

사용자 입력 문자에 반응하는 군집 로봇 행동 구현

Behavior Realization of Multi-Robots Responding to User's Input Characters

조영래, 이길호, 조성호*, 신인식
(Youngrae Jo¹, Kilho Lee¹, Sungho Jo¹, and Insik Shin¹)

¹Korea Advanced Institute of Science and Technology

Abstract: This paper presents an approach to implement the behaviors of multi-robots responding to user's input characters. The robots are appropriately displaced to express any input characters. Using our method, any user can easily and friendly control multi-robots. The responses of the robots to the user's input are intuitive. We utilize the centroidal Voronoi algorithm and the continuous-time Lloyd algorithm, which have popularly been used for the optimal sensing coverage problems. Collision protection is considered to be applied for real robots. LED sensors are used to identify positions of multi-robots. Our approach is evaluated through experiments with five mobile robots. When a user draw alphabets, the robots are deployed correspondingly. By checking position errors, the feasibility of our method is validated.

Keywords: multi-robot, centroidal Voronoi tessellation, density function, mobile robot, localization

I. 서론

군집 로봇의 목적은 단일 로봇이 하기 어려운 업무를 여러 로봇의 협력을 통하여 보다 효과적으로 수행할 수 있도록 하는 것이다. 여러 로봇의 협력으로 임무 수행을 보다 빨리 할 수 있고 이런 장점은 빠른 시간내 위험 지역 또는 오염 지역 감시 및 탐사에 적합한 방법을 제공할 수 있다. 협력적 분업 수행 등은 여러 산업 현장에 군집 로봇의 적용도 활용도가 클 수 있다.

이와 같은 이유로 군집 로봇에 관련된 연구가 활발하게 진행 중이다. 실제로 협력 작업을 통한 물건 운반[1,2], 환경 탐사[3]와 미술관 문제(art gallery problem) [4] 같은 건물 감시 문제와 군집 로봇의 운행 능력 향상을 위해 에너지 소비량의 최소화 그리고 효율적 센서 사용을 위한 연구[5] 등이 진행되고 있다.

일반적인 상황에서 비 효율적인 협력 업무는 오히려 전체 목적 달성을 방해한다. 효율적인 협력을 위해서는 전체 목적에 적합한 로봇 배치가 이루어지고 능력 향상을 위한 임무 분담을 적절히 하는 것이 중요한 문제이다[6]. 따라서 현재 로봇 형태 제어(formation control)문제와 임무 분담 문제(scheduling)문제가 주요 연구 방향이다. 이 논문에서는 로봇 배치 문제에 관련한 연구를 논한다. 대부분의 군집 로봇의 형태 제어 문제는 배치 형태보다는 로봇간의 관계설정 방식에 초점이 맞추어져 있다[7-9]. 본 연구에서는 관점을 달리하여, 사용자의 입력에 반응하여 군집 로봇들을 자신들끼리의 통신을 최소화하고 적절한 형태로 배치하는 방법을 제안하

고자 한다. 특히, 사용자가 입력 문자를 그렸을 때, 이에 반응하여 군집 로봇들이 직관적으로 적절한 배치를 형성하도록 한다. 제안한 방법을 이용하면 군집 로봇들을 보다 쉽게 그리고 직관적으로 제어할 수 있을 것이다. 또한 확인 가능한 엔터테인먼트적인 요소를 가짐으로써 일반 사용자에게 흥미로운 로봇 행동 구현에 활용할 수 있다.

제안한 방법론은 중심 보로노이 알고리즘[10]과 연속 시간 로이드 알고리즘[11]을 활용한다. 이 알고리즘들은 군집 로봇 센싱 커버리지 문제를 푸는 방법으로 많이 사용되어 왔다 [6,10-14]. 특히 로봇의 크기가 존재하지 않고 각 로봇의 센싱 범위가 일정한 경우[12], 센서 범위가 비등방성인 경우[13] 등에 관한 연구, 실험 환경이 불록한 경우가 아니라 오목한 경우를 고려하여 장애물이 있는 환경에서도 중심 보로노이 알고리즘을 이용할 수 있도록 하는 연구[14] 등이 있었다. 본 연구에서는 센싱 커버리지 관점이 아니라 인터페이스 방법으로 보로노이 알고리즘을 활용하고자 한다.

본 논문은 다음과 같이 구성되었다. II 장에서는 모바일 로봇의 위치 최적화에 대해서 설명한다. 중심 보로노이 조각화 및 연속 시간 로이드 알고리즘에 대해서 언급하고 이에 필요한 밀도 함수를 정의하는 방법에 대해서 설명한다. III 장에서는 로봇 제어기에 대해서 설명한다. IV 장은 로봇이 위치 추정하는 방법에 대해서 설명한다. V 장에서는 실험 환경 구성에 대해서 설명하고 실제 로봇들을 이용한 실험 결과를 정리한다. 마지막 VI 장은 본 연구를 결론 짓는다.

II. 모바일 로봇들 위치 결정

1. 위치 최적화

$P = (p_1, \dots, p_n)$ 를 보로노이 다이어그램(voronoi diagram)의 생성자(generator)로 정의하자[10]. 이 연구에서 각 생성자는 각 로봇 위치를 의미한다. Q 는 R^2 내에 존재하는 불록 다각형이라 정의한다.

* 책임저자(Corresponding Author)

논문접수: 2012. 2. 26., 수정: 2012. 3. 20., 채택확정: 2012. 3. 27.

