

# Occlusion-Based Cooperative Transport for Concave Objects with a Swarm of Miniature Mobile Robots

Sanjuksa Nirgude, Animesh Nema, Aishwary Jagetia

Robotics Engineering Department

Worcester Polytechnic Institute

Worcester, MA, USA

snirgude@wpi.edu, anema@wpi.edu, adjagetia@wpi.edu

**Abstract**—An occlusion based strategy for collective transport of a concave object using a swarm of mobile robots has been proposed in this paper. We aim to overcome the challenges of transporting concave objects using decentralized approach. The interesting aspect of this task is that the agents have no prior knowledge about the geometry of the object and do not explicitly communicate with each other. The concept is to eliminate the concavity of the object by filling a number of robots in its cavity and then carry out an occlusion based transport strategy on the newly formed convex object or "pseudo object". We divide our work into two parts- concavity filling of various concave objects and occlusion based collective transport of convex objects.

**Index Terms**—decentralized approach, collective transport, occlusion based transport, concave object, convex object.

## I. INTRODUCTION AND BACKGROUND

Swarm Intelligence is the collective behavior of decentralized, self-organized systems which could be either natural or man-made. Swarm Intelligence exists in nature such as the behavior of ants, bees, birds which can be used as an inspiration for finding its applications in the field of robotics. One such application is the transportation of objects using a number of mobile robots. Such tasks may seem trivial at first, but can be incredibly complicated, depending on various aspects such as shape of the object, size of the object, visibility and perception. In addition to that, the process is decentralized i.e., every agent behaves independently instead of following a fixed leader. There are various techniques to transport an object using decentralized approach. General structure of a robot's behavior, performing collective transport consists of searching the object, positioning itself around the object and then transporting the object [4].

Collective transport methods can be categorized into three categories: Pulling, Pushing and Caging. Pulling constitutes of complex mechanisms like grasping and lifting the objects, whereas caging requires robots to maintain their formation during dynamic movements. During pushing, robot's pushing positions and speed are the constraints to be addressed. Increasing the number of pushing robots increases the stability of the object as pushing force is distributed over multiple points. Also, due to hardware requirements for pulling and caging strategies, we have preferred pushing strategy for our analysis.

In the paper [4] a simple odometry based co-ordination strategy in combination with omni-directional camera has been

used. There is no communication between robots while performing collective transport task. The transportation strategy in this paper consists of four stages, namely prey discovery, team co-ordination, recruitment and transportation. In [2] a collective transport approach has been proposed by the authors using kilobots and r-ones. The agents have no prior knowledge about the object's shape,size, location of its neighbors or the object. They only know the location of the goal. The agents perceive the direction of the goal by using their light sensor(s) and apply forces on the object in the direction of perceived light. By doing so, they optimally transport objects of complex shapes to the desired location. The r-ones also execute flocking behavior in case some agents are occluded from the light source. In such a case, the robots observe the direction of their neighbors to modify their own directions. The authors proved experimentally the scalability, robustness and optimality of their approach by testing their agents under different circumstances. In [3] the authors have proposed a 5 step approach to successfully transport an object of any shape be it, concave or convex. The tasks assigned to the robots are in the following order. First the robots have to explore the environment and locate the object. Once the robots find the object, they will align themselves with the object and grasp it. The third step is object characterization. i.e., finding the centroid of the object,width,diameter, orientation etc. They do so by determining their own positions and centroid. The objects diameter determines the minimum distance from an obstacle where it is safe to rotate the object. Once the object information has been extracted, the robots then perform a path planning function. During this phase, some of the robots stay attached to the object while others explore a suitable path. The robots then navigate the object through a chosen path. The paper talks about various algorithms for each of it's steps, hence giving an idea about collective transport of complex objects.

Our paper uses an occlusion based collective transport strategy. The concept behind the occlusion based strategy is that each agent searches the object, moves towards it and then looks for the goal. All the robots push the object by moving in a direction perpendicular to the objects surface at their points of contact. This way the motion of the object will be approximately towards the goal. The robots work in a decentralized manner and conduct co-operative transport

without explicitly communicating with each other. Our project is divided into two parts in which we fill the concavity of the objects and also complete the collective transport of convex object separately. In case of collective transport experiment the location of the object is known by the robot along with its own position. In the collective transport experiment the task sequence starts with the robot orienting towards the object and moving towards it. When the robot reaches the object, it checks if the goal in our case the light source is visible from its position, if the goal is visible, it moves around the object and looks for the goal again. If the goal is not visible (or occluded), the robot starts pushing the object. Otherwise the robots moves around object, executing a left-hand-wall-following behavior. This is repeated by every agent collectively, till the object is transported to the goal. While this approach has been successfully tested on convex objects [1], concave objects pose much harder challenges. The agents can easily lose the sense of direction towards the goal and might never transport the object.

