



저작자표시-비영리-동일조건변경허락 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이차적 저작물을 작성할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



동일조건변경허락. 귀하가 이 저작물을 개작, 변형 또는 가공했을 경우에는, 이 저작물과 동일한 이용허락조건하에서만 배포할 수 있습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사학위논문

컨벡스 최적화 기법을 이용한
다중 객체 추적

**Matching and Tracking of Multiple Objects through
Convex Optimization Formulation**

2013 년 8 월

서울대학교 대학원

전기 정보 공학부

김 장 욱

컨벡스 최적화 기법을 이용한 다중 객체 추적

Matching and Tracking of Multiple Objects through Convex Optimization Formulation

지도교수 최 진 영

이 논문을 공학석사 학위논문으로 제출함

2013 년 8 월

서울대학교 대학원

전기정보공학부

김 장 욱

김장욱의 공학석사 학위논문을 인준함

2013 년 8 월

위 원 장 : 김 수 환 (印)

부위원장 : 최 진 영 (印)

위 원 : 오 성 회 (印)

초 록

다중 객체 추적은 이동하는 객체를 탐지하고 같은 객체로 판단되는 객체들끼리 연결하는 과정으로 수행된다. 본 학위 논문에서는 이동하는 객체 추적을 위해 컨벡스 (convex) 문제로 공식화하여 해결하는 방법을 제안하고자 한다. 객체들은 각 프레임마다 탐지되고, 데이터 연관 (data association) 방식으로 연속된 프레임들 간의 객체들을 연결함으로써 경로를 계산한다. 프레임 간의 객체들을 연결함으로써 발생하는 비용(cost)을 수치화하는데, 이 때 객체들의 위치와 색상을 고려하여 비용을 계산한다. 이 비용 값들을 원소로 하는 비용 행렬을 계산하고, 이 비용 행렬을 이용하여 객체들을 연결할 시 가장 비용이 적게 발생하도록 연결하는 문제를 컨벡스 문제로 공식화한다. 실험 결과, 기존에 제안된 기법들보다 성능이 개선되었음을 확인하였고, 탐지 과정에서 발생한 오탐지에 의해 경로가 잘못 계산되는 현상도 줄일 수 있었다.

주요어 : 다중 객체 추적, 컨벡스 최적화

학 번 : 2011-20813

목 차

제 1 장 서 론	3
제 2 장 제안한 방법	6
제 1 절 객체 추적 방법	6
제 2 절 최적화 문제 공식화	11
제 3 절 최적화 문제 해결과정	15
제 3 장 실험 결과	17
제 1 절 PETS2009	17
제 2 절 공 추적	20
제 3 절 TownCentre	23
제 4 장 결 론	25
참고 문헌	26

표 목차

[표 1] 성능 평가 지표	18
[표 2] PETS2009-S2-L1 에서의 기존의 기법들과 제안된 기법의 추적 성능 비교	18
[표 3] 탁구공 동영상에서 기존 기법과의 성능 비교 ...	22
[표 4] TownCentre 동영상에서 d_p 의 값에 따른 추적의 성능 비교	24

그림 목차

[그림 1] 연속된 두 프레임	6
[그림 2] 위치 정보를 계산하는 방법	8
[그림 3] PETS2009-S2-L1 동영상에서의 추적 결과	19
[그림 4] 탁구공 동영상에서의 추적 결과	21
[그림 5] d_p 프레임 동안의 객체의 색상으로 각 객체의 색상 히스토그램을 계산하였을 경우 추적 결과	24

제 1 장 서 론

동영상에서 움직이는 객체를 추적하는 방법 중 한 가지는 데이터 연관 (data association) 방식으로, 독립된 두 단계의 작업으로 구성된다. 첫 번째 단계는 동영상의 각 프레임마다 객체들을 탐지하는 과정이다. 그리고 두 번째 단계는 프레임 간에 탐지된 객체들 중 같은 객체로 판단되는 객체들끼리 연결하여 객체들의 이동 경로를 추적하는 과정이다.

본 논문에서는 기존의 객체 탐지 방법을 사용하여 물체를 탐지하고, 탐지된 다중 객체들을 효율적으로 추적하는 방법을 제시하고자 한다. 추적 과정에서 이 탐지된 객체들을 프레임 간에 적절하게 연결시키는 방법, 즉 탐지된 객체들 중 연속된 프레임에서 같은 객체로 판단되는 객체들을 찾는 방법을 개발하는 것이 목표이다. 그리고 탐지 결과 또한 오탐지 오류가 존재하므로, 추적 과정에서 이 오탐지에 의한 영향을 줄임으로써 추적 정확도를 개선시키는 방법도 고려한다.

다중객체 추적에 관련된 기존 연구로서, [1]에서는 객체들을 탐지하는 과정에서 probabilistic occupancy map[2] 기법을 사용하였는데, 이 기법은 여러 대의 카메라를 사용하여 일부 객체가 다른 객체에 가려지는 현상을 방지하고 평면 지도 위에 탐지된 객체들의 위치를 나타낸 후 추적하는 기법이다. 그리고 프레임 간의 탐지된 객체들을 연결하는 과정을 컨벡스 최적화 문제로 공식화하고 k-shortest paths (KSP) 방식을 사용하여 문제를 해결하였다. [3]에서는 입자 군집 최적화 방식을 사용하여 군중 속의

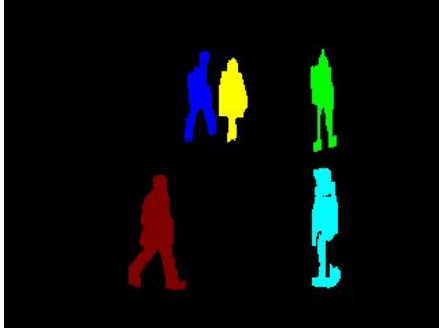
다수의 사람들을 추적하였다. 한 명의 사람을 하나의 입자로 보고, 각 입자들의 위치 정보들을 고려하였다. 그리고 사회적 행동 (사람들 사이에서의 움직임 정보) 등과 같은 고수준의 정보를 사용하였다. 이 정보를 사용함으로써 다수의 사람들이 함께 이동할 시 각각의 사람들의 이동 방향을 예측할 수 있다. 또한 새로운 사람이 화면 안에 등장할 때 분리된 군집을 초기화하였다. [4]에서는 이미 객체들이 충분히 탐지되었다고 가정하고 그 후 연속인 에너지 함수를 최소화하는 알고리즘으로 탐지된 객체들을 추적하였다. 여기서 에너지 함수란 합당한 설정에는 보상을, 합당하지 않은 설정에는 불이익을 부여하는 함수로, 보상이 클수록 에너지의 값이 작아진다. 에너지 함수를 구성하는 항은 객체가 적합한 위치에 존재할 가능성, 이동속도의 방향, 둘 이상의 객체가 동시에 같은 공간에 위치할 때 부여하는 불이익, 객체가 추적 지역에서 갑자기 등장하거나 사라질 때 부여하는 불이익 등이다.