조영래, 이길호, 조성호, 신인식: 한국과학기술원

(joyrae@kaist.ac.kr/phantasia17@naver.com/shjo@cs.kaist.ac.kr/insik.shin@cs.kaist.ac.kr)

본 연구는 지식경제부/한국산업기술진흥원 융복합형로봇전문인력양성사업 및 교과부 일반연구자사업(2011-0003572)의 지원으로 수행되었음.

Q 에서 사건이 발생할 확률이나 정보를 가지고 있는 정도를 나타내는 밀도 함수를 $\rho: Q \rightarrow R_+$ 로 정의한다. 로봇 센서에는 분해능의 손실이나 잡음이 존재하기 때문에 \mathbf{p}_i 에 존재하는 i 번째 로봇이 Q 내에 존재하는 한 점 \mathbf{q} 에서의 감지 능력을 유클리디안 거리 $\|\mathbf{q} - \mathbf{p}_i\|$ 가 클수록 감소한다고 가정하고 이를 비감소 함수 $f: R_+ \rightarrow R_+$ 로 표현한다. $f(\|\mathbf{q} - \mathbf{p}_i\|)$ 는 로봇의 감지 능력을 양적으로 표현한 함수이다. 실제 전자기 센서 또는 음파 센서들의 경우에 신호 대 잡음비는 거리에 역 비례하기 때문에 위와 같은 가정은 적절하다[10].

n 개의 로봇들에 대해 아래의 비용함수[10]를 정의한다.

$$H(P, W) = \sum_{i=1}^n \int_{W_i} f(\|\mathbf{q} - \mathbf{p}_i\|) \rho(\mathbf{q}) d\mathbf{q} \quad (1)$$

여기서 $W = \{W_1, \dots, W_n\}$ 는 Q 의 영역을 나눈 블록 다각형이며 i 번째 로봇은 W_i 영역 내에 존재하기 때문에 센서로 W_i 내의 영역을 감지한다. 그리고 W 의 각각 원소들은 서로 겹치지 않고 W 의 모든 영역의 합은 Q 가 된다.

2. 중심 보로노이 조각화

센서들의 위치 $(\mathbf{p}_1, \dots, \mathbf{p}_n)$ 에 의해 생성되는 보로노이 셀은 Q 의 최적 분할이다[19]. 여기서 보로노이 셀은

$$V_i = \{\mathbf{q} \in Q \mid \|\mathbf{q} - \mathbf{p}_i\| \leq \|\mathbf{q} - \mathbf{p}_j\|, \forall j \neq i\} \quad (2)$$

로 정의되고, 두 보로노이 영역 V_i 와 V_j 가 인접해 있다면 \mathbf{p}_i 는 \mathbf{p}_j 의 보로노이 이웃이라고 하자.

보로노이 알고리즘에서의 일반화된 질량, 무게중심, 극 관성 모멘트는 아래 식들로 각각 정의 된다.

$$\begin{aligned} M_{V_i} &= \int_{V_i} \rho(\mathbf{q}) d\mathbf{q}, \mathbf{C}_{V_i} = \frac{1}{M_{V_i}} \int_{V_i} \mathbf{q} \rho(\mathbf{q}) d\mathbf{q}, \\ J_{V_i, \mathbf{p}_i} &= \int_{V_i} \|\mathbf{q} - \mathbf{p}_i\|^2 \rho(\mathbf{q}) d\mathbf{q} \end{aligned} \quad (3)$$

ρ 는 어떤 밀도 함수를 나타낸다. 이제 다시 위치 최적화 문제를 고려하자.

$$H(P, W) = \sum_{i=1}^n \int_{W_i} \|\mathbf{q} - \mathbf{p}_i\|^2 \rho(\mathbf{q}) d\mathbf{q} \quad (4)$$

위의 식에서 $f(\|\mathbf{q} - \mathbf{p}_i\|) = \|\mathbf{q} - \mathbf{p}_i\|^2$ 으로 가정하면 평행축 정리에 의해서 H_v 와 H_v 의 편미분 결과는 식 (5)로 표현된다.

$$\begin{aligned} H_v(P) &= \sum_{i=1}^n J_{V_i, \mathbf{C}_{V_i}} + \sum_{i=1}^n M_{V_i} \|\mathbf{p}_i - \mathbf{C}_{V_i}\|^2 \\ \frac{\partial H_v}{\partial \mathbf{p}_i}(P) &= 2M_{V_i}(\mathbf{p}_i - \mathbf{C}_{V_i}) \end{aligned} \quad (5)$$

즉, 편미분을 이용해서 H_v 를 최적화 한 결과인 국부적 최소점(local minimum points)은 보로노이 셀의 무게 중심인 \mathbf{C}_{V_i} 이다.

따라서 중심 보로노이 조각화(CVT: Centroidal Voronoi Tessellation)를 위치 최적화 하는데 이용할 수 있음을 알 수

있고 CVT에 의해 로봇을 배치할 수 있다. CVT에서의 밀도 분포 함수는 \mathbf{C}_{V_i} 의 위치에 큰 영향을 미치기 때문에 밀도 분포 함수의 설정은 매우 중요한 문제이다. 이에 대한 자세한 증명 내용은 [15]에 나와있고 밀도 분포 함수를 설정하는 부분은 바로 아래에서 다룰 것이다.

