Cooperative Phototaxis Using Networked Mobile Sensors and Centroidal Voronoi Tessellations

Shelley Rounds, *Student Member, IEEE;* YangQuan Chen, *Senior Member, IEEE*

***Abstract*— The main objective for this research is to design an economical and robust swarm system to achieve phototaxis. The system combines swarm intelligence with centroidal Voronoi tessellations (CVT) to localize and track a dynamic moving light source. A small group of static mesh sensors is used to observe the environment while mobile actuators move according to their CVT. This paper also proposes a solution to decrease the number of sensors needed for CVT convergence with a light estimation algorithm. This new algorithm can find the approximate location of a light source by a general light distribution equation according to light sensor readings. Experiments are conducted both in simulation and on an actual mobile robot platform which show the flexible and robust nature of CVTs over other behavioral swarm algorithms. The complete phototaxis algorithm is presented in a specific swarm design procedure.**

***Index Terms*— centroidal Voronoi tessellations, phototaxis, dynamic tracking, light estimation, swarm**

I. Introduction

Swarm engineering has become an increasingly popular research topic. The idea of using a swarm of simple robots in the place of a single sophisticated robot offers many advantages in military, industrial, and commercial applications. With a group of sensors and actuators it is possible to sense and act on a distributed environment such as temperature, electromagnetic waves, or a cloud of toxic gas. By regularly updating these sensors and actuators, a closed-loop system, also known as cyber-physical systems (CPS) can also characterize and track dynamic environments.

In this project, an array of stationary sensors characterize the environment to build a CVT while the actuators move according to the tessellation to achieve cooperative phototaxis. CVTs include nearest neighbor information which make collision avoidance, cooperative control, and dynamic target tracking possible in one algorithm. CVT is also a non-model based mathematical method which asymptotically converges to a concentrated source [1].

Previous work on CVTs briefly examine the effect of fewer sensors on convergence [2]. Exploring ways to decrease the number of sensors needed for convergence dramatically reduces cost and setup time for a sensor array. A new light estimation algorithm is introduced in the CVT algorithm to assure robot convergence despite fewer sensors.

CVT-based taxis methods can be used in many applications such as chemotaxis, nuclear hazard detection, electromagnetic (EM) radio jammer localization, hot spot detection in forest fire mop-ups, etc. This work advances swarm engineering and Voronoi tessellation algorithms as a whole since to the authors knowledge, dynamic target tracking with CVTs has not yet been applied to actual robots. This paper also introduces the light estimation algorithm as a method to compensate for fewer sensors.

This paper is organized as follows: Section II defines CVTs and its advantages. Section III describes the swarm design approach and follows the phototaxis CVT algorithm through the first three steps of that approach. The details of the phototaxis CVT algorithm is explained in Section IV. The simulation and robot application results are shown in Sections V and VI respectively. Section VII draws a conclusion to this research and proposes future work on the subject.

II. Preliminaries

A. Vorornoi Tessellations

A Voronoi tessellation refers to a region, containing p generating points, separated into cells such that each cell contains one generating point and every point in the cell is closest to its generating point [3]. Voronoi tessellations are mathematically defined as follows. Given a region $\Omega \in \mathbb{R}^N$and a set of generating points $\{p_i\}_{i=1}^k \subset \Omega$, let the Voronoi cell V_i corresponding to the generator p_i be

$$V_i = \{\ q{\in}\ \mathbf{\Omega}|\ |q-p_i| < |q-p_j| \quad j=1,\ldots,k,\ j\neq i\} \tag{1}$$

$$i = 1,\ldots,k,$$

where the set of Voronoi cells $\{V_i\}_{i=1}^k$ creates a Voronoi tessellation on Ω. $|q-p_i|$ denotes the Euclidean distance.

Equation (1) simply compares the distance between points on the region, q, and generators, p. If a point q is closest to the generator p_i, then that point belongs to the Voronoi cell V_i. This concept can be used to create a discrete Voronoi tessellation. An example of Voronoi tessellations created by a random distribution of points can be seen in Fig. 1. Notice that each generator's cell is only affected by its nearest neighbors. Therefore, in order to construct any Voronoi tessellation, each generator should only be aware of its nearest neighbors.

