

눈의 상태 인식을 이용한 디지털 카메라 영상 자동 보정 모듈의 구현

The Implementation of Automatic Compensation Modules for Digital Camera Image by Recognition of the Eye State

전영준*, 신흥섭, 김진일**

Young-Joon Jeon*, Hong-Seob Shin, Jin-II Kim**

요약

본 논문에서는 디지털 카메라를 이용하여 사진을 촬영할 때 눈의 감긴 상태를 확인하여 이를 자동으로 보정하여 출력해주는 모듈의 구현에 관하여 연구하였다. 먼저 촬영된 영상에 대하여 얼굴 및 눈의 영역을 검출하고 눈의 상태를 인식한다. 만약 눈이 감긴 영상이 촬영되었을 때 버퍼에 임시로 저장된 이전 프레임 영상들에 대하여 눈의 상태를 인식한 후, 가장 눈의 상태가 만족스러운 영상을 이용하여 눈을 보정한 후에 사진을 출력한다. 얼굴 및 눈을 정확하게 인식하기 위해서 SURF 알고리즘과 호모그래피 방법을 적용하여 영상을 보정하는 전처리 과정을 수행한다. 얼굴 영역과 눈 영역을 검출하는 것은 Haar-like feature 알고리즘을 이용하였다. 눈을 뜨고 있는 상태인지 감은 상태인지를 눈의 영역에 대한 템플릿매칭을 이용한 유사도를 판단하여 확인한다. 본 연구에서 개발된 기능을 다양한 형태의 얼굴 환경에서 테스트한 결과 얼굴이 포함된 영상에 대하여 효과적으로 보정이 수행됨을 확인하였다.

ABSTRACT

This paper examines the implementation of automatic compensation modules for digital camera image when a person is closing his/her eyes. The modules detect the face and eye region and then recognize the eye state. If the image is taken when a person is closing his/her eyes, the function corrects the eye and produces the image by using the most satisfactory image of the eye state among the past frames stored in the buffer. In order to recognize the face and eye precisely, the pre-process of image correction is carried out using SURF algorithm and Homography method. For the detection of face and eye region, Haar-like feature algorithm is used. To decide whether the eye is open or not, similarity comparison method is used along with template matching of the eye region. The modules are tested in various facial environments and confirmed to effectively correct the images containing faces.

Keywords : SURF Algorithm, Haar-like feature, Face Detection, Eye Detection, Eye State Recognition, Homography, Digital Camera

I. 서론

현재 디지털 카메라 모듈은 다양한 어플리케이션에 적용되고 있으며 카메라 모듈이 적용된 거의 모든 어플리케이션은 기술적으로 지속적인 발전을 하고 있다. 의료 분야, 로봇 분야, 디지털 스틸 카메라 분야, 모바일 단말기 분야 등에서는 앞으로도 관련 기술 경쟁이 치열해질 것으로 전망된다. 디지털 카메라는 이미지 센서를 이용하여 메모리카드에 이미지를 저장하고 촬영한 이미지

는 LCD 모니터로 즉시 확인 가능하며, 필요에 따라 카메라에서 바로 다양한 형태의 영상에 대한 처리가 가능하다. 최근의 디지털 카메라는 1,000만 화소가 넘으며 라이브 뷰 기능, 먼지 제거기능, 손떨림 보정 기능, 터치 스크린 LCD 등 다양한 기능이 탑재되고 있다. 특히 스마트 폰의 대중화와 기술경쟁으로 인하여 스마트 폰에 장착되는 디지털 카메라의 화소도 1,000만 화소급 이상이 되고 있으며 이와 관련된 다양한 어플리케이션이 개발되고 있다. 또한, 최근에는 영상처리 분야에서 다양하게 활용되고 있는 얼굴 및 눈 검출 기술을 활용한 어플리케이션 개발 및 응용들이 다양한 형태로 이루어지고 있다.

얼굴 인식과 관련된 연구에는 영상에서 얼굴 영역을 추출하기 위한 연구, 얼굴을 구성하는 주요소 추출에 관한 연구, 얼굴 영상의 해석에 따른 얼굴의 표정 연구 등이 있다[1]. 얼굴 영역에서 눈을 검출하는 연구는 학습기반 검출 방법과 비 학습기반 검출방

* 동의대학교 부산IT융합부품연구소

** 동의대학교 컴퓨터공학과(교신저자)

투고일자 : 2013. 7. 15 수정완료일자 ; 2013. 7. 29

계재확정일자 : 2013. 8. 1

※ 이 논문은 2011학년도 동의대학교 연구년 지원에 의하여 연구되었음.

법으로 나눌 수 있다. 주성분 분석방법, AdaBoost 분류기를 이용한 방법 등과 같이 통계적인 정보를 이용한 학습기반 눈 검출 방법이 있으며, 비 학습기반 검출 방법에는 투영함수를 이용한 눈 검출 방법, 컬러 정보를 이용한 눈 지도를 생성하는 눈 검출 방법, 영상의 밝기, 에지 정보 등을 조합하여 눈 검출을 시도하는 방법 등이 있다[2]. 본 논문은 얼굴 영역과 눈 검출 기능을 응용하여 디지털 카메라의 기능을 향상시키는 부분에 대한 연구로서 Adaboost 분류기를 이용한 학습데이터를 활용하여 Haar-like feature 알고리즘을 통해 얼굴과 눈 영역을 추출하는 학습기반 검출 방법을 이용하였다.