Our motivation for this project comes from the paper [1]. In which one of the major limitation was its inability to transport concave objects. We propose a method to overcome this drawback of the occlusion based strategy. The object's concavity can be eliminated by filling a number of agents in its concave contour, thus making it more 'convex' like. Another swarm of agents which will be referred to as the 'pushing agents' can then perform an occlusion based approach mentioned above by treating the object as any other convex polygon. Therefore, a concave object can be transported to the goal.

## II. PROPOSED WORK

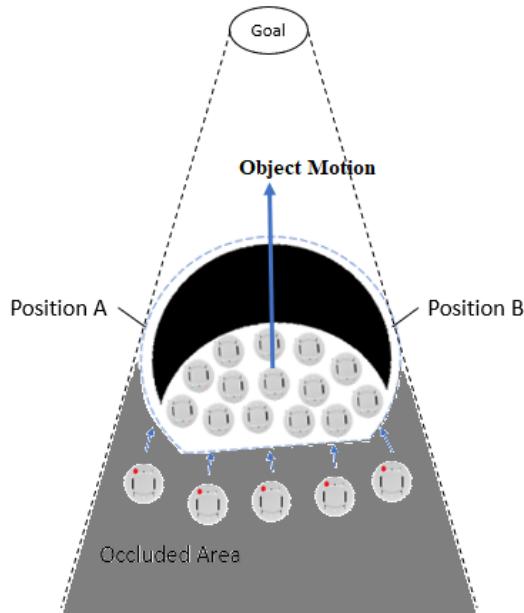


Fig. 1. Occlusion-Based Collective Transport for Concave Object

As described above, the occlusion based collective transport

strategy only works when the object is in convex shape. If the object is concave the strategy fails. Based on our strategy, initially the robot searches for an object while performing a random walk. Once the object is seen, robot approaches the object. When the robot reaches the object, it performs a left-hand-wall-following behavior around the object to check if the object is concave or convex. If the range of the angles where the object is detected relative to each robot is greater than  $\pi$ , the object is identified to be a concave object, else it is a convex object. It can be seen in the figure 2 that the robot is on the convex side of the object. Hence, the angle is less than  $\pi$  and the object will be identified as convex object. Whereas in figure 3 it can be seen that the robot senses the object at an angle more than  $\pi$ , therefore identifying it as a concave object.

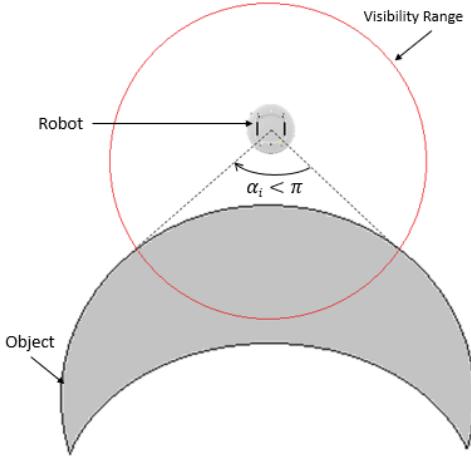


Fig. 2. When robot approaches from convex side

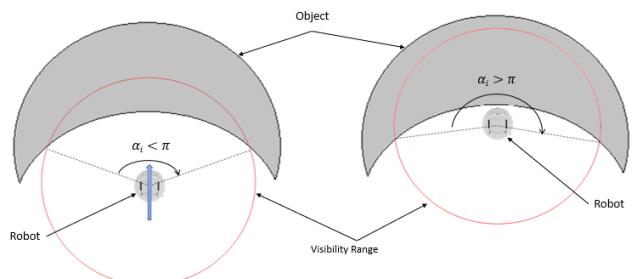


Fig. 3. When robot approaches from concave side

If the object is detected to be concave, the robot will stay at the particular position and change its robot id to "object robots". Hence, filling the concavity of the object. When a next robot arrives behind this robot it will consider the first robot as an object and check if the concavity of the object still

exists. This process will repeat until the concavity of the object is filled and robots collectively form a "**pseudo object**" which incorporates both the actual object and the object robots. Once the pseudo object changes its shape to convex, the remaining robots will change their robot id to "**pushing robots**", and the "**object robots**" will stay intact with the object. Hence, the task of converting the concave object to convex object will be accomplished.

