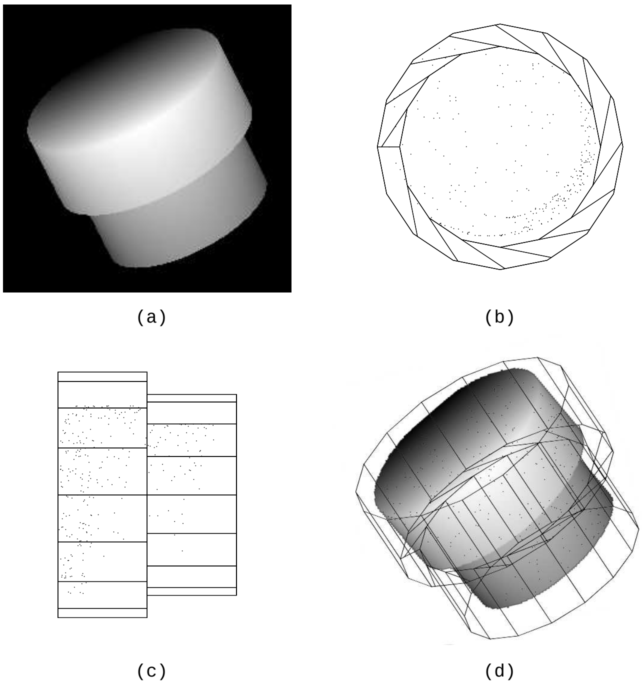


Figure 6.13: Shape matching result for range data (see text).

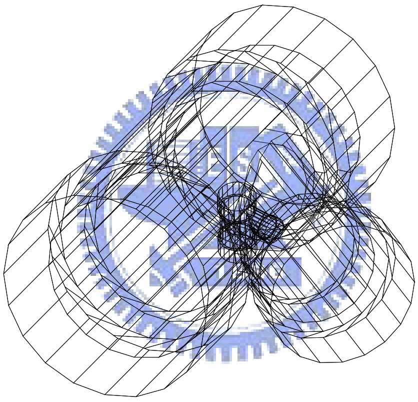


Figure 6.14: Another unsuccessful shape matching result.

There are certain types of objects for which the proposed isotropic scaling scheme will not perform perfectly. For example, for *bigwye* shown in Fig. 6.5, an input object of identical shape will not grow fully after the matching process, as shown in Fig. 6.14. (As a simpler 2-D example of such a problem, consider the process of growing an size-reduced I-shaped object in the corresponding 2-D shape model.)

More involved scaling scheme will need to be developed to overcome such a problem.

6.4 Summary

A novel approach of shape matching and recognition of 3-D objects using generalized potential fields is proposed in this thesis. The potential-based model assumes that 3-D object boundaries are uniformly charged. An initially small input object placed inside a template object can translate and reorient itself to reduce the repulsive potential. If the input object is allowed to increase its size in template objects of different shapes, the template object which allows the largest final size of the input object will correspond to the best match. Such an approach is intrinsically invariant under translation, rotation and size changes of the input object.

For an input object represented by its boundary samples, the above process of potential reduction can be carried out efficiently since the resultant potential gradients, in forms of repulsive force and torque exerted on the object, are analytically tractable. Simulation results show that the potential-based shape matching scheme has the time complexity as a linear function of the boundary samples and works to some extent for a partial view of an object. Moreover, due to the nature of the potential model, such a scheme performs satisfactorily against noise contamination in the input data. The proposed shape matching scheme is not based on any hypothesis of feature correspondence. Therefore, feature extraction for a structured object representation is not needed.

Chapter 7

Conclusions and Future Works

In this thesis, we propose a potential-based algorithm for the path planning of a robot manipulator in 2-D environment. The proposed algorithm uses an artificial potential field to model the workspace wherein object boundaries are assumed to be charged with various source distributions. Simulation results show that a path thus derived is always spatially smooth while its safety level can be adjusted easily by selecting proper charge distributions for the potential-based workspace model. Since the proposed approach uses workspace information directly, it is readily applicable to manipulators of high DOFs. According to the simulations, manipulators with DOFs up to 9 have been tested with satisfactory results.

