2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Maximizing visibility in collaborative trajectory planning

Florian Shkurti[1] and Gregory Dudek[1]

***Abstract*— In this paper we address the issue of coordinating the trajectories of two collaborating robots in environments with obstacles so that visibility between them is maximized in the presence of competing constraints. Specifically, we examine the problem of allowing one robot (the "photographer") to follow another robot ("the subject") through a planar environment while maintaining visual contact to the maximum degree consistent with an efficient traversal. This problem has numerous applications, for instance in scenarios where communication between robots requires line-of-sight. We formalize this problem in the context of centralized kinodynamic planning and we present solutions based on the asymptotically optimal sampling-based RRT* planner. We discuss connections to the traditional formulation of pursuit-evasion games where the analysis typically ends the moment the evader manages to escape the pursuer's visibility region. We also illustrate types of environments and other conditions under which allowing the pair of robots to break the line-of-sight is a better option than always requiring the presence of visual contact.**

I. INTRODUCTION

This paper examines the problem of allowing a pair of robots to circumnavigate an environment while remaining in visual contact. In particular, we examine the case where a target robot ("the movie star") is pursued by second robot ("the photographer") which seeks to maintain visibility. The target wants to assist the photographer (unlike many actual movie stars) but also wants to execute a trajectory of near-maximum efficiency.

The potential applications of planning by taking into account visibility of multiple agents are diverse. One of the primary applications of the formulation presented here is in settings where communication between robots and other agents requires or works best when the sender and the receiver have an unobstructed line-of-sight. This is particularly relevant to the effective range of RF communication links. Another application has to do with automated photographers [1], [2] whose aim is to ensure as long a coverage of a particular visual event as possible. Advantages of taking intermittent visual contact into consideration when planning also include cooperative localization [3].

Depending on the target's behavior and the structure of the environment, it may be impossible for the follower to always maintain visual contact. This has been illustrated in previous work in the pursuit-evasion literature, which has shown that if the target is adversarial there exist settings under which the evader can escape the follower's field of view, no matter what the follower does ([4], [5]). This type of target behavior is the worst-case instance in the spectrum of possible behaviors. The cost function that is considered in traditional pursuit-evasion games is very restrictive, namely the *time of escape* $TE \in [0, T]$, the moment at which line-of-sight breaks for the first time:

$$(p^*, e^*) = \arg\min_{p} \max_{e} TE(p, e) \tag{1}$$

Here T is the duration of the game, and $p : [0, T] \rightarrow X_{\text{free}}$, $e : [0, T] \rightarrow X_{\text{free}}$ are the trajectories followed by the pursuer and the evader within the free space X_{free}. The evader (target) wants to maximize the escape time, while the pursuer (follower) wants to minimize it.

This worst-case assumption on the target's behavior and objective is not always necessary. Loss of visibility can be unavoidable even in cases where the target's motion is independent of the follower's motion, or cooperative under some conflicting constraints[1], due to the structure of the environment. This occurs often when performing visual servoing or, perhaps in a more familiar example, when a car is trying to follow another car in the busy streets of a city, while trying to reach a common destination. In that setting, when one driver is leading another through the city it is often unavoidable, yet acceptable, to briefly lose visual contact. In the case of a cooperative target who is helping the follower, the target might stop or slow down if it appears likely that the follower has fallen too far behind.

In these circumstances choosing TE as the cost function is very limiting for the follower's planning. A more relaxed objective is $T_{\text{occlusion}} \in [0, T]$, which denotes the total time during the execution of the trajectories in which visual contact was absent. An adversarial target tries to maximize it, as opposed to the follower who wants to minimize it.

In this work we make a departure from pursuit-evasion games and consider the case where the target is cooperative in attempting to maintain visual contact with the follower, under constraints that potentially conflict with that objective. One such type of constraint is minimizing the time it takes to reach the goal. What should a cooperative target do when it reaches a corner, wait for the follower so as to prolong the duration of the line-of-sight or quickly continue towards its goal?

