Perceived CT-Space for Motion Planning in Unknown and Unpredictable Environments *

Rayomand Vatcha and Jing Xiao

Department of Computer Science, University of North Carolina - Charlotte
{rvatcha, xiao}@uncc.edu

Abstract: One of the ultimate goals in robotics is to make high-DOF robots work autonomously in unknown changing environments. However, motion planning in completely unknown environments is largely an open problem and poses many challenges. One challenge is that in such an environment, the configuration-time space (CT-space) of a robot is not known beforehand. This paper describes how guaranteed collision-free regions in the unknown CT-space can be discovered progressively via sensing in real time based on the concept *dynamic envelope*, which is not conservative, i.e., does not assume worst-case scenarios, and is robust to uncertainties in obstacle behaviors. The introduced method can be used in general by real-time motion planners for high-DOF robots to discover the existence of guaranteed collision-free future motions efficiently. The utility is further confirmed both in simulation and in real-world testing involving a 5-DOF robot manipulator.

1 Introduction

Most of the existing work addresses robot motion planning in *known* environments, which can be catogorized into the following:

1. Path planning for a robot in a static and known environment to search a collision-free path in the (static) configuration space (C-space) [1] of the robot. Approaches include finding collision-free regions or free space in the C-space [2][3] and sampling-based planners to deal with high-dimensional C-space for robots of high degrees of freedom (DOF)[4][5].
2. Motion planning for a robot in a known dynamic environment to search a trajectory in the CT-space [2] of the robot. Again, sampling-based planners were used here [5] to avoid constructing high-dimensional CT-obstacles. For a mobile robot, the notion of "Inevitable Collision Regions" (ICS) in the CT-space was introduced [6] essentially to characterize the CT-obstacles.

* This work is supported by the National Science Foundation Grant IIS-0742610.

In both cases planning can be done off-line without the need of sensing (sensing is only used to deal with uncertainities during actual execution of motion). Collision detection is usually the most time consuming component of any sampling-based planner. Its complexity increases as the complexity of the robot and environment increases.

An extension to the above basic problems is through adding some obstacles of unknown motion into the largely known environment. This is mostly addressed by on-line revising pre-planned paths with reactive schemes to avoid collisions (e.g., [7][8][9]). These schemes usually assume partially unchanging/known C-space or CT-space to limit the scale of revising/replanning in order to facilitate real-time computation.

A further extension is motion planning in drastically changing environments with unknown obstacle motions. A real-time adaptive planning approach [10][11] for high-DOF robots is very effective, characterized by simultaneous planning and execution based on sensing. However, the approach assumes known obstacles with unknown motions. Thus, planning future motion is based on predicting obstacle motion through tracking and frequently updating the predictions.

Note that in those approaches, unknown changes in an environment are dealt with by repeated computation or recomputation of (some parts of) paths/trajectories, which involve repeated collision checking.

Another extension is motion planning in unknown but static environments. No information about the obstacles is known. Such a problem needs active sensing of the environment. One approach represents an environment in terms of voxels so that obstacle geometry need not be known [12]. Sensing is used to discover which voxels are occupied by obstacles. Such sensor based motion planners [13][14] are often adapted from model based planners to plan paths incrementally, as unknown C-Space becomes known gradually by sensing.

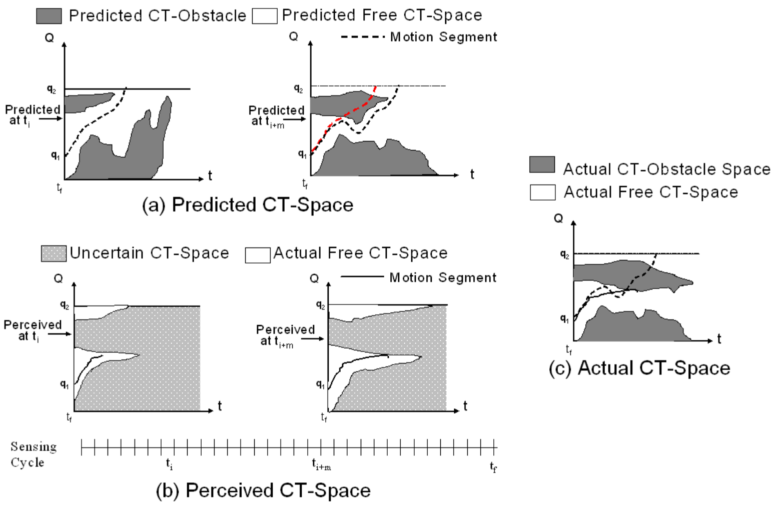
A largely open problem is motion planning in completely unknown environments, where obstacle geometries and states are unknown, i.e., if and when they move or not is not known. In other words, the CT-space of a robot is completely unknown. The existing approaches to deal with known obstacles of unknown motions are not suitable here as obstacles cannot be distinguished, and thus their motions cannot be tracked and predicted.

This paper addresses this open problem. We present a novel approach to discover guaranteed collision-free regions in the unknown and unpredictable CT-space in real-time via sensing. We show how the approach can be used for real-time motion planning in such an environment and test it through simulation and experiments.

2 Perceived CT-space vs. Predicted CT-space

We are interested in discovering true collision-free regions, or free space, in the CT-space of a robot in an unknown and unpredictable environment (i.e., obstacles are not known and whether and how they move are not known).