디지털 카메라를 이용하여 사진을 찍을 때 얼굴이 포함된 영상에서 순간적으로 눈이 감긴 상태로 사진이 찍혀서 다시 촬영하는 경우가 많이 있다. 이러한 문제를 해결하기 위해서 본 논문에서는 인물이 포함된 사진을 촬영하는 경우 인물의 얼굴 영역 및 눈 영역을 검출한 후 눈의 상태를 인식하여 눈이 감겨진 사진이 찍혔을 경우 이를 이전 프레임의 영상을 이용하여 얼굴을 보정한 후 출력하는 기능을 구현하였다. 본 논문의 구현 기능은 먼저 사진 촬영할 때 촬영된 영상과 이전 몇 프레임의 영상이 자동으로 저장된다. SURF(Speeded Up Robust Features) 알고리즘과 호모그래피(Homography) 방법을 이용하여 디지털 카메라에 찍힌 영상을 보정하고 Haar-like feature 알고리즘을 이용하여 얼굴 영역 및 눈 영역을 추출한 후 눈의 상태를 인식한다. 그리고 사용자가 사진을 찍었을 때 눈이 감긴 영상이 찍혔을 경우 버퍼의 프레임 영상들에서 동일한 방법으로 눈의 상태를 찾아서 가장 눈의 상태가 양호한 영상을 이용하여 자동으로 눈을 보정하여 눈이 뜬 상태의 영상으로 출력하도록 하였다.

II. 관련 연구

관련 연구에서는 본 논문에서 영상의 왜곡을 보정하기 위해 전처리 과정에 사용된 알고리즘에 대하여 살펴보고 얼굴 및 눈 영역을 검출하기 위해 사용된 Haar-like-feature 방법의 특징에 대하여 알아본다.

2.1 영상 전처리 알고리즘

본 논문에서는 디지털 카메라를 이용하여 사진을 찍을 때 촬영한 사진의 보정을 위해서 버퍼에 임시로 저장된 이전 프레임의 영상을 이용하는데 이때 버퍼에 저장된 사진이 촬영 사진과 비교하여 카메라 움직임에 의한 프레임 사이의 기하학적 왜곡이 발생하여 영상이 기울어지거나 차이가 나게 된다. 이때 이를 보정하는 전처리 과정을 수행하게 된다. SURF 알고리즘을 이용해서 버퍼 프레임에 대한 특징점을 구하고 난 후, 호모그래피로 마지막 촬영 영상을 기준으로 버퍼에 저장되어 있는 영상 프레임의 특징점에 대한 기울기를 구한다. 구한 기울기로 버퍼 프레임 영상에 대하여 워핑(Warping)과정을 통해 영상 왜곡을 보정한다.

SURF 알고리즘은 여러 개의 영상으로부터 크기(Scale), 조명(Illumination), 시점(Viewpoint) 등의 환경 변화를 고려하여 환경 변화에 불변하는 특징점을 찾는 알고리즘 중 하나이다[3]. 일반적으로 성능이 우수하다고 알려진 SIFT(Scale Invariant Feature

Transform) 알고리즘과 견줄만한 성능을 보이면서 동시에 속도를 크게 향상시킨 알고리즘이다. 영상에서 특징점(Interest point)을 빠르게 검출하기 위하여 원본 영상의 적분이미지를 사용하여 웨이블릿 정보를 추출하고 웨이블릿 정보를 바탕으로 근사화한 헤시안 검출기를 통해 특징점을 추출한다. 헤시안 검출기는 헤시안 행렬에 기반을 둔 특징점 탐색 알고리즘으로서 정확성이 좋다. 찾아낸 각각의 특징점에서 고유 영역을 표현할 수 있는 표현자(Descriptor)를 생성하고 이를 이용하여 소스 영상과 타겟 영상의 매칭 되는 점을 찾는다[4][5]. 그림 1은 SURF 알고리즘의 흐름도를 나타낸다.



그림 1. SURF 알고리즘 흐름도
Fig. 1. Flow chart of SURF algorithm

호모그래피는 왜곡된 영상을 보정하는 방법 중의 하나이다[6]. 카메라에서 촬영되는 이미지는 원근왜곡이 발생할 수 있다. 이러한 원근왜곡을 제거하기 위해 호모그래피를 사용하여 영상을 변형시키는 방법을 사용한다. 호모그래피를 하기 전에 소스영상의 특징점과 표현자, 그리고 타겟 영상의 특징점과 표현자를 이용해 각각의 점들의 매칭 되는 점들을 찾는다. 매칭 되는 점들은 따로 벡터에 담아둔다. 이렇게 찾은 벡터는 짝수 번째에는 소스영상의 특징점과 표현자로 구성되어 있고 홀수 번째에는 타겟 영상의 특징점과 표현자로 구성된다. 호모그래피를 하기 위해 다시 벡터를 짝수와 홀수로 두 개의 벡터로 나눈다. 호모그래피는 구한 두 개의 벡터와 RANSAC(Random Sample Consensus) 알고리즘을 이용해서 구한다. 해당 영상의 특징점과 표현자의 데이터 집합은 노이즈로 인해 이상치(outlier)가 발생할 가능성이 매우 높다. RANSAC 알고리즘을 이용해 원하는 수학적 모델에 가깝게 구할 수 있다. RANSAC 알고리즘은 데이터와 모델과의 거리를 판단해 거리가 임계치(threshold) 이하 값이면 그 모델을 지지하는 데이터로 간주한다[6][7].

