Active visual tracking in multi-agent scenarios

Yiming Wang and Andrea Cavallaro
Centre for Intelligent Sensing, Queen Mary University of London
{yiming.wang, a.cavallaro}@qmul.ac.uk

Abstract

We propose an active visual tracker with collision avoidance for camera-equipped robots in dense multi-agent scenarios. The objective of each tracking agent (robot) is to maintain visual fixation on its moving target while updating its velocity to avoid other agents. However, when multiple robots are present or targets intensively intersect each other, robots may have no accessible collision-avoiding paths. We address this problem with an adaptive mechanism that sets the pair-wise responsibilities to increase the total accessible collision-avoiding controls. The final collision-avoiding control accounts for motion smoothness and view performance, i.e. maintaining the target centered in the field of view and at a certain size. We validate the proposed approach under different target-intersecting scenarios and compare it with the Optimal Reciprocal Collision Avoidance and the Reciprocal Velocity Obstacle methods.

1. Introduction

Robots that actively track a person can offer assistance in shops, airports and museums. Active tracking with a camera aims to maintain the target at a desired position and size on the image plane [12] by either optimization [19] or various feedback controllers [8, 12]. The motion of an active visual tracker is constrained by its limited field of view and the target dynamics. When multiple robots coexist in a scene and track their own target, collisions are likely to occur when the paths of the targets intersect each other. For multi-robot collision avoidance, as robots react to the actions of other robots, reciprocal avoidance strategies are essential to avoid oscillations.

Methods exist that apply human strategies for safe robot navigation, where the strategy can be learned from extracted trajectories [10] or based on cognitive studies [11]. A popular method for multi-robot collision avoidance is Optimal Reciprocal Collision Avoidance (ORCA) [17]. ORCA is a velocity-based method that guarantees collision-free motion in a densely-packed environment. Each robot independently computes the closest collision-avoiding velocity to its preferred velocity (i.e. the velocity the robot would move at if there were no other robots in its way). Assuming that each robot can sense the state of neighboring robots and infer their preferred velocities, ORCA does not require communication among robots.

When targets are intensively intersecting, robots may face the empty-set problem (i.e. without accessible collision-free velocities [17]). The original ORCA adjusts the pair-wise velocity constraints in order to have at least one accessible velocity in the case of an empty-set occurrence [17]. However, by doing so the robot motion is completely dependent on neighboring robots. Instead, we prevent the occurrences of the empty-set problem by adapting the pair-wise responsibilities. The pair-wise responsibility can assign the right of way to the agent with a constant higher priority [6], while we set the pair-wise responsibility at run time to reduce the responsibility in avoiding other robots for a robot with a smaller set of accessible collision-avoiding velocities.

In this paper, we propose an adaptive ORCA (A-ORCA) method that addresses the empty-set problem under challenging target-intersecting scenarios. The proposed method adapts the pair-wise responsibility with the objective of increasing the accessible collision-avoiding velocities of a pair of robots in a fair way. The final collision-avoiding control is achieved via an optimal controller that accounts for both view performance and motion smoothness, and satisfies the velocity constraints derived with A-ORCA. We compare the proposed method with ORCA [17] and RVO [16] under different target-intersecting scenarios.

2. Related work

ORCA is based on the concept of Velocity Obstacle (VO). VO was originally proposed for a robot to avoid moving obstacles with known paths by defining the set of velocities that can lead to collisions within a time horizon [7]. In multi-robot scenarios, each robot is a reactive moving obstacle for other robots. Therefore oscillations are likely to occur if a robot avoids other robots without accounting for their reactive nature. Reciprocal Velocity Obstacle (RVO) avoids reactive robots by using the average of the preferred
velocities together with VO, in order to obtain a set of reciprocally collision-avoiding velocities [16]. Based on RVO, ORCA further defines the set of velocities that are both reciprocally collision-avoiding and close to the preferred velocity [17]. Each robot shares equal responsibility to avoid another robot. The original ORCA considers disk-shaped robots with a holonomic model constrained by a speed limit (a circular boundary in the velocity space). Derivative works extend ORCA in terms of robot shapes [3, 15] and robotic kinematics [2, 3, 4, 15].