This is different from the existing approaches that predict the CT-space to deal with obstacles of unknown motion. Such approaches predict an obstacle's motion mostly by tracking its location or assuming constant velocity within a planning period. Motion planning is then done in the *predicted CT-space* of a robot, involving collision checking of the robot's configuration at any future time against the predicted obstacle configurations at the same time instant. However, if obstacles themselves are unknown, they cannot be detected or tracked so that prediction-based approaches are not suitable.



(a) Predicted CT-Space

(b) Perceived CT-Space

(c) Actual CT-Space

Fig. 1. Predicted CT-space vs. Perceived CT-Space

Moreover, predicted CT-space is often not the true CT-space and only matches closely to it within a short period immediately after the time when the prediction is made. It requires repeated modification as new sensing information becomes available. Hence, motion planning based on prediction will lead to re-computation of motions, and the planned motions may fail to be collision-free due to inaccurately predicted CT-space. It is too conservative to assume worst-case obstacle motion in order to have guaranteed collision-free motions of the robot with the CT-obstacle space exaggerated.

We use the term *perceived CT-space* to call the CT-space discovered by our approach via sensing, which includes actual (i.e., guaranteed) collision-free regions discovered that will not turn false later as sensing continues. Therefore, it is not the same as a predicted CT-space. Figure 1 illustrates the difference between the two. It compares predicted vs. perceived vs. actual CT-space in a 2-D example. Both predicted CT-space and perceived CT-space will change as sensing/time progresses. However, unlike predicted CT-space, where a point predicted collision-free may not be actually collision-free, the perceived CT-space consists of *actual* collision-free regions that can only grow over time and *uncertain* regions, which can be either free or CT-obstacle regions.

3 Perceived CT-Space

We now show how the actual CT-free space can be perceived over time.

3.1 Atomic obstacles

First of all, with obstacles completely unknown in an environment, it will be difficult to try to distinguish different geometric obstacles from sensing. For static environments, this problem can be addressed by, for example, representing an environment in terms of voxels without knowing obstacle geometry [12]. However, if the environment is changing *drastically*, this is computationally costly as the entire set of voxels are only valid for the current sensing interval and have to be re-computed at each subsequent sensing interval. Hence, it makes sense to use the lower-level data from sensors *directly* to represent obstacles without ever performing elaborate sensor information processing. Without the loss of generality, the lower-level sensory data for obstacles can be treated as *atomic obstacles* of similar and simple geometry at different locations. Collectively the atomic obstacles represent actual obstacles in an environment without distinguishing them (see Section 4). At each sensing instant, only the locations of atomic obstacles (with default geometry) can be sensed. However, we can put an upper bound on the changing rate of the environment in terms of a maximum possible speed v_{max} of each atomic obstacle. Of course, an atomic obstacle may have varied actual speeds in $[0, v_{max}]$.

3.2 Dynamic envelope

We are interested in discovering collision-free regions via sensing in real-time in the otherwise unknown CT-space for a general (high-DOF) robot. Assume sensing data are obtained/updated at discrete times starting at $t = 0$. For any CT-point $(\mathbf{q}, t)$, we ask the **question**: *will the robot be guaranteed collision-free at* $(\mathbf{q}, t)$ *based on the sensed information at the current time* t_i?

We can answer this question using the concept of *dynamic envelope.*

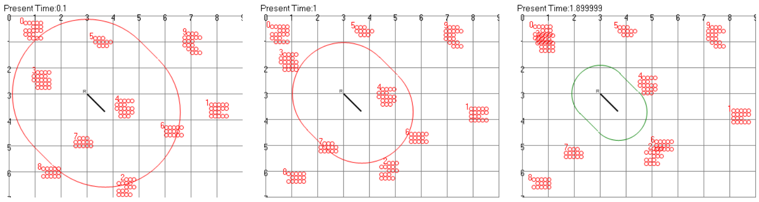
Definition 1: For a CT-point $\chi = (\mathbf{q}, t)$, a *dynamic envelope* $E_{t_o}(\chi, t_i)$, as a function of current time $t_i \leq t$, is a closed surface enclosing the region occupied by the robot at configuration $\mathbf{q}$ in the physical space so that the minimum distance between any point on $E_{t_o}(\chi, t_i)$ and the region is $d_i = v_{max}(t - t_i)$, where $t_o \leq t_i$ is the time when the envelope was created. $t - t_o$ is the maximum *lifespan* of $E_{t_o}(\chi, t_i)$.

The following are general properties of a dynamic envelope $E_{t_o}(\chi, t_i)$. They capture non-worst case scenarios regarding atomic obstacle motions, *without* assuming any particular kinds of obstacle motion.

1. A dynamic envelope shrinks monotonically over sensing time with speed v_{max}, i.e., $E_{t_o}(\chi, t_{i+m}) \subset E_{t_o}(\chi, t_i)$, where $m > 0$, $t_o \leq t_i < t_{i+m} \leq t$.
2. An atomic obstacle not on or inside $E_{t_o}(\chi, t_o)$ will never be on or inside $E_{t_o}(\chi, t_i)$.
3. An atomic obstacle on $E_{t_o}(\chi, t_o)$ will never be inside $E_{t_o}(\chi, t_i)$.
4. An atomic obstacle either on or inside $E_{t_o}(\chi, t_o)$ can be outside $E_{t_o}(\chi, t_i)$, for certain t_i, if not moving *towards* the robot in maximum speed v_{max} *all the time*, i.e., if not moving in the worst case.