호모그래피로 구한 데이터를 이용해 소스영상의 화소에 적용하여 워핑 방법을 이용하여 처리할 수 있다. 영상 워핑은 인공위성이나 다른 우주계획의 일원으로부터 보내져 온 일그러진 영상을 올바르게 만드는 데 처음 사용하였다. 일그러짐 현상은 렌즈의 변형이나 평평한 판에 의해서보다는 구체 모양으로 된 다른 부품들 때문에 발생한다. 기하학적 변형에 의한 영상 워핑은 이러한 일그러짐을 복구시키는 데 사용된다. 영상 워핑은 특정한 규칙에 따라 입력 영상의 크기, 길이, 두께 등 형태를 변형하는 기술로 화소의 위치를 이동하는 기하학적 처리 방법으로써 일그러진 영상을 올바르게 보이도록 만든다든지 한쪽 부분을 늘이거나 줄인다든지

하는 효과를 줄 때에 사용한다. 즉 한 영상의 화소들을, 미리 사용자가 정한 위치변환 규칙에 의해서 다른 위치로 옮겨주는 것이다. 본 논문에서는 손떨림에 의한 영상의 왜곡이나 기울어진 영상을 보정하는 것에 사용한다.

2.2 Haar-like-feature 알고리즘

얼굴 및 눈 영역을 검출하기 위한 알고리즘에 대한 연구는 다양하게 이루어져 왔다. 이 가운데 본 논문의 기능 구현에 사용된 방식은 Haar-like-feature를 얼굴을 인식하는 인식자로서 사용하였다. Haar-like feature는 Viola가 처음 얼굴 검출에 적용한 것으로 단순 합 이미지를 이용하여 특징 값을 표현하는 것이다[8]. 이는 위치, 모양 크기에 따라서 수많은 형태를 가질 수 있다. 이를 AdaBoost 학습 알고리즘을 사용하여 약한 인식자들을 합쳐서 강한 인식자의 그룹으로 만든다. 이 때 생성된 특징값은 얼굴의 특징을 잘 포용하는 장점을 가진다. Haar-Like feature와 AdaBoost에 기반한 객체검출 기술은 높은 정확도와 빠른 처리 속도를 보여준다. 계산 방식이 단순 합이기 때문에 동영상에서의 적용이 매우 용이하다[8][9].

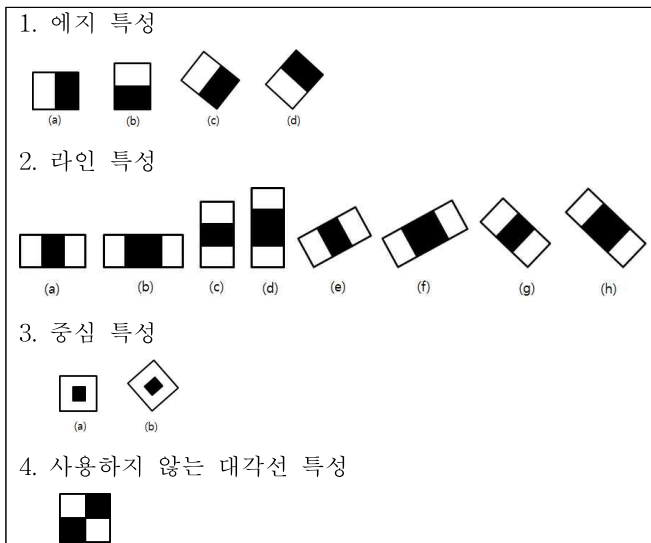


그림 2. Haar-like feature 프로토타입
Fig. 2. Haar-like feature prototype

Haar-like-features 알고리즘은 사람의 얼굴 영역에 그림 2와 같은 Haar-like-feature를 이용하여 사각형을 겹쳐 놓은 다음 사각형에 포함되는 픽셀의 평균에서 사각형내의 검은 색의 영역에 포함되는 픽셀값의 평균에 대한 차이를 구한 뒤, 그 차이가 어떠한 임계치(threshold)를 넘으면 사람 얼굴에 대해 Haar-like feature가 있다고 판단한다. 이러한 방식으로 에지 특성과 라인 특성, 중심 특성을 비교 한 후 모두 통과하게 되면 이를 원하는 객체로 판단하고 해당 객체영역을 반환한다. 미리 학습된 데이터를 바탕으로 입력 영상에 대해 학습된 데이터의 모든 샘플 수에 대하여 그림 2의 모든 특성을 비교한다. 비교는 순차적으로 진행되며, 각 단계 별로 하나라도 만족하지 못하면 객체가 아니라고 판단하고 그 다음 샘플을 비교한다[10][11].

III. 눈 상태 인식 및 영상 자동 보정

본 논문에서는 디지털 카메라에서 촬영한 영상에서 눈이 감긴 영상인 경우 자동으로 보정하는 기능을 구현하였다. 기능 구현을 위하여 영상 획득 모듈, 영상 전처리 모듈, 얼굴 및 눈 영역 검출 모듈, 눈 상태 인식 모듈, 영상 보정 출력 모듈 등 5개의 세부 모듈로 나누어 구현하였다. 그림 3은 영상 자동 보정을 위한 전체 기능 모듈 구성도를 나타낸 것이다.

