Controlled Coverage Using Time-Varying Density Functions [★]

Sung G. Lee [*] **Magnus Egerstedt** [*]

[*] *Georgia Institute of Technology, Atlanta, GA 30332 USA*
(e-mail: {slee656, magnus}@gatech.edu)

Abstract: A new approach for controlling a system of multiple agents by choosing a time-varying density function is presented, employing optimal coverage ideas. In this approach, we specify a time-varying density function that represents where it is that want the agents to monitor, and how important it is for each point to be covered. A new algorithm is presented under which the agents track the time-varying density function while providing optimal coverage of the density function. Results from robot implementation show that the proposed algorithm guides the agents well over the chosen density functions, and that the effectiveness of the coverage is higher than other comparable algorithms.

Keywords: Coverage control, centroidal Voronoi tessellations, time-varying systems, tracking systems, control algorithms, mobile robots

1. INTRODUCTION

In this paper, we present an approach to controlling a system of multiple agents by choosing a time-varying density function. A control law that causes the agents to track the density function while providing optimal coverage of the density function is proposed. The intention is to control the multi-agent system to provide surveillance over the domain of interest by employing optimal coverage ideas on general time-varying density functions. This approach can have applications in diverse situations where we specify what time-varying region in the domain we want the robots to pay more attention to. One example is monitoring of oil spills. A search and rescue scenario is another example, where the density function represents the probability of a lost person being at a certain point in an area. A variety of military applications may exist as well.

We employ Voronoi tessellations of the domain in our design of control law to provide optimal coverage over a time-varying function. Voronoi tessellations have received considerable attention for their usefulness in diverse application areas, such as in image processing, statistics, and animal behaviors among others (Du et al. (1999), Aurenhammer (1991)). In particular, Voronoi tessellations have important implications in mobile sensor networks and optimal coverage where the concept of centroidal Voronoi tessellation (CVT) arises naturally. Given its practical importance, various algorithms that guarantee convergence to CVT have been proposed (Lloyd (2006), Cortés and Bullo (2005), Du and Emelianenko (2006), Liu et al. (2009)).

Relatively little work has been done for the case when the density function is time-varying. Some work done on the time-varying case include Cortes et al. (2002), Lekien and Leonard (2009). Cortes et al. (2002) presents an algorithm for optimal coverage of time-varying density functions. Although the purpose of the algorithm is identical to the one presented in this paper, the algorithm in Cortes et al. (2002) assumes some properties about the time evolution of the density function and that is not generally met. Also, some contributions from time derivatives resulting from agent movements are ignored. In Lekien and Leonard (2009), optimal coverage control of density functions in generalized Voronoi tessellations using cartograms is discussed. The authors define optimal in a slightly different sense that each agent must have equal amount of 'resources' in their generalized Voronoi cells. Nevertheless, the density function is required to change slowly enough with time for stability to be guaranteed for the algorithm presented in Lekien and Leonard (2009).

In this paper, we propose a new algorithm that causes the agents to maintain optimal coverage of the density function. This is achieved in the sense that the agents are to first converge to CVT under time-invariant choice of density function, and then the density function is switched to a time-varying one. Under the proposed algorithm, the agents maintain CVT for time-varying density functions if the agents were in CVT initially. We use the proposed algorithm to achieve desired behavior of the network of agents while viewing the choice of the time-varying density function as an input. The proposed algorithm places no additional assumptions on the time evolution of the density function, at the expense of becoming a centralized control law.

2. PROBLEM STATEMENT

Let $D \subset \mathbb{R}^N$ be a convex domain. Let $f : D \to \overline{\mathbb{R}}^+$ be a non-decreasing, piecewise continuous function. Let $\phi : D \times \overline{\mathbb{R}}^+ \to \mathbb{R}^+$ be bounded and continuously differentiable. Also, let $p_i \in D$ be the position of the ith agent in the domain. Let $P = \{P_1, \cdots, P_n\}$ be a partition of D such that $p_i \in P_i$ for all $i = 1, \cdots, n$. A multi-agent system can be said to be optimally covering a domain with respect to the density function ϕ if they are in a configuration that minimizes the cost function

$$H(p,P,t) = \sum_{i=1}^{n} \int_{P_i} f(\|q - p_i\|)\phi(q,t)dq.$$

Therefore, it is of interest to find the control dynamics $\dot{p}_i$ such that the agents are driven to minimize the given cost function.

[★] This work was sponsored by the AFOSR through grant number FA9550-13-1-0029.

