영상처리를 이용한 물체추적 알고리즘

강해용 ○, 김용태, 이군석
LG DAC 연구소
lhaeyongg@lge.com

Object Tracking Algorithm on Vision

Haeyong KANG, Yongtae KIM, Gunseok LEE
LG DAC LAB.

요 약

카메라로부터 얻어지는 화상정보를 처리하여 사람을 검지하는 기술은 많은 분야에 적용될 수 있다. 실제로 많은 어플리케이션에 적용되고 있다. 현재 Tracking 기술에 관련한 다양한 논문과 방법들이 존재한다. 본 논문에서는 스테레오비전이 아닌 2-D조건에서 움직이는 물체와 움직이지 않는 물체를 구분하여, 구분된 영역에서 탬플릿 매칭을 통하여 사람 검지여부를 결정하는 알고리즘을 제안한다.

1. Introduction

카메라로부터 얻어지는 화상정보를 처리하여 사람검지 기술은 많은 분야에 적용될 수 있다. 보안분야에서는 어떤 특정구역의 사람검지 판단, 공공시설에서의 사람 출입 측정, 마켓이나 백화점에서의 사람 동선분석 등 적용할 수 있는 많은 어플리케이션이 존재한다. 최근에는 흔히 접할 수 있는 디지털카메라에도 화상처리기술이 적용된 사실을 알 수 있다. 셔터를 누르면 사람의 얼굴을 Tracking하여 얼굴의 밝기조절을 한다. 이외에도 가전제품, 자동차 전후방에 존재하는 사람 감지, 교통정보시스템 등이 있겠다.

흔히 말하는 Object tracking은 Computer Vision분야서 많은 연구가 진행 되어 왔다. 물체의 외곽을 추출하여 특징을 추출하여 기존의 감산처리로부터 얻을 수 있는 결과보다 더 향상된 기술보고서[1], 사람 형체 모델링을 통한 사람검지시스템[2], 클러스터링 기법을 이용한 사람 유동측정 시스템[3]등 다양한 Object Tracking 방법이 존재한다.

저자가 제안하는 Object Tracking Algorithm은 크게 3가지로 나누어 진다 (Fig. 1). 첫 번째는 Image Capture 하여 Robust한 데이터를 얻기 위한 전처리 과정, 두 번째는, 감산처리에 근거한 움직이는 물체와 고정된 물체의 Segmentation 하는 과정, 마지막으로, 움직이는 물체로 판단한 영역에서 Template Matching을 하여 사람 판별을 하는 과정이다. 실제로 본 논문을 통하여 Object Tracking Algorithm에 대해서 설명한다.

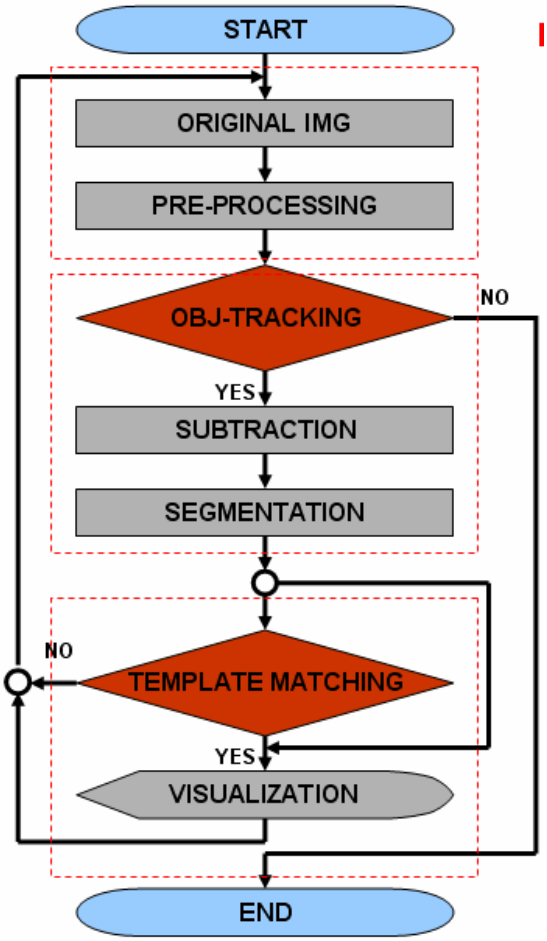

Fig. 1 the Flow of Algorithm

본 논문의 구성은 제2장에서는 저자가 제안하는 Object Tracking Algorithm에 대해서 언급한다. 제 3장에서는 제2장의 Object Algorithm을 토대로 세밀하게 검토 및

분석하여 고찰한다. **Algorithm**에 바탕을 둔 프로그램으로 실험을 하는 단계이기도 하다. 마지막 제 **4**장에서는 정확한 **Tracking**이 이루어 질 수 있도록 개선방향과 **Multi-Tracking Method**에 대해서 언급한다.