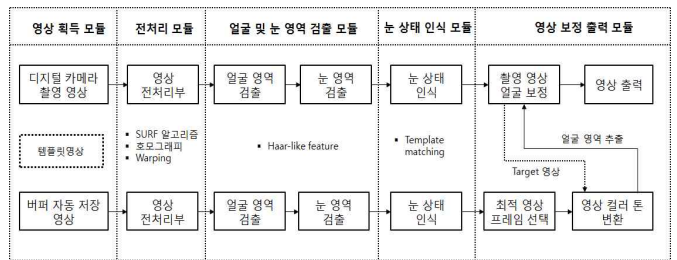


그림 3. 전체 기능 모듈 구성도

Fig. 3. Diagram of overall function modules architecture

SURF 알고리즘과 호모그래피 방법을 이용한 전처리 과정을 통하여 영상을 보정한다. 전처리 과정 후 얼굴 영역과 눈 영역을 검출하기 위해서 앞에서 설명한 Haar-like-feature 알고리즘을 이용한다. 검출된 눈 영역에 대하여 템플릿매칭 알고리즘을 이용하여 눈의 상태를 인식한다. 그 결과 촬영 영상이 눈이 감긴 경우 이전 프레임 영상에서 눈을 뜬 영상을 찾아 이를 수정한다. 이때 촬영 영상의 이전 몇 개의 프레임 영상은 사진을 촬영하는 순간 자동으로 버퍼에 임시로 저장되도록 하여 이를 이용하여 촬영 영상을 자동 보정한다.

3.1 전처리 모듈

본 논문의 구현에서는 사진을 촬영하는 순간 찍힌 사진과 함께 그 이전의 몇 개의 프레임을 동시에 저장하여 영상을 획득한다. 본 논문의 구현에서는 10개의 프레임까지 저장할 수 있도록 하였다. 이때 촬영된 사진과 버퍼에 저장된 사진에 대하여 촬영된 사진을 보정하기 위해서는 버퍼에 저장된 이전 프레임의 사진이 촬영된 사진과 비교하여 왜곡이 발생하지 않아야 한다. 따라서 버퍼에 저장된 이전 프레임 영상들이 얼굴이 일부 기울어진 상태이거나 손떨림으로 기하학적 왜곡이 발생하는 것을 전처리 모듈에서 보정을 한다. 이를 위해 SURF 알고리즘과 호모그래피 방법을 이용한 전처리 과정을 통하여 영상을 보정한다. SURF 알고리즘을 이용해 촬영 영상에 대하여 버퍼 프레임 영상에 대한 특징점을 구한다. 그 결과로 특징점에 대한 정보와 몇 개의 특징점이 추출되었는지에 대한 정보도 같이 얻을 수 있다. 해당 특징점 정보는 중심좌표 x, y, 사이즈, 라플라시안 정보가 담겨있다. SURF 알고리즘 결과를 이용하여 호모그래피 방법으로 촬영 영상을 기준으로 버퍼 프레임의 영상에 대하여 특징점에 대한 기울기를 구한다. 그 결과를 이용하여 워핑 방법으로 화소의 위치 변환을 통해 기울어진 영상을 보정한다. 영상의 워핑 방법은 어떤 이미지, I 가

있다고 할 때, I 안의 (x,y) 좌표에 있는 화소를 $I(x,y)$ 라고 하면 이 화소를 미리 지정된 위치변환함수 $U_x(x,y)$, $U_y(x,y)$ 의해서 옮겨주면 그 결과는 $I(U_x(x,y), U_y(x,y))$ 가 된다. $U_x(x,y)$ 는 x 방향의 위치변환을 나타내고, $U_y(x,y)$ 는 y 방향의 위치변환을 나타낸다. 그림 4는 전처리 과정을 나타낸다.



그림 4. 전처리 과정

Fig. 4. Process of preprocessing

3.2 얼굴 및 눈 영역 검출 모듈

얼굴 및 눈 영역 검출 모듈은 촬영 영상과 전처리 과정을 거친 버퍼에 저장된 프레임 영상들에 대하여 얼굴 영역을 검출한 후 눈 영역을 검출한다. 촬영 영상과 버퍼 프레임 영상을 입력으로 하여 얼굴 영역과 눈 영역을 검출할 때 환경 요소에 강인하면서 빠르게 연산이 가능한 Haar-like-feature 알고리즘을 이용한다. 본 논문에서는 Linehart가 얼굴 및 눈 영역의 검출율을 높이기 위해 제안한 확장된 Haar-like feature 알고리즘을 이용하였다 [11]. 그림 5는 Haar-like-feature 알고리즘을 이용하여 얼굴 인식과 눈의 영역을 인식하는 과정을 나타낸 것으로 본 논문에서는 그림 2의 확장된 Haar-like-feature의 프로토타입을 이용하여 AdaBoost 학습 알고리즘으로 학습된 데이터를 기반으로 해서 Haar-like-feature 알고리즘을 이용하여 얼굴 영역을 먼저 검출하고 다음으로 눈 영역을 검출한다. 촬영 영상과 버퍼에 저장된 이전 프레임 영상들에 대하여 얼굴 영역을 검출하고 눈 영역을 검출한 이후 눈의 상태를 인식한다.