Here, $t \geq 0$ represents time, and p is an aggregated variable for p_i, $i = 1, \cdots, n$. Throughout this paper, it will be assumed that $N = 2$. Since we are interested in optimal coverage of the density function of our choice, we are interested in the sensing performance of each agent. The performance of a large class of sensors deteriorate with a rate proportional to the square of the distance (Meguerdichian et al. (2001), Adlakha and Srivastava (2003)). Let us focus on this class of sensors and set $f(x) = x^2$.

$$H(p,P,t) = \sum_{i=1}^{n} \int_{P_i} \|q - p_i\|^2 \phi(q,t)dq. \tag{1}$$

Then f can intuitively be interpreted to represent the decrease in sensing abilities as the distance from the sensor to the point in space increases, while ϕ represents the relative importance of that point at time t. As such, when a configuration of positions of the agents minimizes (1), then the agents can be said to be providing optimal coverage of the domain.

For the case $\phi(q,t) = \phi(q)$, that is, if ϕ is time-invariant, Lloyd's algorithm is known to cause the agents to converge to a configuration such that (1) attains a local minimum. However, as mentioned in the previous section, there does not yet exist an algorithm that guarantees convergence to a local minimum for general time-varying density function ϕ. We wish to be able to use general time-varying ϕ in our approach of choosing density functions as inputs to multi-agent systems.

Problem: Develop an algorithm that drives the multi-agent system to converge to a configuration that minimizes (1) for general time-varying density function ϕ.

3. REVIEW OF CENTROIDAL VORONOI TESSELLATIONS

At each t, (1) is a function of two different variables, the choice of partition P, and the position of the agents, and as such, must be minimized over both variables. However, due to the non-decreasing nature of f, the choice of P that minimizes (1) is (Cortes et al. (2002))

$$P_i = \{x \in D \mid \|x - p_i\| \leq \left\|x - p_j\right\|, i \neq j\}$$

which is the definition of Voronoi tessellations. We denote this choice of partitions with the symbol V_i. Hence, with $P_i = V_i$, (1) is minimized over P and we can rewrite (1) as

$$H(p,t) = \sum_{i=1}^{n} \int_{V_i} \|q - p_i\|^2 \phi(q,t)dq = \sum_{i=1}^{n} H_i(p,t) \tag{2}$$

which is a function of positions of agents only at each t. It is known that (Iri et al. (1984), Du et al. (1999))

$$\frac{\partial H}{\partial p_i} = \int_{V_i} \frac{\partial}{\partial p_i} f(\|q - p_i\|)\phi(q,t)dq.$$

Since we limit ourselves to discussion for the case $f(x) = x^2$ only, we have

$$\frac{\partial H}{\partial p_i} = \int_{V_i} -2(q - p_i)^T \phi(q,t)dq.$$

Note, however, that

$$\frac{\partial H}{\partial p_i} \neq \frac{\partial H_i}{\partial p_i}$$

due to the fact the derivative involves the integration area as well. However, these contributions cancel out when computing the first derivative of H. But, as will be seen, we need to pay attention to these effects once we compute higher order derivatives.

Let us define some quantities by appealing to physical interpretation of the problem at hand.

$$m_i(p,t) = \int_{V_i} \phi(q,t)dq \tag{3}$$

$$c_i(p,t) = \frac{\int_{V_i} q\phi(q,t)dq}{m_i} \tag{4}$$

Since $\phi > 0$, ϕ can be interpreted as a (time-varying) mass density function. Then m_i represents the mass and c_i represents the center of mass of the ith Voronoi cell. With these notations, the partial derivative above can be written simply as

$$\frac{\partial H}{\partial p_i} = 2m_i(p_i - c_i)^T$$

From this expression, we can see that a critical point of (2) is

$$p_i(t) = c_i(p,t), \quad i = 1, \cdots, n$$

and a minimizer for (2) is necessarily in this configuration (Du et al. (2006)). This configuration of agents is defined as the centroidal Voronoi tessellation. Note that CVT is not unique. Also, it is known that agents being in CVT configuration does not imply that the global minimum of (2) is attained (Cortes et al. (2005)).

4. ALGORITHM FOR GENERAL TIME-VARYING DENSITY FUNCTIONS

Let us assume the agents obey a first order behavior given by

$$\dot{p}_i = u_i.$$

It is known that if ϕ (and hence (2)) is time invariant, by setting

$$u_i = -k(p_i - c_i)$$

where k is a positive gain, the multi-agent system is driven to CVT configuration. The way to see this is, as was done in Cortes et al. (2004), is to take (2) as the Lyapunov function candidate itself.

$$\begin{aligned}
\frac{d}{dt}H(p) &= \sum_{i=1}^{n} \frac{\partial}{\partial p_i} H(p,t)\dot{p}_i \\
&= \sum_{i=1}^{n} 2m_i(p_i - c_i)^T(-k(p_i - c_i)) \\
&= -2k\sum_{i=1}^{n} m_i \|p_i - c_i\|^2
\end{aligned}$$

By LaSalle's invariance principle, the multi-agent system asymptotically converges to a configuration $\{\|p_i - c_i\|^2 = 0,\ i = 1, \cdots, n\}$, which is the definition for CVT.