ORCA-related works apply to robots navigating to fixed goal positions [2, 4, 5, 15, 16, 17]. In the field of active visual tracking, the main focus is the control for visual fixation [8, 12] or for minimizing tracking uncertainty [19]. While there is work for multiple robots tracking one target, collision avoidance is achieved via maintaining a predefined formation [13].

To the best of our knowledge, no works have addressed multi-robot collision avoidance among robots that are independently conducting active visual tracking. Collision avoidance is more challenging during view maintenance as the paths of the targets may frequently intersect with each other.

Methods for multi-robot collision avoidance with ORCA [2, 4, 5, 15, 16, 17] are compared in Table 1.

<table>
<thead>
<tr>
<th>Ref</th>
<th>Application</th>
<th>Agent shape</th>
<th>Agent kinematics</th>
<th>Equal responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[2]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[3]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[4]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[5]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[6]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[7]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Prop</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

The robot heading direction. Each $c_i$ aims to fixate its target by maintaining it at a certain distance away in the camera heading direction while avoiding collisions with any other robot $c_j$, i.e. $d_{ij}(t) > 2r, \forall j, j \neq i$.

We assume that the assignment between robots and targets is given and ambient cameras localize targets and robots. These positions are communicated to the robots [4, 14, 18]. At initialization, robots exchange (or are given) their preferred velocities. Using the positions of nearby robots and the target to follow, a robot calculates at each time step, $t$, its preferred velocity for active tracking in order to maintain the target at the desired position in the FoV, i.e. the FoV center [18], in a smooth manner.

Using the positions and preferred velocities of neighboring robots, each $c_i$ derives the set of velocities that guarantees no collisions in $\tau$ time steps and computes the final control within that set. A short time horizon $\tau$ leads to late avoidance, which can result in empty-set cases as the closeness among robots leaves no space to move; whereas a large $\tau$ averts early avoidance that leaves more space to move, but imposes restrictive velocity constraints which can also cause empty sets [4].

### 3.2. The controller

We use a car-like kinematic model [2]. The robotic control $a_i(t)$ consists of speed $v_i(t)$ and steering angle $\omega_i(t)$, and is bounded as $|v_i(t)| \leq v_{\text{max}}$ and $|\omega_i(t)| \leq \omega_{\text{max}}$. We compute the robotic control $a_i(t)$ by minimizing two cost functions that encode the objectives of maintaining the target at the FoV center and motion smoothness.

Let $p_{\text{in}}^d(t + 1) = \frac{2|d_{\text{in}}(t+1)\cdot v_r|}{\tau_v}$ and $\rho^d_{\text{in}}(t + 1) = \frac{2\delta_{\text{in}}(t+1)}{\tau_o}$ be the ratio of the distance and angle deviation to the FoV center at $t + 1$ with respect to half view range $\frac{\psi}{2}$ and half view angle $\frac{\phi}{2}$. We estimate $d_{\text{in}}(t+1)$ and $\delta_{\text{in}}(t+1)$.
based on the predicted target state and predicted robot state given the control \(a(t)\).

The first cost function, \(J_1(t)\), penalizes the deviation of the target position from the FoV center [18]:

\[
J_1(t) = \exp \left( \sqrt{\rho_n^2(t+1)^2 + \rho_d^2(t+1)^2} \right),
\]

\(J_1(t) < \exp(1)\) ensures that the target locates within the FoV. The minimum, \(J_1(t) = 1\), is achieved when the target is at the center of the FoV.

The second cost function, \(J_2(t)\), penalizes accelerations/decelerations induced by \(J_1(t)\):

\[
J_2(t) = \exp \left( \frac{\|v(t+1) - v(t)\|}{v_{\text{max}} + \|v(t)\|} \right),
\]

where \(v_{\text{max}} + \|v(t)\|\) is the maximum speed difference of a robot between two consecutive time steps. The minimum, \(J_2(t) = 1\), is achieved when the robot does not change velocity.