2. Object Tracking Algorithm

Object Tracking Algorithm은 대략 **4**단계에 걸쳐서 이루어 진다.

[STEP 1] Real-time Image Capturing & Pre-Processing
[STEP 2] Background Difference Processing
[STEP 3] Region Segmentation
[STEP 4] Template Matching

화상처리에 있어서 이미지 캡쳐링하는 시스템을 구축하는 것도 아주 중요하다. 또한 본격적인 화상처리에 앞서 전처리에 따라서는 결과값의 신뢰성을 좌우할 수 있다. 저자는 어떤 마스크영역에서 미디언 값을 취하는 필터링을 전처리 과정으로서 넣었다 **[STEP 1]. Object Tracking Processing**은 배경차분법에 기반을 두었다 **(Fig. 2) [STEP 2].** 단순히 배경 차분에 의해서 얻어진 정보 또한 노이즈가 발생하여 원하는 정보를 얻을 수 없다. 그래서 화상처리에서는 일반적으로 필터링의 기법 중의 하나로 침식연산과 팽창연산을 이용한다. 더욱이 응집된 데이터에 라벨링을 하여 어느 임계치 이하의 데이터는 움직이지 않는 물체로 인식처리를 하여 신뢰성을 높였다.

이러한 결과에 의해 얻어진 정보는 하나의 정보로 통일하는 작업을 한다. 이 처리가 바로 **Segmentation** 처리이다**[STEP 3].** 과거의 프레임에 없었던 것과 현재의 프레임에 생겨난 것을 정확하게 구별하는 과정이다. 현재의 영역에서 모델링한 템플릿 매칭이 이루어 진다. 템플릿 매칭에 사용되는 템플릿에 대해서는 **2.5**절에서 설명하겠다 **[STEP 4].**

매칭 결과 혹은 **Segmentation**의 결과를 다이얼로그상에 디스플레이 한다. 매칭 결과는 **Face, Head**에 따라 색깔로 구분될 수 있도록 디스플레이 하였다. 물론 같은 위치에서 같은 매칭이 이루어 질 수도 있다. 그 매칭 결과는 다이얼로그 박스 **(Fig. 6)**에서 확인할 수 있다.

2.1 Difference Processing

도로상에 차량이 존재하지 않았을 때 영상(배경사진)을 현재의 영상으로 감산처리 **(Fig. 2)**하는 기법을 말한다. **Object Tracking Algorithm**에서 **Background Difference Processing (Fig. 2)**을 이용하는 이유로는 우선, 가장 구현하기 쉽다. 그리고, 사전에 **Background Image**를 캡쳐링하여 실시간의 영상과 비교분석을 할 수 있어서 용이하다. 마지막으로, 결과값의 화상분석을 통하여 얻고자 하는 정보를 빨리 취득할 수 있다는 이점이 있다.

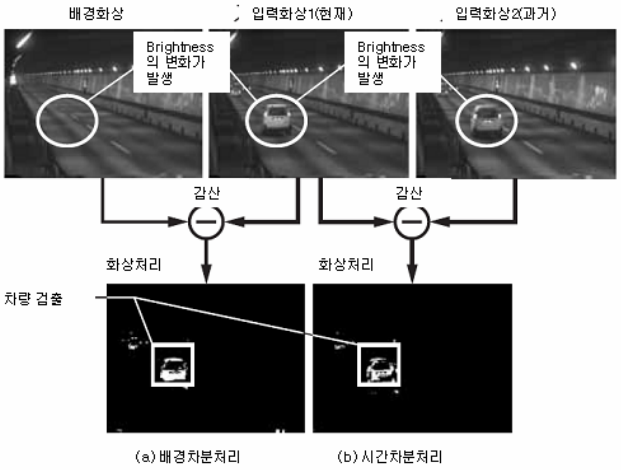

Fig. 2 Difference Processing

2.2 Pre-Processing

원활한 영상처리를 구현하기 위해서는 **Pre-Processing**이 아주 중요하다. 저자는 빈약한 명암 분포를 가진 영상에 적용. **Histogram equalization** (균등화)을 함으로써 **Gray** 레벨의 동적 영역을 증가시켰고, 그 결과 **Image contrast** (이미지 대비)가 증가 되었다. 반대의 효과도 있지만 **Image**의 시각적 외형을 개선할 수 있다. 특정부분 혹은 중앙에 명암이 치우침을 가지는 히스토그램 영상에 좋음은 명암대비스트레칭을 사용하였다.