However, if ϕ is time-varying, the same control law does not stabilize the multi-agent system to CVT. This point can be seen intuitively from the following expression.

$$\begin{aligned}
\frac{d}{dt}H(p,t) &= \sum_{i=1}^{n} \frac{\partial}{\partial p_i} H(p,t)\dot{p}_i + \frac{\partial}{\partial t} H(p,t) \\
&= -2k\sum_{i=1}^{n} m_i \|p_i - c_i\|^2 + \sum_{i=1}^{n} \int_{V_i} \|q - p_i\|^2 \frac{\partial \phi}{\partial t}(q,t)dq \not< 0
\end{aligned}$$

The difficulty in designing a control law for time-varying case arises from the fact that we do not know how the term $\frac{\partial \phi}{\partial t}$ behaves in general. Cortes et al. (2002) attempts to work around this difficulty by assuming some properties about ϕ, but the assumptions made are too restrictive and cannot be expected to hold in general. On the other hand, we propose to specify

ϕ as the input signal, while the control law to be presented ensures that the agents are guided well by the chosen ϕ. To this end, let us define $p^T = [p_1^T \cdots p_n^T]$, $u^T = [u_1^T \cdots u_n^T]$ and $c^T = [c_1^T \cdots c_n^T]$.

Theorem 1. Assume first order dynamics $\dot{p} = u$. Further assume that at some time t_0, $p(t_0) = c(t_0)$. Then for time-varying density function $\phi(q,t)$ that is bounded and continuously differentiable,

$$\|p(t) - c(p(t),t)\| = 0, \quad \forall t \geq t_0$$

under the control law

$$u = \left(I - \frac{\partial c}{\partial p}\right)^{-1} \frac{\partial c}{\partial t}. \tag{5}$$

Proof: Since $p(t_0) = c(t_0)$ at t_0, we wish to pick $\dot{p} = u$ such that

$$\frac{d}{dt}(p(t) - c(p(t),t)) = 0 \quad \forall t \geq t_0.$$

Let us choose

$$\dot{p} = \left(I - \frac{\partial c}{\partial p}\right)^{-1} \frac{\partial c}{\partial t}.$$

It is easy to see that this can be rearranged to

$$\left(I - \frac{\partial c}{\partial p}\right)\dot{p} = \frac{\partial c}{\partial t}$$

$$\dot{p} - \frac{\partial c}{\partial p}\dot{p} = \frac{\partial c}{\partial t}$$

$$\dot{p} = \frac{\partial c}{\partial t} + \frac{\partial c}{\partial p}\dot{p}$$

But

$$\dot{c}(p,t) = \frac{\partial c}{\partial t} + \frac{\partial c}{\partial p}\dot{p}$$

Therefore, $\dot{p} = \dot{c}$. Hence,

$$\frac{d}{dt}(p(t) - c(p(t),t)) = 0 \quad \forall t \geq t_0,$$

as was desired. Therefore,

$$\|p(t) - c(p(t),t)\| = 0, \quad \forall t \geq t_0.$$

□

Note that we treat the density function ϕ as an input to the multi-agent system, so we are able to pick $\phi = \phi(q,t_0)$ such that $\dot{\phi} = 0$ initially. At this point, we run time-invariant algorithms such as the Lloyd algorithm until the agents converge to CVT. Once the agents converge, we switch to (5) and let ϕ vary with time. Note also that in practice, (5) is susceptible to errors due to disturbances and the fact that the agent dynamics were conveniently assumed to be in first order. In order to compensate for these sources of error, we include a proportional term as the following.

$$u = \left(I - \frac{\partial c}{\partial p}\right)^{-1} \left(-k(p-c) + \frac{\partial c}{\partial t}\right) \tag{6}$$

It is known that if the density function is log-concave, then $\left(I - \frac{\partial c}{\partial p}\right)^{-1}$ exists (Du and Emelianenko (2006)). Since this control law requires computation of the nontrivial partial derivative $\frac{\partial c}{\partial p}$, let us now show an expression for it. We need some additional facts in order to do so. We first present a lemma that will allow us to find the partial derivative of a function with respect to p_i over a Voronoi cell (Du and Emelianenko (2006)).