The control is computed by minimizing with brute force the two cost functions:

\[
a_i(t) = \arg\min_{a} \left( \lambda J_1(t) + (1 - \lambda) J_2(t) \right),
\]

where \(\lambda \in [0.5, 1]\). \(\lambda\) starts at 0.5 as the main objective is to maintain the target centered in the FoV. We vary \(\lambda\) based on whether the robot has its target inside the FoV. When the robot loses its target from the FoV, the objective is to re-capture it as soon as possible, and we therefore set \(\lambda = 1\) without accounting for motion smoothness. When the robot has its target inside the FoV, the objective is to maintain the target centered in the FoV with smooth motion. Based on a sensitivity test on \(\lambda\), we experimentally set \(\lambda = 0.6\) as it provides the best trade-off between the decrement of \(J_1(t)\) and the increment of \(J_2(t)\).

### 3.3. Accessible collision-avoiding velocity set

Let us consider a pair of robots \(c_i\) and \(c_j\) at position \(p_i\) and \(p_j\) respectively, aiming to achieve their preferred velocity \(v_i^*\) and \(v_j^*\) (Fig. 2(a)). Let \(\mathcal{O}_{i,j}^\tau(0)\) be the velocity obstacle of \(c_j\) induced by \(c_i\), which is the set of relative velocities of \(c_i\) with respect to \(c_j\) that can lead to a collision within a time horizon \(\tau\) (Fig. 2(b)):

\[
\mathcal{O}_{i,j}^\tau(0) = \{ v \mid \exists t \in [0, \tau], \|v\| \geq \|p_{j} - p_{i} - 2t\| \},
\]

where \(p_{i,j} = p_j - p_i\) is the relative position of \(c_j\) with respect to \(c_i\). The set of collision-avoiding relative velocities for \(c_i\) to avoid \(c_j\) in \(\tau\) time steps, \(\mathcal{A}_{i,j}^\tau(0)\), is therefore:

\[
\mathcal{A}_{i,j}^\tau(0) = \{ v \mid v \notin \mathcal{O}_{i,j}^\tau(0) \}.
\]

\(\mathcal{A}_{i,j}^\tau(0)\) refers to the set of robots within the avoidance range of \(c_i\). The final set of accessible collision-avoiding

---

1To simplify the notation we omit \(t\) from here.
is influenced by the pair-wise responsibility sharing. \( v^*_i \) and \( v^*_j \) are the preferred velocities of \( c_i \) and \( c_j \) respectively. The collision-avoiding velocities of (a) \( c_i \) induced by \( c_j \) and (b) \( c_j \) induced by \( c_i \) with equal responsibility, where \( c_j \) is likely to experience empty-set if surrounded by other robots. The collision-avoiding velocities of (c) \( c_i \) induced by \( c_j \) and (d) \( c_j \) induced by \( c_i \) with adaptive responsibility.

Velocities for \( c_i \), \( A^{*,\tau}_{i,j} \), is

\[
A^{*,\tau}_{i,j} = \left( \bigcap_{\tau \in C_i^4} \bigcap_{\tau \in C_j^4} A^{*,\tau}_{i,j} \right) \cap V_i.
\]

When neighboring robots leave no space for a robot to move (i.e. \( A^{*,\tau}_{i,j} = \emptyset \)), the robot may lose its target. A robot should therefore reduce the occurrences of empty-set cases.

### 3.4. Adaptive responsibility sharing

When a robot has a small set of accessible collision-avoiding velocities induced by a neighboring robot, it is more likely to incur an empty-set case if it is surrounded by additional robots. The choice of the responsibility between a pair of robots influences the total number of accessible collision-avoiding velocities (see Fig. 3).