3. 밀도 분포 함수 설정

사용자가 그린 문자 위에 로봇들이 배치되도록 제어하는 것이 이 연구의 방향이다. 위 2.2절에서 로봇은 국부 최소점인 \mathbf{C}_{V_i} 의 위치로 배치되는 것을 보였다. 그런데 \mathbf{C}_{V_i} 는 무게 중심 정의에 의해 밀도 분포 함수에 영향을 받는다. 즉, 군집 로봇의 움직임은 밀도 분포 함수에 영향을 받는다.

이제 밀도 분포 함수를 설정하자. 이 연구에서는 사용자가 그린 문자 형태의 로봇 배치를 위해서 가우시안 분포 함수를 이용했다. 사용자가 문자를 그리는 동안 지나가는 점을 가우시안 분포 함수의 평균으로 설정하여 로봇이 밀도가 높은 무게 중심에 배치되도록 한다. 따라서 가우시안 분포 함수는 위에서 제시한 로봇 배치에 적절하고 Q 는 R^2 내에 존재하기 때문에 이차원 가우시안 분포 함수를 이용하여 밀도 분포 함수

$$\rho(\mathbf{q}) = \max_{t=0,1,\dots,T} (\rho_t(\mathbf{q})) \quad (6)$$

로 설정하고, $\rho_t(\mathbf{q})$ 는

$$\frac{1}{2\pi\sigma^2} \exp \left\{ - \left(\frac{(x - x_m(t))^2}{2\sigma_x^2} + \frac{(y - y_m(t))^2}{2\sigma_y^2} \right) \right\} \quad (7)$$

여기서 t 는 샘플링 시간을 의미하고 T 는 사용자 문자 입력 종료 시간을 나타낸다. $(x, y) \in Q$, $(x_m(t), y_m(t))$ 는 문자 입력 위치를 의미한다. σ_x 와 σ_y 는 각각 상수 값이고 $\sigma^2 = \sigma_x^2 \sigma_y^2$ 이다. 여기서 우리는 x 와 y 축 가우시안 함수의 폭이 일정하도록 $\sigma_x = \sigma_y$ 로 가정한다.

4. 연속 시간 로이드 알고리즘

로이드 알고리즘(Lloyd's algorithm)은 CVT를 계산하기 위한 고전적인 이산시간 방법이다[11]. 이 방법에서는 각 반복과정에서 다음 3가지 단계를 따르게 된다.

- (1) 보로노이 셀 V_i 를 계산한다.
- (2) 각 보로노이 셀의 무게 중심 \mathbf{C}_{V_i} 를 계산한다.
- (3) 로봇들을 각 보로노이 셀의 중심으로 이동시킨다.

비용 함수 H 를 최소화하는 로봇의 위치를 구하고자 한다. 이 때 사용되는 알고리즘은 연속 시간 로이드 알고리즘(continuous-time lloyd algorithm)이다. 이 알고리즘의 초기 버전은 기존연구[11]에서 자세히 설명되어 있고 [16]에서 전반적인 개요를 설명하고 있으며 [14,15]는 이 알고리즘이 여러 분야에 적용된 결과를 보여 준다. 그리고 연속 시간 로이드 알고리즘은 기울기 감소 방법(Gradient descent method)을 따른다.

로봇의 위치 \mathbf{p}_i 는 일차 동적 거동 $\dot{\mathbf{p}}_i = \mathbf{u}_i$ 을 따른다고 가정하자. 아래의 제어기는 시스템이 \mathbf{C}_{V_i} 에 수렴하는 것을 보장한다[10].

$$\mathbf{u}_i = -k_{prop}(\mathbf{p}_i - \mathbf{C}_{V_i}) \quad (8)$$

여기서 k_{prop} 는 양수이고 $V(P) = \{V_1, \dots, V_n\}$ 는 연속적으로 갱신된다.

III. 제어기

1. 로봇 크기로 인한 충돌

보로노이 영역 안에 자유 보로노이 영역(free Voronoi region) [9]을 설정하자. 여기서 자유 보로노이 영역은

$$F_{V_i} = \{\mathbf{q} \in V_i \mid \|\mathbf{q} - \mathbf{q}_{ev_i}\| \geq r_{p_i}, \forall \mathbf{q}_{ev_i}\} \quad (9)$$

로 정의되고, ∂V_i 는 보로노이 영역의 외곽선을 의미한다. r_{p_i} 는 i 번째 로봇의 반경이다. 즉, 자유 보로노이 영역은 크기가 있는 로봇이 충돌 없이 자유롭게 움직일 수 있는 영역이다. 즉, 그림 1과 같이 로봇이 자유 보로노이 영역 밖으로 이동하고자 하는 경우에 충돌이 발생하지 않도록 제어기를 설정해야 한다. 이 때 투영법을 이용해서 \mathbf{u}_i 가 ∂F_{V_i} 의 접선 방향 성분은 유지되고 ∂F_{V_i} 의 수직 방향 성분은 제거해준다.

이는 아래 수식으로 표현된다.

1) $\mathbf{p}_i \in F_{V_i}$ 경우

$$\mathbf{u}_i = -k_{prop}(\mathbf{p}_i - C_{V_i}) \quad (10)$$

2) 그 외의 경우

$$\mathbf{u}_i = -k_{prop}\pi \left(-\frac{\partial H}{\partial \mathbf{p}_i}, \partial F_{V_i} \right) \quad (11)$$

여기서 $\pi \left(-\frac{\partial H}{\partial \mathbf{p}_i}, \partial F_{V_i} \right) = \left(-\frac{\partial H}{\partial \mathbf{p}_i} \cos \theta \right) \mathbf{t}_{il}$ 이고, \mathbf{t}_{il} 는 ∂F_{V_i} 의

l 번째 외곽선의 단위 접선 벡터이다.