Lemma 1. Let $\Omega = \Omega(U)$ be a region that is a smooth function of U. Also, let Ω have a well-defined boundary. Let

$$F = \int_{\Omega(U)} f(q)dq.$$

Then

$$\frac{\partial F}{\partial U} = \int_{\partial\Omega(U)} f(q)\dot{q} \cdot ndq$$

where $\dot{\imath}$ the derivative of the boundary points of Ω with respect to U, and n is the unit outward normal vector.

From the same paper, we also know that following fact that allows us to find an analytical expression for a needed term when we apply the previous lemma to our problem. Let $S_{ij} = \{v_1, \cdots, v_\gamma\}$ be the set of vertices of the common surface $\partial V_{i,j}$ between two neighboring Voronoi regions generated by p_i and p_j. Let $\lambda_i, i = 1, \cdots, \gamma$ be nonnegative numbers such that

$$\sum_i \lambda_i = 1,$$

and therefore

$$\sum_i \lambda_i v_i \in \partial V_{i,j}.$$

Then

$$\left(\sum_i \lambda_i v_i - \frac{p_i + p_j}{2}\right) \cdot (p_j - p_i) = 0.$$

Taking partial derivatives of the last expression allows us to conclude that for any point $q \in \partial V_{i,j}$ (Du and Emelianenko (2006)),

$$\frac{\partial q}{\partial p_j^{(b)}} \cdot (p_j - p_i) = \frac{1}{2} e_b \cdot (p_j - p_i) - e_b \cdot \left(q - \frac{p_i + p_j}{2}\right)$$

$$\frac{\partial q}{\partial p_i^{(b)}} \cdot (p_j - p_i) = \frac{1}{2} e_b \cdot (p_j - p_i) + e_b \cdot \left(q - \frac{p_i + p_j}{2}\right)$$

where $p^{(b)}$ denotes b-th component of vector p and e_b is elementary unit vector. This expression can be divided by $\|p_j - p_i\|$ and rearranged to

$$\frac{\partial q}{\partial p_j^{(b)}} \cdot n = e_b \cdot \frac{p_j - q}{\|p_j - p_i\|} = \frac{p_j^{(b)} - q^{(b)}}{\|p_j - p_i\|}$$

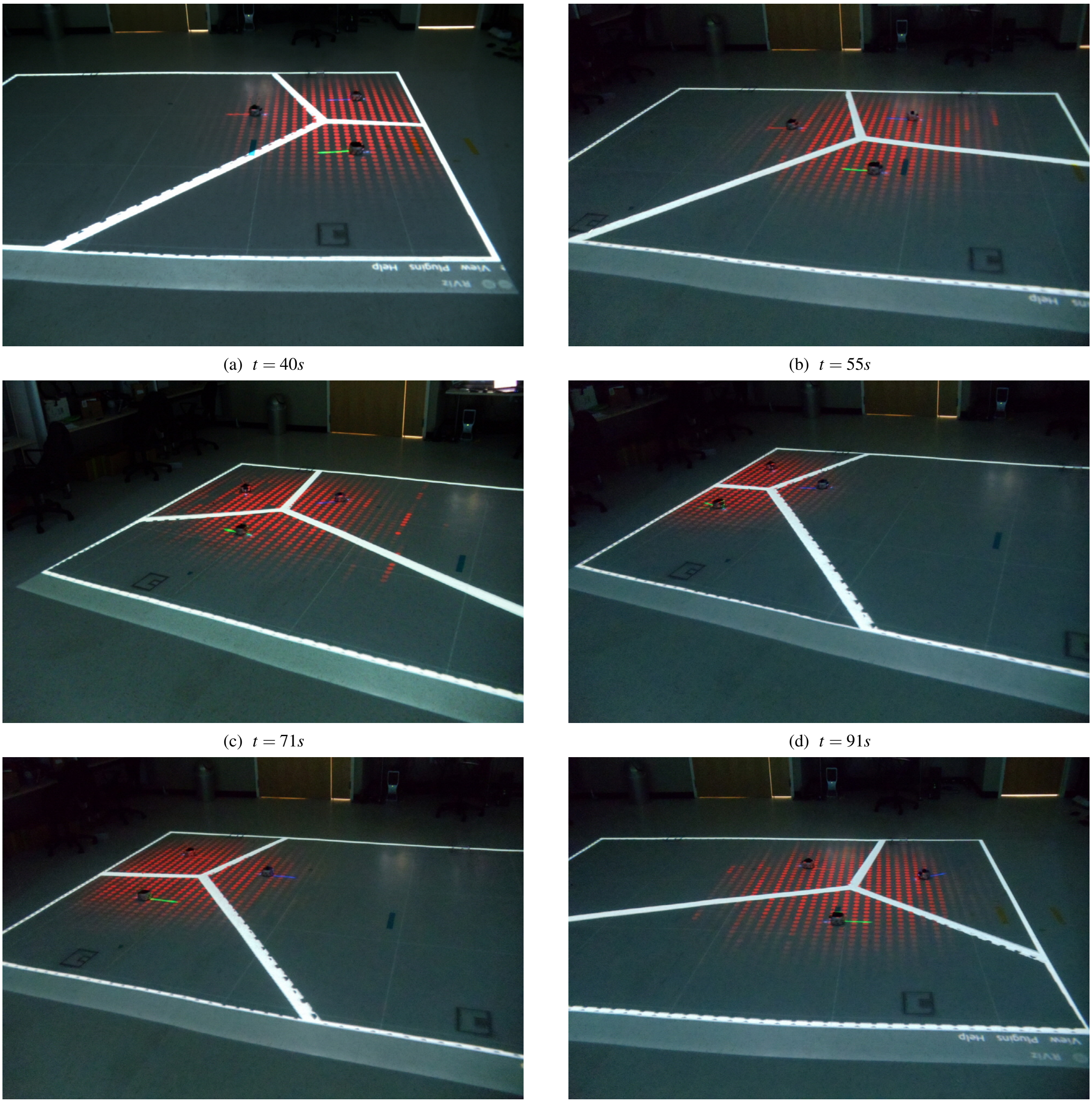
$$\frac{\partial q}{\partial p_i^{(b)}} \cdot n = e_b \cdot \frac{q - p_i}{\|p_j - p_i\|} = \frac{q^{(b)} - p_i^{(b)}}{\|p_j - p_i\|}.$$