From these properties, one can envision the following: suppose some atomic obstacles are on or inside a dynamic envelope $E_{t_o}(\chi, t_i)$ initially; as the dynamic envelope shrinks during its maximum lifespan, no new atomic obstacles will ever enter the dynamic envelope, and the atomic obstacles initially on or inside it will be "squeezed" out of the envelope at some later time during the lifespan *if these atomic obstacles do not always move towards the robot configuration* $\mathbf{q}$ *in* v_{max}*, i.e., under non-worst case scenarios.* Hence, at time t_i, if no atomic obstacle is on or inside the dynamic envelope $E_{t_o}(\chi, t_i)$, $\chi = (\mathbf{q}, t)$ is guaranteed collision-free, i.e., the above **question** is answered. Moreover, all the continuous configuration-time points in the interval $[(\mathbf{q}, t_i), (\mathbf{q}, t)]$ are guaranteed collision-free.

Figure 2 shows an example, where $\chi = ((3, 3), 3)$, $v_{max} = 1$ unit/s.



(a) $E_{0.1}(\chi, 0.1)$ (b) $E_{0.1}(\chi, 1)$ (c) $E_{0.1}(\chi, 1.89)$

Fig. 2. Dynamic envelope of a planar rod robot. In (c), χ is percieved collision free at $t_i = 1.89$s.

In general, as soon as at some t_l, no atomic obstacle sensed is on or inside the dynamic envelope $E_{t_o}(\chi, t_l)$, the envelope is no longer needed, and it can *expire* at t_l, i.e., before its maximum lifespan is reached.

3.3 Collision-free region vs. uncertain region

We have answered in the above that a CT-point $(\mathbf{q}, t)$ can be perceived at $t_i (\leq t)$ as guaranteed collision-free and also explained that if $(\mathbf{q}, t)$ is perceived at t_i as guaranteed collision-free, the hyperline segment $[(\mathbf{q}, t_i), (\mathbf{q}, t)]$ in the CT-space is also guaranteed collision-free.

Now a natural next question is: *given a configuration* $\mathbf{q}$*, what is the longest hyperline segment* $[(\mathbf{q}, t_i), (\mathbf{q}, t)]$*, or the furthest time* t*, that can be perceived at* t_i *as guaranteed collision-free?* The answer to that question depends on the minimum distance $d_{min}(\mathbf{q}, t_i)$ between the robot (if it were) at configuration $\mathbf{q}$ and the closest atomic obstacle sensed at t_i. Let

$$\Delta t(\mathbf{q}, t_i) = \frac{d_{min}(\mathbf{q}, t_i)}{v_{max}}, \tag{1}$$

which is the minimum period before a collision can *possibly* occur at $\mathbf{q}$. Let

$$t_f(\mathbf{q}, t_i) = t_i + \Delta t(\mathbf{q}, t_i) \tag{2}$$

Clearly, as long as t is within the time interval $[t_i, t_f(\mathbf{q}, t_i))$, the hyperline segment $[(\mathbf{q}, t_i), (\mathbf{q}, t)]$ can be perceived at t_i as guaranteed collision-free. Thus, the longest hyperline segment that can be perceived at t_i as guaranteed collision-free is $[(\mathbf{q}, t_i), (\mathbf{q}, t_f(\mathbf{q}, t_i))$.

The union of all the guaranteed collision-free hyperline segments of the CT-space perceived at t_i is the maximum collision-free region (that may include multiple connected continuous regions) perceived at t_i, denoted as $F(t_i)$. $F(t_i)$ consists of only CT-points for $t \geq t_i$. The union of the rest of the regions in the CT-space for time $t \geq t_i$ forms the uncertain region $U(t_i)$.

Theorem 1: For any t_i and t_j, such that $t_i \leq t_j$, if a CT-point $(\mathbf{q}, t)$, where $t \geq t_j$, belongs to $F(t_i)$, then it also belongs to $F(t_j)$. On the other hand, if the point $(\mathbf{q}, t)$ belongs to $U(t_i)$, it may still belong to $F(t_j)$.

Proof: From t_i to t_j, the change in minimum distance at configuration $\mathbf{q}$ can be expressed as:

$$d_{min}(\mathbf{q}, t_j) - d_{min}(\mathbf{q}, t_i) = p v_{max}(t_j - t_i), \ -1 \leq p \leq 1. \tag{3}$$

From equations (1) and (2), and using equation (3), we get

$$\begin{aligned} t_f(\mathbf{q}, t_j) - t_f(\mathbf{q}, t_i) = (t_j - t_i) + \tfrac{d_{min}(\mathbf{q}, t_j) - d_{min}(\mathbf{q}, t_i)}{v_{max}} = (1+p)(t_j - t_i) \\ \Rightarrow t_f(\mathbf{q}, t_j) - t_f(\mathbf{q}, t_i) \geq 0 \end{aligned}$$

That is, if $(\mathbf{q}, t)$ is on the hyperline $[t_i, t_f(\mathbf{q}, t_i))$, then, since $t \geq t_j$, it is also on the hyperline $[t_j, t_f(\mathbf{q}, t_j))$. On the other hand, if $(\mathbf{q}, t)$ belongs to $U(t_i)$, then $t \geq t_f(\mathbf{q}, t_i)$, but as long as $t < t_f(\mathbf{q}, t_j)$, $(\mathbf{q}, t)$ belongs to $F(t_j)$. ■

The significance of the above theorem is that more collision-free CT-space points can be discovered as sensing time progresses, i.e., the collision-free regions can only grow, while uncertain regions can only shrink.