In the following sections, we will formalize this problem as a cooperative kinodynamic planning problem, where we take into account the dynamics of the two players, to plan trajectories that minimize $T_{\text{occlusion}}$ and the total time required to reach the goal region. We will present solutions

[1]F. Shkurti and G. Dudek are with the Center for Intelligent Machines (CIM), School of Computer Science, McGill University, Montréal, QC, Canada (florian@cim.mcgill.ca, dudek@cim.mcgill.ca)

[1]Modeling an "average" behavior for a target remains largely an open issue.

to this problem based on the asymptotically-optimal RRT* planner [6], and we will show cases where allowing the line-of-sight of the two players to break leads to less costly plans than always being required to maintain it.

Throughout this work we make the assumption that the map of the world is known to both players as well as their locations in that map. We do not assume that the environment or its obstacles are polygonal, although the examples that will be presented take place in those settings.

II. RELATED WORK

Our problem is related to maintaining continuous surveillance of a moving, unpredictable target, which has received a lot of attention in the literature of pursuit-evasion games. For most works in this category, issues related to the optimization of visibility based on the structure of the environment are critical, so connections with art-gallery types of problems [7] and similar viewpoint optimization problems [8] are strong.

In [9] the problem of planning the trajectory of a single observer trying to maximize visibility of a fully-predictable or partially-predictable target is considered. The problem setting is augmented in [10] which examines the problem of planning the paths of several observer robots whose goal is to provide continuous visual coverage of several unpredictable targets.

Generally, continuous visibility maintenance has been considered mostly for the case of adversarial targets. Such is the case for example in [5], [4], [11].

Maximum visibility planning that takes into account the dynamics of the observer has been studied in the context of UAV's in [12], where both the time of visibility and the time of execution of the trajectory are taken into account. This is the closest work to the one we are presenting here. The proposed solution is to discretize the state-space into a grid, pre-compute the visibility computations on a visibility table and then plan a path using dynamic programming. The differences with our work are that: we are planning trajectories for both the observer and the target, and we do not discretize the state space, so we do not need a trajectory optimizer that modifies the computed controls (to correct discretization errors) and ensures that they are valid for the vehicle dynamics.

While most existing works that examine this problem assume a known map of the environment, there are approaches that do not, such as [13]. In this case it becomes inherently impossible to do offline planning, so the authors propose local, online control laws based on the surrounding obstacles. Departure from the frequent assumption of centralized planning is also done in [14].

Kinodynamic planning takes into account the dynamics of the vehicle at hand, which is what we are interested in. This topic has been addressed multiple times in the literature, from [15] to more practical, sampling-based approaches, such as the kinodynamic RRT [16], and more recently, the kinodynamic RRT* with asymptotic optimality guarantees, both for holonomic and some classes of non-holonomic systems [6], [17].

III. MAX VISIBILITY PLANNING

A. Problem Formulation

We consider the problem of planning trajectories for two robots with different maximum velocities. We will designate the target as the leader (l) and the other one the follower (f). We are going to limit our discussion to trajectories that lie on the two-dimensional Euclidean plane, despite the fact that our conclusions and the associated algorithms generalize readily to higher dimensional spaces.

Formally, a trajectory is a tuple $\gamma = (x, u, T)$ where $x : [0, T] \to X$ is a timed path in the state-space X, $u : [0, T] \to U$ is a control function in the control space U, and T is the time of completion of the trajectory. We are therefore searching over the space of all possible trajectories for the leader $\gamma_l = (x_l, u_l, T_l)$ and the follower $\gamma_f = (x_f, u_f, T_f)$ so that the visibility of the two robots is maximized while they are performing another task that might oppose that objective, although not directly, as in the case of the adversarial target in pursuit-evasion. For this work we have selected this task to be reaching a goal state in shortest-time, although other similar tasks include multi-robot exploration or coverage.