From (4) and lemma 1, we know that

$$\frac{\partial c_i^{(a)}}{\partial p_j^{(b)}} = \left(\int_{\partial V_{i,j}} \phi(q) q^{(a)} \frac{\partial q}{\partial p_j^{(b)}} \cdot ndq\right) \Big/ m_i -$$
$$\left(\int_{\partial V_{i,j}} \phi(q) \frac{\partial q}{\partial p_j^{(b)}} \cdot ndq\right) \left(\int_{V_i(P)} \phi(q) q^{(a)} dq\right) \Big/ m_i^2.$$

Substituting the result above, we obtain

$$\frac{\partial c_i^{(a)}}{\partial p_j^{(b)}} = \left(\int_{\partial V_{i,j}} \phi(q) q^{(a)} \frac{p_j^{(b)} - q^{(b)}}{\|p_j - p_i\|} dq\right) \Big/ m_i$$
$$- \left(\int_{\partial V_{i,j}} \phi(q) \frac{p_j^{(b)} - q^{(b)}}{\|p_j - p_i\|} dq\right) \left(\int_{V_i(P)} \phi(q) q^{(a)} dq\right) \Big/ m_i^2. \tag{7}$$



(a) $t = 40s$

(b) $t = 55s$

(c) $t = 71s$

(d) $t = 91s$

(e) $t = 109s$

(f) $t = 135s$

Fig. 1. The proposed algorithm in action. The time constant was chosen to be $\tau = 20$. An overhead projector is visualizing pertinent information. The thick, white lines represent the Voronoi cells. The red dots on the floor visualizes the density function at each t - the brighter red a point is, the higher is the density at the point. Lastly, the arrows accompanying each agent shows the position and the orientation of the agent.

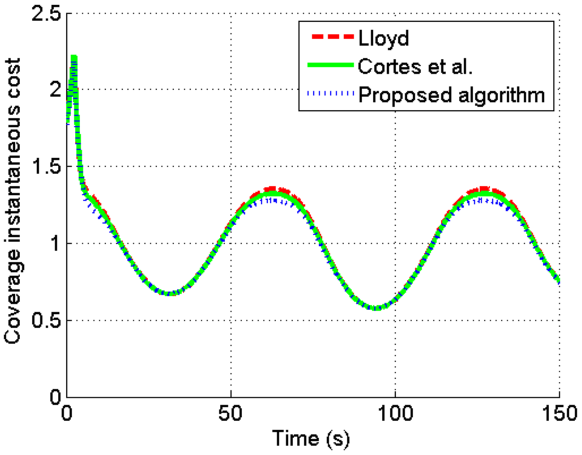


Fig. 2. Computed instantaneous cost with $\tau = 20s$.