그림 5. Haar-like feature를 이용한 얼굴 및 눈 영역 검출
Fig. 5. Face and eye regions detection by Haar-like-feature

3.3 눈의 상태 인식 모듈

눈의 상태 인식 모듈에서는 카메라 촬영 영상과 버퍼에 임시로 저장된 이전 프레임 영상에 대하여 눈 영역을 검출한 후 눈의 상태를 인식한다. 영상에서 눈이 감긴 상태인지 뜬 상태인지를 판단

하기 위해서 템플릿매칭을 사용하였다[12]. 실시간으로 사용자의 눈을 뜬 사진과 감은 눈 사진을 찍어서 템플릿 영상으로 저장한다. 이때 저장된 템플릿 영상에 대하여 Haar-like-feature를 통해 눈 영역을 검출한다. 검출된 템플릿 영상의 눈 영역 영상과 촬영 영상 및 버퍼 저장 영상들 사이의 상관관계를 이용한 템플릿 매칭을 수행한다. 템플릿 매칭은 식 1과 같이 두 이미지 간의 상관계수 값을 이용하여 유사도를 측정하였고 이 값은 0~1 사이 값을 가지게 된다. 1에 가까울수록 유사도가 높다[12][13]. 검출된 눈 영역에 대하여 눈을 뜬 상태와 눈을 감은 상태의 영상에 대한 템플릿 매칭을 각각 수행하여 해당 영역 내의 가장 큰 유사도를 구하여 비교한 후 상대적으로 유사도가 큰 값으로 눈의 상태를 판단한다. 즉 눈을 뜬 상태의 이미지와 유사도가 높으면 눈을 뜬 상태이고, 감은 상태의 이미지에 대한 유사도가 높으면 감은 상태로 인식한다.

$$R(s,t) = \frac{\sum_{x,y} [I(x,y) - \bar{I}(x,y)] [T(x-s,y-t) - \bar{T}]}{\sqrt{\sum_{x,y} [I(x,y) - \bar{I}(x,y)]^2 \times \sum_{x,y} [T(x-s,y-t) - \bar{T}]^2}} \quad (1)$$

단, 입력 영상($M \times N$) : $I(x,y)$,
 템플릿 영상($M' \times N'$) : $T(x,y)$,
 정규화교차상관계수 : $R(s,t)$, $-1 \leq R(s,t) \leq 1$.
 $s=0, 1, 2, \dots, M-1, t=0, 1, 2, \dots, N-1$,
 \bar{T} : $T(x,y)$ 의 평균, $\bar{I}(x,y)$: $I(x,y)$ 의 평균

눈의 상태를 인식하는 알고리즘은 그림 6과 같다. 눈의 상태를 인식한 후 버퍼에 저장된 영상 중에서 눈을 뜬 상태의 영상에 대하여 가장 유사도가 높은 영상을 선택하여 촬영 영상을 보정한다.

```

while (Haar-like-feature 알고리즘을 통해 얻은 얼굴 인식 개수)
{
    1. 눈을 뜬 상태의 영상과 눈을 감은 상태의 영상을 입력하여 패턴 추출
    2. 입력 : Haar-like-feature 알고리즘을 통해 얻은 눈 영역 이미지 데이터
    3. 템플릿매칭을 이용한 유사도 검사
        {
            눈을 뜬 상태의 영상과 입력 영상에 대한 유사도 검사
            눈을 감은 상태의 영상과 입력 영상에 대한 유사도 검사
        }
    4. 유사도 비교
        if(눈을 뜬 상태 유사도 > 눈을 감은 상태 유사도)
            { 눈을 뜬 상태로 인식 }
        else
            { 눈을 감은 상태로 인식 }
}
    
```

그림 6. 눈의 상태 인식 알고리즘
Fig. 6. Recognition algorithm for eye state

3.4 눈의 자동 보정 및 출력 모듈

눈의 자동 보정 및 출력 모듈은 눈의 상태 인식 결과로 이미지 버퍼에 담겨져 있는 최적의 프레임 영상을 가져와서 사용자가 활

영한 영상에 대하여 눈 영역을 편집한다. 이 때 버퍼에서 가져온 영상은 조명이나 외부 영향으로 촬영한 영상과 컬러가 다를 수 있어서 촬영 영상을 보정할 때 사용하는 경우 영상이 왜곡되는 현상이 발생하게 된다. 이러한 현상을 보정하기 위해 촬영 영상인 타겟 영상을 이용하여 버퍼에서 가져온 최적의 프레임 영상인 소스 영상에 대하여 컬러 변환 기법을 통한 보정을 수행한다[14]. 먼저, 영상의 화소값을 RGB 컬러 공간에서 *lab* 컬러 공간으로 변환한다. 식 2를 이용하여 소스 영상과 타겟 영상의 RGB 컬러 공간에서 LMS 컬러공간으로 변형한다.