We formulate the cost function that we are trying to minimize as a linear combination of the time spent by the robots to achieve their tasks, and the time spent without line-of-sight:

$$J(\gamma_l, \gamma_f) = T_l + T_f + \alpha\, T_{\text{occlusion}}(\gamma_l, \gamma_f) \tag{2}$$

where the duration of mutual occlusion is:

$$T_{\text{occlusion}}(\gamma_l, \gamma_f) = \int_0^T \mathbb{1}_{[x_l(t) \text{ cannot see } x_f(t)]} dt \tag{3}$$

with $T = \max\{T_l, T_f\}$. As α goes to infinity, the problem becomes to plan a pair of trajectories that always maintain mutual visibility. For computational purposes, the integral can be approximated as a sum where the line-of-sight indicator function is evaluated over a set of points that sample the two trajectories in equal time intervals. This formulation of the cost function J means that there is no need to compute the visibility region for every point in each of the two trajectories, which would be computationally demanding. Instead, all that is needed to compute the visibility indicator function is whether the line segment from $x_l(t)$ to $x_f(t)$ intersects any of the obstacles in the state-space X. For polygonal representations of obstacles this is a well-understood operation in computational geometry. For occupancy-grid representations it is also easily computable. Our problem formulation can therefore be stated as follows:

$$\begin{aligned}
(\gamma_l*, \gamma_f*) &= \arg\min_{\gamma_l} \min_{\gamma_f} J(\gamma_l, \gamma_f) \qquad (4)\\
\text{subject to} & \\
\dot{x}_l(t) &= L(x_l(t), u_l(t)) \\
\dot{x}_f(t) &= F(x_f(t), u_f(t)) \\
x_l(t), x_f(t) &\in X_{\text{free}} \\
x_l(T_l) &\in X^l_{\text{goal}} \\
x_f(T_f) &\in X^f_{\text{goal}}
\end{aligned}$$

Fig. 1. The follower is blue (solid) and the leader is red (dashed). For appropriate choice of the difference in maximal speeds between the two robots and the weight of the visual occlusion term, the cost of following the shortest path to the common goal (green rectangle) is higher than the cost of going to the other side of the obstacle where the leader is. So, the cost J violates the triangle inequality.

where L and F describe the dynamics of the leader and the follower respectively, which makes this an instance of a kinodynamic planning problem. Frequently used models for the dynamics equations are: omnidirectional systems, differential drive, Dubins and Reeds-Shepp cars. The examples we have used in this work assume omnidirectional robots, each having a 2-D state describing its position on the plane, with dynamics $L(x_l, u_l) = u_l$, and upper-bounded speed $U_l = [0, V_l]$. Similarly for the follower $F(x_f, u_f) = u_f$ and $U_f = [0, V_f]$. This was done because the complexity required in the planning of exact, optimal control trajectories for the other (non-holonomic) models is higher.

B. Properties of the cost function

Due to the presence of the mutual occlusion term, the cost function J is not separable in each trajectory, neither additively, nor multiplicatively. This means that we can not split the optimization problem into two independent problems. Also, viewed from the standpoint of a binary relation, it is obvious that the line-of-sight indicator function is not transitive, which means that even if both robots can see a point in X that does not imply that they can see each other.

If $\gamma_1 + \gamma_2$ denotes the concatenation of the two trajectories then $J(\gamma_1, \gamma) \leq J(\gamma_1 + \gamma_2, \gamma)$ and similarly for the follower's dimension, so J is monotone.

It does not satisfy the triangle inequality, due to the presence of obstacles. Consider the example shown in Fig. 1 where the two robots have the same maximum speeds, and the same goal, but there is an obstacle in between their shortest paths to the goal. So, the cost of one robot taking a detour is lower than that of taking the shortest path to the goal.

The cost of the shortest path is at most $\alpha + 2$ times bigger than the optimal cost, so finding the shortest-time trajectories is an $\alpha + 2$ approximation algorithm.

C. Variations of the formulation

The formulation presented in Eq. (4) is aimed at offline centralized planning by both robots. There are many scenarios where the roles of the leader and the follower are inherently distinct, such as in the case of a robot photographer that is performing visual servoing after a moving leader. The goal of the follower is to maximize its visibility of the target, without having a specific goal to reach and without having prior knowledge of the leader's goal. If the leader is standing still the follower is happy to do the same, and is only reactively forced to move if the leader escapes its field of view.

