SAFETY-ASSURED OFFLINE-ONLINE PATH PLANNING WITH REDUCED STORAGE REQUIREMENTS IN SIMULATED SURGICAL TRAINING

Aakarsh Rao
Michael Valenzula
Jerzy W. Rozenblit
Dept. of Electrical and Computer Engineering
University of Arizona
1230 E Speedway Blvd, Tucson, AZ, USA
aakarshrao7@email.arizona.edu, mvalenz@email.arizona.edu, and jr@ece.arizona.edu

ABSTRACT

Optimal path planning and re-planning are essential requirements for efficient robotic and laparoscopy surgical education as they guide instruments through the operating field. Ensuring collision free paths and training under uncertainty, with limited dependence on an expert surgeon's assistance, are key aspects to be considered. In this paper, a hybrid path planning method is introduced to assist in surgical training that combines offline and online approaches for trajectory generation. The proposed offline-online path planner integrates features of offline approaches that guarantee safe navigation with real-time re-planning capability of online planners. It reduces memory storage requirements of the trajectories for online path planning, by more effectively saving them as segments, catering to the portability of the surgical systems. The efficacy of the proposed approach is demonstrated in a simulated training environment for hand-eye coordination tasks on the Computer Assisted Surgical Trainer (CAST) platform.

Keywords: path planner, simulated surgical training, offline and online.

1 INTRODUCTION

Minimally Invasive Surgery (MIS) (which is often referred to as laparoscopic surgery) is a procedure in which long, thin instruments are inserted through small incisions in the body, while visualizing the operative field on a display monitor. Typical challenges of MIS procedures include a restricted field of vision, hand-eye coordination difficulty, lack of depth perception, and limited flexibility of laparoscopic instruments. For these reasons, much research has been done to develop effective, intelligent training techniques to improve laparoscopy skills of practicing surgeons and trainees. A variety of surgical training systems have been proposed and developed to serve as the underlying practice platforms (Ordonez and et. al. 2007)(Pham et al. 2004)(Stylopoulos et al. 2004)(Derossis et al. 1998). Computer-aided, intelligent surgical training is a novel direction of research in the current age of robotically assisted laparoscopic surgery. In our work, we aim to provide continuous visual guidance on optimal, collision-free navigation paths with minimum supervision of an expert surgeon to improve the outcome of surgical training(Derossis et al. 1998).

The Computer Assisted Surgical Trainer (CAST) (Derossis, Fried, Abrahamowicz, Sigman, Barkun, and Meakins 1998)(Rozenblit 2014)(Rozenblit 2012) as shown in Fig. 1, is a training system that provides the above capabilities along with precise assessment metrics, intuitive visual assistance and haptic guidance for the execution of surgical training tasks. The central component of CAST is the optimal motion planning method (optMIS) (Nikodem et al. 2012)(Napalkova et al. 2014). However, this offline strategy generates strictly fixed paths in static environments with poor computational performance (both memory and time-

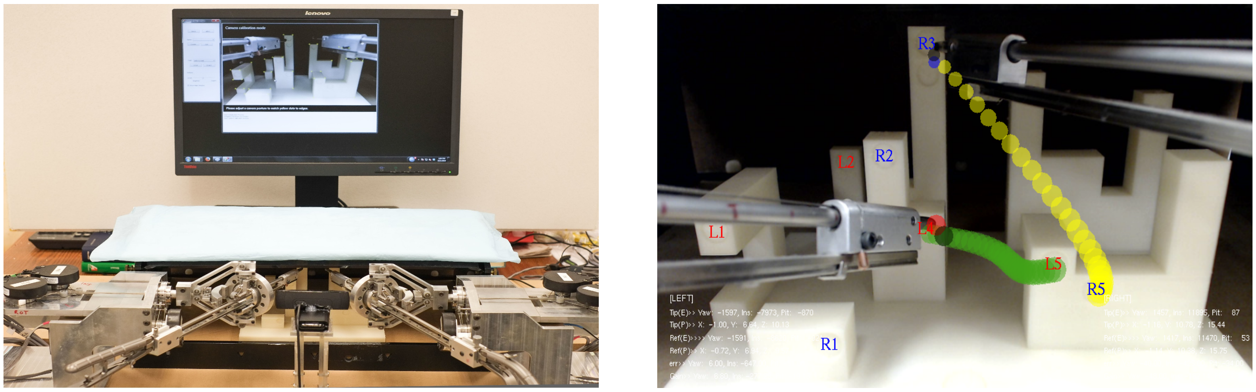

Figure 1: The Computer Assisted Surgical Trainer with the hand-eye coordination "block world" task. The left and right instruments follow the optimal path generated by the hybrid offline-online path planner rendered on the monitor.

wise). Hence, our work derives from this initial surgical movement planner which generates offline, shortest, collision-free trajectories for instruments in a training environment.