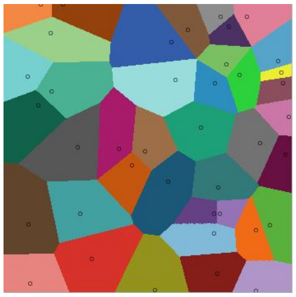


Fig. 1. Random Voronoi tessellation created in NetLogo, a multi-agent simulation software

Shelley Rounds and YangQuan Chen are with the Center for Self-Organizing and Intelligent Systems (CSOIS), Department of Electrical and Computer Engineering, 4160 Old Main Hill, Utah State University, Logan UT 84322-4160, USA. Corresponding author: Prof. YangQuan Chen, **T**: (435)797-0148, **F**: (435)797-3054, **W**: `http://mechatronics.ece.usu.edu/mas-net/`, **E**: `yqchen@ieee.org`.This work was supported in part by NSF Grants # 0540179 (DDDAS) and # 0552758 (REU Site).

978-1-4244-4524-0/09/$25.00 ©2009 AACC

B. Centroidal Voronoi Tessellations

Given a density over the region of interest, a CVT is a Voronoi tessellation in which each generator is the centroid (center of mass according to the given density) of its corresponding cell [4]. The mathematical definition is as follows. Given a density function $\rho(q) \geq 0$ defined on Ω, the mass centroid p_i^* for each Voronoi cell V_i is given by

$$p_i^* = \frac{\int_{V_i} q\rho(q)\mathrm{d}q}{\int_{V_i} \rho(q)\mathrm{d}q} \text{ for } i = 1, \ldots, k. \tag{2}$$

The tessellation constructed by (2) is a centroidal Voronoi tessellation, provided that $\int_{V_i} \rho(q)\mathrm{d}q \geq 0$, if and only if

$$p_i = p_i^* \text{ for } i = 1, \ldots, k.$$

In other words, the points p_i are both the generators and mass centroids for the Voronoi cells V_i [5].

CVTs typically create simple diagrams where the concentration of generators can be controlled by the given density function. CVTs have many applications which include territorial animal behavior, image and data compression, multi-dimensional integration, partial differential equations, and optimal sensor and actuator locations [4].

C. Energy Function

CVTs also minimize the energy function

$$\mathcal{H}_{\mathcal{V}}(p) = \sum_{i=1}^{k} \int_{V_i} |q - p_i|^2 \rho(q)\mathrm{d}q, \tag{3}$$

which is also known as the variance, cost, or error function [4]. This energy function evaluates the location error of the generators, p_i, according to the density $\rho(q)$. The proof in [6] shows that CVT is a necessary condition to minimize the energy function (3).

III. Swarm Design Approach

Global swarm behavior can be complex and volatile. Scientific procedures for swarm behavior have been introduced to break down the complicated process for much more predictable results. For this project, a detailed swarm design approach is used based on S. Kazadi's original procedure [7], [8]. The detailed procedure is below:

1) Identify desired emergent behavior,
2) Select or devise a set of behaviors and motivators,
3) Choose appropriate input for the above behavior motivators,
4) Generate an algorithm to combine behavior motivators,
5) Simulate global behavior,
6) Apply global behavior to robots.

All steps in this process, with the exception of the first, depends on the previous steps. If one step does not work, the previous step should be modified for a successful swarm. This ladder dependency organizes a swarm design and helps identify problem areas in the design.

These procedures were originally created for behavioral algorithms. Mathematical models were previously believed to be much too difficult to develop using swarm procedures because of the complex interactions and unknown variables global swarm behavior brings [8]. However, this project successfully follows the six-step swarm design approach using a mathematical non-model based algorithm without compromising general behaviors or complexity. The remainder of this paper will explain the phototaxis CVT design through each step of the swarm design approach.

A. Identify Emergent Behavior

Step one of the design process defines the overall desired robot behavior. This emergent behavior must be clearly defined before proceeding to the second step. Examples of swarm behavior, which can be combined to create an emergent behavior, include: aggregation, flocking, foraging, following, dispersion, homing, and herding [8]. The main goal of this project is to find and track a moving light source where the location is unknown. Specific tasks to achieve this goal are:

1) *Rendezvous* at a light source,
2) *Follow light gradient* towards light source,
3) *Collision avoidance*.

With these behaviors in mind, selecting motivators is straightforward whether using behavioral or mathematical algorithms.

B. Select Behavior Motivators

Behavior motivators are algorithms that can be run by the individual agents or that allow each agent to react to local knowledge. Motivators constitute the individual agent's behavior and can react to a single or multiple inputs. A combination of motivators creates an emergent swarm behavior [8].

As mentioned in Section I, the CVT algorithm has many behaviors "built in." For this project the robots act as the generators for the CVTs. Because Voronoi cells are convex polygons that never overlap, one robot or generator cannot collide with another. This inherent property of CVTs covers the collision avoidance requirement [9], [10].