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.3811 & 0.5783 & 0.0402 \\ 0.1967 & 0.7244 & 0.0782 \\ 0.0241 & 0.1288 & 0.8444 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2)$$

식 3을 이용하여 LMS 컬러공간에서 다시 인간의 색채 지각에 대한 *lab* 컬러공간으로 변형한다.

$$\begin{bmatrix} l \\ a \\ b \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -2 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix} \quad (3)$$

lab 컬러공간에서 소스 영상의 각 채널별 평균, 표준편차, 분산을 각각 타겟 영상의 각 채널별 평균, 표준편차, 분산과 일치하도록 수정한다. 두 이미지 사이의 색상이 유사한 느낌의 이미지로 표현하기 위해 평균과 분산을 각각 구한다. 그리고 타겟 이미지에 대하여 평균값을 뺀다.

$$\begin{aligned} l^* &= l - \langle l \rangle \\ a^* &= a - \langle a \rangle \\ b^* &= b - \langle b \rangle \end{aligned} \quad (4)$$

다음으로, 식 5와 같이 타겟 영상과 소스 영상의 분산 비율을 곱한다.

$$\begin{aligned} l' &= \frac{\sigma_t^l}{\sigma_s^l} l^* \\ a' &= \frac{\sigma_t^a}{\sigma_s^a} a^* \\ b' &= \frac{\sigma_t^b}{\sigma_s^b} b^* \end{aligned} \quad (5)$$

식 5를 이용하여 *lab*의 값을 보정한 뒤 식 6과 식 7을 다시 원래의 컬러공간인 RGB값으로 되돌린다. 이와 같은 방식으로 소스 영상을 타겟 영상에 가깝게 표현함으로써 촬영 영상을 편집할 때 두 이미지 간의 컬러의 차이를 보정할 수 있다.

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -2 & 0 \end{bmatrix} \begin{bmatrix} \frac{3}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{6}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{2}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} l \\ a \\ b \end{bmatrix} \quad (6)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 4.4679 & -3.5873 & 0.1193 \\ -1.2186 & 2.3809 & -0.1624 \\ 0.0497 & -0.2439 & 1.2045 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix} \quad (7)$$

버퍼에서 가져온 눈의 상태가 최적인 영상에 대한 컬러 보정이 끝난 후 수정된 버퍼 영상을 이용하여 촬영 영상을 보정한다. 즉, 촬영 영상의 얼굴 영역에 대하여 버퍼 영상의 얼굴 영역을 가져와서 보정을 수행한 후 출력한다.

IV. 실험 및 고찰

본 논문에서 제안한 기능을 구현하여 테스트하기 위하여 웹 캠 카메라를 이용하였다. 웹 카메라를 동작시키고 사진을 찍는 순간 이전 10개의 프레임은 버퍼에 자동 저장되도록 하였으며, 최종 찍힌 사진 영상을 분석하여 눈의 상태를 확인한 후 눈이 감긴 상태인 경우에 본 논문에서 제안된 알고리즘에 의하여 보정 작업을 수행하도록 하여 테스트하였다. 테스트를 위해 촬영된 영상과 버퍼에 저장된 영상의 크기는 640×480이다. 본 논문에서는 처리를 빠르게 하기 위하여 버퍼에 있는 영상들과 촬영 영상에 대하여 동시에 처리할 수 있도록 프로그램을 구현하였다. 주요 학습데이터는 OpenCV의 라이브러리를 이용하였다.

본 논문에서 제안한 기능의 구현에 있어서 전처리 모듈은 SURF 알고리즘과 호모그래피 및 워핑 방법을 이용하여 버퍼에 저장된 이전 프레임을 보정한다. 먼저, 전처리 과정에서 SURF 알고리즘을 수행하게 되는데 그 결과로 특징점에 대한 정보와 몇 개의 특징점이 추출되었는지에 대한 정보도 같이 얻을 수 있다. SURF 알고리즘 수행 결과는 그림 7과 같은 형태이며, 중심좌표 (x, y)에 사이즈만큼 원을 표시하여 나타내었다.

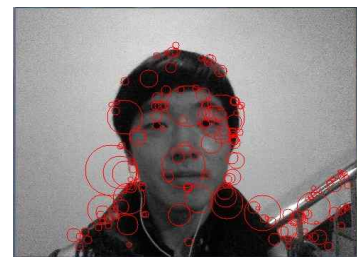


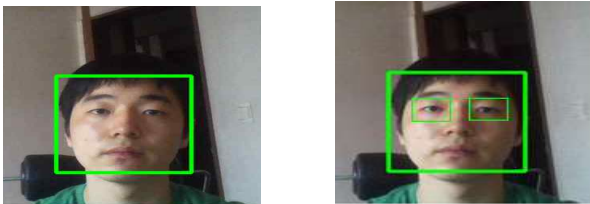
그림 7. SURF 알고리즘을 이용한 특징점 추출 영상
Fig. 7. Image of the SURF based feature extraction

그림 8은 호모그래피를 이용해 촬영 영상을 기준으로 버퍼 이미지에 대하여 기울기를 구한 후 이미지를 워핑 처리한 결과 화면이다. 그림 8에서 두 이미지 중 위의 이미지는 촬영된 사진으로서 기준이 되는 프레임이다. 아래 이미지는 틀어진 이미지로 워핑 방법을 적용하기 전에는 흰 사각형 영역과 같이 기울어져 있었던 사진이다. 촬영 사진을 기준으로 SURF 알고리즘으로 구한 특징점과 표현자를 이용해서 아래 이미지의 기울기를 구하고 워핑 방법을 적용했다. 그 결과 기준이 되는 사진 프레임과 거의 유사한 프레임으로 사진 각도가 틀어지는 문제를 해결할 수 있다.