We posit that surgical path planners should dynamically update the path, for instance, to reinforce proper reactions to unanticipated complications (*e.g.*, bleeding). Current state of the art online path planning algorithms (Geraerts and Overmars 2004)(LaValle and Kuffner 2001) as applied to surgical trainers, do not assure collision avoidance during rapid re-planning especially under strict timing constraints (in an unforeseen medical emergency), nor do they take into account potential memory storage requirements for storing pre-computed roadmaps (Yershov and LaValle 2011)(Torres, Baykal, and Alterovitz 2014). Memory (and computation speed in cases of rapid re-planning needs) requirement is envisioned to be a key consideration in path generation, especially in portable surgical trainers (like CAST) and mobile robotics, in general. This is attributed to laparoscopic instruments having multiple degrees of freedom operating in a 3D space along with several constraints to be considered for collision avoidance. This motivates the work presented here.

In what follows, we present a hybrid, flexible offline-online path planner with safety assurance and reduced memory demands for surgical training systems. Safety-assurance refers to the guarantee of collision-free navigation in the work space even during dynamic re-planning. We also mathematically model the optimal segment length for consistent memory savings. The proposed approach is intended to improve training capability by incorporating quick dynamic re-planning while assuring collision avoidance. The wider aspect of this approach, *optMIS*, has been presented and validated in (Napalkova et al. 2014). We evaluate our path planner in a simulated environment on a hand-eye coordination training task in a "block world" environment for the CAST (see Fig. 1).

2 RELATED WORK

Motion and path planning has been a subject of substantial research in the past few decades. Discrete, combinatorial and sampling based path planning algorithms are widely utilized in the community (LaValle 2006). Sampling based algorithms are especially powerful for offline planning as they probe the configuration space for obstacles with an independent collision detection unit. Sampling based algorithms have been developed for large state space dimensions and dynamic, cluttered and unpredictable environments (LaValle 2006). For such environments the performance cost and computation time is high, thus, variants of these algorithms were introduced in (Ichnowski and Alterovitz 2012)(Yershov and LaValle 2011)(Salzman and Halperin 2016) that improve those metrics. Discrete combinatorial algorithms are effective as the state space is static and predictable while being exact, since they find (or fail to find) a path for a given problem instance. Probablistic Roadmaps (PRM) (Kavraki et al. 1996) are appropriate in this reference, where a roadmap of the state space is created and a path is queried from this roadmap. Popular search methods such as A*, D* and Dijkstra's are used to query the constructed roadmap/graph (Bennewitz and Burgard 2000)(Yershov and LaValle 2011)(Stentz 1994). Our offline path planner i.e. *optMIS* is based on PRM, tailored to the constraints of CAST with an additional smoothing algorithm to result in near-optimal collision-free smooth trajectories.

Motion/path planning for medicine and surgery have critical requirements of safety (obstacle avoidance) and precision. The approach to planning in such domains is based on (van den Berg, Ferguson , and Kuffner 2006), where a priori information of the environment, including static and dynamic elements are taken into account (offline) and continuous improvements to the initial trajectory are made, based on the agent and environmental changes (online). Offshoots of this core approach have been validated and demonstrated in simulated medical and surgical environments in (Torres, Baykal, and Alterovitz 2014)(Sun and Alterovitz 2014)(Berenson, Abbeel, and Goldberg 2012). Our hybrid path planner is influenced by these approaches.

Storage is a prohibitive factor in portable surgical training systems like CAST. It is desirable to store several paths for several training scenarios in such systems. Most of the references mentioned do not consider storage requirements for their roadmaps. (Shaharabani et al. 2013)(Dobson and Bekris 2014) propose algorithms to reduce the size of path planning roadmaps. However, these algorithms applied in the context of surgical training will result in loss of coverage, connectivity and quality. In addition, these methods require collision checking during sparsification of the roadmap impeding performance. Our proposed path planner reduces storage requirements while retaining coverage of the state space, improving connectivity, maintaining path quality and requiring no additional collision checking.

In the remainder of this paper, we discuss the details of the hybrid offline-online path planner and model the segmentation of roadmap to reduce memory requirements. We evaluate an implementation for a simulated hand-eye co-ordination laparoscopic training task in CAST and conclude with a discussion of future work.

3 PROPOSED APPROACH

Our approach of the hybrid offline-online path planner is divided into three main phases: (a) offline path planning, (b) repository of segmented paths, and (c) online path planning, as shown in Fig. 2. The following

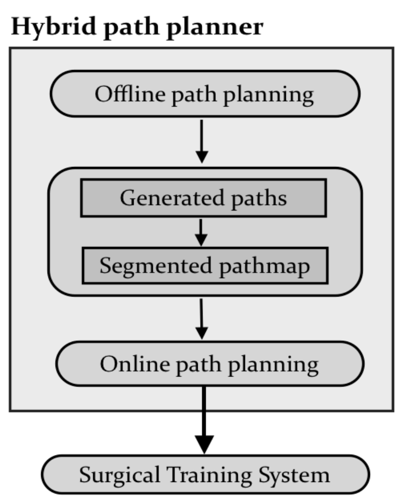

Figure 2: Diagram of the Offline-Online approach.