Also, if a region of interest is given a concentrated density, the CVT will cause the robots to aggregate or rendezvous toward the maximum peak of that density. During this aggregation the robots' paths follow the gradient of the density. This behavior satisfies the rendezvous and light follow requirements given the maximum of the density function is placed at the location of the light.

An additional observation shows that the robots can also converge to the light source simultaneously. [6] contains a proof that to minimize the energy function in (3) the generators simultaneously approach a CVT configuration. Granted, each generator may approach their CVT location at different rates depending on how far they are from that location and the location of nearest neighbors. However, if the robots kinematics respond well enough to commands, a simultaneous rendezvous can be achieved. Considering these advantages to CVTs only one rendezvous motivator is needed for cooperative phototaxis.

C. Choose Inputs

Currently a CVT is calculated on a base station computer that can communicate with all robots and sensors. In the future, a more decentralized design will be extremely beneficial. Specific inputs will change as the algorithm gradually becomes more decentralized. For now, inputs required for the rendezvous motivator with CVTs include:

1) Array of light readings,
2) Position at each light reading,
3) Position of each robot.

Light readings are gathered by wireless mesh sensors. Robots positions are gathered by a combination of pseudo-GPS and encoders.

IV. Generate CVT Phototaxis Algorithm

Perhaps the most common and basic algorithm to construct discrete CVTs is the *Lloyd's algorithm*. This algorithm is a clear-cut iteration between building Voronoi tessellations and computing their centroids [11].

Lloyd's method requires few iterations, but each iteration is expensive to find the precise Voronoi tessellation and mass centroid. A second commonly used method for computing CVTs is the *MacQueen algorithm*. This algorithm does not require any precise construction of Voronoi tessellations or mass centroids; thus, taking advantage of discrete CVTs. Despite the absence of these calculations the algorithm still converges to a CVT [12]. However, each iteration only moves one generator and many iterations are needed for convergence. A combination of the few iterations of Lloyd's method and the cheap computation of the MacQueens method can help create a faster converging CVT algorithm for robots.

A. CVT Algorithm for Robots

A detailed explanation of the setup for the MASnet platform is discussed in Section VI, but a preliminary description is needed to understand the CVT algorithm. An array of sensors is placed evenly over a platform which serves as the region of interest. The number of sensors is not a concern at this point. These sensors measure light density over the platform at their corresponding positions to construct the discrete density function $\rho(x,y)$. The light readings and robot positions, p_i, are gathered to build a Voronoi tessellation. Note that the density function $\rho(x,y)$ is only sampled at certain points q_j, where $j = 1,\ldots,l$, and l is the total number of sensors. The combined Lloyd-MacQueen method for robots is as follows:

Given a region Ω, a density function $\rho(x,y)$ defined for all $x \in \Omega$, and positive integers k and l,

1) Select an initial set of k points $\{p_i\}_{i=1}^{k}$(robot starting positions) as the generators;
2) Select the sampled points $\{q_j\}_{j=1}^{l} \in \Omega$ where $\rho(q_j) = \rho(x_j,y_j)$;
3) Find the generator p_i closest to the point q_j for each sampled point; assign the set of points q_j with closest generator p_i to V_i (this builds the discrete approximation of the Voronoi cells$\{V_i\}_{i=1}^{k}$);
4) Find the discrete mass centroid of each Voronoi cell. These centroids become the new set of generators;
5) Give the robots command to move to the new generating points;
6) If the new generating points meet a given convergence criterion, terminate; otherwise, return to step 2.

Step three of this algorithm can be obtained by looping though sensor and robot data. This step is also the application of (1) which compares the distances between generators, p_i, and points, q to assign the Voronoi cell V_i. Notice that a single iteration of this new Lloyd-MacQueen method changes all generators p_i at once, as in the basic Lloyd method, while an explicit computation of Voronoi tessellations is not needed, much like the MacQueen method. The best of both algorithms are now combined for faster convergence. It is important to note, however, that robot synchronization has the disadvantege of reducing the swarm's flexibility. Future work on a more decentralized system will help analyze this tradeoff.

B. The Density Function

Before the CVT algorithm can be applied, a density $\rho(x,y)$ is required. Originally, the readings from the sensors served as the density. However, interference from other light sources and sensor noise caused sporadic results. Secondly, the light source could be too broad or asymmetrical for the robots to aggregate neatly around the source. Third, robots will follow unique light sources in different ways. Each light source has its own distribution characterized by intensity and drop-off rate.