Such a reactive follower is endowed with two abilities: to visually track the leader when line-of-sight is unobstructed, and to estimate the leader's position if it disappears after turning around a corner[2]. We can include both of these behaviors in the follower's reactive planner $\gamma_f = G(\gamma_l)$. Given this deterministic reaction to the leader's trajectory the burden of planning in this scenario falls on the leader. The leader, being cooperative, has to choose a trajectory that will minimize $J(\gamma_l) = T_l + \alpha T_{\text{occlusion}}(\gamma_l, G(\gamma_l))$ under the constraints expressed in Eq. (4), with the exception that the follower does not care where it ends up, i.e. $x_f(T_l) \in X_{\text{free}}$.

Another variation of the problem presented in Eq. (4) has to do with the cost function. If we take into account the orientation of the two robots and make the pragmatic assumption that they have limited field of view, both in angle and in maximum radius, then both of these requirements can be incorporated in the visibility indicator function.

IV. ALGORITHMS

The problem formulation in Eq. (4) could be optimized locally by resorting to numerical differentiation or generic optimal control solvers. These methods, however, make it difficult to describe the shape of arbitrarily complex obstacles in the environment, and also frequently fail in the presence of discontinuous or non-smooth objective functions. This is one of the reasons we choose to examine solutions to the maximum visibility planning problem that are based on the RRT*, because the only operations it requires are: to probe the structure of the environment via a curve-in-obstacle test which can be approximated by discrete point-in-obstacle tests, and to concatenate locally optimal trajectories. The other reason it is a better choice in our setting is that it was recently shown to be asymptotically optimal for kinodynamic planning for dynamical systems that satisfy a few assumptions (see [18], [6]):

(a) The cost function J must be monotonic, which was addressed above, and upper-bounded by the length of the path. The latter requirement is true for maximum visibility planning. Given the fact that one term in the cost is the last time of arrival to the goal, at each moment in time, at least one of the robots is travelling at full speed. Therefore, $J(\gamma_l, \gamma_f) \leq (\alpha + 2)\ \max\{T_l, T_f\} \leq (\alpha + 2)\ \frac{\max\{||x_l||, ||x_f||\}}{\min\{V_l, V_f\}}$ where $||x||$ denotes the length of the path prescribed by trajectory γ.

(b) The dynamical system must belong to a class that is a superset of locally controllable systems. In our case this requirement is satisfied for omnidirectional robots, but it is also satisfied for many non-holonomic systems.

[2]As long as the follower is able to have a single-homotopy hypothesis of the leader's position, so that it does not have to search for the leader. If the leader travels along a branch of an intersection with self-similar branches then the follower has to perform search.

The critical improvement that RRT* offers over the classical RRT [16] formulation is essentially a rewiring step in which existing nodes around a small neighborhood of a newly-inserted node are probed to check if they would benefit in terms of cost to reach the root by replacing their existing parent with the newly-inserted node. If so, the local trajectory connecting the old parent to its child node is replaced by the trajectory from the newly inserted node to said child node. This rewiring step enables the sequence J_n, which represents the minimum cost of a state node on the tree after n iterations, to converge almost surely to the optimal cost.

A. Joint Planning

We initially set up the centralized planning problem for the RRT* based on the joint 4-D state that contained the planar positions of the two robots. Since the state is the Cartesian product of the individual robots' state spaces this proved to require many iterations to converge in practice. To contrast this setting with planning a 2-D state on the Euclidean plane, let p_n be the probability that at the n^{th} iteration we will sample a 2-D state that will improve the trajectory of the first robot with respect to the second. By sampling the Cartesian product of the individual state spaces the probability that we will sample a joint 4-D state that improves both robots' trajectories is p_n^2, which leads to slower convergence.