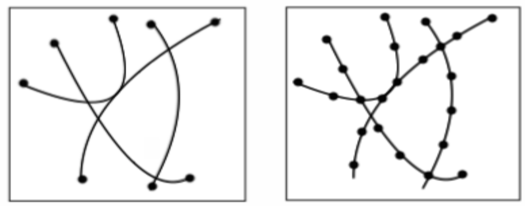

Figure 3: Representation of optimal paths with its *segmented pathmap*.

expands the background stages i.e. offline path planning and repository of segmented paths for the online path planner.

3.1 Offline path planner

We adopt the offline path planner implemented in (Napalkova et al. 2014) named *optMIS* that robustly generates optimal paths for the laparoscopic instruments, provided the input of a 3D training model. In *OptMIS*, Delaunay triangulation algorithm splits the 3D space into tetrahedrons to derive the search space $\mathscr{C}'$ for path planning. Given the initial and goal point, Dijkstra's algorithm finds the shortest path. Cubic splines smooth the shortest paths with a brute-force collision check with all the obstacles present in the space, which is essential in the medical context. The resulting optimal paths are generated as an array of 3D points. However, generating optimal paths for a moderately sized 3D model as in Fig. 1, is significantly slow and hence, improved algorithms were incorporated for faster performance. To limit the scope of the paper, we do not delve into these algorithms. We choose points of interest based on the training space and task to be performed. These points can be selected by the surgeon/trainee or can be randomly generated in the search space roadmap. The instrument tip is assumed to be a sphere of diameter w_{tip} (width of the instrument tip). Thus points of interest can be chosen with a tolerance distance of up to w_{tip}. The array of paths $\mathbf{P}$ is constructed from the improved *optMIS* using the points of interest $\mathbf{p}$ as the input. Enumerative paths are computed between all points in $\mathbf{p}$ resulting in $\binom{N_{\mathbf{P}}}{2}$ paths in $\mathbf{P}$, where$N_{\mathbf{p}}$ is the total number of points of interest chosen. $\mathbf{P}$ is assured to consist of collision-free and hence, safe paths as the offline path planner has been formally verified in (Napalkova et al. 2014).

3.2 Repository of segmented paths

To store the generated trajectories $\mathbf{P}$ efficiently, we segment the paths and find common segments. Only a single copy of the common segments found are stored along with the unique segments, embedding the neighbor information of the segments to maintain generated paths. As $\mathbf{P} \in \mathscr{C}'$, especially in medical scenarios where $\mathscr{C}'$ is restricted, the probability of finding common paths is high; which is an opportunity to reduce memory requirements. A leeway of w_{tip} to find aligning common segments is considered as it is a reasonable assumption in laparoscopic surgical training. This compresses the original path roadmap to what we call a *segmented pathmap*. The *segmented pathmap* has three main purposes: 1. it reduces the amount of memory required for storage without loss of information, 2. it expands search space connectivity by allowing finer re-planning without the need for additional computation, and 3. the search space can be easily scaled for other training tasks or scenarios. As illustrated in Fig. 3, we see that the connectivity increases, as there are many more possible paths from the *segment pathmap* as compared to just a repository of paths, which implies more training scenarios for a surgical trainee.
The memory savings factor m can be given by:

$$m = \frac{\text{length}\,(\mathbf{P})}{|\mathbb{S}|}, \text{where } \mathbb{S} \text{ is the } \textit{segmented pathmap} \tag{1}$$

The segment size n_s is the key factor in obtaining an increased m. The details of the mathematical formulation to deduce an optimal n_s is detailed in Section 4.

4 OPTIMAL SEGMENT SIZE DEDUCTION

We consider left and right hand laparoscopic instruments in a 3D environment space $\mathscr{C}$ with obstacle space $\mathscr{O}$ (such as the one shown in Fig. 1). The free space $\mathscr{C}_{free}$ that avoids collisions of the instruments with the obstacles is defined as $\mathscr{C}_{free} = \mathscr{C} - \mathscr{O}$. Based on (Feng et al. 2009), in surgical training systems the actual task training space $\mathscr{C}'$ is bound around the obstacle space such that $\mathscr{C}' \subseteq \mathscr{C}_{free} \cup \mathscr{O}$.

4.1 Repository of path segments

Points of interest, $\mathbf{p}$, are chosen based on the task to be performed in the training environment and are bound in the space $\mathscr{C}'$. Let the total number of points of interest be $N_{\mathbf{p}}$, then, the total number of paths generated by offline path planning is $N_P = \binom{N_{\mathbf{p}}}{2}$ and represented by the array of paths

$$\mathbf{P} = [\mathtt{P}_1, \ldots, \mathtt{P}_{N_P}] . \tag{2}$$

Each path $\mathtt{P}_i$ is an array of 3D points $[p_{1_{x,y,z}}, \ldots, p_{l_{x,y,z}}]$, where l represents the total number of points in a path. The paths are divided into N_s segments of n_s points each.