The desired response requires robots to find and track any light source identically, regardless of the light characteristics. To accomplish this, the density is modeled after a Gaussian distribution. The center of the distribution, (x_c,y_c), occurs at the location of the light. Below is the Gaussian function

$$\rho(x,y) = c\exp^{-\sigma\left[(x-x_c)^2+(y-y_c)^2\right]}, \qquad (4)$$

where c is the intensity of the Gaussian, and σ is the drop-off rate. A large σ is desired for concentrated densities. The general Gaussian density in (4) achieves extremely robust rendezvous behavior. Results of using this density can be seen in the following sections.

C. Light Estimation Algorithm

In order to plot the density function, the location of the light source must be known. Unfortunately, the number of sensors is limited and the source may not lie directly over any sensor. Therefore, a light estimation algorithm is needed to approximate the light source's location. A precise characterization of the light is typically done to create an equation fit, piece-wise function, or zone intensity levels of the light distribution. However, such exact calculations are only useful for that particular light. Assuming the light source is unknown, the light estimation algorithm must find any light source within the region of interest.

This project introduces such an algorithm. Instead of characterizing the distribution of a specific light, consider the general intensity equation for the spreading of light [13],

$$i(x,y) = \frac{c}{\left[(x-x_c)^2+(y-y_c)^2\right]^{\sigma/2}}. \qquad (5)$$

Again, c is the intensity and σ is the drop-off rate .

In this case, the intensity at each sensor reading, $\{i(x_j,y_j)\}_{j=1}^{l}$ is known. Through signal processing and least mean squares technique the intensity, c, the drop-off rate, σ, and the location of the light, (x_c,y_c) can be found. First, take the log of the intensity equation

$$\lambda(x,y) = \log c - \frac{1}{2}\sigma\log\left[(x-x_c)^2+(y-y_c)^2\right],$$

$$\text{where } \lambda = \log(i).$$

Combine all known sample readings in matrix form

$$\begin{bmatrix} \lambda(x_1,y_1) \\ \lambda(x_2,y_2) \\ \vdots \\ \lambda(x_l,y_l) \end{bmatrix} = \begin{bmatrix} 1 & -\frac{1}{2}\log\left[(x_1-x_c)^2+(y_1-y_c)^2\right] \\ 1 & -\frac{1}{2}\log\left[(x_2-x_c)^2+(y_2-y_c)^2\right] \\ \vdots & \vdots \\ 1 & -\frac{1}{2}\log\left[(x_l-x_c)^2+(y_l-y_c)^2\right] \end{bmatrix} \begin{bmatrix} \log c \\ \sigma \end{bmatrix}$$

$$\Rightarrow \bar\lambda = \mathbf{A}(q_c)\bar{\mathbf{b}}(c,\sigma),$$

$$\text{where } q_c = (x_c,y_c).$$

Because we are dealing with measured values, noise and interference are introduced and the squared error $\left\|\bar\lambda - \hat{\mathbf{A}}(q_c)\hat{\bar{\mathbf{b}}}(c,\sigma)\right\|^2$, where a hat indicates an estimated matrix or vector, must be minimized

$$\min_{\hat{\bar{\mathbf{b}}},q_c}\left\|\bar\lambda - \hat{\mathbf{A}}(q_c)\hat{\bar{\mathbf{b}}}(c,\sigma)\right\|^2 = \min_{\hat q_c}\left(\bar\lambda^T\mathbf{P}(q_c)\bar\lambda\right) = \min_{\hat q_c}\mathbf{E}(q_c), \qquad (6)$$

$$\text{where} \qquad \mathbf{P}(\hat q_c) = \mathbf{I} - \hat{\mathbf{A}}(\hat{\mathbf{A}}^T\hat{\mathbf{A}})^{-1}\hat{\mathbf{A}}^T = \mathbf{I} - \hat{\mathbf{A}}\left(\hat{\mathbf{A}}^{-1^*}\right),$$

$$\text{and} \qquad \hat{\mathbf{A}}^{-1^*} \text{ is the pseudoinverse of } \hat{\mathbf{A}}.$$

The right hand side of (6) finds the x and y positions where $\mathbf{E}(q_c)$ is an absolute minimum. The minimum value can be found by iterating through x and y positions and calculating $\mathbf{E}(q_c)$ for each iteration. Finally, light intensity c and drop-off rate σ are derived from the matrix $\hat{\mathbf{A}}$ associated with the minimum value

$$\hat{\bar{\mathbf{b}}} = \begin{bmatrix} \log \hat{c} \\ \hat{\sigma} \end{bmatrix} = \left(\hat{\mathbf{A}}^{-1^*} \right) \bar{\lambda}. \tag{7}$$

Because the phototaxis project only requires an estimation of the location of the light (x_c, y_c), (7) is not used, but can be extremely useful in identifying different light sources.