In addition, it is unclear whether there is a way to efficiently incorporate a 2-D point in the standard iteration that improves the 4-D state, in such a way that the new point improves either of the two robots' trajectories. In the standard formulation of the RRT* a new joint state (x_l^1, x_f^1) is going to connect to the closest joint state (x_l^0, x_f^0) so the cost $J(\gamma_l^{0\to1}, \gamma_f^{0\to1})$ can be computed as in Eq. 2. If we only sample, say x_f^1, and connect it to x_f^0 the cost of this trajectory is going to be dependent on future samples of the leader, so, efficiently keeping track of the dependencies between these costs is required. This situation occurs almost always during the rewiring step of the RRT*, and it is due to allowing different ending times for a joint state transition. It implies that, unless particular data-structures are used, even the operation of simply checking whether a node would benefit from changing its parent becomes non-local and linear in the number of nodes in the worst case. This is one of the main reasons why we did not use 4-D states in our solution.

One of the basic operations of the RRT* that is employed in this algorithm is Steer$((x_l^0, x_f^0) \to (x_l^1, x_f^1))$ which returns a trajectory to the newly-sampled state (x_l^1, x_f^1). This is the function that essentially performs optimal control from one sampled state to the other, while respecting the dynamics of the vehicle. In the case of omnidirectional robots with bounded speeds this function allows both robots to travel at their respective full speeds when the optimal trajectories from the current state to the next are either fully unobstructed, or fully obstructed. Optimization is only necessary in partially obstructed trajectories, in which case the planner must allow at least one of the robots to go full speed at any given time, and search for the optimal speed of the other robot. This is better illustrated in Fig. 2. It is worth mentioning at this point that the best response trajectory of the follower to a given fixed target trajectory is not unique. Rather, the best responses form an equivalence class in terms of cost, which is better illustrated in Fig. 3. If it is deemed that one of the robots has to slow down at a corner to increase its time of visibility to the other robot, the three following trajectories: waiting in place for some time and then moving at full speed; going on a straight line at the required speed; taking a detour while still seeing the target, are all equivalent responses as long as the robot ends up at its next state by the same time with the same cost. So, under the proposed formulation of the maximum visibility planning, the best responses are not unique.

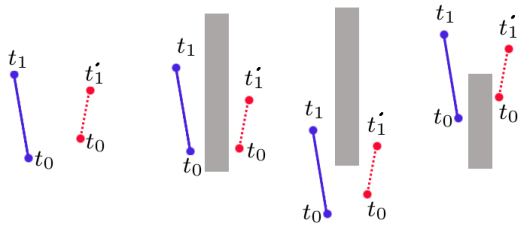


Fig. 2. There are four possible transitions to the next pair of states. In the first the two trajectories are fully visible, while in the second they are fully occluded. In the first two cases both robots should go full speed. In the third case one robot might be required to slow down. In the fourth case both robots should go full speed, but the paths might not be straight lines.

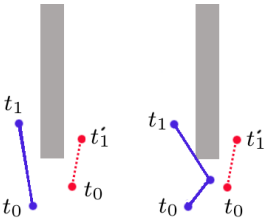


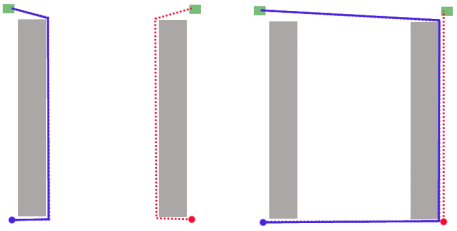
Fig. 3. The transition from mutually visible states to occluded states. One robot is going to travel at full speed, while the other might have to slow down to trade visibility time for travel time (in this case the blue robot). The trajectory of the blue robot is under-constrained. It can wait in place for some time and then go straight at full speed, as is shown on the left, or it can make a detour and then reach its next state as it does on the right.