필터링 기법으로는 마스크 **(3 x 3)** 내의 중간 값을 화소값으로 지정 (잡음이 생기거나 왜곡이 될 수 있을 경우 사용)하는 미디언 필터링기법과 이진영상에 대한 침식 **(Erosion) & 팽창(Dilation)**연산하므로 **Noise** 제거하는 모폴로지 연산, 그리고 각 영역을 라벨링처리하여 영역의 화소의 총합이 임계치**(Threshold)**에 미치지 못하는 영역 라벨을 제거**(Noise** 제거**)** 할 수 있는 라벨링에 의한 필터링을 사용하였다 **(Fig. 3).** 라벨링에 의한 필터링 기법은 **Parameter** 공간에서 데이터의 클러스터에 라벨을 붙여, 어느 임계치 이하이면 잡음으로서 인식처리 과정이라 생각하면 이해하기가 쉽다.

2.3 Color Extraction Processing

사람의 얼굴 색을 판별하기 위해서는 **2.2**절의 **Segmentation** 영역에서 색상을 추출할 필요가 있다. 화상처리에 있어서 일반적인 기법이기도 하지만 잠깐 설명

한다.

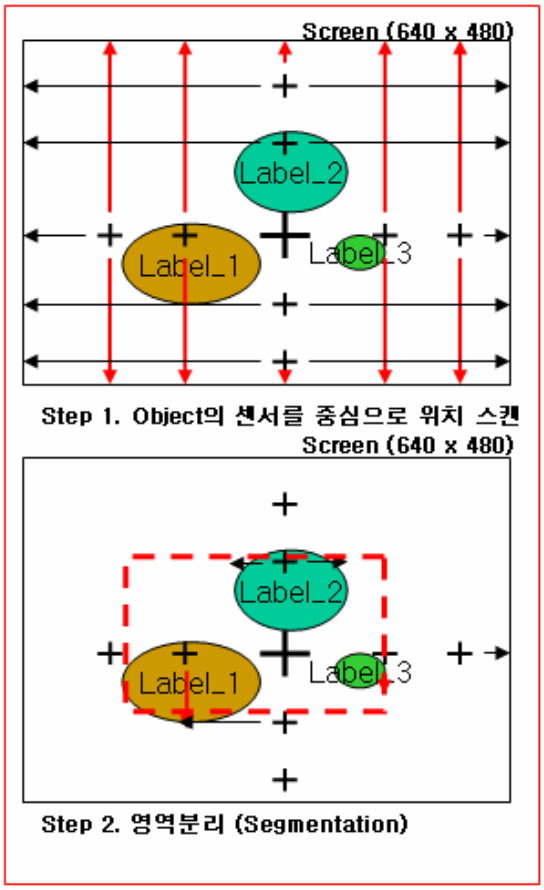

Fig. 3 Region Segmentation

RGB영상을 HSV영상으로 변환한다. 채도를 나타내는 V의 임계치110 이상은 Skin Color에 가깝다. 따라서 110 이상의 의 Pixel 좌표를 버퍼에 저장을 함으로써 Color 정보를 추출 할 수 있다.

2.4 Region Segmentation (Fig. 3)

영역분리를 하기 위해서는 우선 Background Subtraction에 의해 얻어진 화소를 바탕으로 고속 라벨링을 한다. 고속 라벨링을 하여 어느 임계치 이하는 움직이는 물체로서 인정하지 않는다(Fig. 3). 각 라벨의 좌표값을 이용하여 Label 1, 2, 3의 중심좌표를 산출한다.

Step 1에서 Object의 센터(중심)를 중심으로 어느 일정한 픽셀 (가로: 50픽셀, 세로 40픽셀)간격으로 4방면으로 스캔을 시작한다. 0이 아닌 다른 값을 만날 때 하나의 영역으로서 인정한다. Step 2에서 Step 1의 결과를 가지고 영역분리를 하게 된다. 이 영역 안에서 템플릿 매칭을 하게 된다.