Unfortunately, iterating through the entire region of interest may take too much time. The algorithm can be much faster if only a fraction of all x and y positions are evaluated. A recursive light position estimation algorithm helps reduce the number of iterations needed. The algorithm focuses or "zooms in" on the critical area and ignores the rest of the region:

Given a region Ω, an array of light intensity readings, $\{i_j\}_{j=1}^{l}$ and position at each reading q_j,

1) Find the log of each reading, $\lambda_j(q) = \log\left[i_j(q)\right]$;
2) Begin with a large step size Δ and *critical area* Ω;
3) Iterate through x and y positions by step size Δ over the *critical area* to find the minimum $\mathbf{E}(q)$;
    a) $\hat{\mathbf{A}} = \begin{bmatrix} 1 & -\frac{1}{2}\log\left[(x_1-x)^2+(y_1-y)^2\right] \\ 1 & -\frac{1}{2}\log\left[(x_2-x)^2+(y_2-y)^2\right] \\ \vdots & \vdots \\ 1 & -\frac{1}{2}\log\left[(x_l-x)^2+(y_l-y)^2\right] \end{bmatrix}$;
    b) $\mathbf{P} = \mathbf{I} - \hat{\mathbf{A}}(\hat{\mathbf{A}}^T\hat{\mathbf{A}})^{-1}\hat{\mathbf{A}}^T = \mathbf{I} - \hat{\mathbf{A}}\left(\hat{\mathbf{A}}^{-1^*}\right)$;
    c) $\mathbf{E}(q) = \min\left(\bar{\lambda}^T\mathbf{P}\bar{\lambda}\right)$;
    d) $\Delta \leftarrow \frac{\Delta}{10}$; *critical area* $\leftarrow q \pm 5\Delta$;
    e) If step size Δ reached the minimum step, continue to step 4; otherwise, repeat step 3.
4) Return the estimated light position $q_c \leftarrow q$.

By combining light location estimation, the Gaussian density function, and the basic CVT algorithm for robots a successful cooperative phototaxis can be achieved. The algorithms are first tested in a simulated environment and finally applied to physical robots.

V. Simulate Behavior with Diff-MAS2D

A new simulation platform called `MAS2D`, derived from `Diff-MAS2D` [14], is used for testing these algorithms. `MAS2D` is designed to receive any moving or static distribution over the region $\Omega = [0,1]^2$. Robots are modeled as a particle by second order dynamics [15]

$$\ddot{p}_i = u_i, \tag{8}$$

where u_i is the control law. To minimize the function in (3), the control law is set to follow a CVT

$$u_i = k_p(p_i - p_i^*) - k_d\dot{p}_i, \tag{9}$$

where p_i^* is the mass centroid of V_i, and both k_p and k_d are positive constants. The final term in (9) introduces viscus damping [16]. k_d is the damping coefficient and $\dot{p}_i$ is the velocity of robot i. This term eliminates oscillation as the robot approaches its destination.

In this simulation nine static sensors are evenly placed over Ω to show phototaxis performance with a limited amount of sensors. Four robots are also evenly placed over Ω. The control law for the robots is set to

$$u_i = 3(p_i - p_i^*) - 3\dot{p}_i.$$

The light source is set to the bottom left corner $(0,0)$ with the distribution

$$i(x,y) = \frac{1}{(x^2+y^2)^{0.3/2}}.$$

The Gaussian density function for CVT rendezvous is set to

$$\rho(x,y) = \exp^{-100\left[(x-x_c)^2+(y-y_c)^2\right]}.$$

After the center of the light source is calculated, Fig. 2 shows the Gaussian density with the center of the estimated light location at $(0,0)$.