본 논문에서는 동영상의 모든 프레임에서 객체들을 탐지하여 얻은 결과를 데이터 연관 방식으로 연속된 여러 프레임 간의 객체들을 연결함으로써 객체들을 추적하였다. 연속된 프레임 사이의 두 객체를 같은 객체로 판단하는 기준은 임의의 두 객체를 연결할 시 발생하는 비용(cost)으로 정하였다. 이 비용을 계산하기 위해 두 객체의 위치 정보와 색상 정보를 사용하였다. 그리고 이 비용들을 계산하여 비용 행렬(cost matrix)을 구하였고, 이 비용 행렬을 이용하여 가장 적합한 연결을 찾기 위해, 객체들을 연결할 시 발생하는 비용을 최소화하는

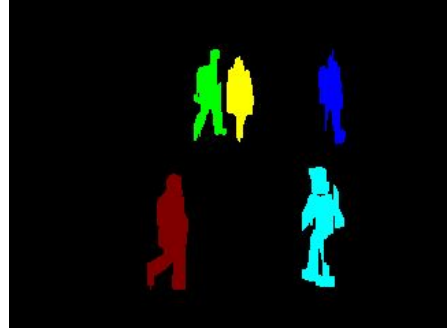
컨벡스 (convex) 최적화 문제로 공식화하고, 이 문제의 해를 구함으로서 다중객체 추적문제를 해결하였다.

실제 동영상에서 알고리즘을 적용하고 성능을 분석하여 기존에 제안된 기법들과 성능을 비교하였다. 그 결과 기존에 제안된 기법들에 비해 제안된 기법이 더 성능이 높음을 확인하였으며, 가려짐 현상의 영향도 줄일 수 있음을 확인하였다.

제 2 장 제안된 방법



(a) Frame t



(b) Frame $t+1$

그림 1. 연속된 두 프레임

제 1 절 객체 추적 방법

동영상의 모든 프레임에서 모든 객체들이 충분히 잘 탐지되었다고 가정한다. 탐지 과정이 끝난 후, 이전 프레임에서 탐지된 객체가 현재 프레임에서 어디로 이동하였는지를 추정하기 위해, 각각의 프레임에서 탐지된 두 객체를 서로 연결할 시 발생하는 비용을 수치화하여 그 값을 이용한다. 비용이 적은 객체일수록 그 두 객체가 같은 객체일 가능성이 크다.

그림 1 은 연속된 두 프레임을 나타낸다. 이 영상들에서 탐지된 객체들은 서로 매우 유사하게 생겼으나, 각 객체들 사이의 거리로 보아 각 객체들은 가장 가까운 객체로 이동하였음을 추정할 수 있다. 각각의 객체들이 화면 안에서 서로 충분히 멀리 떨어져 있으면 문제가 되지 않으나, 객체들이 서로 충돌하거나, 일부 객체가 화면 밖으로 나가거나,

또는 새로운 객체가 화면 안으로 들어오는 경우 등이 발생하면 객체 쌍을 찾기가 어려워진다. 따라서 이 문제를 최적화 문제로 공식화하여 해결해보는 방법을 생각할 수 있다.

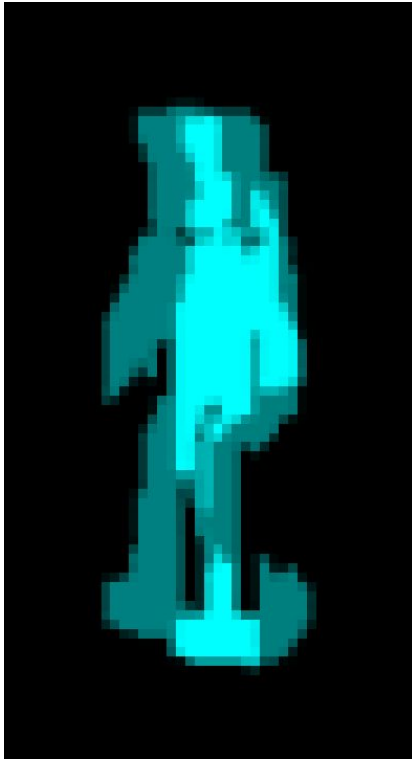
이와 같이 두 객체 사이의 거리를 비용으로 고려할 수도 있으며, 객체의 색상을 고려할 수도 있다. 서로 유사한 색을 가진 객체일수록 그 두 객체가 서로 같은 객체일 가능성이 높은 것은 자명하다. 그러나 약간의 변화에도 색상은 쉽게 변하기 때문에, 화면에 나타날 수 있는 색상들을 비슷한 색상들끼리 몇 가지로 분류하여 같은 분류에 속하는 색상들은 서로 같은 색상이라고 가정한 후에 색상들을 비교하는 것이 좋다. 비용을 c 라 하면, 다음과 같이 식을 세울 수 있다.

$$c(W_t^i, W_{t+1}^j) = \rho_1 c^{(p)}(W_t^i, W_{t+1}^j) + \rho_2 c^{(c)}(W_t^i, W_{t+1}^j) \quad (1)$$

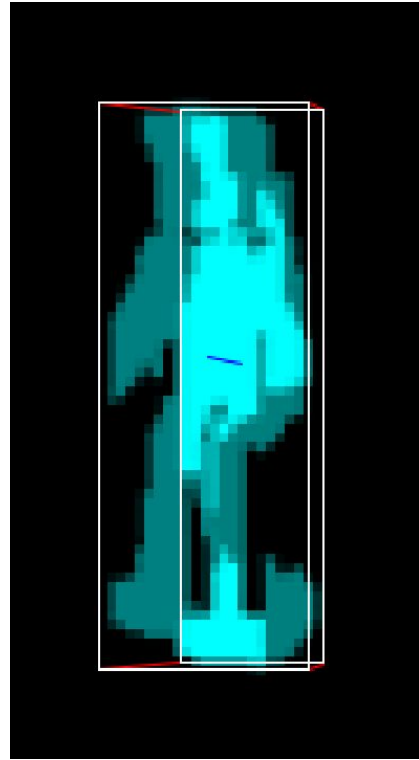
여기서 W_t^i 는 프레임 t 에서 탐지된 객체 i 를 의미하고, $c^{(p)}$ 와 $c^{(c)}$ 는 각각 위치 정보와 색상 정보를 사용하여 계산된 비용, ρ_1 , ρ_2 는 가중치(weight)를 의미한다.