An extended potential-based algorithm for a robot manipulator is proposed to solve path planning problems of 3-D workspace. The approach is similar to that done in 2-D manipulator, except the DOFs of joints and adopted potential model. In a 3-D environment, to give a general direction of the path, a sequence of GPs instead of guide lines (GLs) to be reached by the manipulator are assumed to be given in advance in the workspace. As a GP is an intermediate or final goal for the end-effector of a manipulator to reach, it also helps to establish certain motion constraints for adjusting manipulator configuration during path planning. According to such constraints, the proposed approach derives the path for a manipulator by adjusting its configurations at different locations along the path to minimize the potential using the above force and torque. Simulations show that a path thus derived is always spatially smooth and effective. While only the static environments are considered in this thesis, the proposed approach can be extended to workspace of moving obstacles with essentially no change in the path planning algorithm.

Furthermore, a potential-based path planning is also proposed for articulated robots with moving bases. The proposed approach is similar to that done in 3-D manipulator.

According to the adopted potential model, surfaces of workspace obstacles are assumed to be uniformly charged and the links of the articulated robot are represented as a set of charged sampling points. The repulsive force and torque between links and obstacles thus modeled are analytically tractable. While the base of a manipulator is fixed, the base of an articulated robot is movable. Since the base of an articulated robot is not fixed, the force exerted on a link due to the torque of other links does not longer exist. Thus, a sequential planning strategy is adopted to speedup the computation. Simulation results show that a path thus derived always stays away from obstacles and is spatially smooth.

Finally, a novel approach of shape matching and recognition of 3-D objects using generalized potential fields is also proposed in this thesis. The potential-based model assumes that 3-D object boundaries are uniformly charged. An initially small input object placed inside a template object can translate and reorient itself to reduce the repulsive potential. If the input object is allowed to increase its size in template objects of different shapes, the template object which allows the largest final size of the input object will correspond to the best match. Such an approach is intrinsically invariant under translation, rotation and size changes of the input object. For an input object represented by its boundary samples, the above process of potential reduction can be carried out efficiently since the resultant potential gradients, in forms of repulsive force and torque exerted on the object, are analytically tractable. Simulation results show that the potential-based shape matching scheme has the time complexity as a linear function of the boundary samples and works to some extent for a partial view of an object. Due to the nature of the potential model, such a scheme performs satisfactorily against noise contamination in the input data. Moreover, the approach is also practicable to objects with incomplete surface descriptions, e.g., due to a partial view.

Appendix A

A Potential-Based Generalized Cylinder Representation

The generalized cylinder (GC)-based shape representation is a popular representation for 3-D objects. A GC is a solid defined by its axis, a cross-section curve, and a scaling function. The solid is obtained by sweeping a planar region, its cross-section, as the region is moved and deformed according to the scaling function along a space curve, its axis. These cross-sections are also good candidates of initial GPs for path planning discussed in Chapter 4 and 5.

A.1 GC Axis Generation

Since the GC axis usually lies in the middle of an object, which is in a way similar to an object skeleton, the approach proposed in [62] adopts the potential-based skeletonization algorithm presented in [18], as reviewed next, to obtain the GC axis. The potential-based skeletonization algorithm identifies the potential valleys connecting several input seed points, which are selected as the convex vertices of the object (i.e., end points of the derived object skeleton), to obtain the object skeleton. The algorithm for generating the GC axis is essentially the same as the skeletonization algorithm except that the input seeds are selected as the centroids of two cross-sections of the object instead the convex vertices. With the boundary of a polyhedron charged according to the generalized potential model described in the previous section, the basic requirement for skeletonization, i.e., the potential will diverge at boundaries of polyhedral objects, is guaranteed. Inside the object boundary, the potential valley corresponds to 3D locations where the potential has locally minimal value in the 2D subspace perpendicular to the direction of the potential gradient. The skeletonization approach proposed in [18] obtained the object skeleton by, starting from selected seed points, traversing the corresponding potential valleys until a potential minimum is reached. The basic computation of the MAT skeleton, starting with one of the seed point, is described in the following algorithm :

Algorithm Potential_Skeleton

Step 1: Follow the direction of the force to traverse the skeleton (potential valley) until a zero force is obtained, i.e., a potential minimum is reached.

Step 2: Repeat Step 1 for each of the seed points.

Step 3: End the skeleton computation if there is only one potential minimum.

Step 4: Derive additional skeleton branches by identifying potential valleys connecting neighboring potential minima.