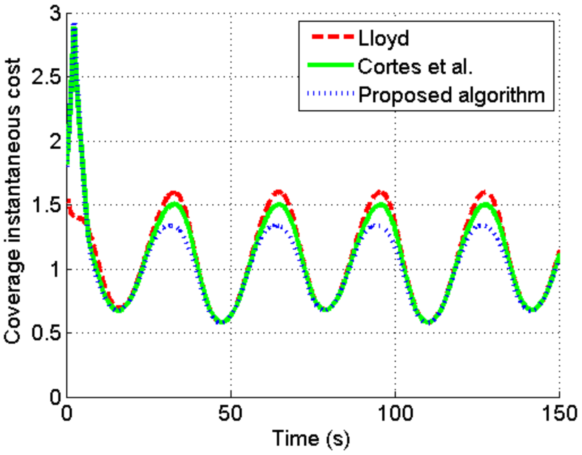


Fig. 3. Computed instantaneous cost with $\tau = 10s$.

Similarly,

$$\frac{\partial c_i^{(a)}}{\partial p_i^{(b)}} = \left(\int_{\partial V_{i,j}} \phi(q) q^{(a)} \frac{q^{(b)} - p_i^{(b)}}{\|p_j - p_i\|} dq\right) \Big/ m_i - \left(\int_{\partial V_{i,j}} \phi(q) \frac{q^{(b)} - p_i^{(b)}}{\|p_j - p_i\|} dq\right) \left(\int_{V_i(P)} \phi(q) q^{(a)} dq\right) \Big/ m_i^2.$$

Note that for each $\frac{\partial c_i^{(a)}}{\partial p_i^{(b)}}$, if N_{V_i} denotes the set of Voronoi neighbors of agent i, then the contribution from each $j \in N_{V_i}$ must be considered in $\frac{\partial c_i^{(a)}}{\partial p_i^{(b)}}$. That is,

$$\frac{\partial c_i^{(a)}}{\partial p_i^{(b)}} = \sum_{j \in N_{V_i}} \left[\left(\int_{\partial V_{i,j}} \phi(q) q^{(a)} \frac{q^{(b)} - p_i^{(b)}}{\|p_j - p_i\|} dq\right) \Big/ m_i - \left(\int_{\partial V_{i,j}} \phi(q) \frac{q^{(b)} - p_i^{(b)}}{\|p_j - p_i\|} dq\right) \left(\int_{V_i(P)} \phi(q) q^{(a)} dq\right) \Big/ m_i^2 \right]. \tag{8}$$

We now have the complete expression for $\frac{\partial c}{\partial p}$. From (4),

$$\frac{\partial c_i}{\partial t} = \frac{m_i \int_{V_i} q \dot{\phi}(q,t) dq - \frac{\partial m_i}{\partial t} \int_{V_i} q \phi(q,t) dq}{m_i^2} \tag{9}$$

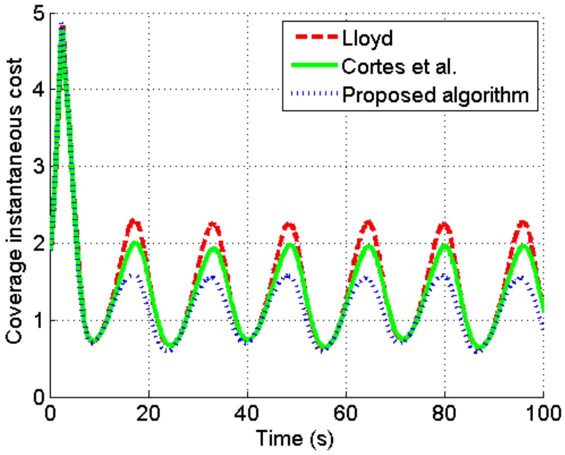


Fig. 4. Computed instantaneous cost with $\tau = 5s$.

and

$$\frac{\partial c}{\partial t}^T = [\frac{\partial c_1}{\partial t}^T \cdots \frac{\partial c_n}{\partial t}^T].$$

With this, we have all the expressions needed to compute the control law (5) and (6).

5. ROBOTIC IMPLEMENTATION

In this section, an algorithm based on (6) is proposed. The algorithm maintains CVT configuration given that initial positions of the agents start close to a CVT. The utility of the algorithm is demonstrated with a robotic implementation.