The total number of segments N_s of $\mathbf{P}$ is

$$N_s = N_u + \sum_{i=1}^{N_c} \mathbf{x}_{c_i},$$

where N_u is the total number of unique segments and N_c is the total number of common segments with $\mathbf{x} = \{r_{c_1}, \ldots, r_{c_{N_c}}\}$ repetitions each. So the number of elements in (2) is

$$\begin{aligned} \mathrm{length}\,(\mathbf{P}) = & \,\mathrm{length}\left(\left[s_{u_1}, \ldots, s_{u_{N_u}}\right]\right) + \\ & \mathrm{length}\left(\left[s_{c_1}, \ldots, s_{c_{N_c}}\right] \odot \mathbf{x}\right) \end{aligned} \tag{3}$$

where $\{s_{u_1}, \ldots, s_{u_{N_u}}\}$ is a set of unique segments, $\{s_{c_1}, \ldots, s_{c_{N_c}}\}$ is a set of common segments with $\mathbf{r}$ repetitions each, and $\odot$ is a repeat operator (s_{c_i} is repeated in memory r_{c_i} times). $\mathbb{S}$ represents the segments of the *segmented pathmap*:

$$\begin{aligned} \mathbb{S}_u &= \{s_{u_1}, \ldots, s_{u_{N_u}}\}, \\ \mathbb{S}_c &= \{s_{c_1}, \ldots, s_{c_{N_c}}\}, \\ \mathbb{S} &= \mathbb{S}_u \cup \mathbb{S}_c. \end{aligned}$$

We assume these segments exist in the training space, *i.e.,* $s \in \mathscr{C}' | s \in \mathbb{S}$.

Taking no additional action, the memory required for $\mathbf{P}$ will be on the order of:

$$\mathtt{M}' = \left(3 \;\mathrm{length}\,(\mathbf{P}) + \binom{N_{\mathbf{p}}}{2}\right), \tag{4}$$

for the 3D vectors and the references to where each path starts. If we remove the duplicate segments, the memory requirement (for the segments and for the connections between segments) is:

$$\mathtt{M} = (3\, n_s + B)\ |\mathbb{S}| . \tag{5}$$

n_s is a constant describing the number of points in a segment and B is the memory requirement to store a reference to a successor or predecessor segments. The factor of three follows from each segment being composed of a 3D point. Equation 5 is valid as the common segments need to be stored only once when storing the successor and predecessor segment's information B.

4.2 Segmentation of paths' modeling

If $\mathtt{P}_{i_k}$ is the kth point on the ith path, let n_s be the number of points in a segment. Our task is to find an optimal or near optimal segment size for memory storage compression. Complicating this matter is the alignment of segments (*e.g.,* [B,C]-[D,E]-[F,G] versus [A,B]-[C,D]-[E,F]). Two approaches to address alignment is to using padding, which itself will incur memory overhead, or to explicitly take into account the probability of alignment. We chose the latter.

Let $\mathrm{Ind}\colon \mathtt{P}_{i_k} \to \{1, 2, \ldots, n_s\}$. Ind maps from a point to its index in a segment. Consider

$$\mathrm{Ind}(\mathtt{P}_{i_k}) = k \bmod n_s. \tag{6}$$

If the segment length is short relative to the path length and the resulting indices are independent of the path, then the function Ind outputs a discrete uniform distribution. In other words, if

$$n_s \ll \text{length}(\mathtt{P}_i) \wedge \mathrm{Ind}(\mathtt{P}_{i_k}) \perp\!\!\!\perp \mathtt{P}_i \tag{7}$$

then

$$\mathrm{Ind}(\mathtt{P}_{i_k}) \sim \mathcal{U}(1, n_s). \tag{8}$$

Based on (8) the probability that two distinct independent paths $\mathtt{P}_i$ and $\mathtt{P}_j$ are aligned along a common segment, given the random variables $X = \mathrm{Ind}(\mathtt{P}_{i_k})$ and $Y = \mathrm{Ind}(\mathtt{P}_{j_k})$, is

$$\begin{aligned} P(X = Y) &= \sum_{x \in \{1,2,\ldots,n_s\}} P\left(X = Y \mid X = x\right) \cdot P\left(x = X\right) \\ &= \sum_{x \in \{1,2,\ldots,n_s\}} P\left(x = Y\right) \cdot \left(\frac{1}{n_s}\right) \\ &= \sum \frac{1}{(n_s^2)} \\ &= \frac{1}{n_s}, \end{aligned} \tag{9}$$

where $P\left(x = X\right) = 1/n_s$ by definition of $\mathcal{U}(1, n_s)$.

This result has two key consequences, showing the following. First, segments of length 1 (individual points) are always aligned. Second, the expected number of encoded points is independent of n_s. This means $n_s \cdot |\mathbb{S}_c|$ is mostly independent of n_s, assuming (7).

When trying to optimize (5), we need to keep three things in mind. Equation (9) shows expected number of encoded points is independent of n_s. Larger segments mean there will be fewer segments and hence less overhead. However, larger segments run the risk that $\mathrm{Ind}(\mathtt{P}_{i_k})$ is something other than uniformly distributed. We consider the variance of the memory usage to break this tie.