2. 모바일 로봇 제어

이 연구에서는 4개의 바퀴가 달린 모바일 로봇을 다루기 때문에 회전이나 직진 움직임을 위해서 각 바퀴의 속도를 결정해주어야 한다. 그 식 [18]은 아래와 같고 비례제어 방식을 이용했다.

$$\begin{aligned} v_l &= k_d e_d - k_\theta e_\theta \\ v_r &= k_d e_d + k_\theta e_\theta \end{aligned} \quad (12)$$

여기서 v_l 과 v_r 은 각각 로봇의 왼쪽, 오른쪽 바퀴의 속도이다. k_d 와 k_θ 은 각각 거리와 각도에 대한 P 이득이고 e_d 과

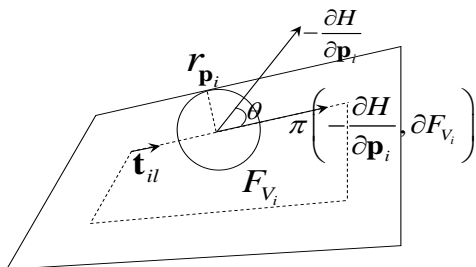


그림 1. 로봇 충돌을 방지 위한 자유 보로노이 영역.
Fig. 1. Free Voronoi region for collision projection.

e_θ 의 자세한 계산 과정은 다음장에서 다룰 것이다.

IV. 로봇의 위치 추정

1. 위치 추정

로봇은 CVT를 이용해서 목적지를 계산하고 목적지로 이동한다. 이 때 정확한 제어를 위해서 로봇은 자신의 위치를 파악하여 목적지에 도착한지를 판단해야 한다. 따라서 로봇의 위치 추정 문제가 로봇 제어 시 중요한 문제가 된다.

그림 2는 시스템 구성도를 보여준다. 천장에 붙여진 9개의 적외선 태그를 기반으로 위치 추정을 한다. 본 시스템은 로봇이 적외선 LED 인식을 필요로 하는데 CBlob 라이브러리 [17]를 이용하여 해결하였다. 카메라에서 얻은 이미지에서 중심이 로봇의 상대적 위치이다. 이미지 내에서 중심과 가장 가까운 태그를 선택해서 위치 추정을 한다. 로봇은 카메라로부터 이미지를 얻어서 위치 추정을 위한 계산을 한다. 그렇기 때문에 이 이미지 내에서 기준이 되는 실 좌표계를 구성할 수 있어야 한다. 그림 3은 로봇에 달려있는 카메라에서 얻은 이미지이고 위치 추정 방법에 대해서 보여주며 그림 3에서 오각형은 적외선 태그를 의미한다. O를 실 좌표계의 원점이라 하자. 그림 3에서 head 부분이 실 좌표계의 x 축을 향한다. 따라서 이미지에서 인식한 head의 방향을 \mathbf{x}_t 라 하고 이는 실 좌표계의 x 축과 동일한 방향이다. \mathbf{x}_t 에 수직인 방향인 \mathbf{y}_t 는 실 좌표계의 y 축과 동일한 방향이다.

로봇 위치 벡터는 아래와 같이 표현된다.

$$\mathbf{p} = \mathbf{t} + \mathbf{r} \quad (13)$$

여기서 \mathbf{t} 는 태그 위치 벡터이므로 알고 있는 값이다. 그리고 \mathbf{r} 는 아래의 식을 통해서 구한다.

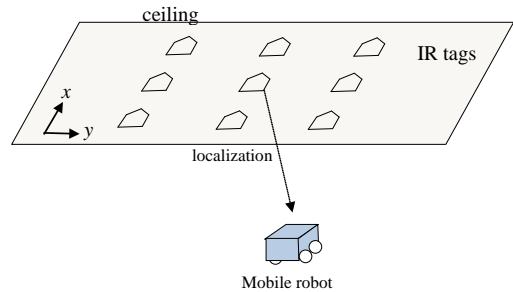


그림 2. 위치 인식을 위한 시스템 구성도.
Fig. 2. System for position estimation.

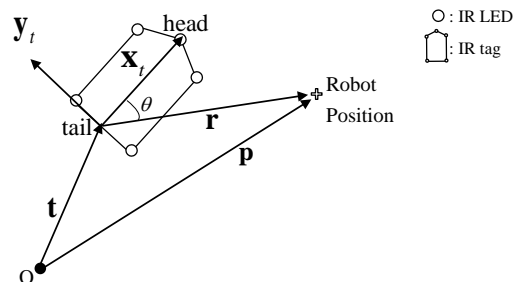


그림 3. 모바일 로봇의 위치 및 방향각 추정.
Fig. 3. Position and orientation of a mobile robot.

$$\begin{aligned} r_x &= |\mathbf{r}| \cos \theta = \frac{\mathbf{r} \cdot \mathbf{x}_i}{|\mathbf{x}_i|} \\ r_y &= |\mathbf{r}| \cos(\pi - \theta) = \frac{\mathbf{r} \cdot \mathbf{y}_i}{|\mathbf{y}_i|} \end{aligned} \quad (14)$$

여기서 \mathbf{r} 은 로봇 위치가 영상의 중심임을 이용하면 쉽게 구할 수 있다. 여기서 구한 단위는 이미지 내의 픽셀 단위이다. 아래 식을 통해서 픽셀 단위를 cm단위로 바꾼다.

$$\mathbf{p}_{cm} = \frac{1}{3} \mathbf{p} \quad (15)$$

여기서 위 식의 상수 $\frac{1}{3}$ 은 실 좌표계 범위 130 cm가 390 pixel에 해당함을 나타내고 \mathbf{p}_{cm} 는 pixel 단위로 표현되는 \mathbf{p} 를 cm단위로 나타낸 것이다.