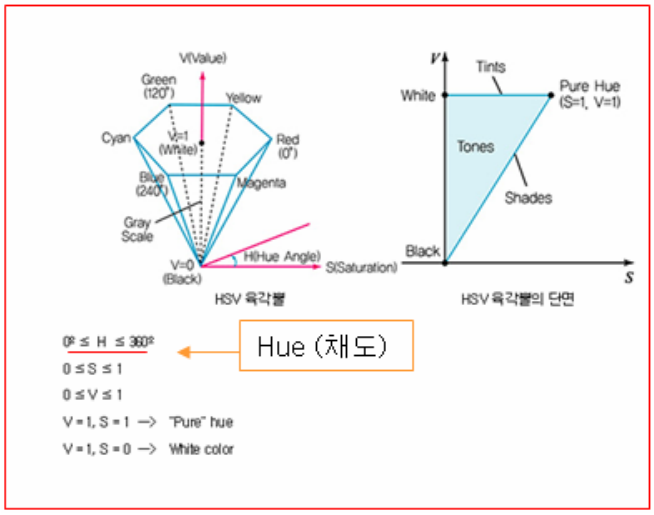

Fig. 4 Color Extraction

2.5 Template Matching Method

영역분리에서 얻어진 부분에서 모든 화소값 50 x 50 pixels 를 가우스분포로 변환하여 Template(Head and Face) Matching 이 이루어진다. 단, 얼굴 매칭은 2.3절에서 언급한 컬러정보(Hue)와 Background Subtraction 을로부터 얻어진 Binary Data를 이용한다. Template Matching 방법은 Template의 정의뿐만 아니라 Threshold값의 영향을 많이 받는다. 저자는 템플릿을 가우스 분포화하여 Template Matching을 제안한다 (Fig. 4).

저자가 생각한 마스크는 Image Processing 에서 Biometrics Recognition에서 자주 사용되는 Matching 기법 중의 하나이다. 미리 촬영한 얼굴 원형상에 수십 가지의 Variation을 두어 서로의 특징 점들을 확률적으로 계산하여 인식을 한다. 저자는 이 점을 이용하여, Head와 Face의 형상 (Fig. 5 (a))을 미리 Paint Shop Tool을 이용하여 준비한 후 가우스 분포에 의한 Mask (Fig. 5 (b))를 작성하였다. 여기서 간단히 Mask 작성법에 대하여 설명하겠다.

$$G_2(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad \text{-------- (1)}$$

가우스 분포는 Eq. 1에 의해 계산이 된다. 사용하는 마스크는, 최대 50 x 50 pixels이기 때문에, x 와 y의 범위는 -25에서 +25까지 이다. 실제 프로그램에서 가우스 분포에 의한 마스크 작성 함수는 아래와 같다. Template Mask 50 x 50 pixels로 둔 이유는, USB-CAM으로부터 1~3m 거리에서 평균적인 사람의 머리크기를 고려했기 때문이다. 사용자의 편의에 따라 적절하게 변경할 수 있

겠다.

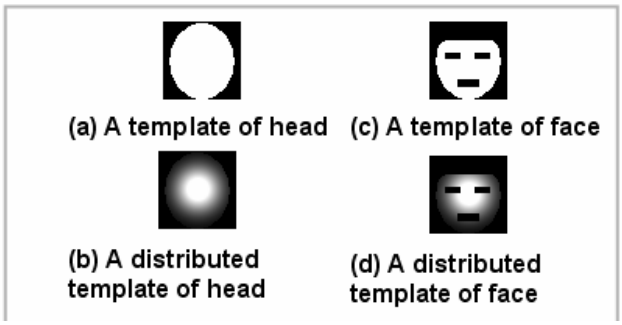

Fig. 5 Templates

3 Experiments

X-좌표축은 대략 -134.81~1548.84[mm]이다. 실제공간 좌표로부터 X좌표의 원점으로부터 각각360[mm], 620[mm], 950[mm]에 위치할 때, 카메라 좌표의 값과 비교 측정하였다. 사람의 높이(HEIGHT)는 측정할 필요는 없으므로 Y좌표는 불필요하다. 카메라의 렌즈로부터 비교측정 위치까지의 거리는 1900[mm] 떨어진 곳에 위치한다.