In Step 1, the force following is performed inside the polyhedral boundary. For a closed polyhedron, there exists at least one potential minimum since the potential and thus the force field will diverge at its surface [1]. If the number of potential minima obtained from the above force following procedure is equal to one, the complete skeleton is generated and Step 4 is not necessary. Otherwise, the potential minima need to be connected to form the complete MAT skeleton. To estimate the potential skeleton between these potential minima, we first connect each pair of neighboring minima with a virtual line segment[1]. Equally spaced point samples are then selected along the line segment as initial guesses of point samples along the associated skeleton branch. The potential valley between the two minima is then estimated by searching the minimum potential location in the 2D subspace perpendicular to the virtual line segment at each of these samples.

Fig. A.1(a) shows the skeleton of a rectangular hexahedron obtained with *Potential_Skeleton*. The seed points, where the force following procedure of the algorithm start, are selected as the eight convex corners of the object. (In the computer implementation, the selected seed points are located inside the object boundary and are slightly away from those convex corners.) The derived skeleton is closely related to the media surface derived

[1] In some complex cases where the line segment intersects the object boundary, a more refined estimation based on the visibility graph method is necessary. See [18] for more detailed discussion.

in [63], as discussed in [18]. Obviously, the derived skeleton differs from the GC axis which is defined in Sec. 1 as the principal axis in reflecting the topology of an object. However, if we select the seed points as the centroid points of the two ends (cross-sections) of the object instead of the eight corners, a GC axis will be obtained directly with the same algorithm, as shown in Fig. A.1(b)[2].

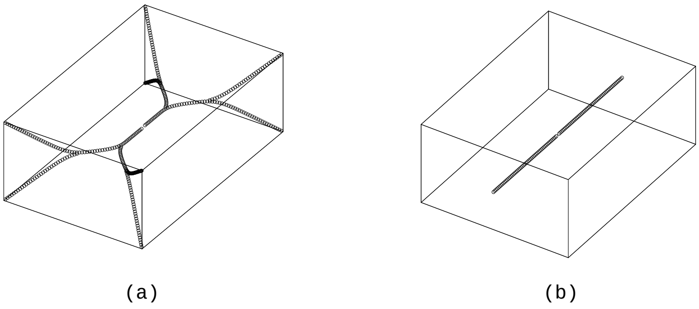


Figure A.1: (a) A rectangular hexahedron and its MAT skeleton. (b) A GC axis of the rectangular hexahedron.

The main difference between the GC axis and the object skeleton in describing the object shape can be stated most conveniently by considering prismatic objects. While the axis and the cross-section of GC will simply be the principal axis and the cross-section of a prism, respectively, the object skeleton obtained with the proposed approach are more complex near the two ends of the prism where the topology of the skeleton is determined through the selection of seeds (which are convex vertices of the cross-sections in our case). In general, for an object consists of elongated parts, both representations will reflect similar topological (skeletal) properties of each part of the object, except for the above differences, and in turn those of the whole object.

In addition, different from the object skeleton, the GC representation is not unique in general. For example, the GC axis can have either horizontal or vertical direction for a rectangle in the 2D space. As shown in Fig. A.2(a), we may select points X and Y as two seeds and generate the GC axis $\overline{XY}$. Similarly, points M and N can also be selected as seeds to generate the GC axis in a different direction, as shown in Fig. A.2(b). Figs. A.2(c) and (d) show the GC cross-sections corresponding to $\overline{XY}$ and $\overline{MN}$, respectively. In Sec. A.2.1, some suggestions for seed point selection will be given from the observation on the different GC axes and cross-sections.

[2] Theoretically, the MAT skeleton and GC axis are both continuous curves, however, they are searched step by step in our implementation and then yield a discrete form of MAT skeleton and GC axis.