이 연구에서 사용한 로봇의 움직임에는 직선 움직임과 회전 움직임이 존재한다. 그렇기 때문에 방향각을 추정해야 목표 지점을 향하는 회전 움직임을 가질 수 있다. 방향각을 추정하기 위해서 아래의 식을 이용한다.

$$\begin{aligned} \theta_{p_i} &= \arccos\left(\frac{\mathbf{r} \cdot \mathbf{x}_i}{|\mathbf{r}| |\mathbf{x}_i|}\right) \text{ if } r_y \geq 0 \\ \theta_{p_i} &= 2\pi - \arccos\left(\frac{\mathbf{r} \cdot \mathbf{x}_i}{|\mathbf{r}| |\mathbf{x}_i|}\right) \text{ if } r_y < 0 \end{aligned} \quad (16)$$

2. 추정된 위치의 오차 계산

그림 4는 실 좌표계를 픽셀 단위로 표현한 것이고 작은 네모는 한 픽셀을 의미한다. 실제 (p_x, p_y) 는 이미지에서 구한 로봇의 위치에 해당하는 픽셀 값이다. R_{pix} 은 (p_x, p_y) 픽셀의 영역을 의미하고 영역 R_{pix} 내에 존재하는 정확한 로봇의 위치를 알 수 없다. 이와 같은 불확실성을 줄이고자 아래 식으로 위치 추정 시 발생하는 오차를 측정한다.

$$e = \min_{r \in R_{pix}} d(\mathbf{p}_i, \mathbf{g}) \quad (17)$$

여기서 함수 d 는 두 벡터의 거리를 구하는 함수이다.

3. 로봇 제어시 오차 계산

식 (12)에 따르면 로봇 제어시 오차 계산이 요구된다. 거리 오차는

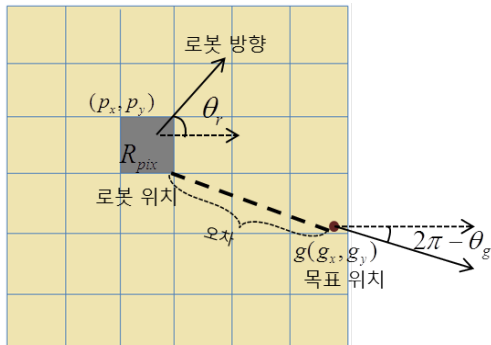


그림 4. 위치 추정 오차 계산.
Fig. 4. Position estimation error.

$$e_d = d(\mathbf{p}_i, C_{V_i}) \quad (18)$$

이고, \mathbf{p}_i 는 현재 위치 추정의 결과로써 식 (13)를 이용해서 구하고, C_{V_i} 는 각 보로노이 셀의 무게중심으로써 목적지를 의미하며 d 함수는 식 (17)를 이용하였다. 각 오차는

$$e_\theta = \theta_{C_{V_i}} - \theta_{p_i} \quad (19)$$

여기서 θ_{p_i} 은 식 (16)을 통해서 현재 로봇의 방향을 구하고, $\theta_{C_{V_i}}$ 은

$$\begin{aligned} &\arccos\left(\frac{\mathbf{p}_i \cdot C_{V_i}}{|\mathbf{p}_i| |C_{V_i}|}\right) \text{ if } C_{V_i,y} - p_{iy} \geq 0 \\ &2\pi - \arccos\left(\frac{\mathbf{p}_i \cdot C_{V_i}}{|\mathbf{p}_i| |C_{V_i}|}\right) \text{ if } C_{V_i,y} - p_{iy} < 0 \end{aligned} \quad (20)$$

으로 구한다. 여기서 $C_{V_i,y}$ 와 p_{iy} 는 각각 C_{V_i} 와 \mathbf{p}_i 의 y 좌표값을 의미한다.

V. 실험 결과

1. 실험 환경

그림 3(a)는 실제 실험에 사용된 로봇이며, 총 5대로 실험을 하였다. 로봇의 크기는 12.5 cm x 13.7 cm x 18 cm이다. 다음으로 로봇의 구성은 ARM 보드, AVR 보드, Cam과 모터로 구성된다. 그림 3(a)를 보면 로봇을 각 층으로 나누어서 구분하였고, 각 층마다 ARM 보드, AVR 보드, CAM, 그리고 배터리와 모터가 있다. 여기서 ARM 보드는 OMAP3 프로세서인 Gumstix Overo Fire, AVR 보드는 ATmega328 인 Arduino Duemilano, 그리고 CAM은 640x480 해상도의 Logitech C300을 사용하였다. 마지막으로 모터는 Pololu의 micro metal gear motor 제품을 사용했다. Cam 가시광선 필터를 사용해서 적외선 LED에서 나오는 적외선을 감지하기 위한 이미지를 얻는다. 이 이미지로 ARM보드에서 영상 처리를 해서 위치 추정을 한다. 그리고 서버, AVR보드와 통신하는데 서버와는 로봇의 위치와 목적지를 주고 받고 제어를 통해 결정된 모터의 PWM을 AVR보드로 전송한다. 그러면 AVR보드는 이 PWM을 받아서 모터를 제어한다.