4 Representing Unknown Obstacles of Unknown Geometry: An Example

There is much research on how to explore an unknown (mostly static) environment using robots with sensors mounted, and issues studied include how to move a robot to maximize sensing views (i.e., minimize occlusions) [15] and how to map an environment accurately through repeated exploration (e.g., SLAM [16]). For sensor-based robot navigation, different kinds of sensors are used either mounted in the environment to provide a world view or mounted on a robot to provide a robot-centric, local view. However, these issues regarding the arrangement and style of sensing is out of the scope of this paper. It is important to note that the concepts introduced in Section 3 are independent of any kind of sensing style.

In general, the lowest level data points of whatever sensor (e.g., laser range finders, sonar, etc.) constitute atomic obstacles (as mentioned in Section 3). As a concrete example, consider that an overhead stereovision sensor is used to provide a view of an unknown environment. The stereovison sensor provides an image of the environment. Every pixel (i, j) of that image maps to a surface region R_{ij} of 3-D points in the physical world. We can further obtain the 3-D point (x, y, z) in R_{ij} that is closest to the image plane. This mapping between 3-D point (x, y, z) and pixel (i, j) is a one-to-one mapping.

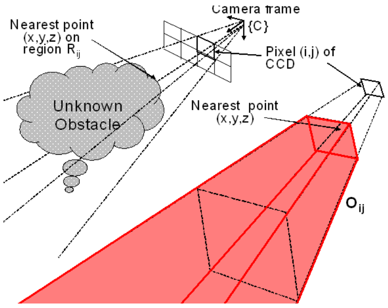


Fig. 3. An atomic obstacle O_{ij} from stereo vision

Since R_{ij} occludes the space behind it, from it one cannot tell if and how there are objects in that occluded space. Therefore, to be safe, R_{ij} and the infinite volume of points it occludes can be viewed as an atomic obstacle O_{ij} that a robot cannot collide with. O_{ij} is associated with a pixel (i, j) of the image, which starts from the point (x, y, z) extending towards infinity. It can be viewed as a trapezoidal ray originated from (x, y, z) as shown in Figure 3. The 3-D environment can now be viewed as consisting of *only* these atomic obstacles O_{ij} for all (i, j)'s in the image. Let $M \times N$ be the size of the image.

Note that since actual obstacles in the environment are entirely *unknown* and can move/change unpredictably, we cannot relate the atomic obstacles from an image taken at time t_k to those from a image taken at t_{k+1}. Thus, the low-level sensory data from t_k is only useful within that sensing interval and should be replaced entirely by the sensory data obtained from t_{k+1}. In other words, there is no need for accumulating sensory data, and the space complexity for storing sensory data is simply a constant $C = M \times N$.

Now consider the time complexity of using the stored data. Since the sensed atomic obstacles are used for perceiving collision-free or uncertain CT-points, only those atomic obstacles enclosed in a dynamic envelope (of the robot at the considered CT-point) need to be considered. For this example, atomic obstacles are *directly* indexed by (i, j) in a one-to-one mapping between a physical 3-D point (x, y, z) to a pixel (i, j). Projecting a dynamic envelope onto the image plane, we can obtain the indices (i, j) of the atomic obstacles enclosed and consider only them for collision test. Let n indicate the number of such indices, then for any CT-point $(\mathbf{q}, t)$, the time complexity of collision test is a function of n, which is usually much smaller than C. As the dynamic envelope shrinks overtime, so is the time for collision test.

Of course, as viewing directions change, the atomic obstacles as defined above change too, but that does not matter because we are not concerned here with what the actual obstacles look like. Also note that the atomic obstacles do not have to have the same size. It is important that the atomic obstacles come directly from sensory data and are of simple shapes.

5 Computing Motions in the Perceived CT-space

The concept of dynamic envelope introduced in section 3 can be used by motion planners to discover collision-free regions in the CT-space for future motions efficiently, which do *not* require re-computation or revision.

Let $\mathbf{q}_1$ and $\mathbf{q}_2$ be two configurations of a robot. Let $(\mathbf{q}_1, t_s)$ and $(\mathbf{q}_2, t_e)$ be two points in the CT-space, and let the current time be $t_o < t_s$. We are interested in finding whether a trajectory segment connecting $(\mathbf{q}_1, t_s)$ and $(\mathbf{q}_2, t_e)$, where $t_s < t_e$, is collision-free or not, based on sensing at each $t_i \in [t_o, t_s)$, $i = 1, 2, ...$ The trajectory segment can be represented by a sequence Γ of CT-points between $(\mathbf{q}_1, t_s)$ and $(\mathbf{q}_2, t_e)$ through interpolation. If the resolution for interpolation is chosen such that the maximum gap between the robot put at two consecutive CT-space points (after interpolation) is smaller than the known size of an atomic obstacle, then if the two CT-space points are guaranteed collision-free, the CT-points in between are also guaranteed collision-free. In that sense, the sequence Γ truly represents a continuous motion segment in the perceived CT-Space.

For every point $\chi_k = (\mathbf{q}_k, t^k)$ in the sequence Γ, where $t^1 = t_s$, $t^m = t_e$, and $1 \leq k \leq m$, we need to check if it is collision-free or not. This can be done by creating the dynamic envelope $E_{t_o}(\chi_k, t_o)$ at t_o for each k. Note that for a robot consisting of many links, a dynamic envelope can be built for each link with a simple shape. We next observe all $E_{t_o}(\chi_k, t_i)$'s along Γ shrink as the sensing time t_i progresses from t_o. If for every dynamic envelope $E_{t_o}(\chi_k, t_i)$, there exists a time $t_i^k \in [t_o, t_s)$ when $E_{t_o}(\chi_k, t_i^k)$ is free of atomic obstacles, then it means Γ is a guaranteed collision-free motion segment, discovered before its starting time t_s. Moreover, if the starting time of Γ is moved to t'_s, such that $\max(t_i^k) \leq t'_s < t_s$ (i.e., earlier than t_s), the shifted Γ (along the time axis) is also guaranteed collision-free. This means that the whole sweped region of Γ along the time axis from $\max(t_i^k)$ to t_s in the CT-space is now discovered to be collision-free.