At this point, the occlusion based collective transport strategy comes into effect. Now the robot checks if the goal can be seen from its position. If it is not visible or if the robot is in occluded region the robot must start pushing the object. Otherwise the robots should move around object, executing a left-hand-wall-following behavior.

As shown in figure 1 the robots after filling the concavity of the object now start applying the occlusion based collective transport strategy for moving the object. The robots with red LED's are now the "**pushing robots**" in the occluded area of the object. The position A and the position B are the positions of the extreme observer robots from where the goal is visible.

### III. EXPERIMENTS

Our Experiments are broadly divided into two sub-experiments. We have implemented the concave filling experiment in Buzz programming language [5] and collective transport experiment in C++. We have simulated both the experiments in ARGoS Simulator [6].

#### A. Experiment 1: Concave Filling

Concave filling for different concave shapes is shown in the figure 4. Our environment setup is shown in 7, 14 and 15 for different shapes.

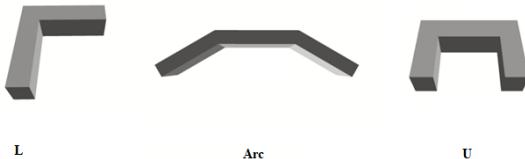


Fig. 4. Experimental Objects

*1) Experimental Setup:* We have experimented on three different shapes and size objects as shown in figure 4. In order to develop concave object we have used small convex objects to form one big concave object. Using the Concave filling algorithm discussed below, we have experimented concavity filling on these different shape objects.

*2) Algorithm:* On initialization, the robots perform a random walk to look for the object and their LED's are switched off. Once in a certain range of the object, the robots turn red indicating that it has started using their proximity sensors to detect the object geometry while performing left hand wall behavior. As soon as 4 proximity readings (theoretically it should be 5 readings) are detected simultaneously, the robot turns blue, signaling that it has turned into a '**pseudo object**'.

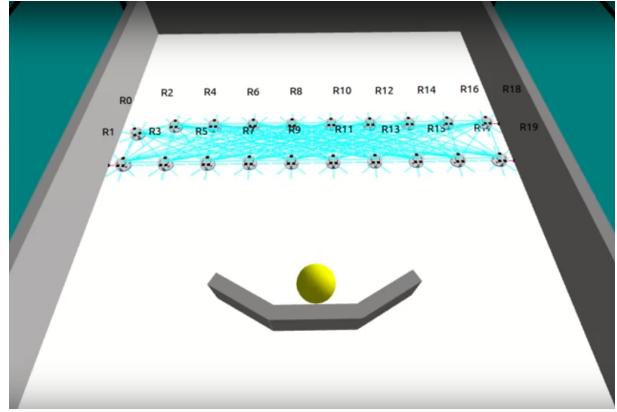


Fig. 5. Experimental Setup: Proposed Filling to form Convex Object

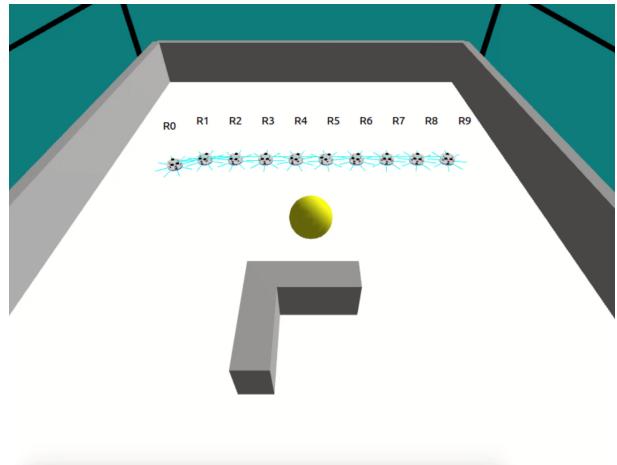


Fig. 6. Experimental Setup: Proposed Filling to form Convex Object

#### Concave Filling

```

Rotate about own axis
if Object not in range then
    | Perform Random walk;
else
    | Move to the Object;
end
if alpha greater than pi then
    | Attach to Object;
    | Change id == Object;
else
    | Perform Left Hand Wall Rule;
end

```

*3) Parameters:* We tested the following parameters for the concave filling.