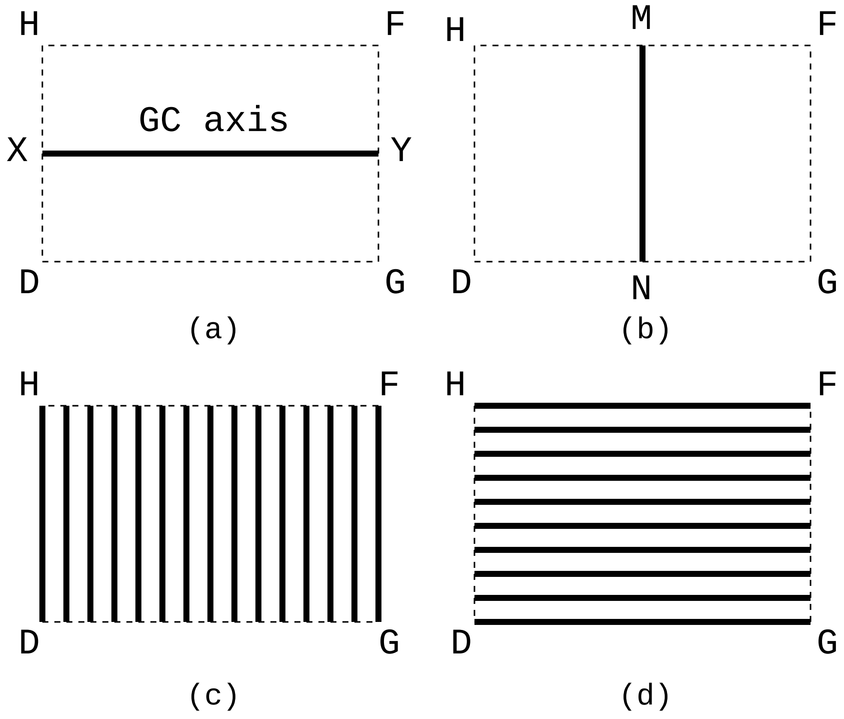


Figure A.2: The GC axis and cross-sections of a rectangle in the 2D space.

A.2 Cross-Sections Generation

The algorithm for obtaining the cross-sections of a generalized cylinder is based on the definition proposed by Binford : a cross-section of an object is called a cross-section of a generalized cylinder if its surface normal is parallel to the GC axis. For example, polygon $ABCD$ shown in Fig. A.3 is one of the GC cross-sections of the rectangular solid since the normal of $ABCD$ is parallel to the GC axis.

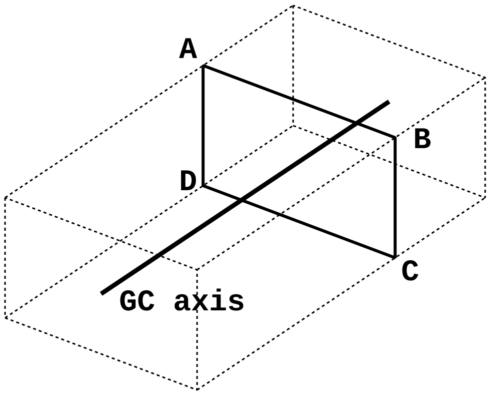


Figure A.3: One GC cross-section $ABCD$ of a rectangular solid.

Given object surface and the derived GC axis, a simple procedure for obtaining the GC cross-sections for the sequence of n sampling points S_i, $1 \leq i \leq n$, of the derived GC axis can be summarized as follows.

Algorithm Cross_Section_Gen

```
Variables: $S_i$ Sampling Point $i$; $P_i$ Plane $i$; $n$ number of sampling points;
Step 1: Set $i = 1$.
Step 2: Find the plane $P_i$ which is perpendicular to $\overrightarrow{S_i S_{i+1}}$ at $S_i$.
Step 3: Compute all intersection points and line segments of $P_i$ and the object surfaces.
Step 4: Connect these intersections sequentially to form a polygon, i.e., one cross-section
        of the object.
Step 5: $i \leftarrow i + 1$, repeat Step 3 if $i \leq n$.
Step 6: End of the algorithm.
```

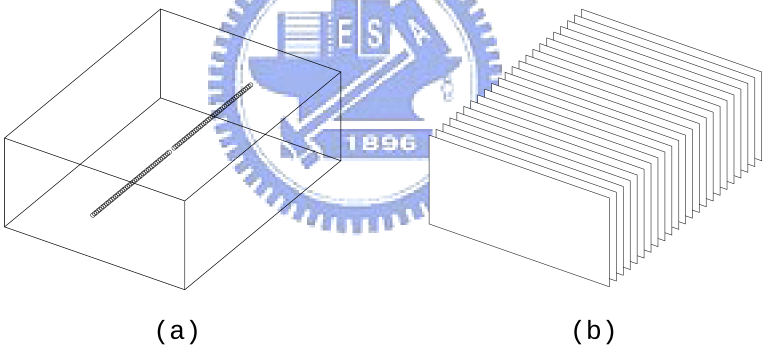


Figure A.4: GC representation of rectangular hexahedron: (a) GC axis, (b) cross-sections.