Our general algorithm to check whether a CT-point $\chi = (\mathbf{q}, t)$ is collision-free or not based on the (shrinking) dynamic envelope $E_{t_o}(\chi, t_i)$, for $t_i \in [t_o, t)$, is called the *collision-free perceiver* (CFP) as shown in Algorithm 1. CFP is quite efficient for real-time operation because of the following: (1) CFP returns a boolean value and does not require (more expensive) minimum distance computation. (2) As shown in Section 4, CFP only needs to consider a subset of the atomic obstacles sensed. The number n of such atomic obstacles is related to the size of the dynamic envelope, which shrinks over time. (3) Both the atomic obstacles and the dynamic envelope are of simple shapes. A time-limit for CFP can be further imposed.

Algorithm 1 Collision-Free Perceiver (CFP)

```
Input configuration-time point $\chi = (\mathbf{q}, t)$, current time $t_o$
$i = 1$, $t_i = t_o$
Create dynamic envelope $E_{t_o}(\chi, t_i)$
while $t_i < t$ and (not time-limit) do
  if no atomic obstacle is on or inside $E_{t_o}(\chi, t_i)$ then
    $E_{t_o}(\chi, t_i)$ expires
    return $\chi \in F(t_j), \forall t_i \leq t_j \leq t$, ($\chi$ is guaranteed collision-free)
  else
    $i = i + 1$ (Next sensing moment)
  end if
end while
return $\chi$ may not be collision-free
```

The above method of computing guaranteed collision-free motion segments in CT-space using CFP can be employed by any real-time motion planner seeking collision-free motion in an unknown and unpredictable environment. As the robot moves along a perceived collision-free trajectory segment, the planner can continue finding subsequent collision-free motion segments. Since the segment currently followed by the robot is truly collision-free, the robot can safely stay on it until it needs to execute a subsequent collision-free motion segment provided by the planner. As the planner does not need to worry if the current segment being executed by the robot will become infeasible, it can solely focus on planning the next motion segment. In section 7, we will show concrete simulation and real-world examples of real-time planning using CFP.

6 Robustness of Approach over Exaggerated v_{max}

As v_{max}, the maximum speed of an atomic obstacle, is the only known or estimated parameter we assume in our approach dealing with completely unknown environment, it is necessary to investigate how robust our approach of perceiving collision-free CT-space points is with respect to very inaccurate v_{max}. Specifically, it is natural to over-estimate v_{max} to be safe, i.e., the estimated v'_{max} satisfies: $v'_{max} > v_{max}$. The effect of such over-estimation can be stated in the following theorem.

Theorem 2: Let $v'_{max} = cv_{max}, c > 1$, and let $(\mathbf{q}, t)$ be a collision-free CT-point. If t'_k and t_k are the respective time instants when $(\mathbf{q}, t)$ is perceived to be collision-free, then, they satisfy: $t_k < t'_k \leq t$.

Proof: Suppose at time t_o, the dynamic envelopes $E_{t_o}((\mathbf{q}, t), t_o)$ and $E'_{t_o}((\mathbf{q}, t), t_o)$ for ($\mathbf{q}$, t) were created with respect to v_{max} and v'_{max} respectively, where

$$d_o = v_{max}(t - t_o) \text{ , and}$$
$$d'_o = v'_{max}(t - t_o) = cv_{max}(t - t_o)$$

Clearly for any time $t_o \leq t_i < t$, $E'_{t_o}((\mathbf{q}, t), t_i)$ is greater than $E_{t_o}((\mathbf{q}, t), t_i)$. Suppose further that at least one atomic obstacle was on or inside $E_{t_o}((\mathbf{q}, t), t_o)$. Thus, it was also on or inside $E'_{t_o}((\mathbf{q}, t), t_o)$.

Suppose at time t_k, where $t_o \leq t_k \leq t$, the dynamic envelope $E_{t_o}((\mathbf{q}, t), t_k)$ has shrunk enough to just "squeeze out" atomic obstacles, i.e., the CT-point $(\mathbf{q}, t)$ is perceived collision-free. Recall that $d_{min}(\mathbf{q}, t_k)$ is the minimum distance between the robot if put at $\mathbf{q}$ and the atomic obstacles. Thus,

$$d_k = v_{max}(t - t_k) = d_{min}(\mathbf{q}, t_k) - \epsilon \tag{4}$$

where $\epsilon > 0$ is very small. Clearly at t_k, $E'_{t_o}((\mathbf{q}, t), t_k)$ still has atomic obstacles because it is larger than $E_{t_o}((\mathbf{q}, t), t_k)$. Later, suppose at time instant t'_k, where $t_k < t'_k \leq t$, $E'_{t_o}((\mathbf{q}, t), t'_k)$ has shrunk enough to "squeeze out" atomic obstacles in it, perceiving the CT-point $(\mathbf{q}, t)$ as collision-free. Thus,

$$d'_k = cv_{max}(t - t'_k) = d_{min}(\mathbf{q}, t'_k) - \epsilon. \tag{5}$$

From (4) and (5), we have $d'_k - d_k = d_{min}(\mathbf{q}, t'_k) - d_{min}(\mathbf{q}, t_k)$. From the above equation and equation (3), we have