2.1.1 위치 정보로 비용 계산

두 객체를 연결할 시 발생하는 비용을 계산하는데 위치 요소와 색상 요소가 사용된다. 위치 요소는 객체들의 2차원 좌표를 사용하는데, 두 객체 사이의 유클리디안 거리(Euclidean distance)를 2차원 좌표상에서 계산한다. 단, 객체의 중심을 객체의 좌표로 사용하기보다는 객체가 완전



(a)



(b)

그림 2. 위치 정보를 계산하는 방법. (a) 연속된 프레임에서 탐지된 두 객체를 겹친 형태. (b) 두 객체 사이의 거리를 나타낸 그림. 붉은 색 선은 두 객체의 각각의 네 구석의 거리를, 푸른 색 선은 두 객체의 각 중심 사이의 거리를 나타낸다.

이 포함되는 최소한의 크기의 직사각형을 각각 그렸을 때, 그 각각의 직사각형의 각 꼭지점들 사이의 거리를 유클리디안 거리로 구하여 전부 더한 값을 위치에 대한 비용으로 계산한다. 이 방법으로 계산하면 객체의 위치뿐 아니라 객체의 크기도 고려될 수 있다. 그림 2 는 탐지된 두 객체 사이의 거리를 구할 시, 각 객체의 각각의 네 구석 사이를 구한 결과(붉은 색 선)와 단순히 각 객체의 중심 사이를 구한 결과(푸른 색

선)를 나타낸다. 붉은 색 선이 푸른 색 선에 비해 길이가 매우 긴 것을 알 수 있는데, 이는 두 객체의 크기 차이도 붉은 색 선의 길이에 영향을 미치기 때문이다. 따라서 단순히 각 객체의 중심 좌표만을 고려하기보다는 각 객체의 네 구석의 좌표를 고려하는 것이 객체들의 위치 정보를 더 효과적으로 이용할 수 있음을 확인할 수 있다.

2.1.2 색상 정보로 비용 계산

탐지된 객체들의 색상 정보는 위치 정보보다 더 중요한데, 이는 시각적으로 객체들의 구분이 더 쉽기 때문이다. 그러나 객체의 이동이나 주변 환경의 변화 등의 약간의 변화에도 객체들을 이루는 색상은 매우 쉽게 변하므로, 색상 정보를 그대로 이용하기보다는 다음과 같이 약간의 계산을 수행한 후에 이용하는 것이 좋다. 이 계산 방법은 [7]에서 STARS 측정법이라는 명칭으로 제안되었다.

객체들의 색상은 hue, saturation, value (HSV) 값으로 표현한다([8]). 그러면 색상의 hue 값, saturation 값, value 값에 따라 분류를 하는데, 분류하는 가짓수를 각각 N_h 가지, N_s 가지, N_v 가지라 하면 총 분류의 가짓수는 $N = N_h N_s N_v$ 가짓가 될 것이다. 그러므로 탐지된 하나의 객체를 구성하는 색상들을 전부 이 방법으로 분류하면 하나의 객체의 색상 히스토그램을 구할 수 있다. 그리고 어떤 색이 어느 위치에 있는지도 중요할 것이다. 일반적으로, 탐지된 객체의 중심 쪽에 있는 색상이 중요할 가능성이 크므로, 객체의 중심에 가까운 색상일수록 가중치를 높게 부여하는 것이 좋다. 따라서 다음과 같이 가중치를 계산한다.

$$\omega(d) = \begin{cases} 1-d^2, & d < 1 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

여기서 d 는 한 픽셀에서 객체의 중심까지의 거리인데, 객체의 한 구석에서 객체의 중심까지의 거리를 1 로 두었을 때의 상대적인 거리를 나타낸다. 그러면 한 객체의 색상 히스토그램은 다음과 같다.

$$p_t^{(n)} = K \sum_{u \in W_t} \omega(d(u)) \delta(b_t(u) - n), \quad n = 1, \dots, N. \quad (3)$$

여기서 $b_t(u)$ 는 픽셀 u 의 HSV 값이고, $K = \frac{1}{\sum_u \omega(d(u))}$ 는 $\sum_{n=1}^N p_t^{(n)} = 1$ 을 만족시키기 위한 정규화 (normalization) 계수이다. 그리고 이렇게 계산된 색상 히스토그램으로 두 객체의 색상 유사도(color similarity)는 다음과 같이 구한다.

$$\text{Color similarity}(W_t^i, W_{t+1}^j) = \sum_{n=1}^N \sqrt{p_t^{(n)} p_{t+1}^{(n)}} \quad (4)$$

여기서 $p_t^{(n)}$ 와 $p_{t+1}^{(n)}$, $n = 1, \dots, N$ 는 각각 W_t^i 와 W_{t+1}^j 의 색상 히스토그램이다. 이 값이 1 에 가까우면 히스토그램이 서로 유사하다는 의미이다. 그러나 비용 값은 작을수록 서로 같은 객체일 가능성이 높다는 의미이므로, 1 에서 색상 유사도 값을 뺀 값을 비용으로 사용해야 한다.

그리고 이와 같이 계산하면 객체들이 서로 너무 가까이 존재할 경우, 가려짐 현상에 의해 한 객체가 다른 객체의 영역에 침범함으로써 원래 서로 다른 객체였어도 색상 정보가 거의 비슷하게 얻어질 가능성이 크다. 이렇게 되면 색상 정보를 효과적으로 이용하기 어렵기 때문에 색상 정보를 이전의 단 한 프레임에서만 얻기보다는 최근의 일련의 프레임에서

얻은 색상 정보를 모두 이용하는 것이 더 좋을 수 있다. 가려짐 현상이 발생하기 전의 색상 정보도 사용하게 되기 때문이다.