One can readily see the second moment of the path alignment problem is:

$$\sum_{x \in \{1,2,\ldots,n_s\}} P(X = Y)\, n_s^2 = n_s.$$

In this case, the variance (the second moment subtract the first moment squared) is dominated by the second moment. The larger n_s, the more variable the memory savings will be. For consistency, we choose a small value of n_s.

The smallest value we can choose for n_s is 1. This means that we should only save points not segments. However, this ignores three critical details. First, this ignores the influence of B in (5). Second, the smaller n_s, the larger the set $\mathbb{S}$; smaller segments mean more work is required to reconstruct optimal paths. Last, we want to use "natural" cubic splines (Hagan and West 2006) without having to reconstruct larger segments first. This implies that the smallest segment length we can use is a length of three. Thus, we choose $n_s = 3$. This leads to good, consistent memory savings. The paths are hence divided in segments of three points. End segments may consist of different number of points depending on the value of $l \bmod 3$. The derivation of the probability of finding overlapping segments in a specific space is a noteworthy factor, but will be pursued in future work to limit the scope of this paper.

5 ONLINE PATH PLANNER

The input to online path planning is the constructed *segmented pathmap*. Online path planning is performed by the A* graph search algorithm (Hart, Nilsson, and Raphael 1968) employing a specific heuristic for our problem as discussed below.

1. *Constructing input graph:* Each segment of the *segmented pathmap* represents a node η in the input graph. The neighbor information is stored as a pointer to the adjacent segments in the path. The end segments of the paths are the terminal nodes of the graph. A delimiter field is included in the start and terminal nodes of each path to maintain the path sequence. An iteration is performed through the data structure to detect the common segments. Copies of the common segments are deleted and their neighbor pointers from/to predecessor/successor nodes respectively are redirected.
2. *A* parameters design:* Admissible heuristics (defined in (Hart, Nilsson, and Raphael 1968)) improve the efficiency of A* algorithm by exploring fewer nodes and never overestimating the distance to the goal. For the current node η

$$\begin{aligned} f(\eta) =& g(\eta) + h^*(\eta), \\ \text{such that, } & h^*(\eta) \leq h(\eta) \end{aligned} \tag{10}$$

   where $f(\eta)$ is the evaluation function of η, $g(\eta)$ is the cost from the start node η_{start} to n, $h(\eta)$ is the actual cost to reach the goal node η_{goal} from η and $h^*(\eta)$ is the heuristic underestimated cost to reach the goal node η_{goal} from n.
   To satisfy (10), we choose the Euclidean heuristic (Rayner, Bowling, and Sturtevant 2011) in 3D space. For the current segment node $\eta_{cur} \in \mathbb{S}$, the Euclidean heuristic function is the Euclidean distance between the initial point of the segment η_{cur_1} and the initial point of the segment in goal node η_{goal_1}, *i.e.,* $h^*(\eta_{cur}) = \|\mathbf{r}(\eta_{cur_1}) - \mathbf{x}(\eta_{goal_1})\|_2$, where $\mathbf{x}(\eta_{cur_1})$ and $\mathbf{x}(\eta_{goal_1})$ are the 3D positions of the initial points of the segments. To simplify $g(\eta)$ we assume the curve segments to be rectifiable, as the summation of lengths of its constituent linear segments. Fig. 4 illustrates our parameter calculations for the A* algorithm.

The online path planning is implemented given the initial $\mathbf{x_{init}}$ and goal position $\mathbf{x_{goal}}$. Segments corresponding to $\mathbf{x_{init}}$ and $\mathbf{x_{goal}}$ are found. If multiple goal segments exist, they are reduced to a single goal segment, since the objective is to find the shortest path to $\mathbf{x_{goal}}$ irrespective of its contained segment. A* algorithm is executed on the *segmented pathmap* input with the heuristic discussed above. The planner computes near-optimal (approximation of length of curve segments do not always guarantee shortest trajec-

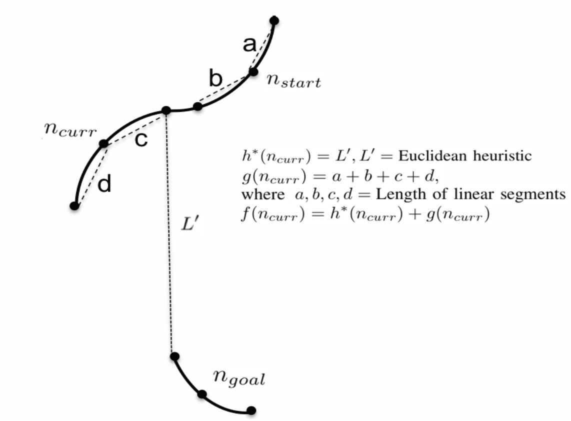

Figure 4: Heuristic calculation.