$$d'_k - d_k = pv_{max}(t'_k - t_k), \ -1 \leq p \leq 1 \tag{6}$$

From the equations (4), (5), and (6), we can further obtain

$$t'_k - t_k = (\frac{c-1}{c+p})(t - t_k) \leq t - t_k \tag{7}$$

Hence, $t_k < t'_k \leq t$. ∎

The significance of the theorem is that, if a CT-point $(\mathbf{q}, t)$ is collision-free, then it will be perceived as collision-free *no later than* time t no matter how badly the actual v_{max} is overestimated as v'_{max}. Moreover, $t'_k = t$ only in the very scenario when, at any time $t_i \in [t_k, t]$, the nearest atomic obstacle at t_k originally inside $E_{t_o}((\mathbf{q}, t), t_o)$ moves towards the robot's configuration $\mathbf{q}$ with v_{max}, and in all other cases, $t'_k < t$. This shows the robustness of the CFP.

7 Implementation and Experimental Results

We have tested our approach in both simulation and real-world experiments.

7.1 Test in simulation

A planar rod robot was considered in the test. It can only translate on a plane with a fixed orientation $\theta = -45°$. Thus the robot has two translational degrees of freedom, with reference position set at the top point of the rod. As shown in Figure 4, the robot is initially at a collision-free CT-point $(\mathbf{S}, t_o)$ and needs to reach a goal configuration $\mathbf{G}$ in this completely unknown environment, where there are unknown obstacles of arbitrary shapes formed by what the robot can only sense as identical red circles (called the atomic obstacles). The sensing frequency is 20 Hz. The obstacles can either be static or move randomly with changing speeds no greater than v_{max} units/s, which can be overestimated by the robot as $v'_{max} > v_{max}$.

We want to check how effective the collision-free perceiver (CFP) of Algorithm 1 can be used by a real-time motion planner to guide the rod robot to its goal while avoiding obstacles. While the robot waits at the starting configuration $\mathbf{S}$, the planner can explore the perceived CT-space to find a motion segment for the robot to move. Different planners can be used here, and the difference is only that they will provide different candidate motion segments for CFP to check for feasibility (i.e., if guaranteed collision-free or not). A motion segment can consist of multiple straight-line segments.

We use the real-time adaptive motion planner (RAMP) [10] to provide CFP candidate motion segments. The planner is implemented in C#, on Dell Optiplex GX620. RAMP can simultaneously establish a diverse set of trajectories starting from the robot's current location for the CFP to check and let the robot to execute the best feasible trajectory segment. While the robot executes the guaranteed collision-free trajectory segment, RAMP continues planning subsequent feasible trajectory segments, using CFP. Thus, once the robot finishes executing the current segment, it can hopefully move to the next guaranteed collision-free trajectory segment seamlessly without stop.

Figure 6 shows the snapshots of an example run, when the rod robot of unit length moves from position $\mathbf{S}= (1, 1)$ to the goal $\mathbf{G}= (9, 9)$ with speed 5 units/s. $v_{max} = 1$ unit/s, and the obstacles can change velocities instantly. The sequences of green (or dark in B/W) reference positions show the perceived

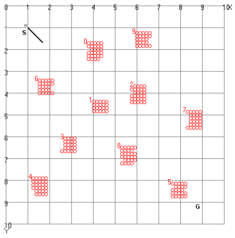


Fig. 4. Simulation environment

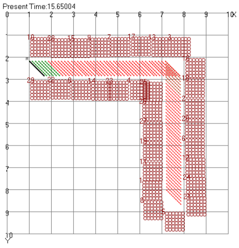
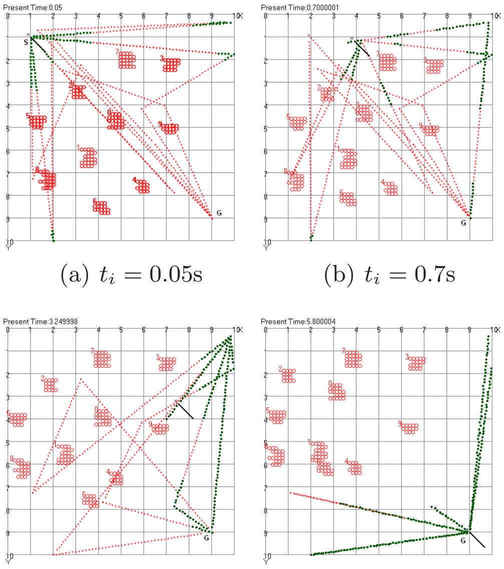


Fig. 5. Static narrow passage of width approximately 1 unit



(a) $t_i = 0.05$s

(b) $t_i = 0.7$s

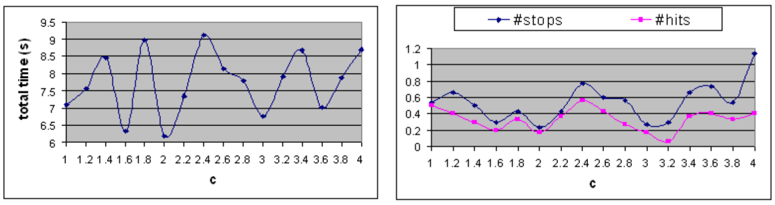
(c) $t_i = 3.25$s

(d) $t_i = 5.8$s

Fig. 6. Snapshots of an example run in simulation

collision-free trajectory segments (without showing the time instants). The sequences of red reference positions indicate uncertain trajectory segments at each moment of perception, which may or may not be collision-free. The robot executes the best green option found. Note that the robot never hits an obstacle *while moving along a green trajectory* because it is guaranteed collision-free. The attached movie shows the process in four examples with increasing number of obstacles and v_{max}. The robot may only get hit by an obstacle (and momentarily change color to blue) when it cannot find a green trajectory so fast and has to stop its motion.