제 2 절 최적화 문제 공식화

이전 프레임의 라벨의 개수를 I 라 하고, 현재 프레임의 라벨의 개수를 J 로 정의하고, 두 객체를 연결할 시 발생하는 비용을 원소로 갖는 비용 행렬을 C 로 정의한다. C 는 $(I+J) \times (I+J)$ 행렬이며, 행과 열은 각각 이전 프레임과 현재 프레임의 라벨의 번호를 나타낸다. 즉,

$$C = [c_{i,j}], \quad i, j = 1, \dots, I+J \quad (5)$$

이고, 여기서 원소 $c_{i,j}$ 는 비용이다. 이 행렬에서 처음 I 개의 행은 이전 프레임의 객체들이며, 나머지 J 개의 행은 최적화 문제의 제약 조건을 만족시키기 위해 이전 프레임에 추가된 가상의 객체들이다. 마찬가지로, 처음 J 개의 열은 현재 프레임의 객체들이며, 나머지 I 개의 행은 현재 프레임에 추가된 가상의 객체들이다.

각 프레임에 가상의 객체들을 추가하여 비용 행렬을 만든 이유는 일부 객체는 어떤 객체와도 연결되기 적합하지 않는 경우에 대비하기 위함이다. 일반적으로 연속된 프레임들 사이의 모든 객체들은 다른 객체와 연결이 가능하지만, 객체가 화면 밖으로 사라졌을 경우나 새로운 객체가 화면 안으로 나타났을 경우에 함께 연결될 객체가 없다. 그러나 앞으로 공식화할 최적화 문제에서는 ‘연속된 프레임에서 모든 객체들은 다른 프레임 안의 단 하나의 객체와 연결되어야 한다’는 제약 조건이 있으므로, 이를 만족시키기 위해 위와 같은 경우에 속하는 객체들은 추가된 가상의

객체에 연결시킨다. 이전 프레임의 화면에 탐지되었던 I 개의 객체들이 모두 사라지면서 동시에 현재 프레임의 화면에 J 개의 객체들이 전부 나타나는 경우까지 고려하기 위해, 이전 프레임에는 J 개의 가상의 객체를 추가하고 현재 프레임에는 I 개의 가상의 객체를 추가하는 것이다.

비용 행렬 안의 비용 값들이 전부 계산되면, 이제 최적의 비용 값들을 선택해야 한다. 각각의 비용 값들의 선택 여부를 나타내기 위해, 또 다른 $(I+J) \times (I+J)$ 행렬 $B=[b_{i,j}]$ 를 사용한다. 비용 행렬 C 의 한 비용 값을 선택할 경우에 1 을, 선택하지 않을 경우에 0 을 행렬 B 에 나타낸다. 예를 들어, 이전 프레임의 i 번째 객체를 현재 프레임의 j 번째 객체와 연결시킬 경우, 비용 값 $c_{i,j}$ 를 선택한다는 의미이므로 $b_{i,j}=1$ 이다. 반면 이 두 객체를 연결시키지 않을 경우, $b_{i,j}=0$ 이다. 따라서 연속된 프레임들 간에 객체들을 연결시킬 시 발생하는 총 비용은 $\sum_{i,j} \{C \circ B\}_{i,j}$ 이다. 여기서 연산자 \circ 은 두 행렬의 Hadamard 곱(같은 위치에 있는 원소끼리의 곱)을 나타낸다.

행렬 B 의 원소의 값들을 결정할 시에 제약 조건이 존재한다. 이전 프레임 안의 가상의 객체를 포함한 모든 객체들은 현재 프레임 안의 가상의 객체를 포함한 모든 객체들과 반드시 하나씩만 연결되어야 하므로, 행렬 B 에서 각 행이나 각 열에서 단 하나의 원소만 선택되어야 한다. 여기서 제약 조건을 식으로 나타내면 다음과 같다.

$$\begin{aligned}
\sum_{j=1}^{I+J} b_{i,j} &= 1, & i &= 1, \dots, I+J, \\
\sum_{i=1}^{I+J} b_{i,j} &= 1, & j &= 1, \dots, I+J, \\
b_{i,j} &\in \{0, 1\}, & i, j &= 1, \dots, I+J.
\end{aligned} \tag{6}$$

즉, 행렬 B 의 원소는 0 또는 1 이어야 하므로, 행렬 B 의 각 행이나 각 열에 속한 원소들의 합이 1 이면 각 행이나 각 열의 속한 원소들 중 단 하나의 원소만 값이 1 이고 나머지 원소들은 값이 0 이 되어야 한다. 물론 객체들 중 다른 프레임의 어떤 객체들과도 연결되기가 적합하지 않은 경우, 그 객체는 가상의 객체와 연결되어 위 제약 조건들을 만족한다.

그리고 약간 드문 경우로, 화면 밖으로 사라진 객체들의 개수와 새로 화면에 등장한 객체들의 개수가 같아서 결과적으로 화면 안에 탐지된 객체들의 수가 같을 수도 있다. 이러한 경우에도 한 프레임의 실제 객체들이 이전 또는 다음 프레임의 실제 객체들과 모두 연결되어서는 안 된다. 이를 방지하기 위해 허용 가능한 최대의 비용을 정한다. 이는 실제 객체가 다른 실제 객체와 연결될 때 발생하는 비용이 허용 가능한 최대의 비용보다 클 경우, 실제 객체와 연결되기보다는 가상의 객체와 연결되도록 하기 위함이다. 일반적으로, 사라지기 직전의 객체나 새로 등장한 직후의 객체는 다른 객체와 연결될 경우, 상당한 크기의 비용이 발생할 것으로 가정할 수 있기 때문이다. 이 허용 가능한 최대의 비용을 C_{\max} 로 나타내면, 다음과 같은 제약 조건을 추가할 수 있다.

$$0 \leq c_{i,j} b_{i,j} \leq C_{\max}, \quad i, j = 1, \dots, I+J. \tag{7}$$

이 식에서, 객체 i 와 객체 j 를 연결할 시 발생하는 비용 $c_{i,j}$ 이 C_{\max} 보다 크면, $b_{i,j}$ 는 반드시 0이어야 한다. 즉, 이 두 객체는 서로 연결되어서는 안 된다. 한편, 한 가상의 객체를 다른 객체(가상의 객체 포함)와 연결할 시 발생하는 비용은 전부 C_{\max} 로 정한다. 따라서 어떤 한 객체가 다른 프레임의 어떤 실제 객체와 연결되어도 그 비용이 C_{\max} 보다 크면 그나마 비용이 작은 가상의 객체와 연결되게 된다. 또한 이 연결 결과로 사라진 객체와 새로 나타난 객체를 인식할 수 있다.