Fig. A.4(a) shows the GC axis of a rectangular hexahedron obtained by using *Potential_Skeleton* in Phase 1. The two centroids of the two ends of the object are selected as seeds. Based on the derived GC axis, a complete shape description of the object is then obtained in Phase 2 using *Cross_Section_Gen*, as shown in Fig. A.4(b). Obviously, the selection of seed points, will greatly affect the result of the derived GC axis and cross-sections. A more detailed discussion about this issue will be given in the next subsection.

A.2.1 Selection of Seed Points

In this subsection, we will discuss the influence of seed point selection on the derived GC axis and cross-sections. Consider the rectangular hexahedron shown in Fig. A.5. The MAT skeleton of the hexahedron obtained by *Potential_Skeleton* using eight convex corners as seed points is shown in Fig. A.5(a). As for the GC axis of the object, Fig. A.5(b) shows the axis obtained using two of the eight corner points as seeds whereas centroids of the front and back end surfaces of the object are selected for the axis shown in Fig. A.5(c). The corresponding cross-sections obtained by *Cross_Section_Gen* for Figs. A.5(b) and (c) are shown in Figs. A.5(d) and (e), respectively.

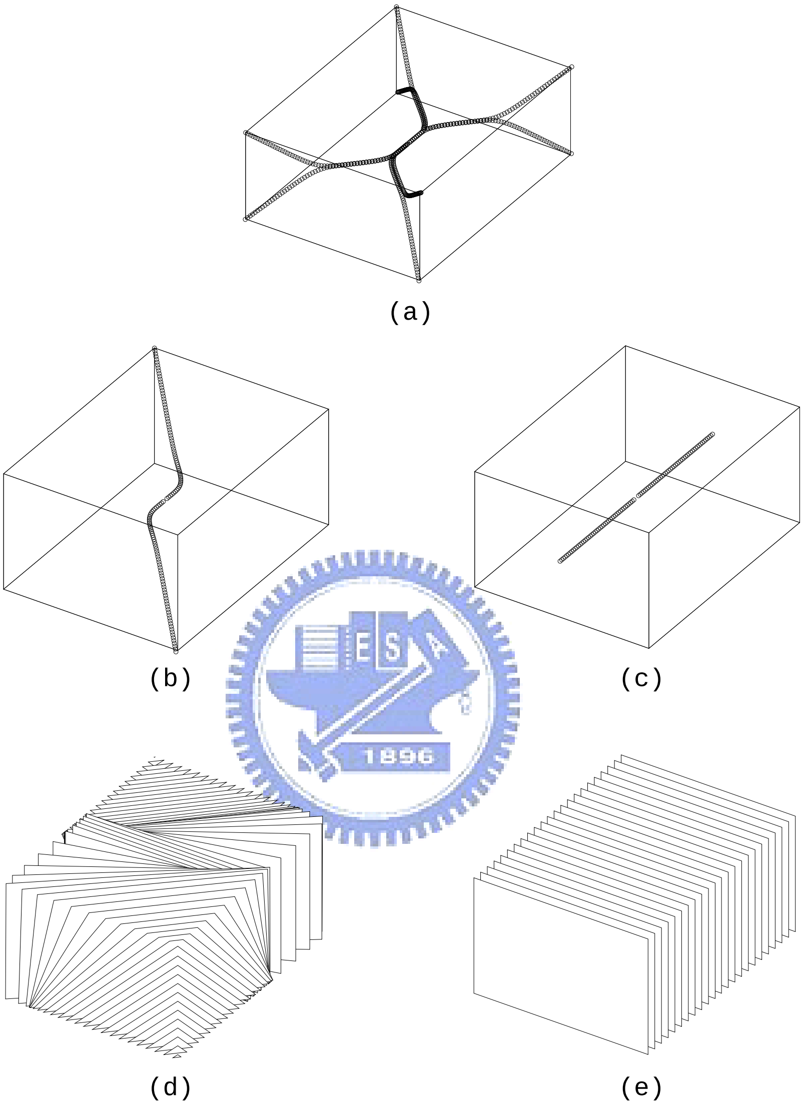


Figure A.5: The MAT skeleton and different GC representations of a rectangular hexahedron (see text).