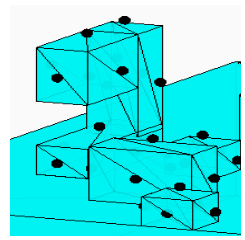

Figure 5: Selection of points of interest.

tories) trajectories but ensures collision-free complete paths which is our main emphasis. Real-time path re-planning can be accomplished by providing different $\mathbf{x_{goal}}$ on the fly while performing the task.

6 SIMULATIONS AND RESULTS

We demonstrate our offline online path planner in a simulated training environment of CAST. The simulated training scenario is a hand eye coordination task in a "block world" environment. We chose a tolerance radius of w_{tip} = 0.04cm in our experiments as this represents the diameter of the laparoscopic instrument tip used in CAST (assumed as a sphere). We perform our simulations on an Intel i7 2.93GHz quad-core computer with 4GB memory.

6.1 Simulation Setup

CAST consists of two laparoscopic instruments mounted on fixtures composed of a gimbal, one for the left hand and one for the right hand. As in laparoscopic surgery, the gimbal allows four degrees of freedom, all centered around one single entry corresponding to the incision. The hand eye coordination experimental task involves touching target positions close to the objects ("blocks") in the workspace using the tips of the instruments. The target points are selected such that they are reachable by the laparoscopic instruments in the setup (see Fig. 1). This task is modeled in MATLAB and experiments are performed using our hybrid path planner. Fig. 5 illustrates the points of interest chosen for the "block world" hand-eye coordination task. The points of interest are selected in this fashion, since the task involves maneuvering the laparoscopic instruments to pre-assigned target points in $\mathscr{C}'$ while avoiding collisions with obstacles. The selection is normally done by an expert surgeon.

Eighty points of interest are selected and improved *optMIS* is used to generate optimal paths between these points to result in about 1,300 optimal paths. The effective number of resulting optimal paths is lesser than $\binom{80}{2}$, due to the elimination of points of interest unreachable by the laparoscopic instruments in the given setup. The resulting optimal path trajectories take about seven hours to generate which is significantly less from the expected fourteen hours without the improved algorithms. The path trajectories are stored in a file with delimiters to signify start and end of each path and occupy 1.46MB of memory space. These trajectories are divided into segments for memory savings.

We conducted experiments to reaffirm our deduction of the segment length to be three for good and consistent memory savings from Section 4.2 . The path trajectories are divided into segments of length two,

Table 1: Storage memory requirements of different sized segments.

Segment Size	Common Segments (in KB)	Unique Segments (in KB)	Segmented Pathmap (in KB)
2	113	153	266
3	147	272	419
4	167	373	540
5	167	447	614

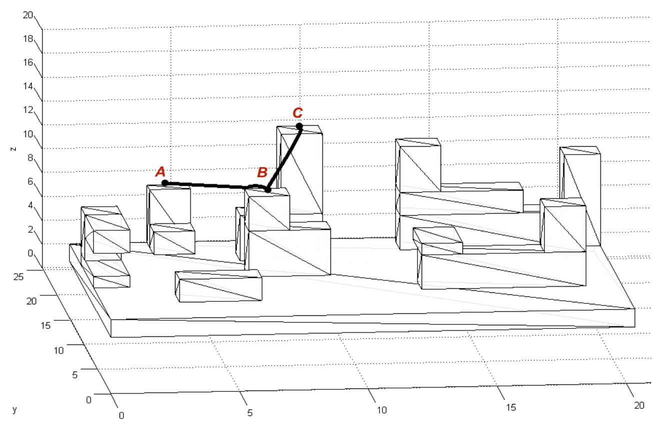

Figure 6: Illustration of an instance of the hybrid offline-online path planner for the "blocks world" laparoscopic training scenario.

three, four and five. Iterating through the segmented paths, the common segments are detected based on a correspondence of the segment end points, while considering a tolerance of w_{tip}. The common segments identified contain varying redundant repetitions, which are eliminated. Then, a single copy of each common segment along with the unique segments are saved in memory with adjacent neighbor segments pointers. Storage memory values of the compressed *segmented pathmap* for each of these segment sizes are collected and results shown in Table 1.

As observed in Table 1, segments with three points provide the greatest compression savings for the *segmented pathmap* while considering the critical details discussed in Section 4.2. It occupies 419KB of memory with 4KB of neighbor information and delimiters. Based on (1), we attain a memory savings factor of 3.45. The percentage of common segments present in the array of segmented paths with segment size three was 50.54% reaffirming our deduction. Another interesting observation is that the space occupied by common segments of segment size four and five are the same. This can be attributed to the fact that, the common segments after a point start saturating. Also, we can see the space of the unique segments start to grow slower. An iterative approach to find adaptive segment lengths for maximum savings for any given problem can be deduced but it is beyond the scope of the paper and we will pursue it in future work.