따라서 최종적으로 공식화된 최적화 문제는 다음과 같다.

$$\begin{aligned}
& \text{Minimize } \sum_{i,j} \{C \circ B\}_{i,j} = \sum_{i,j} c_{i,j} b_{i,j} \\
& \text{subject to } \sum_{j=1}^{I+J} b_{i,j} = 1, \quad i = 1, \dots, I+J, \\
& \quad \sum_{i=1}^{I+J} b_{i,j} = 1, \quad j = 1, \dots, I+J, \\
& \quad 0 \leq c_{i,j} b_{i,j} \leq C_{\max} \quad i, j = 1, \dots, I+J, \\
& \quad b_{i,j} \in \{0, 1\}, \quad i, j = 1, \dots, I+J.
\end{aligned} \tag{8}$$

여기서 변수로 작용하는 $b_{i,j}$ 는 0 또는 1이어야만 하므로, 이 최적화 문제는 NP-hard 문제이다. 그러나 목적 함수와 나머지 3개의 제약 조건은 선형이거나 컨벡스 형태이다. 따라서 마지막 제약 조건을 다음 식과 같이 느슨하게 조정하면, 이 제약 조건도 컨벡스 형태가 되어 최적화 문제가 컨벡스 형태가 된다.

$$0 \leq b_{i,j} \leq 1, \quad i, j = 1, \dots, I+J. \tag{9}$$

위와 같이 제약 조건이 변경되더라도, C 의 같은 행 또는 같은 열의 비용 값들이 전부 서로 다르면 $b_{i,j}$ 가 0 또는 1이 되는 경우 중에서만 목적

함수가 최적의 값이 된다. 그리고 비용 계산 시 여러 요소를 고려하여 충분히 복잡한 계산을 수행하므로, 한 객체가 다른 프레임의 서로 다른 객체들과 연결되는데 각각 발생하는 비용이 같을 가능성은 없다. 단, 가상의 객체에 연결될 때의 비용은 C_{\max} 로 전부 같지만 가상의 객체는 실제 존재하는 객체가 아니므로 이 경우는 고려할 필요가 없다.

제 3 절 최적화 문제 해결 과정

주어진 최적화 문제를 해결하기 위해 라그랑지안 승수법(Lagrangian multiplier method)을 사용한다.

$$\begin{aligned} L(B, \mu_I, \mu_J, \Lambda^{(1)}, \Lambda^{(2)}) = & \sum_{i,j} c_{i,j} b_{i,j} + \sum_{i=1}^{I+J} \mu_{J,i} \left(\sum_{j=1}^{I+J} b_{i,j} - 1 \right) + \sum_{j=1}^{I+J} \mu_{I,j} \left(\sum_{i=1}^{I+J} b_{i,j} - 1 \right) \\ & + \sum_{i,j} \lambda_{i,j}^{(1)} (c_{i,j} b_{i,j} - C_{\max}) + \sum_{i,j} \lambda_{i,j}^{(2)} c_{i,j} b_{i,j}. \end{aligned} \quad (10)$$

여기서 $\mu_I = (\mu_{I,1}, \dots, \mu_{I,I+J})^T$, $\mu_J = (\mu_{J,1}, \dots, \mu_{J,I+J})^T$ 는 각각 첫 번째와 두 번째 제약 조건에 대한 라그랑지안 승수 벡터이고, $\Lambda^{(l)} = [\lambda_{i,j}^{(l)}]$, $l=1,2$, $i,j=1, \dots, I+J$ 는 세 번째 제약 조건에 대한 라그랑지안 승수 행렬이다. 위 식의 B 에 대한 그래디언트를 계산하면 다음과 같다.

$$\begin{aligned} \nabla_B L(B, \mu_I, \mu_J, \Lambda^{(1)}, \Lambda^{(2)}) = & C + \begin{bmatrix} \mu_{J,1} \mathbf{1}_J^T \\ \vdots \\ \mu_{J,I+J} \mathbf{1}_J^T \end{bmatrix} + \begin{bmatrix} \mu_{J,1} \mathbf{1}_J & \cdots & \mu_{J,I+J} \mathbf{1}_J \end{bmatrix} \\ & + (\Lambda^{(1)} + \Lambda^{(2)}) \circ C \end{aligned} \quad (11)$$

여기서 $\mathbf{1}_N$ 은 원소가 전부 1인 $N \times 1$ 크기의 벡터를 의미한다.

$\nabla_B L(B, \mu_I, \mu_J, \Lambda^{(1)}, \Lambda^{(2)}) = 0$ 을 만족하는 B 를 B^* 라 하면 다음이 성립한다.

$$\begin{aligned} \nabla_B L(B^*, \mu_I, \mu_J, \Lambda^{(1)}, \Lambda^{(2)}) &= C + (\Lambda^{(1)} + \Lambda^{(2)}) \circ C \\ &+ \begin{bmatrix} \mu_{I,1} + \mu_{J,1} & \cdots & \mu_{I,I} \\ \vdots & \ddots & \vdots \\ \mu_{I,1} + \mu_{J,I+J} & \cdots & \mu_{I,I+J} + \mu_{J,I+J} \end{bmatrix} = O. \end{aligned} \quad (12)$$

따라서 dual 함수는 다음과 같이 구할 수 있다.

$$\begin{aligned} g(\mu_I, \mu_J, \Lambda^{(1)}, \Lambda^{(2)}) &= \inf_B L(B, \mu_I, \mu_J, \Lambda^{(1)}, \Lambda^{(2)}) \\ &= \begin{cases} -\sum_{i=1}^{I+J} \mu_{J,i} - \sum_{j=1}^{I+J} \mu_{I,j} - \sum_{i,j} \lambda_{i,j}^{(1)} C_{\max} & \text{if } \mu_{I,j} + \mu_{J,i} = (\lambda_{i,j}^{(1)} + \lambda_{i,j}^{(2)} + 1)c_{i,j}, \forall i, j, \\ -\infty & \text{otherwise.} \end{cases} \end{aligned} \quad (13)$$

그리고 dual 문제는 다음과 같다.

$$\begin{aligned} &\text{Maximize } g(\mu_I, \mu_J, \Lambda^{(1)}, \Lambda^{(2)}) \\ &\text{subject to } \lambda_{i,j}^{(1)} \geq 0, \quad \lambda_{i,j}^{(2)} \geq 0, \quad \forall i, j. \end{aligned} \quad (14)$$