태그는 그림 5(b)이며, 총 9개의 적외선 센서 태그를 천장에 부착하였는데 그림 5(c)에서 보여주고 있다. x 축 방향의 태그는 91 cm마다 부착되어 있으며, y 축 방향의 태그는 102

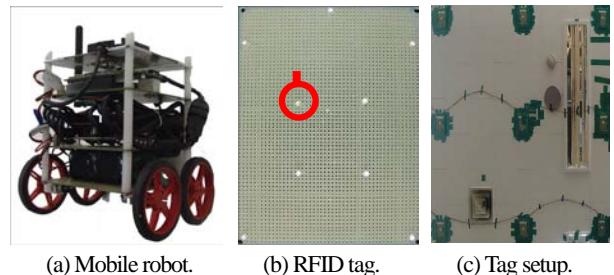


그림 5. 실험 환경.
Fig. 5. Experimental setup.

cm마다 부착되어 있다. 그리고 로봇의 활동 좌표인 실 좌표계의 범위는 130 cm x 130 cm으로 설정하였다.

2. 실험 결과

우리는 로봇의 크기에 의해 발생하는 충돌을 방지하고자 자유 보로노이 영역을 설정하였다. 그 결과를 그림 6에서 확인할 수 있다. 그림에서 보여주듯이 (a)의 마지막 그림은 충돌 부분을 확대해서 보여주고 있다. (b)는 자유 보로노이 영역을 설정해 주었을 때 충돌이 일어나지 않는 것을 확인할 수 있다.



(a) Collision without free Voronoi region.



(b) Collision protection with free Voronoi region.

그림 6. 자유 보로노이 영역 설정에 따른 충돌 실험.
Fig. 6. Experiment for collision protection.

표 1. 각 알파벳 인풋 반응 결과의 오차 (단위: cm).
Table 1. Positional errors of experimental results.

문자	E	H	B	K	C	O	N	D	V	F
1회	4.712	5.104	4.829	5.661	4.974	4.685	6.959	5.658	5.643	6.029
2회	8.269	4.144	4.201	6.363	5.331	6.362	5.127	4.508	5.557	5.426
3회	8.120	2.985	5.808	5.582	7.541	6.354	6.864	4.832	3.587	4.393
4회	6.684	4.779	5.587	5.376	5.868	7.852	3.969	6.636	5.390	2.736
5회	5.237	3.484	3.922	7.692	5.135	6.335	6.331	7.692	7.742	6.115
평균	6.604	4.099	4.869	6.135	5.770	6.317	5.850	5.865	5.584	4.940
표준편차	1.451	0.787	0.741	0.847	0.936	1.003	1.145	1.173	1.318	1.262
전체 평균									5.604	
전체 표준편차									1.307	

수 있다. 위에서 제시한 인터페이스를 이용하여 실제 군집 모바일 로봇에 적용하여 이를 분석하였다. 사용자가 알파벳을 그렸을 때, 군집 로봇의 움직임을 분석하였다.

보다 구체적으로 제시한 방법을 검증하기 위해서 10개의 영어 알파벳 문자를 썼을 때 이에 5대의 로봇들이 반응하여 배치되는 실험을 하였다. 각 알파벳 당 총 5번씩의 실험을 하였다.

표 1은 총 10개의 알파벳 문자에 대해서 각각 5번씩 반복하여 얻은 데이터이다. 표 1에서 횟수는 한 문자에 대해서 반복한 횟수를 의미하며 표에 나온 데이터 값은 각 5대의 로봇이 가지고 있는 오차 값을 평균한 값이다. 이를 각 문자에 대해서 평균과 표준편차를 구했으며, 최종적으로 총 10개의 문자에 대해 평균과 표준편차를 구했다. 총 평균 오차 값은 5.604 cm로 로봇의 크기의 절반보다 작은 값을 가지고 실 좌표계와 비교해도 4.31 %정도에 해당하기 때문에 오차의 크기가 크지 않음을 알 수 있다. 표준 편차 또한 큰 값이 아니므로 여기서 구한 오차의 안정성을 보장해준다. 이를 통해서 위치 추정 결과가 적합하다는 것을 알 수 있다.

그림 7은 각 문자에서 임의의 한 데이터를 선택해서 총 10개의 데이터에 대해서 시간에 따른 로봇의 움직임을 보여준다. 빨간 점은 CVT의 C_V 를 의미하고 최종 그림에서 사용자가 그린 문자를 보여주고 있다.

중간과정에서 로봇들이 반복적으로 구해지는 C_V 를 따라가고 마지막 그림에서 로봇이 최종 수렴한 결과를 보여준다. 각 알파벳 문자를 적절히 분할하여 로봇이 배치됨을 확인할 수 있다. 충분한 로봇 대수를 이용하면 보다 뚜렷한 문자 형상화가 가능할 것이다.

그림 8은 각 문자에 대해 5번씩 실험한 결과를 평균하였고 경과시간에 따른 오차 크기를 보여주는 그래프이다. 초기 위치에서는 적은 오차 값을 가지지만 경과 시간이 흐르면서 오차가 증가하는 경향을 보인다.

이는 로봇들이 적합한 자신의 위치를 찾는 과정으로 해석된다. 경과시간 60 %를 지나면서 점점 오차가 감소하는 것을 확인할 수 있다. 이는 로봇들이 올바른 위치에 도달하고 세부 위치 조정을 하는 단계이다.