We tested the effects of overestimating the speed bound v_{max} of obstacles in the same environment of the example run. Figure 7 shows the average results over 30 runs for each c. The *total time* is the average total time for the robot to plan and move simultaneously from the start position to the goal position. If the robot cannot find a collision-free trajectory segment to move to when it reaches the end of the current collision-free trajectory segment, it has to stop its motion until a new collision-free segment is found. Thus, the *# stops* means the average number of stops the robot has to make during its journey from the start position to the goal position. The *# hits* is the average number of times when the robot got hit by obstacles during its stops (i.e., before the next collision-free trajectory segment is found). The results show that increasing c, or the level of over-estimation of v_{max}, has little effect on those performance parameters. The ups and downs in the curves reflect the randomness in the environment.



(a) c vs. total time (sec)

(b) c vs. # hits and # stops

Fig. 7. Effects of Over-estimating v_{max} as $v'_{max} = cv_{max}$, $c \geq 1$

We also tested how our approach works in a static environment ($v_{max} = 0$) with a narrow passage shown in figure 5, where the robot has to move through the narrow passage to a goal, shown in every position of the path. We performed experiments for varied over-estimation $1 \leq v'_{max} \leq 4$. In all cases, the travel time for the robot from the start to the goal position was constant: 2.49s. This, in fact, experimentally verified Theorem 2.

7.2 Real-world experiments with a high DOF robot

We have also tested the CFP (Algorithm 1) by embedding it in a simple real-time motion planner for a real desktop 5-DOF robot manipulator with revolute joints in an unknown and unpredictable environment, sensed via an overhead stereovision sensor (Figure 8). Our real-time motion planner finds a collision-free straight-line segment in the CT-space as the next-step motion for the robot to execute, with a search method compromising randomized and greedy search[1] and using CFP for discovering collision-free motion. As the robot moves, the planner simultaneously finds again the subsequent next step until the goal is reached.

The planner was implemented in C++ on a low-end PC (Dell Optiplex GX260). The 5-DOF manipulator is made from the Robix Rascal RC6 kit. The stereo vision camera is PGR's Digiclops. The obstacles are blocks unknown to the robot, which can be moved in ways also unknown to the robot. Table 1 shows the input parameter values to the planner, where **S** and **G** are the starting and goal configurations respectively. $\dot{q}_{-ve}$ and $\dot{q}_{+ve}$ are the negative and positive bounds on the joint speeds of the robot. Note that the number of atomic obstacles of an actual obstacle increase if the obstacle is close to the origin of the camera frame.

[1] In this way the planner is able to overcome local minima, but the details of search and handling local minima is not the focus of this paper since a number of different strategies can be used.

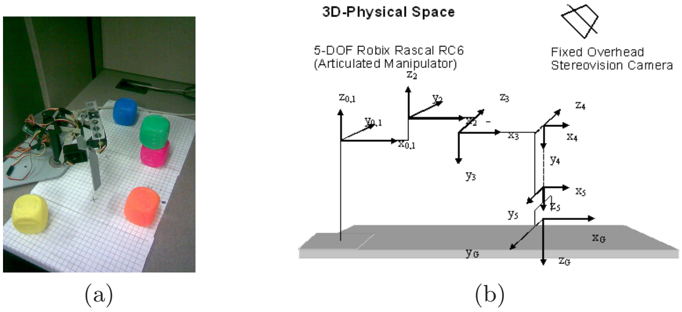


Fig. 8. Experimental setup. Fig(a) shows Robix Rascal RC 6 and obstacles of unknown geometry in its workspace and (b) shows kinematics of the manipulator

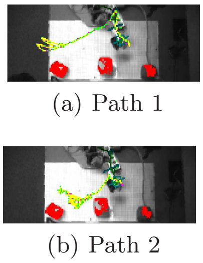
The atomic obstacles generated from stereo vision are as described in section 4. The atomic obstacles representing the robot itself and the known desk surface (as "floor") were filtered out. The shape of an atomic obstacle was approximated as a straight-line ray. The shape of a link of the robot was simplified by a cylindrical bounding volume. The number of atomic obstacles in the test environment were in the range of 345–800. The average rate of collision checking in the CFP computation was 1430.64 CT-points/second.

Table 1. Input Parameters and Values

S, **G** (degrees)	v_{max} cm/sec	$(\dot{q}_{-ve}, \dot{q}_{+ve})$ (degrees/sec)	*Sensor* *Image resolution*	$min\#O(i,j)$ *per obstacle*
$[-70, 45, 0, 0, 0]^T$ $[70, -45, 0, 0, 0]^T$	1	$([-6, -6, -5, -6, -7]^T,$ $[\ 6,\ 6,\ 5,\ 6,\ 7]^T)$	160 × 120	115

Figure 9 shows a test environment and two different resulting paths that the robot traveled. The environment had 4 blocks as obstacles, where two were placed at the corners and two were stacked together to form a taller obstacle in between. The taller obstacle created a local optima for the given robot structure with limited dexterity, which our planner was able to overcome.