이 때, $\mu_{I,j} + \mu_{J,i} = (\lambda_{i,j}^{(1)} + \lambda_{i,j}^{(2)} + 1)c_{i,j}$ 인 경우에만 g 가 bounded 값을 갖기 때문에, 이 경우만을 고려한다. 그리고 C_{\max} 의 값은 양수이므로, g 의 값을 최대화하기 위해서는 모든 i, j 에 대하여 $\lambda_{i,j}^{(1)} = 0$ 이어야 한다. g 의 값을 최대화하는 최적의 $\Lambda^{(1)}$ 과 $\Lambda^{(2)}$ 를 각각 $\Lambda^{(1)*}$ 와 $\Lambda^{(2)*}$ 라 하면, 다음을 얻는다.

$$g(\mu_I, \mu_J, \Lambda^{(1)*}, \Lambda^{(2)*}) = -\sum_{i=1}^{I+J} \mu_{J,i} - \sum_{j=1}^{I+J} \mu_{I,j}. \quad (15)$$

따라서, 총 비용의 최솟값은 $-\sum_{i=1}^{I+J} \mu_{J,i} - \sum_{j=1}^{I+J} \mu_{I,j}$ 가 될 것이다.

제 3 장 실험 결과

지금까지 설명한 객체 추적 기법의 성능을 알아보기 위해 실제 동영상에 적용하고, 그 결과를 확인하였다.

실험을 위해 사용된 동영상 데이터셋(dataset)은 PETS2009([5])와 [1]에서 사용된 탁구공 동영상, 그리고 TownCentre([10])이다. 각 기법의 성능은 multiple object tracking accuracy(MOTA)([6])와 MT, ML, Frag, IDS([9])를 사용하였고, MOTA 는 전체 프레임에 존재하는 객체들 중 올바르게 탐지된 객체 수의 비율로, 다음과 같이 계산된다.

$$\text{MOTA} = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t}. \quad (16)$$

여기서 g_t 는 시간 t 에 해당하는 프레임 안에 실제로 존재하는 객체 수, m_t 는 시간 t 에서의 감지 실패 수, fp_t 는 시간 t 에서의 오탐지 수, mme_t 는 시간 t 에서의 mismatch (객체 간 충돌이나 가려짐 현상 등에 의해 추적하는 객체가 바뀌는 경우) 수를 의미한다. 그리고 MT, ML, Frag, IDS 의 정의는 표 1 과 같다.

제 1 절 PETS2009

이 동영상에서는 여러 명의 보행자들이 움직이고 있으며, 두 명이상의 보행자들이 마주쳤을 때 이야기를 하는 등의 사회적 행동을 취하기도 한다.

MT (%)	Mostly tracked. Ground truth 경로의 전체 길이 중 80% 이상으로 추적을 성공한 경로의 비율. 높을수록 좋음.
ML (%)	Mostly lost. Ground truth 경로의 전체 길이 중 20% 미만으로 추적을 성공한 경로의 비율. 낮을수록 좋음.
Frag	Fragments. 추적된 경로가 중간에 끊어지는 횟수. 적을수록 좋음.
IDS	ID switches. 둘 이상의 추적된 경로에 각각 상응하는 객체가 중간에 서로 바뀌는 횟수. 적을수록 좋음.

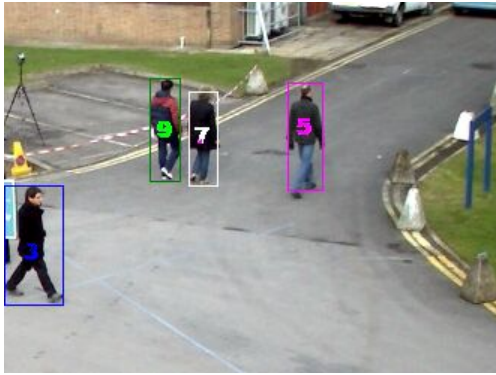
표 1. 성능 평가 지표.

	MOTA	MT	ML
KSP [1]	79.0%	-	-
Thida et al. [3]	83.0%	94.7%	5.3%
Andriyenko et al.[4]	81.4%	82.6%	0.0%
제안된 방법	83.6%	88.2%	3.7%

표 2. PETS2009-S2-L1 에서의 기존의 기법들과 제안된 기법의 추적 성능 비교.

탐지 과정에서는 sliding window 기법으로 화면을 훑은 후 각각의 window 에 대해 HOG([11]) 특징점들을 추출하여 보행자들을 탐지하였다. 그리고 추적 과정에서는 이 탐지 결과들로 경로를 계산하였다.

표 2 는 기존에 제안되었던 기법들([1], [3], [4])과의 성능을 비교한 표이



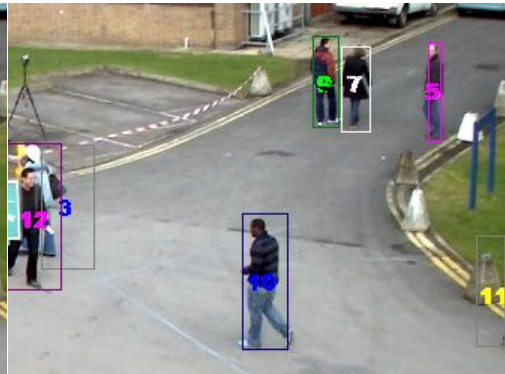
(a) Frame # 112



(b) Frame # 123



(c) Frame # 129



(d) Frame # 150

그림 3. PETS2009-S2-L1 동영상에서의 추적 결과.

다. 성능 테스트를 위해 사용한 동영상은 PETS2009-S2-L1 이며, 단일 카메라 영상만을 사용하였다. 성능을 비교한 결과, 제안된 기법은 기존의 기법들보다 MOTA 가 높음을 확인하였고, MT 나 ML 도 다른 기법에 비해 크게 뒤떨어지지 않는았다.

제 2 절 공 추적

지면을 위에서 내려다본 시점에서 모양의 차이가 거의 없고 이동하는 속력이 빠른 탁구공들이 화면을 가로질러 움직이는 동영상이다. 움직이는 객체(탁구공)와 배경의 색의 차이가 명백하므로 배경최적화 기법으로도 매우 높은 정확도로 움직이는 객체들을 탐지할 수 있다. 그러나 탁구공들의 모양이 서로 매우 유사하여 색상 정보는 거의 효력이 없고, 속력이 보행자에 비해 빠르기 때문에 탐지 과정에서 어려움이 있을 수 있다. 이 동영상은 [1]에서 제안한 기법과 성능을 비교하였다.