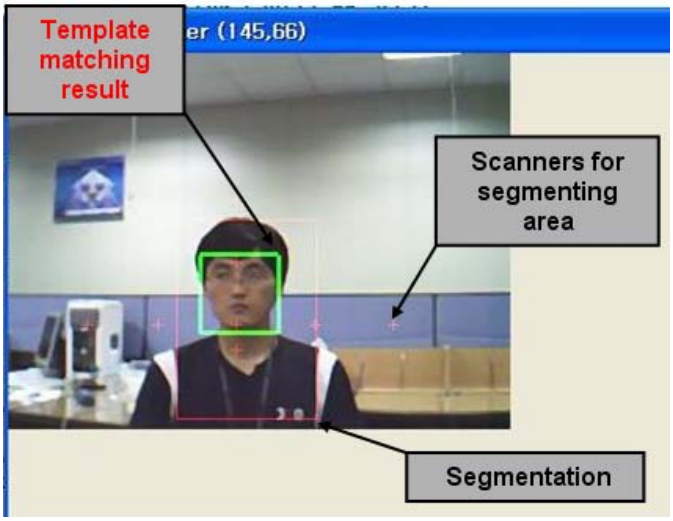

Fig. 6 an Experiment of Object Tracking

참고로, 사람은 카메라 좌표로부터 1900[mm] (Z좌표) 떨어 진 위치에서 Tracking한다. 각각의 위치에서 10회, 총 30회의 반복실험을 하였다. Fig. 6은 실제 본 논문에서 제안한 알고리즘을 바탕으로 구현한 모습이다. 평균적인 A Tracking Error Rate는 5.15[mm], 전체적인 A Tracking Error Rate는 150[mm]이하의 결과를 얻었다. Fig. 7에서는 실험 결과를 통계적으로 나타낸다.

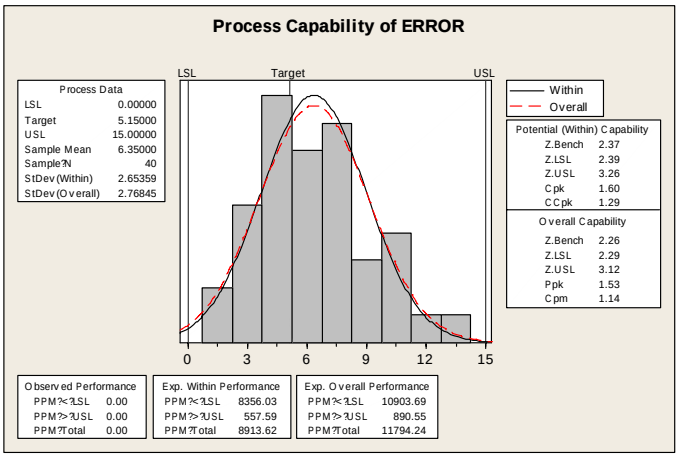

Fig. 7 an Error Rate of Tracking Object

4. Conclusion

본 논문에서 다루는 Tracking Algorithm은 Background Difference Processing을 바탕을 두었다. Difference Subtraction으로부터 얻어진 Binary Data를 고속 라벨링 처리를 하여 잡음처리가 가능하였다. 분리된 영역 내에서 템플릿 매칭을 통하여 사람의 검지 시스템을 구축하였다.

좀 더 정확한 템플릿 매칭이 이루어 지기 위해서는 Background Subtraction Processing에서 얻어진 Binary Data의 실루엣 처리가 필요하다. 현 단계의 모폴로지연산만으로는 정밀한 데이터 처리를 하는 데 무리가 따른다.

향후 Multi-Tracking을 구현하고 자 하였을 때는, 각 라벨에 대해 서로의 독립된 영역으로서 가정하여 템플릿 매칭을 하면 어느 정도 구현이 가능할 것으로 판단된다. Multi-Tracking에 관련한 연구 역시 많이 존재한다[4, 5, 6]. 마스크의 정의, 모델링, 특징추출, 클러스터링, 주성분분석 (통계적 접근) 등의 다양한 개념을 이용한다면 Tracking에 있어서도 만족할 만한 기대를 얻을 수 있을 것이다.

References

[1] Akinoshita Haruki, Suzuki Yoshihiko, " Image Processing System MPEC-2 Specifications", Toshiba Review Vol. 61 No.5, 2006

[2] Takeshi Shoji, Takashi Iketani, Yoshinobu Asokawa, "스테레오비전을 이용한 인간검지 시스템", OMRON TECHNICS Vol. 45 No. 1, 2005

[3] Baba Kenji, Enohara Takaki, Yuasa Yuichiro,

“ Pedestrian Traffic Measurement System Using Computer Vision”, Toshiba Review Vol. 61 No. 12, 2006

[4] Mei Han, Wei Xu, Hai Tao, Yihong Gong, “An Algorithm for Multiple Object Trajectory Tracking”, IEEE 2004

[5] D.B. Reid, “An Algorithm for Tracking Multiple Targets”, AC,24(6): 843-854, December 1979.

[6] I.J. Cox and S.L. Hingorani, An Efficient Implementation of Reid’s Multiple Hypotheses Tracking Algorithm and Its Evaluation for the Purpose of Visual Tracking”, IEEE Trans. Pattern Analysis and Machine Intelligence, 18(2): 138-150, Feb 1996.