Algorithm 1

```
Require: $p(t_0) = c(t_0)$
Ensure: $p(t) = c(t), \quad t \geq t_0$
while $p(t) \neq c(t)$ do
    Set $u = -k(p - c)$ (Lloyd algorithm) with $\phi(q,t), t = t_0$.
end while
while $t_0 \leq t$ do
    Compute the Voronoi tessellations
    for $i = 1 : n$ do
        Compute $m_i$, $c_i$ according to (3), (4)
        Compute $\frac{\partial c_i}{\partial p_j}$ for $j \in N_{V_i}$ according to (7) and (8)
        Compute $\frac{\partial c_i}{\partial t}$ according to (9)
    end for
    Form $\frac{\partial c}{\partial p}$ and $\frac{\partial c}{\partial t}$.
    Set $u$ according to (6)
end while
```

The proposed algorithm was implemented for robotic experiments. The ROS (Robot Operating System, version Diamondback) framework running on Ubuntu (version 11.04) machine with Intel dual core CPU 2.13GHz, 4GB memory was used to implement the algorithm and send control signals to individual agents over a wireless router. Three Khepera III robots from K-team were used as the team of mobile agents for the experiment. The Khepera III robots each have a 600MHz ARM processor with 128Mb RAM, embedded Linux, differential drive wheels, and a wireless card for communication over a wireless router. 10 Optitrack S250e motion capture cameras provide very accurate position and orientation data for the agents, which were used to provide the information required for the algorithm and the computation of Voronoi partitions. All codes were written in C++. All integrations were performed numerically. The rviz package in ROS was used to visualize many pertinent information regarding the problem, such as the position and the heading of the robots, the density function, and the Voronoi partitions. The visualization was overlapped with the real physical environment to give a real-time visual representation (Fig. 1).

The Khepera III robots are modeled as unicycles

$$\begin{aligned} \dot{x}_i &= v_i \cos\theta_i \\ \dot{y}_i &= v_i \sin\theta_i \\ \dot{\theta}_i &= \omega_i \end{aligned}$$

where we choose

$$\begin{aligned} v_i &= \|\dot{p}_i\| \\ \omega_i &= [-\sin\theta_i \;\; \cos\theta_i] \cdot \frac{\dot{p}_i}{\|\dot{p}_i\|}. \end{aligned}$$

Here, the $\cdot$ symbol denotes the dot product. The time-varying density function used was

$$\phi(q,t) = e^{-(q_x - 2sin\frac{t}{\tau})^2 - q_y^2},$$

where τ is a time constant that was varied to see how performance of different algorithms changed as τ changed. Three algorithms were compared: (i) Lloyd's algorithm, (ii) the algorithm from Cortes et al. (2002) for time-varying case, and (iii) the algorithm proposed in this paper. A brief review of the other two algorithms run for the experiments is as follows.

Algorithm (i) is the simplest of all three. It has the advantage of being computationally cheap and the capability to be implemented in distributed fashion. It involves simply setting $\dot{p}_i = -k(p_i - c_i)$. Note that this algorithm was not developed to be used in time-varying applications, but we use it as a baseline to compare the performance of other algorithms.

Algorithm (ii) is designed to be used in time-varying applications. It can also be implemented in distributed fashion. It involves setting

$$\dot{p}_i = c_{i,t} - (k + \frac{m_{i,t}}{m_i})(p_i - c_i),$$

where

$$m_{i,t} = \int_{V_i} \dot{\phi}(q,t)dq, \quad c_{i,t} = \frac{1}{m_i}\left(\int_{V_i} q\dot{\phi}(q,t)dq - m_{i,t}c_i\right).$$

Note that $m_{i,t}$ approximates $\frac{d}{dt}m_i$ and $c_{i,t}$ approximates $\frac{d}{dt}c_i$.

For all runs, the robots started at the same initial position and orientation, not necessarily in CVT. The proposed algorithm was tested starting from line 4.

Fig.2-4 shows the graph of instantaneous coverage cost numerically calculated by (2). For each graph, all three algorithms were run with a fixed time constant τ. Table.1 shows the total cost from each experiment, obtained by numerically integrating the graphs on Fig.2-4. In essence, the entries of Table.1 are obtained from numerically evaluating

$$\int_0^T H(t)dt \tag{10}$$

where H is the instantaneous cost (Fig.2-4). For $\tau = 20s$ and $\tau = 10s$, $T = 150s$. For $\tau = 5s$, $T = 100s$.

It can be seen from Fig.2-4 that in all three cases, (ii) performs better than (i) in the sense that that the instantaneous cost for running (ii) is less than the instantaneous cost for running (i). Similarly, (iii) performs better than (ii) in terms of the instantaneous cost. The total cost agrees with this observation, as can be seen from Table.1. The total cost for running (iii) is always less than the total cost for running (ii). Similarly, the total cost for running (ii) is smaller than the total cost for running (i), with the exception for the case $\tau = 10$. But for this case, inspection of Fig.3 shows that the agents converged under (i) unusually quickly, and is very likely that this is an anomaly. As time constant decreases – meaning as ϕ moves more vigorously – the difference in performance between these algorithms increases, measured by both instantaneous and total costs. This results in the relative performance of the proposed algorithm increasing as the time constant decreases. This shows that the proposed algorithm is more suited for optimal coverage when density functions are the more 'dynamic' in time.