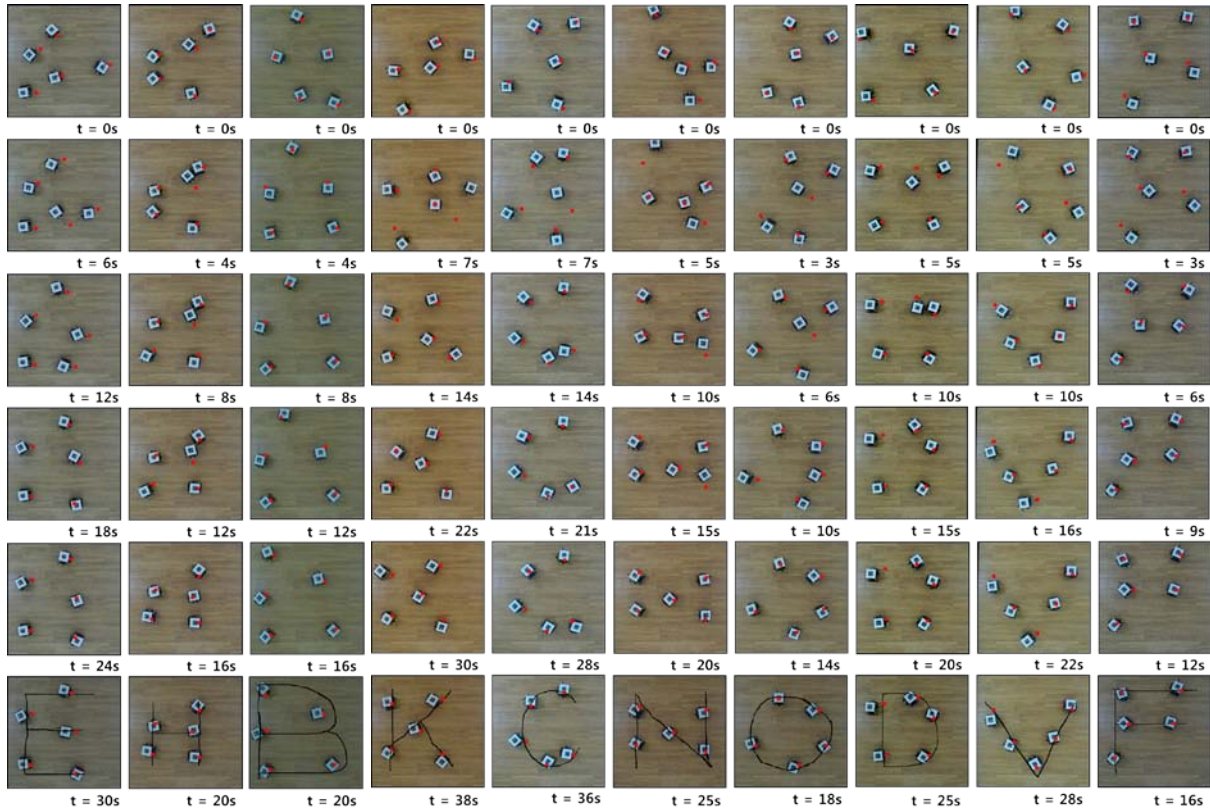


그림 7. 군집 로봇으로 표현한 알파벳.

Fig. 7. Experimental results.

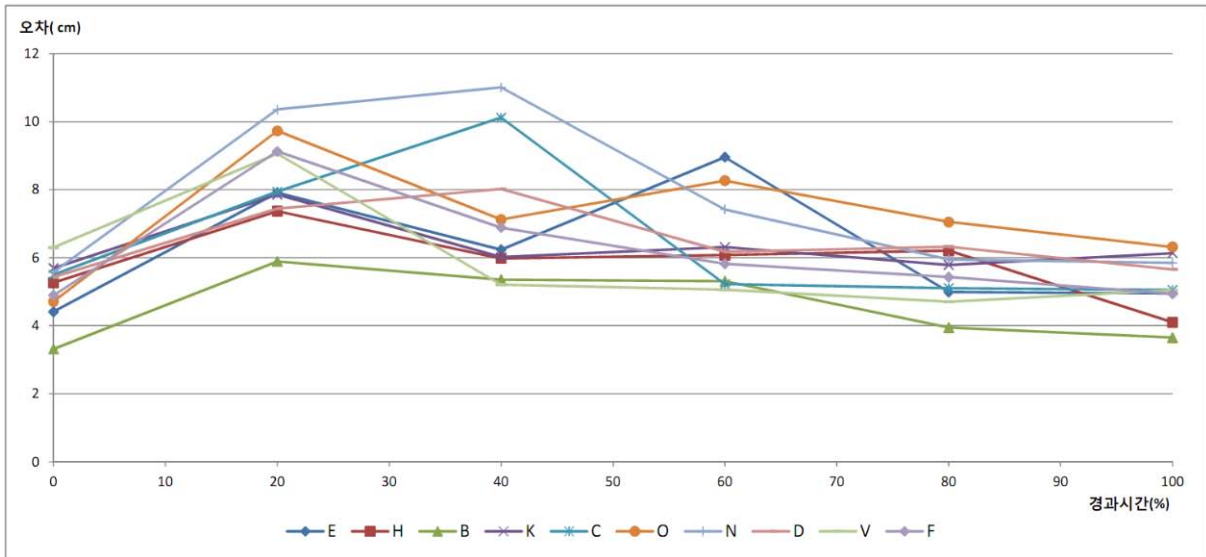


그림 8. 경과 시간에 따른 오차 분석.

Fig. 8. Error values over time.