- Distribution of the robots:

The distribution of the robots under a certain range slightly affected the performance of the system. It can be said that the distribution of the robots affects the concave filling but in most cases, the robots were still able to accomplish the task.

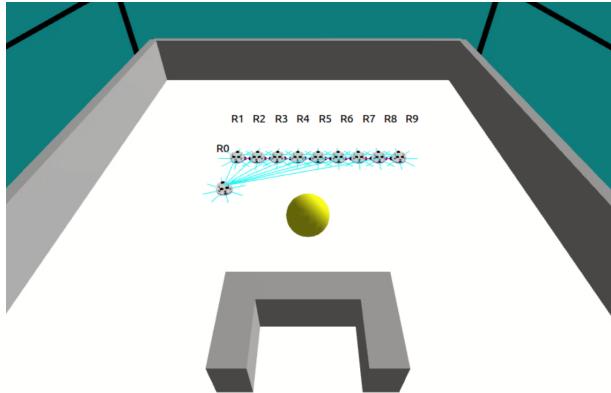


Fig. 7. Experimental Setup: Proposed Filling to form Convex Object

However, placing the robots too close to the object during initialization or orienting the robots away from the light source doesn't result in a good performance.

- Number of robots executing Concave filling:  
The number of robots affect the performance in two ways.

- 1) The total number of robots executing the task determine the degree of the filling in the sense that less number of robots would be incapable of filling the object depending on it's size. Therefore, enough robots have to be deployed to completely fill the object.
- 2) The number of robots simultaneously deployed affect the performance because the proximity sensors detect the neighbor robots yet do not distinguish them from the object. The robots are programmed to stop as soon as 4 sensors simultaneously detect readings.

- Light Source above the object:

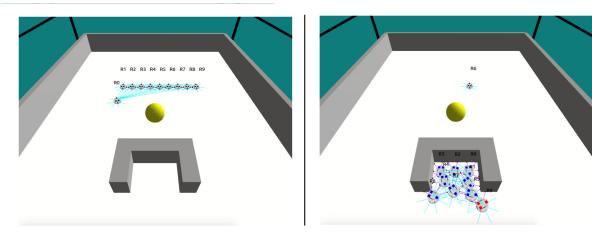
The light source above the object does not really affect the performance in a given range. The robots do not use the light source during the concave filling, they just use the light source to move towards the object. Once, in a certain range, they only use their proximity readings. On changing the location of the light source, the robots may approach the object differently though.

- Shape and size of the object:

The time and accuracy of concave filling varies according to the shape of the object. However, for the the same shape, it's size doesn't necessarily affect the performance drastically. The increase in size however may require additional robots.

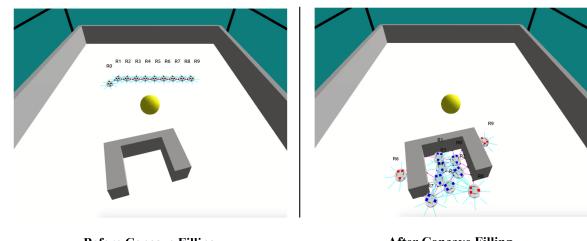
#### 4) Results:

- 1) Analysis 1: U shape as shown in figure 8
- 2) Analysis 2: U shape with 15 degree orientation as shown in figure 9
- 3) Analysis 3: L shape as shown in figure 10
- 4) Analysis 4: L Shape: Changing the Robots Initial Ori-



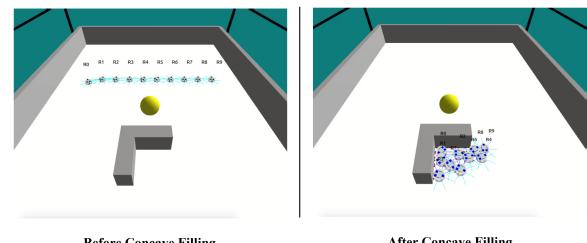
Before Concave Filling      After Concave Filling

Fig. 8.