We implement the online path planner in C++ to provide real-time capability. The input graph is generated from the *segmented pathmap* for online path planning. A trainee/surgeon provides the initial and goal points of a desired path. The training work space of "blocks world" is modeled in MATLAB for visualization. Fig. 6 provides a snapshot of an instance of the online path planner for a surgical training session. Point A represents the initial position and point B the intermediate goal position for a training path for one of the instruments. Once the trainee reaches the intermediate goal position B, a replanned path is generated in almost real-time to the final goal position C. Detailed evaluation of the online path planner for dynamic path re-planning will be performed in future work.

We averaged the execution time of generating paths over three runs. Execution time of the online path planner for our instance required 90.0ms on overage. The average time of execution for a single path is around 44.0ms. Our offline-online path planner hides the poor time performance of offline only path planner (*optMIS*), that takes 67.6s for the same instance. It assures completeness and collision free paths, which are the main requirements in a surgical training scenario that may involve unanticipated emergency procedures.

7 CONCLUSIONS

We have presented a novel hybrid offline-online path planner that can compute approximately optimal paths for surgical systems. With medical training scenarios becoming more complicated and training systems leading to portability, storage memory as well as exectution efficiency are key concerns. We have addressed this issue by incorporating segmentation of the generated offline paths by storing in memory only a single copy of the detected common segments. Our method integrates safety assurance and completeness provided by discrete path planners with the real-time re-planning potential of online path planning, thus enhancing the computational performance of discrete path planning, notably the one developed in (Napalkova et al. 2014). We demonstrated our planner and its efficiency on the Computer Assisted Surgical Trainer with a hand-eye coordination training scenario.

Our offline-online path planner is scalable to other complicated surgical training models, with the one time overhead of generating the trajectory repository. Additional obstacles in a particular training scenario can be added and repository updated to reflect the same. This can be envisioned in surgical training systems with synthetic body parts and tissues. However, we still have to evaluate the offline-online path planner in a real-training scenario. Since our segmented path map is static, it restricts online path planning, failing to generate paths from points not present in the input graph. We wish to pursue these issues in future work. We will also design more complicated training scenarios (by adoption of the Fundamentals of Laparoscopic Surgery (SAGES 2016) standards) and investigate advanced offline and online path planning algorithms to reduce generation time of paths further.

ACKNOWLEDGMENT

The authors would like to thank Mr. P. Czapiewski of Manufacturing Systems Solutions, LLC for his insightful feedback and mechanical design of the Computer Assisted Surgical Trainer. We wish to thank Allan J. Hamilton, MD (Arizona Simulation Technology and Education Center (ASTEC), University of Arizona, AZ, USA) for the clinical input. This work was supported by Raymond J. Oglethorpe Endowment.

REFERENCES

Bennewitz, M., and W. Burgard. 2000. *An Experimental Comparison of Path Planning Techniques for Teams of Mobile Robots*, pp. 175–182. Berlin, Heidelberg, Springer Berlin Heidelberg.

Berenson, D., P. Abbeel, and K. Goldberg. 2012, May. "A robot path planning framework that learns from experience". In *2012 IEEE International Conference on Robotics and Automation*, pp. 3671–3678.

Derossis, A. M., G. M. Fried, M. Abrahamowicz, H. H. Sigman, J. S. Barkun, and J. L. Meakins. 1998. "Development of a Model for Training and Evaluation of Laparoscopic Skills 1". *The American Journal of Surgery* vol. 175 (6), pp. 482 – 487.

Dobson, A., and K. E. Bekris. 2014, January. "Sparse Roadmap Spanners for Asymptotically Near-optimal Motion Planning". *Int. J. Rob. Res.* vol. 33 (1), pp. 18–47.

Feng, C., J. W. Rozenblit, A. J. Hamilton, and A. Wytyczak-Partyka. 2009, April. "Defining Spatial Regions in Computer-Assisted Laparoscopic Surgical Training". In *2009 16th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems*, pp. 176–183.

Geraerts, R., and M. H. Overmars. 2004. *A Comparative Study of Probabilistic Roadmap Planners*, pp. 43–57. Berlin, Heidelberg, Springer Berlin Heidelberg.

Hagan, P. S., and G. West. 2006. "Interpolation Methods for Curve Construction". *Applied Mathematical Finance* vol. 13 (2), pp. 89–129.

Hart, P. E., N. J. Nilsson, and B. Raphael. 1968, July. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". *IEEE Transactions on Systems Science and Cybernetics* vol. 4 (2), pp. 100–107.

Ichnowski, J., and R. Alterovitz. 2012, Oct. "Parallel sampling-based motion planning with superlinear speedup". In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1206–1212.

Kavraki, L. E., P. Svestka, J.-C. Latombe, and M. H. Overmars. 1996. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces". *IEEE transactions on Robotics and Automation* vol. 12 (4), pp. 566–580.

LaValle, S. M. 2006. *Planning algorithms*. Cambridge university press.

LaValle, S. M., and J. J. Kuffner. 2001. "Randomized kinodynamic planning". *The International Journal of Robotics Research* vol. 20 (5), pp. 378–400.