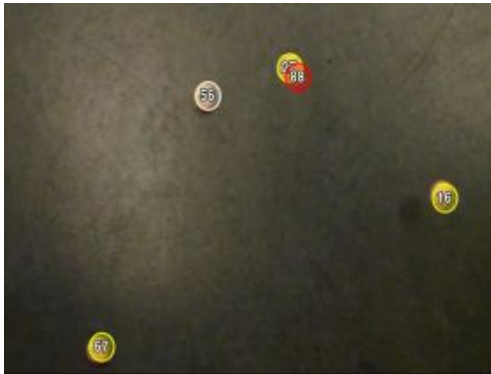
성능을 비교한 결과, 제안한 기법은 KSP 방식에 비해 MOTA, MT, ML, IDS 에서 더 높음을 확인하였으며, 특히 KSP 방식을 사용한 경우 IDS 의 값이 크게 나타났다. 이는 KSP 를 사용할 시 위치를 2 차원 노드(node)로 표현하여 나타냈기 때문에 위치 값이 그다지 정확하지 않았기 때문이다. 구체적으로 결과를 확인하기 위해 그림 4 을 보면, KSP 방식을 적용한 경우인 그림 4-(a)에서는 감지되지 않은 객체가 있었고, 그림 4-(b)에서는 하나의 객체에 두 개의 라벨이 연결된 것을 볼 수 있다. 반면 이 논문에서 제안한 기법을 적용한 경우인 그림 4-(e)와 그림 4-(f)에는 오류가 발견되지 않았다. 또한 세 번째 행과 네 번째 행은 연속된 프레임에서의 영상이다. 이 논문에서 제안한 기법을 사용한 경우에는 오류가 발생하지 않은 반면, KSP 방식을 사용한 경우 두 객체가 서로 가까이 접근할 때 라벨이 서로 바뀌는 오류가 발생하였다.



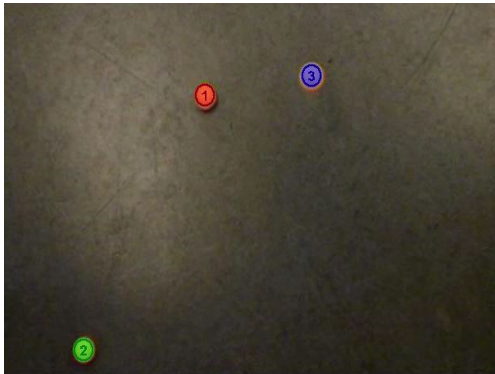
(a) KSP, Frame # 571



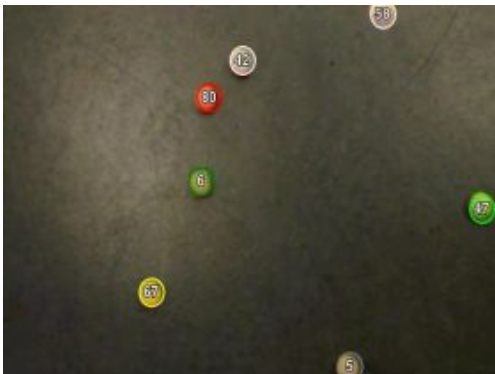
(e) 제안, Frame # 569



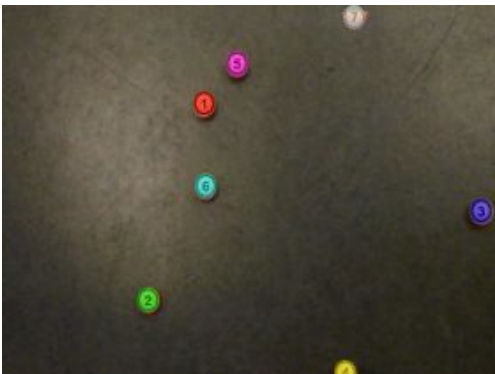
(b) KSP, Frame # 742



(f) 제안, Frame # 737



(c) KSP, Frame # 550



(g) 제안, Frame # 550



(d) KSP, Frame # 551



(h) 제안, Frame # 551

그림 4. 탁구공 동영상에서의 추적 결과. (a) ~ (d)는 KSP [1] 기법을, (e) ~ (h)는 이 논문에서 제안한 기법을 각각 적용한 결과의 일부이다. 같은 행에 위치한 프레임의 시간은 거의 비슷하다.

	MOTA	MT	ML	Frag	IDS
KSP	96.0%	92.9%	7.1%	0	36
제안	98.8%	96.0%	2.8%	0	4

표 3. 탁구공 동영상에서 기존 기법과의 성능 비교.

제 3 절 TownCentre

이 동영상에서는 매우 많은 사람들이 거리를 보행하고 있다. 주로 양쪽에서 사람들이 마주 보고 지나가기 때문에 오고 있는 사람들을 지나칠 시 가려짐 현상이 자주 발생한다.

탐지 과정에서는 역시 sliding window 기법으로 HOG 특징점 추출 기법으로 보행자들을 탐지하였다. 주어진 탐지 결과에서는 그림자 등에 의한 오탐지가 많이 발생하였는데, 이러한 오탐지가 다른 보행자들과 가려짐 현상을 일으켜 경로 추적에 방해가 되기 쉽다.

이 경우에 대비하기 위해 객체들의 색상 히스토그램을 최근의 일련의 프레임 동안의 객체의 색상을 이용하여 계산해 볼 수 있다. 그림 5는 단순히 이전 프레임의 객체의 색상만으로 객체의 색상 히스토그램을 구한 후의 추적 결과와 최근의 일련의 프레임 동안 객체의 색상들로부터 객체의 색상 히스토그램을 구한 후의 추적 결과를 보여준다. 고려하는 최근의 프레임 개수를 d_p 라 하면, 그림 5-(a)와 그림 5-(b)에서 $d_p=1$ 은 최근의 가장 마지막 프레임만을 고려했다는 의미이다. 이 경우 18번 객체는 오탐지인 22번 객체에 의해 IDS가 발생하였다. 반면 $d_p=10$ 인 그림 5-(c)와 그림 5-(d)에서는 17번 객체가 오탐지인 24번 객체와 마주쳐도 경로가 계속 유지되었다.

표 4는 d_p 의 값에 따른 추적 결과를 나타낸다. $d_p=10$ 인 경우가 $d_p=1$ 인 경우보다 정확도가 높고 오류가 적다는 것을 확인할 수 있다.