Before Concave Filling      After Concave Filling

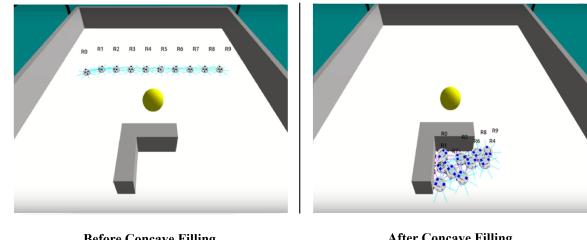
Fig. 9.



Before Concave Filling      After Concave Filling

Fig. 10.

entation as shown in figure 11



Before Concave Filling      After Concave Filling

Fig. 11.

- 5) Analysis 5: L Shape: Changing the number of Robots as shown in figure 12
- 6) Analysis 6: Arc Shape as shown in figure 13

#### B. Experiment 2: Collective Transport

In our second experiment we replicated the Occlusion-based convex object based experiments implemented in [1] for convex object.

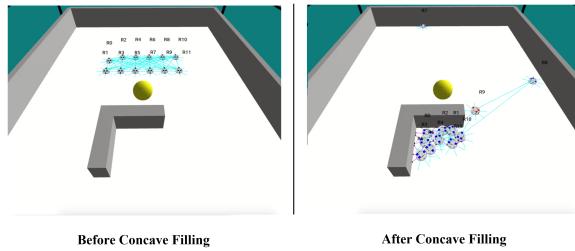


Fig. 12.

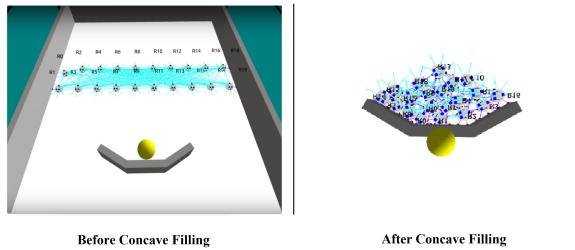


Fig. 13.

*1) Experimental Setup:* The environment of the experiment as shown in figure 15 is a rectangular arena that is bounded by the walls. The floor of the arena has white color, and its walls are painted in gray. The goal is a light source, depicted as a yellow sphere, placed at a certain height to avoid robots collision with it.

The initial configuration of a trial is illustrated in figure 14. We also have initial configurations of trials changed to randomly distributed robots in the environment. The objects centroid and the orientation was positioned randomly as shown. The sequence of the implementation is explained with the algorithm.

*2) Algorithm:* The algorithm for Occlusion based experiment is implemented for convex objects in this paper. Further it can also be performed on concave objects after filling their concavity. The experiments starts with finding the object position. The robots in the environment also have their own 'X' and 'Y' positions along with the objects position. The object position is dynamic in nature, hence as the object moves we get its updated position.

With the help of both these positions the robot can find the angle at which the object is placed from it. Once it gets the orientation direction of the object, it keeps checking if its current orientation is directing towards the object. If the orientation is towards the object, it updates the status of it being oriented. If not, it continues the process of rotating to orient itself.

Once the robot is oriented, it starts moving towards the object. After reaching the object, the robots use the proximity sensors to maintain distance from the object and follow the left hand wall rule. While performing the left hand wall rule the robots keep checking for light using their light sensors. As the robot reaches the occluded region on the object it doesn't

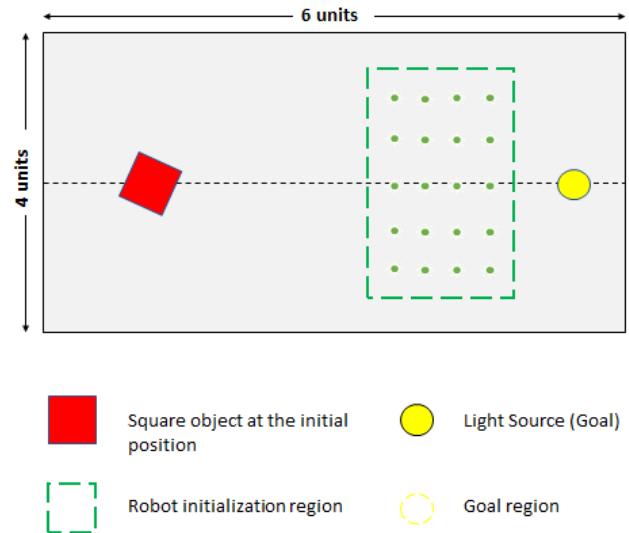


Fig. 14. Experimental setup. The initially the robots can be placed in a grid formation or in a randomly distributed manner.