B. Alternating Trees (Block Coordinate Descent)

To overcome the issues of slower convergence and handling of distinct termination times for the two robots, we decided to split the planning process into two RRT*'s, one for each robot. Informally, the main idea is that we keep the current best trajectory for one robot fixed, while the other is using its 2-D RRT* to optimize for time and mutual occlusion. Then they alternate and repeat this iteration, essentially performing block coordinate descent. The process, outlined in Alg. 1, is guaranteed to converge because J_n is a decreasing sequence bounded below by zero. Formally determining whether the limit is a local minimum is left as future work, however, it is clear that it is not a global minimum. The reason for this is illustrated in

Algorithm 1 Block coordinate descent

Let $\tau_l :=$ shortest path of leader to X^l_{goal}
Let $\tau_f :=$ shortest path of follower to X^f_{goal}
Use RRT* to compute γ_f given τ_l
Use RRT* to compute γ_l given τ_f
if $J(\tau_l, \gamma_f) < J(\gamma_l, \tau_f)$ **then**
  $\gamma_l \leftarrow \tau_l$
else
  $\gamma_f \leftarrow \tau_f$
end if
for $i := 1...N$ **do**
  Use RRT* to compute τ_f given γ_l
  $\gamma_f = \tau_f$ if $J(\gamma_l, \tau_f) < J(\gamma_l, \gamma_f)$
  Use RRT* to compute τ_l given γ_f
  $\gamma_l = \tau_l$ if $J(\tau_l, \gamma_f) < J(\gamma_l, \gamma_f)$
end for



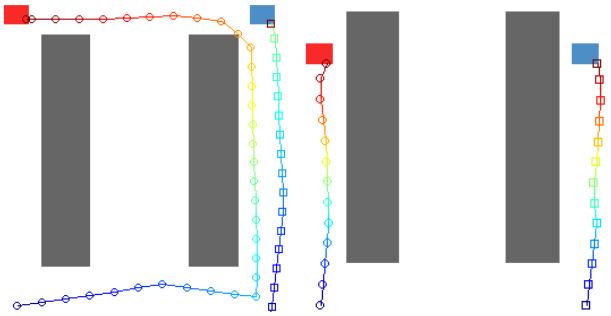
(a) Global minimum (b) Local minimum

Fig. 4. Example of a potential local minimum: the two robots have equal maximal speeds and a high weight on the visual occlusion cost. Due to the initialization, after the leader crosses to the follower's side, none of the two will have a reason to consider the middle hallway during their iteration, thus reaching a fixed point (b). However, the optimal pair of trajectories is for both to go through the middle as shown in (a).

Fig. 4. Essentially, the pair of trajectories that the algorithm outputs might end up belonging to either of two homotopy classes: the one exemplified by the leader's shortest path or the one exemplified by the follower's shortest path –due to the initialization– and will ignore the rest, thus losing the optimum. To help avoid this issue, a search in the space of possible homotopy classes is required, a representation of which has been explained in [19].

The basic operations involved in this block coordinate descent algorithm are all conditional on the other robot's fixed trajectory. This means that every node (representing a state) on the tree needs to keep track of the time at which it was reached by its parent. With that timing information it can lookup the other robot's position at that time to check for visibility. These basic operations are minor modifications of the standard methods found in RRT*:

Steer$(x_0 \rightarrow x_1|\gamma)$, which returns the local trajectory of a robot from one state to the next, taking into account the dynamics of the vehicle and the fixed trajectory of the other robot. It is the function that performs optimal control locally, without checking if the trajectory collides with obstacles. It is computable even for some cases of non-holonomic systems, such as the Dubins car [20], [21]. Whether the trajectory



(a) (b)

Fig. 5. The follower's trajectory is annotated by circles and the leader's is annotated by squares. Both have the same maximum speed. The color gradient indicates the time of arrival at a location (blue is zero). (a) The penalty on visual occlusion is $\alpha = 10$. Instead of heading directly to the goal, the follower does a wide detour to maximize its visual contact with the leader. (b) The penalty on visual occlusion is lowered to $\alpha = 1$. In this case there is no incentive for the follower to take a detour.

returned by Steer is going to be accepted depends on the collision-checking function.

A distance function dist$(x_0 \rightarrow x_1|\gamma)$ which returns the minimum cost J for a robot to reach state x_1 from x_0. Based on this distance Nearest$(x|\gamma)$ returns the closest node on the tree and NearVertices$(x, r|\gamma)$ performs a range search and returns all the nodes on the tree that are reachable from x at cost less than r.