Let us re-write the notation \( A^{*,\tau}_{i,j} \) to \( A^{*,\tau}_{i,j}(a) \) to indicate the dependence on the responsibility. Our objective is to maximize the total number of optimal reciprocal collision-avoiding velocities, while maintaining the fairness, i.e. a balanced distribution of accessible collision-avoiding velocities between a pair of robots.

We compute the percentage of optimal reciprocal collision-avoiding velocities for \( c_i \) induced by \( c_j \) as

\[
\rho_{i,j}^{*,\tau}(a) = \frac{|A^{*,\tau}_{i,j}(a)|}{|A^{*,\tau}_{i,j}|}, \quad \text{where } |\cdot| \text{ is the cardinality of a set.}
\]

The responsibility for \( c_i \) to avoid \( c_j \) is computed by maximizing the average percentage of optimal reciprocal collision-avoiding velocities while accounting for fairness:

\[
a_{i,j} = \arg\max_a \left( f_{i,j}(a) + \rho_{j,i}^{*,\tau}(a)(1 - a) \right) / 2,
\]

where \( f_{i,j}(a) \) indicates the fairness of the percentage of optimal reciprocal collision-avoiding velocities between a pair of robots. In order to have a continuous and bounded fairness measure to combine with the average ratio, we adopt the Jain’s fairness measure [9]:

\[
f_{i,j}(a) = \frac{(\rho_{j,i}^{*,\tau}(a) + \rho_{j,i}^{*,\tau}(a))^2}{2(\rho_{j,i}^{*,\tau}(a)^2 + \rho_{j,i}^{*,\tau}(a)^2)}. \tag{9}
\]

Note that the responsibility can be a negative value as long as the responsibilities shared by a robot pair sums to 1.

We compute the optimal \( a_{i,j} \) by searching in a discretized space bounded by \([-a_{\max}, a_{\max} + 1]\). We compute the adaptive responsibility only if a robot has less than half of accessible collision-avoiding velocities when sharing equal responsibility. The adaptive responsibility \( a_{j,i} \) for \( c_j \) to avoid \( c_i \) is computed in the same way. We can guarantee \( a_{i,j} + a_{j,i} = 1 \) due to the symmetric property of the objective function.

Note that this scheme reduces the chances of the occurrence of empty-set cases but does not prevent empty-set cases. When an empty-set case occurs, we use a simple braking-recompute strategy: the robot with an empty set stops and all other robots recompute the collision-avoiding velocity sets. The process is repeated until no robot is in an empty-set situation.

### 4. Validation

We validate the proposed framework, A-ORCA, in terms of the influence of collision avoidance on target viewing performance and compare it with the avoidance strategies based on RVO [16] and ORCA [17].

We quantify the empty-set occurrences, \( \varepsilon \), as the empty-set ratio

\[
\varepsilon = \frac{1}{MT} \sum_{i=1}^{M} \sum_{t=1}^{T} b^i_t(t), \tag{10}
\]

where \( T \) is the experiment time steps, \( M \) is the number of robots and \( b^i_t(t) \) is a binary value indicating whether the set of accessible collision-avoiding velocities is empty (\( A^{*,\tau}_{i,j}(t) = \emptyset \)). Similarly, we quantify the target

![Figure 3: The accessible collision-avoiding velocity set (gray area) is influenced by the pair-wise responsibility sharing.](image-url)
The color of the trajectory indicates the target ID. (a) Scenario I with a pair of targets intersecting at different angles ranging from $0.1\pi$ to $\pi$ with step size $0.1\pi$. The target tracked by $c_1$ is fixed at angle $0$ and we vary the angle of the target tracked by $c_2$. (b) Sample Scenario II with 10 targets with randomly distributed intersecting angles between a pair of neighbors. (c) Sample Scenario III with 20 targets that intersect at random times and locations. (d) Scenario IV with 10 trajectories extracted from a video sequence of the PET2009 dataset.

viewing performance of robots, $\eta$, by the viewing ratio

$$\eta = \frac{1}{MT} \sum_{i=1}^{M} \sum_{t=1}^{T} b^v_i(t),$$

where $b^v_i(t)$ is a binary value for the presence of target within FoV. Results are averaged over 10 independent runs.