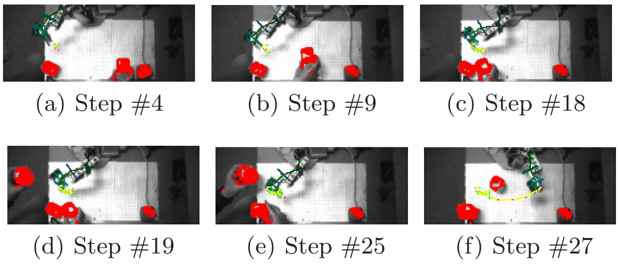
Figure 10 shows a sequence of selected snapshots of the robot motion in another test environment, where there are four obstacles, and two of them are dynamic, moved by the two hands of a human operator. (Note that a grid of 1×1 cm^2 squares on the desk was used as a guidance to move obstacles close to $v_{max} = 1$cm/s.) As shown, the operator first moved one block towards the robot. Between step 9 to step 18, the planner tried to get most of links closer to their goal positions while avoiding the moving block and the block at the bottom left corner. After step 18, the moving block decreased its speed, and a new block was moved into the visible robot workspace. The planner noticed the reduced speed of the first moving block in time due to the non-conservative



(a) Path 1

(b) Path 2

Fig. 9. An environment (Env1) and two traveled paths by the robot



(a) Step #4

(b) Step #9

(c) Step #18

(d) Step #19

(e) Step #25

(f) Step #27

Fig. 10. Selected steps taken by the robot in an unknown dynamic environment (Env2). In (a), robot is near the configuration **S** and in (f), robot is at configuration **G**

nature of the dynamic envelopes and simply guided the robot to pass by the moving block and the static block, while moving away from the newly entered block to reach the goal in step 27. Table 2 shows the resulting statistics characterizing the planner performance in the two task environments.

Table 2. Average results from two environments (for the same start and goal configurations of the robot)

Env	**Path length** (deg)	**#Steps**	**Total time** (*sec*)	**#Sensing cycle** (*Hz*)
Env1	514.924	62.2	82.8	3.5
Env2	235.98	27	49	4.58

8 Conclusions and Future Work

The paper introduces the notion of perceived CT-Space for a robot, which characterizes what truely collision-free regions can be perceived from sensing in an otherwise completely unknown and unpredictable environment. Through the novel concept of dynamic envelopes complemented by low-level atomic obstacles directly from sensing, the paper presents an approach to discover guaranteed collision-free motion segments to facilitate real-time robot motion planning in completely unknown and unpredictable environments. The approach is in essence efficient because it does not assume worst-case behaviors but rather operates based on perceiving the actual obstacle behaviors, and no re-computation is needed for the already found collision-free motion segments. The approach is also proven robust with respect to unknown maximum velocities of obstacles. It can be used by different motion planners regardless of specific planning strategies. It is tested in both simulation and real experiments with a real 5-DOF robot manipulator.

We will further take into account the robot's position and control uncertainty in producing guaranteed collision-free motions and further test the approach in experiments to see how fast the obstacles have to move relative to the robot for the approach to be infeasible.

References

1. T. Lozano-Perez, "Spatial planning: A configuration space approach," in *IEEE Transaction on Computers*, pp. 108–120, 1983.
2. J. Latombe, *Robot Motion Planning.* Kluwer Academic Publishers, 1991.
3. J. Ward and J. Katupitiya, "Free space mapping and motion planning in configuration space for mobile manipulators," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4981–4986, 2007.
4. L. Kavraki, P. Svestka, J. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," in *IEEE Transactions on Robotics and Automation*, pp. 566–580, 1996.
5. S. M. LaValle, *Planning Algorithms.* Cambridge University Press, 2006.
6. T. Fraichard and H. Asama, "Inevitable collision states. a step towards safer robots?," in *Advanced Robotics*, pp. 1001–1024, 2004.
7. Y. Yang and O. Brock, "Elastic roadmaps: Globally task-consistent motion for autonomous mobile manipulation in dynamic environments," in *Proceedings of Robotics: Science and Systems II*, 2006.
8. M. Zucker, J. Kuffner, and M. Branicky, "Multipartite rrts for rapid replanning in dynamic environments," in *IEEE ICRA*, pp. 1603–1609, 2007.
9. P. Leven and S. Hutchinson, "A framework for real-time path planning in changing environments," in *Int. J. of Robotics Research (IJRR)*, pp. 999–1030, 2002.
10. J. Vannoy and J. Xiao, "Real-time adaptive motion planning (ramp) of mobile manipulators in dynamic environments with unforeseen changes," in *IEEE Transactions on Robotics*, Oct. 2008.
11. J. Vannoy and J. Xiao, "Real-time motion planning of multiple mobile manipulators with a common task objective in shared work environments," in *IEEE ICRA*, pp. 20–26, April 2007.
12. Y. Yu and K. Gupta, "An efficient on-line algorithm for direct octree construction from range images," in *ICRA*, pp. 3079–3084, 1998.
13. Y. Yu and K. Gupta, "Sensor-based probabilistic roadmaps: experiments with an eye-in-hand system," in *Advanced Robotics*, pp. 515–536, 2000.
14. F. Yili, J. Bao, W. Shuguo, and C. Zhengcai, "Real-time sensor-based motion planning for robot manipulators," in *IEEE ICRA*, pp. 3108–3113, 2005.
15. Y. Yu and K. Gupta, "C-space entropy: A measure for view planning and exploration for general robotsensor systems in unknown environments," in *IJRR*, pp. 1197–1223, 2004.
16. S. Thrun, "Exploring artificial intelligence in the new millenium," ch. Robotic mapping: A survey, Morgan Kaufmann, 2002.