A CostToGo$(x|\gamma)$ function is required that returns an underestimate of the real cost of reaching the goal from state x. Its role is to reject the insertion of nodes that are definitely not going to be parts of the minimal path. In our case we use the time of the straight-line trajectory from x directly to the goal, ignoring any obstacles and visual occlusion.

A rewiring function that modifies the parents of certain nodes when a new node is inserted to the tree. For each node that replaces its parent to improve its cost, this function needs to update the time that each descendant in the subtree of said node was reached. When the time of reach of each descendant is updated, their line-of-sight with the appropriate point on the fixed trajectory needs to be re-evaluated, which is a costly operation. It differs, however, from the rewiring operation in the case of 4-D state because determining whether a node would benefit from a parent replacement is done in constant-time, and is a local operation.

V. EVALUATION

We evaluated the block coordinate descent algorithm on simulated examples that illustrate the behavior of wide turning around corners and slowing down whenever necessary – indicative of a cooperative target. In particular, Fig. 5(a) shows this plan, where the faster robot performs a wide turn to keep the slower robot visible which is heading to its goal at full speed. When the robots have equal maximal speeds and the penalty of the visual occlusion term is low then both robots perform their shortest-path trajectories to the common goal region. This is shown in Fig. 5(b). The other two example runs, shown in Figs. 6 and 7, also illustrate the

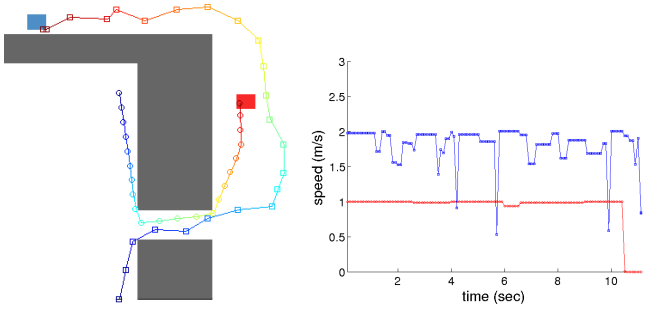


Fig. 6. The follower's trajectory is annotated by circles. The leader is 2x faster than the follower and the penalty for occlusion is $\alpha = 10$. (a) The leader takes very wide turns and slows down at corners as shown in the plot of controls (b). The colors indicate the times at which locations were reached.

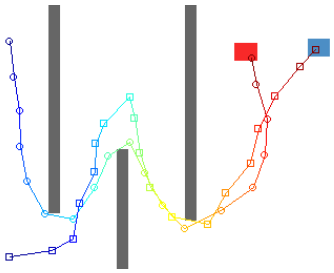


Fig. 7. Both robots have the same maximal speed, and the penalty for occlusion is $\alpha = 10$. The leader performs a wide turn around the middle obstacle and maintains full visibility until they reach the goal.

wide turning behavior and slow-down of the faster robot that is equivalent to waiting at the corner.

In all of these example runs the cost of the alternative strategy of maintaining continuous line-of-sight while heading towards their goals is higher than the trajectories shown here. This is because if we motion plan for the two robots as if they were the endpoints of a bar having variable length, in such a way that the bar always remains in free space, the speed difference between the two robots would not matter and the two robots would likely travel at the speed of the slowest robot.

VI. CONCLUSIONS

We presented a formulation of the problem of allowing one robot (the "photographer") to follow another robot ("the subject") through a planar environment while maintaining visual contact to the maximum degree consistent with an efficient traversal, namely the shortest path to a goal. In certain environments these two objectives can be conflicting, so we proposed an objective function that balances the two and which can be optimized for instance by sampling-based approaches. We formulated an algorithm for the optimization of this objective function that is based on the asymptotically-optimal RRT*, which takes into account the dynamics of the vehicle, and we showed preliminary results of pairs of centrally-planned trajectories that show collaborative behavior around corners, where visibility might be lost.

ACKNOWLEDGMENTS

The authors would like to thank NSERC for supporting this work through the NSERC Canadian Field Robotics Network (NCFRN), as well as the Walter Sumner Foundation.