VI. 결론

이 논문에서는 사용자의 입력에 반응해서 로봇들이 적절히 배치되도록 하는 인터페이스 방법을 제안하였다. 실제 사용자가 알파벳을 쓰면 이에 대응하여서 로봇들이 적절한 위치로 배열하는 것을 확인하였다. 분석을 통해서 위치 추정에 의한 오차가 큰 영향을 못 미치는 것을 확인했다. 따라서 사용자가 원하는 배치 형태를 입력하면 로봇이 이에 맞는 적절

한 배치가 이루어질 수 있었다.

이 논문에서는 중심 보로노이 알고리즘을 이용하였다. 이를 활용하고 밀도함수 분포를 사용자의 인풋 정보를 나타내는 표현으로 해석함으로써 사용자의 의도대로 여러대의 로봇들을 배치할 수 있는 방법론을 제안하였다. 각각의 로봇에 일일이 명령을 내리지 않고 쉽고 직관적인 방법으로 군집 로봇들의 행동을 구현하기에 유용할 것으로 기대한다.

참고문헌

[1] J. Sletzer, A. Das, R. Fierro, C. Taylor, V. Kumar, and J. Ostrowski, "Cooperative Localization and control for multi-robot manipulation," *International Conference on Intelligent Robots and Systems*, vol. 2, pp. 631-636, 2001.

[2] D. Vallejo, P. Remagnino, D. N. monekosso, L. Jimenez, and C. Gonzalez, "A multi-agent architecture for multi-robot surveillance," *Proceeding of International Conference on Computational Collective Intelligence, Semantic Web, Social Networks and Multiagent Systems*, vol. 5796, pp. 266-278, 2009.

[3] M. N. Rooker and A. Birk, "Multi-robot exploration under the constraints of wireless networking," *Control Engineering Practice*, vol. 15, no. 4, pp. 435-445, 2007.

[4] R. N. Lass, M. J. Grauer, E. A. Sultanik, and W. C. Regli, "A decentralized approach to the art gallery problem," *Proceeding of the 17th Fall Conference on Computational Geometry*, pp. 81-82, 2007.

[5] W. Li and C. G. Cassandras, "Distributed cooperative coverage control of sensor networks," *Proceeding of the IEEE Conference on Decision and Control*, pp. 2542-2547, 2005.

[6] G. J. Fan and S. Y. Jin, "Coverage problems in wireless sensor network: a survey," *Journal of Networks*, vol. 5, no. 9, pp. 1033-1040, 2010.

[7] G. L. Mariottini, F. Morbidi, D. Prattichizzo, N. Vander Valk, N. Michael, G. Pappas, and K. Daniilidis, "Vision-based Localization for leader-follower formation control," *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1431-1438, 2009.

[8] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, "A vision-based formation control framework," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 813-825, 2002.

[9] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926-939, 1998.

[10] J. Cortes, S. Martinez, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243-255, 2004.

[11] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, pp. 129-137, 1982.

[12] L. C. A. Pimenta, V. Kumar, R. C. Mesquita, and G. A. Pereira, "Sensing and coverage for a network of heterogeneous robots," *Proc. of IEEE Conference on Decision and Control*, pp. 3947-3952, 2008.

[13] A. Gusrialdi, S. Hirche, T. Hatanaka, and M. Fujita, "Voronoi based coverage control with anisotropic sensors," *American Control Conference*, pp. 736-741, 2008.

[14] A. Renzaglia and A. Martinelli, "Distributed coverage control for a multi-robot team in a non-convex environment," *Proceeding of IEEE IROS Workshop on Planning, Perception and Navigation for Intelligent Vehicles*, pp. 76-81, 2009.

[15] Q. Du, V. Faber, and M. Gunzburger, "Centroidal voronoi

tesselleations: applications and algorithms," *SIAM Review*, vol. 41, no. 4, pp. 637-676, 1999.

[16] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Transactions on Information theory*, vol. 44, no. 6, pp. 2325-2383, 1998.

[17] <http://opencv.willowgarage.com/wiki/cvBlobsLib/>

[18] J.-H. Kim, et al., *Robot Soccer Engineering (in Korean)*, KAIST Press, 2002.

[19] A. Okabe, B. Boots, K. Sugihara, and S. N. Chju, *Spatial Tesselations: Concepts and Applications of Voronoi Diagrams*, Wiley Series in Probability and Statistics, John Wiley & Sons, 2000.

[20] <http://simbad.sourceforge.net/>



조영래

2009년 경북대학교 물리학과(이학사). 2011년 한국과학기술원 로봇학제전공(공학석사). 관심분야는 군집로봇시스템.



이길호

2010년 아주대학교 정보 및 컴퓨터 공학부(공학사). 2012년 한국과학기술원 전산학과(공학석사). 2012년~현재 한국과학기술원 전산학과 박사과정. 관심분야는 컴퓨터 시스템, 시스템 소프트웨어, 운영체제, 모바일 플랫폼, 실시간 시스템.



조성호

1999년 서울대학교 기계공학부(공학사). 2001년 MIT 기계공학과(공학석사). 2006년 MIT 전자전산학과(공학박사). 2008년~현재 한국과학기술원 전산학과 조교수. 관심분야는 로봇공학, 뇌-기계 인터페이스, 지능시스템.



신인식

1994년 고려대학교 전산학과(이학사). 1998년 스탠포드대학교 전산학과(공학석사). 2006년 펜실바니아대학교 전산학과(공학박사). 2008년~현재 한국과학기술원 전산학과 조교수. 관심분야는 실시간 내장형 시스템.