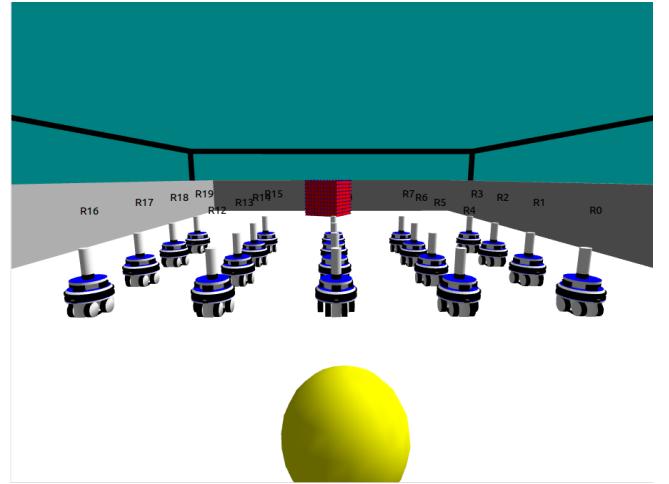


Fig. 15. Experimental setup in ARGoS environment with Footbot robots

get any readings on the light sensor.

Now it is declared that the robot is in occluded area of the object. The next task is to push the object. In order to push the object, the robot again orients itself to the object following the initial orientation steps using the current position of the object and its own current position. Once oriented it starts moving towards the object, but this time it is not constrained by the proximity sensor hence it touches the object and pushes it. As the occluded region is decided by light source the robots push towards the light source which is our goal.

*3) Parameters:* For the occlusion based transport, we tested the following parameters:-

- Distribution of the robots:

One of the most important parameters to test the robustness of the experiment is to test whether the robots

### Occlusion Based Algorithm

```

Find Robot Position
Find Object Position
Calculate Orientation
if Orientation is true then
| Move in Oriented Direction;
else
| Change Orientation to the Object;
end
Reached Object
Perform Left Hand Wall Rule
if Light Source is Visible then
| Continue Left Hand Wall Rule;
else
| Rotate to align with the object ;
| Start Pushing the Object ;
end
if Object touches the wall then
| STOP ;
else if Object reached at the Light then
| Experiment successful - STOP ;
else
| Time limit crossed - STOP ;
end

```

are able to perform the task irrespective of how they have been placed in the environment. We carried out experiments by placing the robots randomly in the environment and then placing them in a formation as shown in 15 to compare the results obtained. When the robots were placed randomly the result were better than when placed in the formation. As robots initially try to orient themselves towards the object and then start moving towards the object the random distribution enables them to avoid other robots which have not yet being oriented hence giving better results. But when the robots are placed in formation, all the robots orient together hence start to move towards the object together. As the condition of reaching the object is analyzed based on the proximity values, the robots are misguided when they come across other robots. Hence, considering this other robot as an object and performing left hand wall rule. This happens at a large scale when the robots are placed in grid formation.

- Position of the object:

The position of the object also influences the performance efficiency of the experiment. When the object was placed near walls it was noticed that the robots approach the object but they get lost. When near the wall the robots start applying the left hand wall rule on the wall instead of the object. Hence, taking rounds along the wall of the arena. Hence, affecting the success rate of the trials.