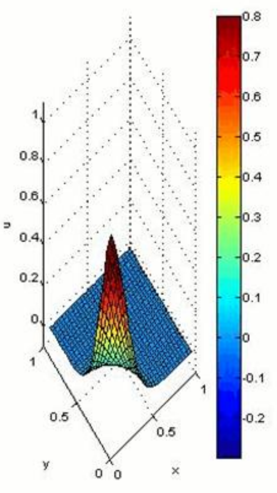
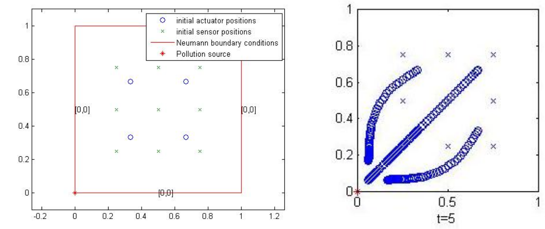


Fig. 2. Plot of the Gaussian function with estimated light location $(0,0)$

The time step is set to 0.05 seconds. Robots compute desired positions every 0.2 seconds. Progression of the simulation is shown in Fig. 3. The X's indicate sensors, O's indicate robot paths and the red asterisk indicates the estimated position of the light source. Notice how the robots drive toward the source while keeping their square formation. This is achieved by the nature of CVTs; no formation control consensus algorithms are used. The robots converge to a CVT and arrive simultaneously at the source after 5 seconds. Similar behaviors occur at different light locations and with dynamic light sources. For the robots to gather closer to the light, σ in the Gaussian density should be increased. Proof of convergence for these simulations can be found in [17].



(a) Initial postitions

(b) Robot motion

Fig. 3. Progression of simulated robots for phototaxis

VI. Apply to Robots with the MASnet Platform

Now that phototaxis works in simulation, the algorithm is ready to be tested on physical robots.

A. MASnet Platform

A wireless sensor network combined with a platform encapsulates the ongoing MASnet project that began in 2003. Several robots, with limited communication and sensing abilities, can move on top of the MASnet platform. Despite restricted communication, robots are able to coordinate with each other to perform swarm tasks. A picture of the actual platform used for this research is in Fig. 4.

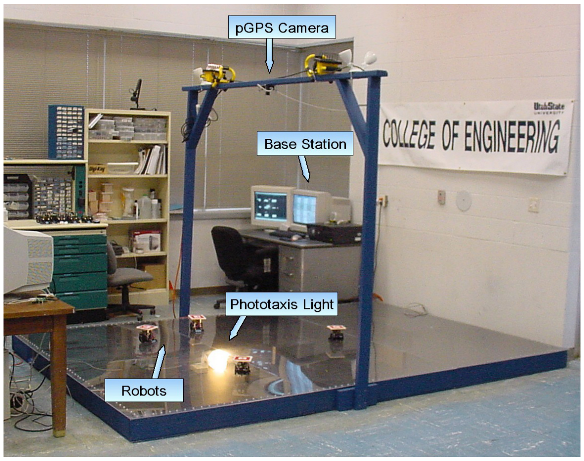


Fig. 4. MASnet (mobile actuator and sensor networks) platform

The robots perform commands on the platform, the pGPS camera monitors the robots position, the camera information is displayed on the base station computer, and the base station sends commands back to the robots. The base station functionality is performed in a program called RobotCommander written in C++ exclusively for MASnet. The platform is made of off-the-shelf products and open source software to keep the system flexible and low cost. More detail on MASnet platform development is described in [18], [19], [2]. In the CVT phototaxis experiment, the light is held above the platform for the sensors, under the platform, to measure its distribution.

B. Hardware and Software Description

Each robot is a small, two-wheel, differentially driven robot built of mainly commercial parts; see Fig. 5. The robots are intended to be simple, compact, and easily redesigned. The base station and robots operate by a MicaZ programming board, or MicaZ mote developed by Crossbow.

Nine TmoteSky boards, previously developed by Moteiv, comprise the sensor array. Each Tmote is programmed to gather and send visible light readings over the 2.4 GHz Radio [20]. The MicaZ base station mote then receives and processes the Tmote data for CVTs. The base station computer calculates CVTs and sends mass centroid commands to each robot every 2 seconds.

The Tmote and MicaZ motes use TinyOS, an event-driven operating system designed for wireless sensor networks with limited memory. The TinyOS system is developed in nesC, an extention of C, which is primarily used for embedded systems such as this wireless sensor network [21], [22], [23], [24].

C. Results

Nine experiments were conducted with a stationary light source at different locations on the platform. For these experiments, an incandescent light was held above the platform by hand. The initial configuration is setup similar to the simulation; see Fig. 6(a). Robots are indicated by red circles, Tmotes are indicated by teal squares, the region of interest is indicated by a blue rectangle, and the light sits on the top right corner. The average time it takes to surround the light source is 13.5 seconds. See Table I for a table of average convergence times according to light location.