We use a real scenario and design three synthetic scenarios to investigate the influence of the intersecting angles between targets and the number of intersecting targets on the methods (Fig. 4). Targets are initialized along a circle with their initial velocity heading towards the center of the circle. Each target moves at a speed of $1m/s$ with additional Gaussian noise bounded within $[-0.1m/s, 0.1m/s]$. Scenario I increases the intersecting angle between a pair of targets from $0.1\pi$ to $\pi$ with step size $0.1\pi$ (Fig. 4(a)). Scenario II increases the number of targets from 2 to 10 and randomly sets the intersecting angle between two neighboring targets (Fig. 4(b)). Scenario III increases the number of targets from 2 to 30 and creates sparser target intersections by varying the velocity direction over time (Fig. 4(c)).

Additionally, we test the methods in Scenario IV with real people trajectories extracted from the PETS2009 dataset. The trajectories of 10 people of 60s duration is extracted from the S2L1 sequence and map to the ground plane with the provided camera calibration (Fig. 4(d)).

In all scenarios, robots follow a car-like model and are initialized with their target centered in their camera FoV. Each camera has the viewing angle $\phi = 90^\circ$ and viewing range $r_v = 5m$. We set the robot avoidance range to $2v_{\text{max}}$ as it is the worst case for a collision between a pair of robots in the next time step. We set $\tau = 6$ for Scenario II and $\tau = 3$ for the other scenarios. A larger $\tau$ allows for earlier aversion when robots are densely intersecting (Scenario II), and improves the viewing ratio.

Under Scenario I, all methods successfully avoid collisions and robots keep their target in their FoV regardless of the variation in target intersecting angles. Under Scenario II, the viewing ratio deteriorates as the number of robots, $M$, increases. When $M > 7$, robots using ORCA achieve a limited average viewing ratio (60%) with a small standard deviation because all robots get stuck at the intersecting position (Fig. 5(a)). Robots using A-ORCA achieves a higher viewing ratio than those using ORCA, as empty-set occurrences are reduced by almost half (Fig. 5(b)). The pair-wise optimization does not provide guarantee for non-empty sets, and this explains that A-ORCA fails with $M = 10$. In-
terestingly, robots using RVO may achieve higher viewing ratios than ORCA. Compared to ORCA, RVO has more restrictive velocity constraints which prevent robots from getting too close to each other and therefore provides more space for robots to avoid each other and recapture their target (see [1]).

Under Scenario III, ORCA-based methods outperform RVO as the target trajectories require less effort for avoidance (Fig. 5(c)). Although A-ORCA reduces the empty-set ratio in this scenario, it may achieve a slightly worse overall average viewing ratio compared to ORCA when $13 \leq M \leq 16$. This is because A-ORCA adapts the responsibility sharing among robots when there is an empty-set case, in order to achieve a non-empty set of collision-avoiding velocities. A-ORCA may end up with a new velocity that causes the robot to temporally lose its target from the FoV, while ORCA may simply stop and move at next time step without losing the target.

Finally, in Scenario IV people cross each other, walk in parallel and stay together without moving (see [1]). No collision occurred and the viewing ratios (empty-set ratios) achieved by RVO, ORCA and A-ORCA are 0.96 (0.032), 0.98 (0.018) and 0.96 (0.017), respectively. The results are consistent with those obtained with the synthetic trajectories when $M = 10$ (Fig. 5(c)).

5. Conclusion

We presented an ORCA-based framework for active visual tracking in multi-robot scenarios. We proposed a controller that accounts for both viewing performance and motion smoothness. We improved ORCA with an adaptive scheme to optimally split the pair-wise responsibility to reduce the empty-set cases under target intersecting scenarios. This solution leads to longer target viewing.

As future work, we will include a target avoidance functionality and validate the framework with real robots.

References