Napalkova, L., J. W. Rozenblit, G. Hwang, A. J. Hamilton, and L. Suantak. 2014, November. "An Optimal Motion Planning Method for Computer-assisted Surgical Training". *Appl. Soft Comput.* vol. 24 (C), pp. 889–899.

Nikodem, J., G. Hwang, J. W. Rozenblit, L. Napalkova, and P. Czapiewski. 2012, April. "Technical realization of the optimal motion planning method for minimally invasive surgery". In *2012 IEEE 19th International Conference and Workshops on Engineering of Computer-Based Systems*, pp. 195–200.

Ordonez, and et. al.. 2007, Aug. "A Ubiquitous Context-Aware Environment for Surgical Training". In *2007 Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking Services (MobiQuitous)*, pp. 1–6.

Pham, T., L. Roland, K. A. Benson, R. W. Webster, A. G. Gallagher, and R. S. Haluck. 2004. "Smart tutor: a pilot study of a novel adaptive simulation environment.". *Studies in health technology and informatics* vol. 111, pp. 385–389.

Rayner, C., M. Bowling, and N. Sturtevant. 2011. "Euclidean Heuristic Optimization". In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11, pp. 81–86, AAAI Press.

Rozenblit, J. W. 2012. *Models and Techniques for Computer Aided Surgical Training*, pp. 233–241. Berlin, Heidelberg, Springer Berlin Heidelberg.

Rozenblit, J. W. e. a. 2014. "The Computer Assisted Surgical Trainer: Design, Models, and Implementation". In *Proceedings of the 2014 Summer Simulation Multiconference*, SummerSim '14, pp. 30:1–30:10. San Diego, CA, USA, Society for Computer Simulation International.

SAGES 2016. "Fundamentals of Laparoscopic Surgery".

Salzman, O., and D. Halperin. 2016, June. "Asymptotically Near-Optimal RRT for Fast, High-Quality Motion Planning". *IEEE Transactions on Robotics* vol. 32 (3), pp. 473–483.

Shaharabani, D., O. Salzman, P. K. Agarwal, and D. Halperin. 2013. "Sparsification of motion-planning roadmaps by edge contraction". In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 4098–4105. IEEE.

Stentz, A. 1994, May. "Optimal and efficient path planning for partially-known environments". In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pp. 3310–3317 vol.4.

Stylopoulos, N., S. Cotin, S. Maithel, M. Ottensmeyer, P. Jackson, R. Bardsley, P. Neumann, D. Rattner, and S. Dawson. 2004. "Computer-enhanced laparoscopic training system (CELTS): bridging the gap". *Surgical Endoscopy And Other Interventional Techniques* vol. 18 (5), pp. 782–789.

Sun, W., and R. Alterovitz. 2014, Sept. "Motion planning under uncertainty for medical needle steering using optimization in belief space". In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1775–1781.

Torres, L. G., C. Baykal, and R. Alterovitz. 2014, May. "Interactive-rate motion planning for concentric tube robots". In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1915–1921.

van den Berg, J., D. Ferguson , and J. Kuffner. 2006, May. "Anytime Path Planning and Replanning in Dynamic Environments". In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2366 – 2371.

Yershov, D. S., and S. M. LaValle. 2011. "Simplicial Dijkstra and A* algorithms for optimal feedback planning". In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3862–3867. IEEE.

AUTHOR BIOGRAPHIES

AAKARSH RAO is currently a PhD student in the Electrical and Computer Engineering Department at the University of Arizona. He works in the Model Based Design Laboratory on medical device security/design and path planning for surgical training systems. He received his Masters in January 2015 from the same university and B.E in Electronics and Communication from M.S. Ramaiah Institute of Technology, India in 2009. His email is aakarshrao7@email.arizona.edu

MICHAEL VALENZUELA received his BSc (2008), MSc (2010), and his PhD (2016) from the University of Arizona during which he contributed to this work. He received the International Engineering Consortium's William L. Everitt Student Award of Excellence (University of Arizona, Electrical and Computer Engineering Departmental Honors, 2008). While at the University of Arizona he worked on novel machine learning techniques, focusing on exploiting the No Free Lunch Theorems for search, optimization, and machine learning. Other academic research interests include modeling, simulations, applied mathematics, and artificial intelligence. Also while at the University of Arizona, he was the chief research associate of a sponsored research project called "Asymmetric Threat Response and Analysis Program (ATRAP). He currently works in Raytheon Missile Systems' Signal Processing department. His email is mvalenz@ece.arizona.edu

JERZY W ROZENBLIT is a University Distinguished Professor, Raymond J. Oglethorpe Endowed Chair in the Electrical and Computer Engineering (ECE) Department, with a joint appointment as Professor of Surgery in the College of Medicine at the University of Arizona. During his tenure at the University of Arizona, he established the Model-Based Design Laboratory with major projects in design and analysis of complex, computer-based systems, hardware/software codesign, and simulation modeling. He presently serves as Director of the Life-Critical Computing Systems Initiative, a research enterprise intended to improve the reliability and safety of technology in healthcare and life-critical applications. His email address is jr@ece.arizona.edu.