Obviously, the cross-sections shown in Fig. A.5(d) are too complex due to the large curvature of the GC axis at two locations, as shown in Fig. A.5(b). On the other hand, the GC axis shown in Fig. A.5(c) goes straight from the front to the back end of the object while maintaining maximum distances from the other four sides of the object. Thus, the cross-sections derived for the GC axis shown in Fig. A.5(c) are much simpler (all of identical shape and size) and more appropriate as the representation of the rectangular solid.

According to the above discussion, one can see clearly that, the selection of seed points plays a crucial role in the GC algorithm proposed in this thesis. In general, seed points should be selected such that a GC axis with smaller curvature can be generated and the corresponding cross-sections will be simpler and neater for representing the object. This is especially the case for elongated object with cross-sections of identical size and shape. However, a systematic way of identifying optimal seed points for an object of arbitrary shape is yet to be established.

Bibliography

[1] J.H. Chuang, "Potential-based modeling of three-dimensional workspace of the obstacle avoidance," *IEEE Trans. on Robotics and Automation*, vol. 14, no. 5, pp. 778–785, 1998.

[2] C.H. Tsai, J.S. Lee, and J.H. Chuang, "Path planning of 3-D objects using a new workspace model," *IEEE Trans. on Sys., Man, Cybern., Part B: Cybernetics*, vol. 31, no. 3, pp. 405–410, Aug. 2001.

[3] T. Lozano-Perez, "Spatial planning: a configuration space approach," *IEEE Trans. on Computers*, vol. C-32, no. 2, pp. 108–120, Feb. 1983.

[4] R. Brooks and T. Lozano-Preez, "A subdivision algorithm in configuration space for findpath with rotation," *IEEE Trans. on System, Man and Cybernetics*, vol. 15, no. 2, pp. 224–233, March-April 1985.

[5] J. Barraquand and J.C. Latombe, "Robot motion planning: A distributed representation approach," *International Journal of Robotics Research*, vol. 10, no. 6, pp. 628–649, 1991.

[6] L. Kavraki and J.C. Latombe, "Randomized preprocessing of configuration space for fast path planning," *Proc. IEEE Intl. Conf. on Robotics and Automation*, pp. 2138–2139, 1994.

[7] J.C. Latombe L.E. Kavraki, P. Svestka and M.H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, Aug. 1996.

[8] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, Spring 1986.

[9] J.H. Chuang and N. Ahuja, "An analytically tractable potential field model of free space and its application in obstacle avoidance," *IEEE Trans. on Systems, Man, and Cybern., part B: Cybernetics*, vol. 28, no. 5, pp. 729–736, 1998.

[10] A. Thanailakis, P.G. Tzionas, and P.G. Tsalides, "Collision-free path planning for a diamond-shaped robot using two-dimensional cellular automata," *IEEE Trans. on Robotics and Automation*, vol. 13, pp. 237–250, 1997.

[11] Y.K. Hwang and N. Ahuja, "Gross motion planning a survey," *ACM Computation Survey*, vol. 24, no. 3, pp. 219–291, 1992.

[12] P. Khosla and R. Volpe, "Superquadric artificial potentials for obstacle avoidance and approach," *Proc. IEEE Intl. Conf. on Robotics and Automation*, pp. 1778–1784, 1988.

[13] J.H. Chuang, C.H. Tsai, W.H. Tsai, and C.Y. Yang, "Potential-based modeling of 2-d regions using nonuniform source distribution," *IEEE Trans. on Systems, Man, and Cybern., part A: Systems and Humans*, vol. 3, no. 2, pp. 197–202, 2000.

[14] K.P. Valavanis, T. Herbert, and N.C. Tsourveloudis, "Mobile robot navigation in 2-D dynamic environments using electrostatic potential fields," *IEEE Trans. on Sys., Man, Cybern., Part A: Systems and Humans*, vol. 30, no. 2, pp. 187–196, 2001.

[15] N.C. Tsourveloudis, K.P. Valavanis, and T. Herbert, "Autonomous vehicle navigation utilizing electrostatic potential fields and fuzzy logic," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 4, pp. 490–497, 2001.

[16] J.C. Latombe, *Robot Motion Planning*, MA: Kluwer, Boston, 1991.