(a) $d_p = 1$, Frame # 255

(b) $d_p = 1$, Frame # 285



(c) $d_p = 10$, Frame # 255

(d) $d_p = 10$, Frame # 285

그림 5. d_p 프레임 동안의 객체의 색상으로 각 객체의 색상 히스토그램을 계산하였을 경우 추적 결과. (a)(b) 18 번 객체가 22 번 객체의 영향으로 IDS 가 발생하였다. (c)(d) 17 번 객체가 24 번 객체의 영향을 받지 않고 제대로 추적되었다.

	MOTA	MT	ML	Frag	IDS
$d_p = 1$	81.8%	83.5%	10.1%	21	22
$d_p = 10$	83.7%	84.1%	9.5%	21	14

표 4. TownCentre 동영상에서 d_p 의 값에 따른 추적의 성능 비교.

제 4 장 결 론

단일 카메라로 촬영한 영상에서 탐지된 객체들을 추적하는 기법을 고안하였다. 각각의 프레임에서 탐지된 객체들을 추적하기 위해, 연속된 프레임 사이에서 두 객체를 연결할 시 발생하는 비용을 수치화하였고, 이 비용 값들을 고려하여 총 비용을 최소화하는 방향으로 각각의 객체들을 연결함으로써 객체들의 이동 경로를 계산하였다. 비용을 계산하는데 고려되는 요소는 객체들의 위치 정보와 색상 정보이며, 위치 정보로는 탐지된 객체들의 네 구석의 좌표를 이용하였고, 색상 정보로는 탐지된 객체들의 색상을 몇 개의 작은 분류로 분류하여 히스토그램을 계산한 후, 이 히스토그램을 객체들 간의 유사도를 계산함으로써 사용하였다. 총 비용을 최소화하도록 객체들을 연결하는 문제를 최적화 문제로 공식화하여 해결하였다. 성능을 분석한 결과, 제안된 알고리즘으로 기존의 기법들보다 뛰어난 성능을 보임을 확인하였으며, 탐지 과정에서 오탐지가 발생하더라도 추적 과정에서 이동 경로를 계산하는데 미치는 좋지 않은 영향도 줄일 수 있었다.

참고 문헌

- [1] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua, “Multiple object tracking using k-shortest paths optimization,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1806-1819, Sept. 2011.
- [2] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, “Multi-camera people tracking with a probabilistic occupancy map,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 267-282, Feb. 2008.
- [3] M. Thida, H.-L. Eng, D. N. Monekosso, and P. Remagnino, “A particle swarm optimization algorithm with interactive swarms for tracking multiple targets,” *Applied Soft Computing*, May 2012.
- [4] A. Andriyenko, K. Schindler, “Multi-target tracking by continuous energy minimization,” *IEEE CVPR 2011*, pp. 1265–1272, June 2011.
- [5] PETS - Performance Evaluation of Tracking and Surveillance, 2009, <http://www.cvg.rdg.ac.uk/PETS2009>.
- [6] CLEAR - Classification of Events, Activities and Relationships, 2007, <http://www.clear-evaluation.org>
- [7] X. Cao, C. Wu, J. Lan, P. Yan, and X. Li, “Vehicle detection and motion analysis in low-altitude airborne video under urban environment,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 10, pp. 1522-1533, Oct. 2011.
- [8] V. Vapnik, “Statistical learning theory,” *New York: Wiley*, 1998.

- [9] Y. Li, C. Huang, and R. Nevatia, “Learning to associate: hybridboosted multi-target tracker for crowded scene,” *IEEE CVPR 2009*, June 2009.
- [10] B. Benfold and I. Reid, “Stable multi-target tracking in real-time surveillance video,” *IEEE CVPR 2011*, pp. 1265–1272, June 2011.
- [11] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” *IEEE CVPR 2005*, June 2005.

ABSTRACT

Matching and Tracking of Multiple Objects through Convex Optimization Formulation

Jangwook Kim

Department of Electrical and Computer Engineering

The Graduate School

Seoul National University

Multiple object tracking is performed by detecting objects and matching the objects in consecutive frames. In this paper, a tracking method is proposed where a convex optimization problem is formulated and solved to track moving objects. Objects are detected in every frame, and these objects are matched in consecutive frames, thus the trajectories of the objects are obtained. This method is called data association. We calculate costs from matching two objects. To calculate the costs, we consider the positions and the colors of the objects. Then we create a cost matrix that contains the cost values as its elements. By using the cost matrix, we formulate the convex optimization problem to minimize the total cost. Experimental results show that the proposed method outperforms the conventional method, and the method can reduce ID switch errors due to false positives.

주요어 : 다중 객체 추적, 컨벡스 최적화

학 번 : 2011 – 20813

ACKNOWLEDGEMENTS

지난 1 년간 석사학위를 받게 되기까지 주위로부터 많은 도움을 받았습니다. 이 지면을 빌어 감사의 말씀을 드리고자 합니다.

먼저 본 논문이 있기까지 부족한 저를 지도해 주시고 이끌어 주신 최진영 교수님께 감사를 드리며, 바쁘신 와중에도 흔쾌히 논문 심사를 맡아서 지도해 주신 김수환 교수님과 오성희 교수님께 감사를 드립니다. 또한 입학 후 석사 과정을 마칠 때까지 저에게 참된 배움의 길로 나아가도록 인도해 주신 여러 교수님들께도 감사를 드립니다.

석사과정 동안 함께 연구하고 기쁨과 슬픔을 나누었던 인지지능연구실 선후배님들께 진심으로 감사를 드립니다. 바쁘신 중에도 항상 따뜻한 조언과 격려를 아끼지 않으셨던 민수 형, 형진이 형, 선정이 누나, 광무 형, 수완이 형, 한주 형, 하욱이 형, 기민이, 문섭이에게 감사를 드립니다. 또한 영준이, 상두, 지윤이, 병호에게도 감사를 드리며, 이제 연구실 생활을 시작한 지 얼마 되지 않은 기경이와 병주에게도 좋은 결과가 있기를 바랍니다.

항상 부족한 아들을 위해 한없는 희생과 사랑을 주신 부모님께도
진심으로 감사를 드립니다. 부모님 기대에 어긋나지 않는 든직하고
자랑스러운 아들이 되도록 항상 노력하겠습니다. 저의 친
누나에게도 동생을 위한 그 마음에 고마움을 전합니다.

그간 제게 많은 관심과 격려, 그리고 힘을 주신 분들께 다시 한
번 큰 감사를 드리며, 이 좁은 가슴 속에서 미처 꺼내지 못한 저를
아껴주시는 모든 분들에게 감사한 마음을 전합니다.

2013 년 8 월

김장욱