그림 8. 호모그래피와 워핑 방법을 이용한 기울기 보정
 Fig. 8. Gradient correction using homography and warping method

그림 9는 Haar-like-feature 알고리즘으로 OpenCV의 Adaboost 알고리즘으로 학습된 얼굴 데이터를 이용해 얼굴을 인식한 결과를 나타내었다[13][15]. 그림 9에서 사각형 영역 부분은 인식된 얼굴 영역이다. 또한 인식된 얼굴에서 Haar-like-feature 알고리즘으로 눈 영역을 검출한 결과도 함께 보여주고 있다. 눈 영역 검출시의 학습데이터 역시 OpenCV의 Adaboost 알고리즘으로 학습된 눈 영역에 대한 데이터를 이용하였다.



(a) 얼굴 영역 추출 (b) 눈 영역 검출

그림 9. Haar-like-feature를 이용한 얼굴 및 눈 영역 검출
 Fig. 9. Image of face and eye region detection by Haar-like feature

눈을 감았다가 뜰 때 사람에 따라 차이가 있지만 테스트 결과 일반적으로 7 프레임 정도가 소요된다. 따라서 촬영 영상이 눈을 감은 상태인 경우 이전 5 프레임 정도에 눈을 뜬 상태의 프레임이 대부분 포함되어 있다. 영상에서 눈이 감았는지 떠있는지에 대한 상태를 판단하기 위해서 템플릿 매칭을 사용하였다. 뜬 눈 사진과 감은 눈 사진을 미리 저장되어 있는 상태에서 이미지에 Haar-like-feature를 통해 얻은 얼굴 데이터 영역의 범위로 템플릿 매칭을 수행한다. 얼굴 데이터 영역 내부에 뜬 눈과 감은 눈의 템플릿 매칭을 수행해 해당 영역내의 가장 큰 유사도를 구한다. 뜬 눈의 유사도와 감은 눈의 유사도를 비교하여 상대적으로 유사도가 큰 값으로 눈의 상태를 판단한다. 그림 10은 일반적인 환경, 안경을 쓴 경우, 뒤에 다른 사람이 있는 환경 등에 있어서 얼굴 영역 및 눈 영역을 인식하고 눈 영역에 대하여 템플릿 매칭을 이용하여 영상의 유사도를 얻고 눈을 뜬 상태의 유사도가 가장 높은 프레임을 선택하여 촬영된 사진을 보정한 결과를 보여준다. 각 영상에서 1번과 4번 그리고 1번 프레임 영상을 각각 선정하여 영상을 보정하였다.

수정 전	버퍼	버퍼	버퍼	버퍼	버퍼	보정된
촬영 영상	프레임1	프레임2	프레임3	프레임4	프레임5	촬영 영상
뜬 눈	0.74436	0.73362	0.71979	0.69991	0.70648	1번 프레임
유사도	0.71900	0.72644	0.72869	0.73187	0.72769	4번 프레임
유사도	0.72896	0.70865	0.72288	0.72537	0.71564	1번 프레임
감은 눈	0.69804	0.69486	0.76534	0.82313	0.77377	
유사도	0.65863	0.66315	0.66442	0.67872	0.73296	
유사도	0.66762	0.65362	0.67705	0.72727	0.79122	

그림 10. 눈의 상태 인식 및 영상 자동 보정 결과
 Fig. 10. Results of eye state recognition and automatic compensation

본 논문에서 개발한 기능의 테스트 과정에서 여러 명이 동시에 찍힌 사진에서 눈의 상태를 인식할 때에는 일부 오류가 발생하는 경우가 있었다. 이것은 눈의 상태 인식에 있어서 템플릿 매칭을 수행할 때 템플릿 영상을 실시간 저장해야 하는 문제로 여러 사람의 얼굴을 동시에 수정하는 것에 있어서는 일부 문제가 있었다. 이 문제는 향후 다양한 얼굴에 대한 눈 영역의 특징을 분석하여 Adaboost 알고리즘 등을 이용하여 학습데이터로 미리 저장하여 이를 활용함으로써 보완할 수 있을 것으로 판단된다. 또한 일부 조명이나 어두운 경우에서 오인식이 발생하는 경우가 있었다.

그림 11은 본 논문에서 구현한 기능을 이용하여 다양한 형태의 얼굴에 대하여 눈의 상태를 인식하여 얼굴을 보정한 테스트를 수행한 결과이다. 대상자들의 눈 상태를 인식하여 얼굴을 정상적으로 보정된 결과들을 보였다.