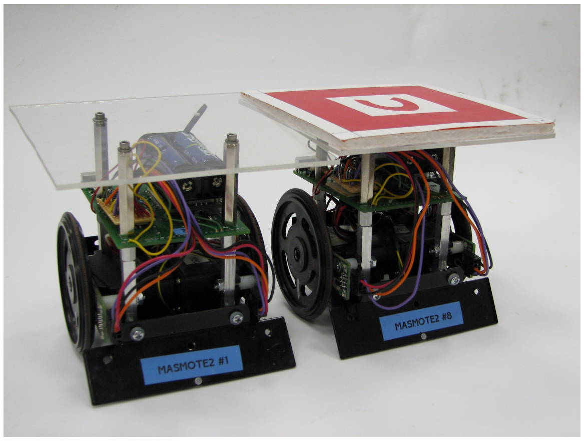


Fig. 5. MASnet Robots

TABLE I
AVERAGE CONVERGENCE TIMES ACCORDING TO LOCATION

Location	Middle	Corners	Edges	Overall
Convergence	10 s	18.5 s	14 s	13.5 s

Similar convergence results occur for dynamic CVTs provided the light does not move faster than the algorithm reconfigures. Screenshots of one particular experiment can be seen in Fig. 6. The upper left robot was intentionally impaired to show the robust nature of the algorithm. It is difficult to tell the exact location of the light from the overhead camera because if its inherent distorted view, but phototaxis behavior can still be observed. The phototaxis CVT algorithm is also easily scalable. Robots can be added at any time and include themselves in the surrounding group. Visit the YouTube channel `http://www.youtube.com/user/MASnetPlatform` for videos of static and dynamic CVT-based phototaxis experiments on the MASnet platform. Proof of convergence for this hardware platform can be found in [17].

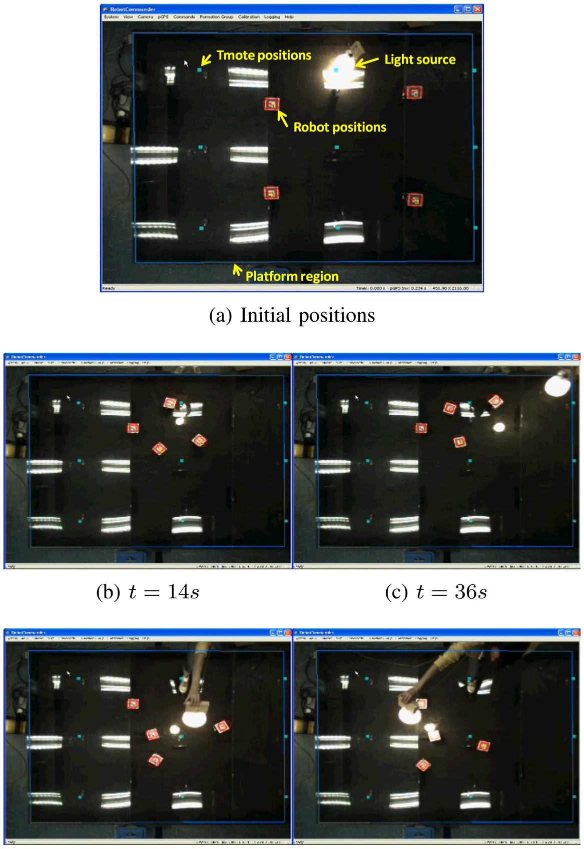
VII. Conclusion

This paper introduces a new algorithm for cooperative phototaxis using CVTs. The results show this mathematical algorithm is extremely reliable, robust, and scalable. To the author's knowledge, this is the first time a CVT algorithm for dynamic target tracking has been applied entirely to a hardware platform. The algorithm also compensates for CVTs with a limited number of sensors.

Future work includes building a more decentralized algorithm for a swarm, such as the algorithms introduced in [25]. Further analysis on starting configurations, synchronized versus non-synchronized robots, and the least number of sensors needed for the light estimation algorithm should be conducted. Finally, CVTs for other measurable elements such as EM waves, or fog will introduce many more applications to robot CVTs.

References

[1] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, pp. 243–255, Apr. 2004.



(a) Initial positions

(b) $t = 14s$

(c) $t = 36s$

(d) $t = 54s$

(e) $t = 58s$

Fig. 6. Progression of MASnet robots for phototaxis

[2] Z. Wang, “Distributed control of distributed parameter systems using mobile actuator and sensor networks,” Master’s thesis, Utah State University, Logan, UT, 2005.

[3] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, ch. 4, pp. 229–287. Hoboken, NJ: John Wiley & Sons, Ltd., 2 ed., 2002.