[17] D.T. Kuan, J.C. Zamiska, and R.A. Brooks, "Natural decomposition of free space planning," *Proc. IEEE Intl. Conf. on Robotics and Automation*, pp. 168–173, 1985.

[18] J.H. Chuang, C.H. Tsai, and M.C. Ko, "Skeletonization of three-dimensional object using generalized potential field," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1241–1251, Nov. 2000.

[19] W.A. Perkins, "A model based vision system for industrial parts," *IEEE Trans. Comput.*, vol. 21, pp. 126–143, 1978.

[20] P. Rummel and W. Beutel, "Workpiece recognition and inspection by a model-based scene analysis system," *Pattern Recognition*, vol. 17, pp. 141–148, 1984.

[21] N. Ayache and O.D. Faugeras, "HYPER: A new approach for the recognition and positioning of two-dimensional objects," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 8, pp. 44–54, 1986.

[22] C.T. Zahn and R.Z. Roskies, "Fourier descriptors for plane closed curves," *IEEE Trans. Comput.*, vol. 21, pp. 269–281, 1972.

[23] E. Persoon and K.S. Fu, "Shape discrimination using Fourier descriptors," *IEEE Trans. Syst. Man Cybern.*, vol. 7, pp. 170–179, 1977.

[24] M.K. Hu, "Visual pattern recognition by moment invariants," *IEEE Trans. Inf. Theory*, vol. 8, pp. 179–187, 1962.

[25] S. Maitra, "Moment invariants," *Proc. IEEE*, vol. 67, pp. 697–699, 1979.

[26] S.S. Reddi, "Radial and angular moment invariants for image identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 3, pp. 240–242, 1981.

[27] J.G. Leu, "Computing a shape's moments from its boundary," *Pattern Recognition*, vol. 24, pp. 949–957, 1991.

[28] S.R. Dubois and F.H. Glanz, "An autoregressive model approach to two-dimensional shape classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, pp. 55–66, 1977.

[29] R.L. Kashyap and R. Chellappa, "Stochastic models for close boundary analysis: representation and reconstruction," *IEEE Trans. Inf. Theory*, vol. 27, pp. 627–637, 1981.

[30] M. Das, M.J. Paulik and N.K. Loh, "A bivariate autoregressive modeling technique for analysis and classification of planar shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, pp. 97–103, 1990.

[31] H. Freeman and L.S. Davis, "A corner finding algorithm for chain coded curves," *IEEE Trans. Comput.*, vol. 26, pp. 297–303, 1977.

[32] W.H. Tsai, and S.S. Yu, "Attributed string matching with merging for shape recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 7, pp. 453–462, 1985.

[33] M. Maes, "Polygonal shape recognition using string-matching techniques," *Pattern Recognition*, vol. 24, pp. 433–440, 1991.

[34] A. Blumenkrans, "2-dimensional object recognition using a 2-dimensional polar transform," *Pattern Recognition*, vol. 24, pp. 879–890, 1991.

[35] C.C. Chang, S.M. Hwang and D.J. Buehrer, "A shape recognition scheme based on relative distances of feature points from the centroid," *Pattern Recognition*, vol. 24, pp. 1053–1063, 1991.

[36] L.G. Shapiro and R.M. Haralic, "Structural descriptions and inexact matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 3, pp. 504–519, 1981.

[37] N. Ueda and S. Suzuki, "Learning visual models from shape contours using multiscale convex/concave structure matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, pp. 337–352, 1993.

[38] P.J. Flynn and A.K. Jain, "3-D object recognition using invariant feature indexing of interpretation tables," *Image Understanding*, vol. 55, no. 2, pp. 119–129, March 1992.

[39] P.J. Flynn, "3-D object recognition with symmetric models: symmetry extraction and encoding," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 16, no. 8, pp. 814–818, 1994.

[40] J. Mao, P.J. Flynn and A.K. Jain, "Integration of multiple feature groups and multiple views into a 3-D object recognition system," *Computer Vision and Image Understanding*, vol. 62, no. 3, pp. 309–325, 1995.

[41] P.J. Flynn and A.K. Jain, "BONSAI: 3-D object recognition using constrained search," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 13, no. 10, pp. 1066–1075, 1991.

[42] K.C. Wong, Y. Cheng and J. Kittler, "Recognition of polyhedral using triangle pair features," *IEE Proceedings-I*, vol. 140, no. 1, pp. 72–85, 1993.