Table 1. Total cost for each cases, obtained from numerical integration of instantaneous costs over time from $t = 0$ to final time. See equation (10).

	$\tau = 5s$	$\tau = 10s$	$\tau = 20s$
Lloyd algorithm (Lloyd (2006))	149.5	112.4	100.5
Cortes et al. (Cortes et al. (2002))	140.0	113.0	99.5
Proposed algorithm	122.3	107.2	97.9

As mentioned before, (i) was not meant to be used for time-varying applications, which is why its performance was relatively low. However, it has a large advantage of being computationally cheap and still yielding results that are not too different from algorithms that are specifically designed for time-varying density functions if the time constant is low. Algorithm (ii) had intermediate performance of the three. It yielded much better results when the density function varies faster with time than (i). While its performance was less than (iii), it must be noted that (ii) has much lower computational complexity and the ability to be implemented in distributed fashion. For some applications, it may not be possible to give up on these traits, so (ii) may be more suitable than (iii).

6. CONCLUSION

In this paper, a new approach to controlling a system of multiple agents by manipulating time-varying density functions was proposed. The time-varying density function is considered as the input, while the proposed control law guides the agents to track the chosen density function. Given that the agents start near a CVT, the agents follow the density function while providing optimal coverage of it on the motion. The choice of the density function is limited only by the condition that it must be bounded and continuously differentiable. The proposed algorithm has the disadvantage of being a centralized control law, and more computationally expensive then other comparable algorithms. The experiments from actual robot implementation show that the proposed algorithm performs better than other comparable algorithms in terms of instantaneous and total cost when the density function is time-varying. The faster the density function varied with time, the better relative performance the proposed algorithm showed.

REFERENCES

Adlakha, S., Srivastava, M. B., 2003. Critical density thresholds for coverage in wireless sensor networks. In: WCNC. IEEE, pp. 1615–1620.

Aurenhammer, F., Sep. 1991. Voronoi diagrams–a survey of a fundamental geometric data structure. ACM Comput. Surv. 23 (3), 345–405.

Cortés, J., Bullo, F., October 2005. Coordination and geometric optimization via distributed dynamical systems. SIAM Journal on Control and Optimization 44 (5), 1543–1574.

Cortes, J., Martinez, S., Bullo, F., 2005. Spatially-distributed coverage optimization and control with limited-range interactions 11, 691–719.

Cortes, J., Martinez, S., Karatas, T., Bullo, F., 2002. Coverage control for mobile sensing networks: Variations on a theme. In: Mediterranean Conference on Control and Automation. Lisbon, Portugal, Electronic Proceedings.

Cortes, J., Martinez, S., Karatas, T., Bullo, F., Apr. 2004. Coverage control for mobile sensing networks. IEEE Transactions on Robotics and Automation 20 (2), 243–255.

Du, Q., Emelianenko, M., 2006. Acceleration schemes for computing centroidal voronoi tessellations. Numerical Linear Algebra with Applications 13 (2-3), 173–192.

Du, Q., Emelianenko, M., Ju, L., Jan. 2006. Convergence of the lloyd algorithm for computing centroidal voronoi tessellations. SIAM Journal on Numerical Analysis 44 (1), 102–119.

Du, Q., Faber, V., Gunzburger, M., Dec. 1999. Centroidal voronoi tessellations: Applications and algorithms. SIAM Review 41 (4), 637–676.

Iri, M., Murota, K., Ohya, T., 1984. A fast voronoi-diagram algorithm with applications to geographical optimization problems. In: Thoft-Christensen, P. (Ed.), System Modelling and Optimization. Vol. 59 of Lecture Notes in Control and Information Sciences. Springer Berlin Heidelberg, pp. 273–288.

Lekien, F., Leonard, N. E., Feb. 2009. Nonuniform coverage and cartograms. SIAM Journal on Control and Optimization 48 (1), 351–372.

Liu, Y., Wang, W., Lévy, B., Sun, F., Yan, D.-M., Lu, L., Yang, C., Sep. 2009. On centroidal voronoi tessellation – energy smoothness and fast computation. ACM Transactions on Graphics 28 (4), 1–17.

Lloyd, S., Sep. 2006. Least squares quantization in pcm. IEEE Transactions on Information Theory 28 (2), 129–137.

Meguerdichian, S., Koushanfar, F., Qu, G., Potkonjak, M., 2001. Exposure in wireless ad-hoc sensor networks. In: Proceedings of the 7th annual international conference on Mobile computing and networking. MobiCom '01. ACM, New York, NY, USA, pp. 139–150.