[4] Q. Du, M. Emelianenko, H.-C. Lee, and X. Wang, “Ideal point distributions, best mode selections and optimal spatial partitions via centroidal Voronoi tessellations,” in *The 2nd International Symposium on Voronoi Diagrams in Science and Engineering*, (Seoul, Korea), pp. 325–333, Oct. 2005.

[5] H. Chao, Y. Chen, and W. Ren, “A study of grouping effect on mobile actuator sensor networks for distributed feedback control of diffusion process using central Voronoi tessellations,” *The Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation*, pp. 769–774, June 2006.

[6] Q. Du, V. Faber, and M. Gunzburger, “Centroidal Voronoi tessellations: Applications and algorithms,” *SIAM Review*, vol. 41, no. 4, pp. 637–676, 1999.

[7] S. Kazadi, “On the development of a swarm engineering methodology,” in *Proceedings IEEE Conference on Systems, Man, and Cybernetics*, pp. 1423–1428, Oct. 2005.

[8] W. Bourgeous, S. Rounds, and Y. Chen, “A swarm engineering approach to mobile sensor network design towards collaborative phototaxis with a slowly moving light source,” in *ASME/IEEE 3rd International Conference on Mechatronics and Embedded Systems Application*, (Las Vegas, NV), Sept. 2007.

[9] M. Lindhe and K. H. Johansson, *Taming Heterogeneity and Complexity of Embedded Control*, ch. 24, pp. 419–434. Hoboken, NJ: John Wiley & Sons, Ltd., Feb. 2007.

[10] C.-Y. Mai and F.-L. Lian, “Analysis of formation control and networking pattern in multi-robot systems: a hexagonal formation example,” *International Journal of Systems, Control and Communications 2008*, vol. 1, no. 1, pp. 98–123, 2008.

[11] H.-C. Lee and S.-W. Lee, “Reduced-order modeling of burgers equations based on centroidal Voronoi tessellation,” in *The 2nd International Symposium on Voronoi Diagrams in Science and Engineering*, (Seoul, Korea), pp. 346–357, Oct. 2005.

[12] J. B. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability* (L. M. L. Cam and J. Neyman, eds.), vol. 1, pp. 281–297, University of California Press, 1967.

[13] M. Hall, “Taking robots to the extreme.” NSF REU 2006 PowerPoint Presentation, http://www.csois.usu.edu, Aug. 2006.

[14] J. Liang and Y. Chen, *Diff-MAS2D (version 0.9) - User’s Manual*. http://mechatronics.ece.usu.edu/reports/USU-CSOIS-TR-04-03.pdf, Utah State University, Logan, UT, 1 ed., Oct. 2004.

[15] H. Chao, Y. Chen, and W. Ren, “Consensus of information in distributed control of a diffusion process using centroidal Voronoi tessellations,” *The Proceedings of the 46th IEEE Conference on Decision and Control*, pp. 1441–1446, Dec. 2007.

[16] A. Howard, M. J. Mataric, and G. S. Sukhatme, “Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem,” in *Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems (DARS02)*, (Fukuoka, Japan), pp. 299–308, June 2002.

[17] S. Rounds, “Distributed control for robotic swarms using centroidal Voronoi tessellations,” Master’s thesis, Utah State University, Logan, UT, 2009.

[18] P. Chen, “Pattern formation in mobile wireless sensor networks,” Master’s thesis, Utah State University, Logan, UT, 2005.

[19] Y. Chen, K. L. Moore, and Z. Song, “Diffusion-based path planning in mobile actuator-sensor networks (MAS-Net): Some preliminary results,” in *Proceedings of SPIE Conference on Intelligent Computing: Theory and Applications II, part of SPIE’s Defense and Security*, (Orlando, FL), Apr. 2004.

[20] Moteiv, “Tmote Sky: Ultra low power IEEE 802.15.4 compliant wireless sensor module.” http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf, Feb. 2006.

[21] Crossbow Technology Inc., “MICAz, Wireless Measurement System.” http://www.xbow.com.

[22] Crossbow Technology Inc., “MoteWorks Getting Started Guide.” http://www.xbow.com, Apr. 2007.

[23] D. Gay, P. Levis, D. Culler, and E. Brewer, “nesC 1.2 Language Reference Manual.” http://www.tinyos.net, Aug. 2005.

[24] University of California at Berkeley, “TinyOS Tutorial.” http://www.tinyos.net/tinyos-1.x/doc/tutorial/index.html, Sept. 2003.

[25] M. Schwager, J. McLurkin, and D. Rus, “Distributed coverage control with sensory feedback for networked robots,” in *Proceedings of Robotics: Science and Systems*, (Philadelphia, USA), Aug. 2006.