REFERENCES

[1] C. Schroeter, M. Hoechemer, S. Mueller, and H.-M. Gross, "Autonomous Robot Cameraman - Observation Pose Optimization for a Mobile Service Robot in Indoor Living Space," in *IEEE ICRA*, 2009, pp. 424–429.

[2] R. Bodor, A. Drenner, M. Janssen, P. Schrater, and N. Papanikolopoulos, "Mobile Camera Positioning to Optimize the Observability of Human Activity Recognition Tasks ," in *IEEE IROS*, 2005, pp. 1564–1569.

[3] I. M. Rekleitis, G. Dudek, and E. Milios, "Multi-robot collaboration for robust exploration," *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1-4, pp. 7–40, 2001.

[4] S. Bhattacharya and S. Hutchinson, "On the Existence of Nash Equilibrium for a Two-player Pursuit–Evasion Game with Visibility Constraints," *The International Journal of Robotics Research*, vol. 29, no. 7, pp. 831–839, Dec. 2009.

[5] S. Hutchinson, "Approximation Schemes for two-player pursuit evasion games with visibility constraints," in *Robotics: Science and Systems (RSS)*.

[6] S. Karaman and E. Frazzoli, "Optimal Kinodynamic Motion Planning using Incremental Sampling-Based Methods," in *IEEE Conference on Decision and Control*, 2010.

[7] J. O'Rourke, *Art Gallery Theorems and Algorithms*. Oxford University Press, 1987.

[8] L.-T. Cheng and Y.-H. Tsai, "Visibility Optimization Using Variational Approaches," *Communications in Mathematical Sciences*, vol. 3, no. 3, pp. 425–451, 2005.

[9] S. M. Lavalle, H. Gonzalez-Banos, C. Becker, and J.-C. Latombe, "Motion Strategies for Maintaining Visibility of a Moving Target," in *IEEE International Conference on Robotics and Automation*, no. April, 1997, pp. 731–736.

[10] R. Murrieta-Cid, B. Tovar, and S. Hutchinson, "A Sampling-Based Motion Planning Approach to Maintain Visibility of Unpredictable Targets," *Autonomous Robots*, vol. 19, no. 3, pp. 285–300, Dec. 2005.

[11] V. Isler, S. Kannan, and S. Khanna, "Randomized pursuit-evasion with limited visibility," in *In Proc. of ACM-SIAM Symposium on Discrete Algorithms (SODA*, 2004, pp. 1053–1063.

[12] K. Lee, "Visibility Maximization with Unmanned Aerial Vehicles in Complex Environments," M.Sc. Thesis, MIT, 2010.

[13] H. Gonzalez-Banos, C.-Y. Lee, and J.-C. Latombe, "Real-time combinatorial tracking of a target moving unpredictably among obstacles," in *IEEE ICRA*, vol. 2, 2002, pp. 1683–1690.

[14] L. E. Parker, "Distributed Algorithms for Multi-robot observation of multiple moving targets," *Autonomous Robots*, vol. 12, no. 3, pp. 231—-255, 2002.

[15] B. Donald, P. Xavier, J. Canny, and J. Reif, "Kinodynamic motion planning," *Journal of the ACM*, vol. 40, no. 5, pp. 1048–1066, Nov. 1993.

[16] S. Lavalle and J. Kuffner, "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[17] S. Karaman and E. Frazzoli, "Sampling-based Optimal Motion Planning for Non-holonomic Dynamical Systems," in *IEEE International Conference on Robotics and Automation*, no. 1, 2013.

[18] ——, "Sampling-based Algorithms for Optimal Motion Planning," *International Journal of Robotics Research (IJRR)*, vol. 30, no. 7, pp. 846–894, 2011.

[19] S. Bhattacharya, M. Likhachev, and V. Kumar, "Identification and Representation of Homotopy Classes of Trajectories for Search-based Path Planning in 3D," in *Robotics: Science and Systems (RSS)*, 2011.

[20] L. Dubins, "On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, 1957.

[21] P. Soueres, J. D. Boissonnat, and J.-P. Laumond, "Optimal trajectories for nonholonomic mobile robots," in *Robot Motion Planning and Control*. Springer, 1998, pp. 93–170.