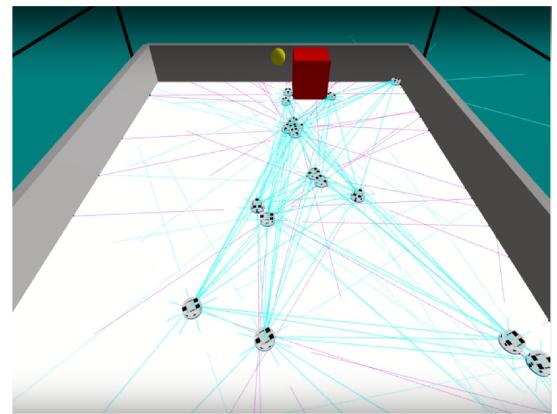
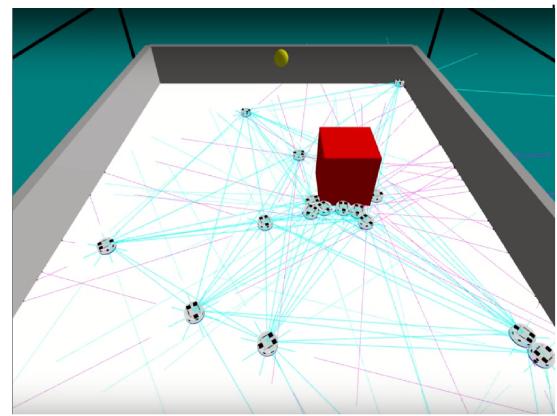
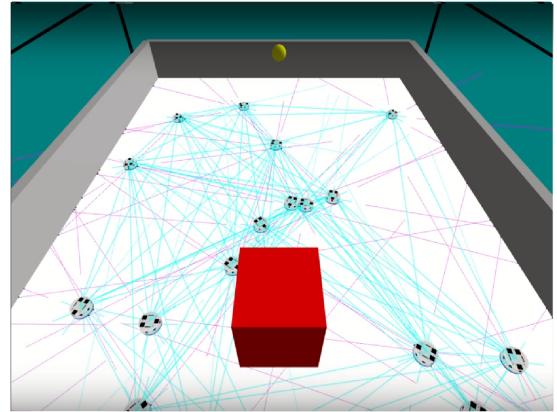


Fig. 16. Occlusion Based Collective Transport of a convex object

## IV. LIMITATIONS

### A. Concave Filling

- The range of proximity sensors affects the performance of the experiments due to the fact that Khepera IV has a limited range. Hence, it is very hard to achieve 5 proximity readings simultaneously ( it signals that the

angle is greater than  $\pi$ ).

- Simultaneously deploying too many robots affects the performance as the robots cannot distinguish between object and neighbor robots. The robots react when they come close to their neighbor as they would for the object.

#### B. Collective Transport

As we are not using camera on the robot we are not able to distinguish between the object to be transported and walls of the arena. Hence causing the robot to perform the left-hand wall rule along the walls of the arena. Another limitation is that objects cannot be placed near the wall. The path followed by the robots is not optimal and may vary from shape to shape.

## V. CONCLUSION

The occlusion based collective transport strategy can be used to perform collective transport when the object and robot position is known. The concave filling algorithm presented can be used to eliminate or minimize the concavity of the object. The robots need not have any prior knowledge about the objects geometry to perform the concave filling. The object can then be transported as a convex object, using the occlusion based transport strategy.

## VI. FUTURE WORK

In the future we would like to perform the same experiment using a camera. This would eliminate the use of light source being placed above the object for concave filling. Also, this would allow to differentiate robots between its neighbors, wall, object and goal. We would also like to integrate both concave filling and occlusion based transport as a single experiment. A path optimization technique can be used so that the robots follow an optimal path. Obstacle avoidance can also be used so that the robots do not collide with its neighbors.

## REFERENCES

- [1] Chen, J., Gauci, M., Li, W., Kolling, A. and Gro, R., 2015. Occlusion-based cooperative transport with a swarm of miniature mobile robots. *IEEE Transactions on Robotics*, 31(2), pp.307-321.
- [2] Rubenstein, M., Cabrera, A., Werfel, J., Habibi, G., McLurkin, J. and Nagpal, R., 2013, May. Collective transport of complex objects by simple robots: theory and experiments. In Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems (pp. 47-54). International Foundation for Autonomous Agents and Multiagent Systems.
- [3] Habibi, G., 2015. Collective Transport of an Unknown Object by Multi Robots with Limited Sensing (Doctoral dissertation, Rice University).
- [4] TORABI, S., 2015. Collective transportation of objects by a swarm of robots (Doctoral dissertation, Ms. Thesis, Chalmers University of Technology).
- [5] Pinciroli, C., Lee-Brown, A. and Beltrame, G., 2015. Buzz: An extensible programming language for self-organizing heterogeneous robot swarms. *arXiv preprint arXiv:1507.05946*.
- [6] Pinciroli, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Di Caro, G., Ducatelle, F. and Stirling, T., 2011, September. ARGoS: a modular, multi-engine simulator for heterogeneous swarm robotics. In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on (pp. 5027-5034). IEEE.
- [7] <https://github.com/Collective-transport-of-concave-objects/Environment.git>