그림 11. 다양한 형태의 얼굴에 대한 자동 보정 결과
 Fig. 11. Automatic compensation results for various face type

V. 결론

본 논문은 디지털 카메라를 이용하여 촬영된 인물 영상의 감은 눈 상태를 인식하여 이전 프레임 영상에서 눈이 감기지 않은 영상을 찾아서 이를 이용하여 촬영한 영상을 자동으로 보정하는 기능을 구현하였다. 촬영 영상과 이전 프레임 영상을 입력으로 하여 Haar-like-feature와 Adaboost 학습 데이터를 이용하여 얼굴 영역과 눈 영역을 검출하였으며, 템플릿매칭 방법으로 눈의 상태를 인식하였다. 촬영 영상의 눈 상태가 눈을 감은 영상인 경우 이를

눈을 뜬 상태의 눈으로 자동으로 보정을 하는데, 이때 사진을 촬영하는 순간 이전 몇 개의 프레임 영상을 이용하여 보정을 수행한다. 본 연구의 구현 기능을 테스트한 결과 여러 가지 다양한 환경에서 촬영한 얼굴 영상을 문제없이 보정할 수 있었다. 단, 눈의 상태 인식에 있어서 템플릿 매칭을 수행할 때 템플릿 영상을 실시간 저장해야 하는 문제로 여러 사람의 얼굴을 동시에 수정하는 것에 있어서는 일부 문제가 있었다. 이에 대하여 향후 눈 영역의 특징에 대한 학습데이터를 획득하여 이용하는 방법으로 알고리즘에 대한 보완을 추가적으로 진행해 나갈 예정이다. 본 연구결과는 인식기술을 바탕으로 한 다양한 콘텐츠 개발연구에 응용할 수 있으며, 더 나아가 보안, 자동차 연구, 사용자 인터페이스 기술 등 여러 분야에 폭넓게 활용할 수 있을 것으로 판단된다.

참고 문헌

[1] 한동일, "실시간 얼굴 검출 기술 연구 동향," IDEC NewsLetter, pp. 08~13, 2011.

[2] 박찬우, "텍스처 기반의 눈 검출 기법," 한양대학교 대학원, 석사학위 논문, 2007. 08.

[3] H.Bay, T.Tuytelaars, and L.V.Gool, "Surf: Speeded up robust features," In Proceedings of the ninth European Conference on Computer Vision, 2006.

[4] 윤현섭, 한영준, 한현수, "길러 불변 특징과 광역 특징을 갖는 확장 SURF 알고리즘," 전자공학회 논문지, 제 46권 sp편 제 6호, pp. 58~67, 2009.

[5] Meng You, Jong-Seok Lim, Wook-Hyun Kim, "Panoramic Image Stitching using SURF," 한국신호처리·시스템학회논문지, v.12, no.1, pp.26~32, 2011.

[6] 장철희, 이기성, 조근식, "평면 호모그래피 정확도 향상을 위한 제약만족문제(Constraint Satisfaction Problem) 기반의 RANSAC 알고리즘," 정보과학회논문지 : 소프트웨어 및 응용 제39권 제11호, pp. 876~888 2012. 11.

[7] Homography, <http://en.wikipedia.org/wiki/Homography>, 2012.

[8] Paul Viola, Michael J. Jones, Fast Multi-view Face Detection, Mitsubishi Electric Research Laboratories, TR2003-096, 2003. 08.

[9] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In Proc. of IEEE Conference on Computer Vision and Pattern Recognition, Kauai, HI, pp. 1~9, 2001. 12.

[10] Philip Ian Wilson, John Fernandez, "FACIAL FEATURE DETECTION USING HAAR CLASSIFIERS" JCS 21, 4, pp.127~133, 2006. 04.

[11] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," In Proc. ICIP(1), pp.900~903, 2002.

[12] Chau, M. and Betke, M., 2005, Real Time Eye tracking and Blink Detection with USB Cameras,

Boston University Computer Science Technical Report No. 2005-12, 2005.

[13] OpenCV Computer Vision Application Programming, <http://opencv.org/>, 2013.

[14] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, "Color transfer between images," IEEE Computer Graphics and Applications, vol.21, no.5, pp.34~41, Sep./Oct. 2001.

[15] Gary Bradski, Adrian Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library, O'Reilly Media, 1st edition, 2008. 1.

[16] 신흥섭, 김진일, "디지털 카메라의 인물사진 보정 알고리즘에 관한 연구", 한국신호처리·시스템학회 2013 하계학술대회 논문집, 14권 1호, pp.87~88, 2013.

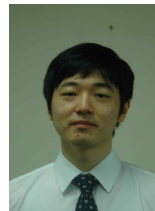


전 영 준 (Young-Joon Jeon)

1998년 2월 동의대 전산통계학과(이학사)
2000년 2월 동의대 전산통계학과(이학석사)
2005년 2월 동의대학교 컴퓨터공학과
(공학박사)

2005년 9월 ~ 2008년 8월 대구대학교 초빙교수
2008년 9월 ~ 현재 동의대학교 부산IT융합부품연구소
연구개발부 팀장

※주관심분야 : 임베디드시스템, 차량통신시스템



신 홍 섭 (Hong-Seob Shin)

2008년 3월 동의대 컴퓨터공학과 입학
2013년 현재 동의대학교 컴퓨터공학과
학부 4학년

※주관심분야 : 영상처리, 임베디드시스템



김 진 일 (Jin-II Kim)

正會員

1980년 2월 경희대 전자공학과(공학사)
1982년 8월 경희대 전자공학과(공학석사)
1995년 8월 서강대학교 컴퓨터공학과
(공학박사)

1982년 ~ 1983년 미국 Bon Scours Systems 연구원
1996년 ~ 1997년 미국 Purdue Univ. 전기및컴퓨터공학부
연구교수

1988년 ~ 현재 동의대학교 컴퓨터공학과 교수
※주관심분야 : 지능형시스템, 퍼지논리, 위성영상응용,
영상인식, ITS/LBS