[43] D.P. Huttenlocher and S. Ullman, "Recognition solid objects by alignment with an image," *Int. Journal of Computer Vision*, vol. 5, pp. 195–212, 1990.

[44] R. Horaud and H. Sossa, "Polyhedral object recognition by indexing," *Pattern Recognition*, vol. 28, no. 12, pp. 1855–1870, 1995.

[45] C.F. Olson, "Probabilistic indexing for object recognition," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 17, no. 5, pp. 518–522, 1995.

[46] K. Arbter, W.E. Snyder, H. Burkhardt and G. Hirzinger, "Application of affine-invariant fourier descriptors to recognition of 3-D objects," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 12, no. 7, pp. 640–647, 1990.

[47] A. Pentland and S. Sclaroff, "Closed-form solutions for physically based shape modeling and recognition," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 13, no. 7, pp. 715–729, 1991.

[48] Y. Chen and G. Medioni, "Surface description of complex objects from multiple range images," *Proc. IEEE Conf. CVPR*, pp. 153–158, 1994.

[49] K. Wu and M.D. Levine, "3-D parts segmentation using simulated electrical charge distributions," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 19, no. 11, pp. 1223–1235, 1997.

[50] J.H. Chuang, "A potential-based approach for shape matching and recognition," *Pattern Recognition*, vol. 29, no. 3, pp. 463–470, 1996.

[51] D.R. Wilton, S.M. Rao, A.W. Glisson, D.H. Schaubert, O.M. Al-Bundak, and C.M. Butler, "Potential integrals for uniform and linear source distributions on polygonal and polyhedral domains," *IEEE Trans. Antennas and Propagation*, vol. AP-32, no. 3, pp. 276–281, 1984.

[52] L. Bers and F. Karal, *Calculus*, New York: Holt, Rinehart and Winston, New York, 1976.

[53] J.C. Latombe, "Motion planning: A journey of robots, molecules, digital actors, and other artifacts," *International Journal of Robotics Research*, vol. 18, no. 11, pp. 1119–1128, Nov. 1999.

[54] T. Lozano-Perez and M.A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Communications. ACM*, vol. 22, no. 10, pp. 560–570, 1979.

[55] T. Laliberte and C. Gosselin, “Efficient algorithms for the trajectory planning for redundant manipulators with obstacle avoidance,” *Proc. IEEE Intl. Conf. on Robotics and Automation*, pp. 2044–2049, 1994.

[56] R. Nevatia and T.O. Binford, “Description and recognition of curved objects,” *Artificial Intelligence*, vol. 8, no. 1, pp. 77–98, 1977.

[57] C.I. Connolly and R.A. Grupen, “Path planning using laplace’s equation,” *IEEE Intl. Conf. of Robotics and Automation*, pp. 2102–2106, 1990.

[58] D.E. Koditschek, “Exact robot navigation by means of potential functions: Some topological considerations,” *IEEE Intl. Conf. of Robotics and Automation*, pp. 1–6, 1987.

[59] O. Brock and L.E. Kavraki, “Decomposition-based motion planning: A framework for real-time motion planning in high-dimensional configuration spaces,” *IEEE Intl. Conf. of Robotics and Automation*, pp. 1469–1474, 2001.

[60] K.K. Gupta, “Fast collision avoidance for manipulator arms: A sequential search strategy,” *IEEE Trans. on Robot. and Auto.*, vol. 6, no. 5, pp. 522–532, Oct. 1990.

[61] C.C. Lin, C.C. Pan, and J.H. Chuang, “A novel potential-based path planning of 3-D articulated robots with moving bases,” *Proc. IEEE International Conference on Robottic and Automation*, pp. 3365–3370, 16-18 Sep 2003, in Taipei, Taiwan, R.O.C.

[62] J.H. Chuang, N. Ahuj, C.C. Lin, C.H. Tsai, and C.H. Chen, “A potential-based generalized cylinder representation,” *accepted by Computers and Graphics*, 2004.

[63] E.C. Sherbrooke, N.M. Patrikalakis, and E. Brisson, “An algorithm for the medial axis transform of 3-D polyhedral solids,” *IEEE Trans. Visualization and Computrer Graphics*, vol. 2, no. 1, pp. 44–61, Mar. 1996.