

특집논문 (Special Paper)

방송공학회논문지 제19권 제5호, 2014년 9월 (JBE Vol. 19, No. 5, September 2014)

<http://dx.doi.org/10.5909/JBE.2014.19.5.578>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

AdaBoost를 이용한 윈도우 영상의 하위 영상 검출

길 종 인^{a)}, 김 만 배^{a)†}

Subimage Detection of Window Image Using AdaBoost

Jong In Gil^{a)} and Manbae Kim^{a)†}

요 약

윈도우 영상은 흔히 컴퓨터에서 응용프로그램을 실행하였을 때, 모니터를 통해 출력되는 화면을 의미하여, 웹페이지, 동영상 플레이어 및 여러 가지 응용프로그램을 모두 포함한다. 웹페이지는 다른 어플리케이션에 비해 다양한 종류의 정보를 다양한 형태로 전달한다. 이러한 웹페이지와 같은 윈도우 영상은 카메라로부터 획득할 수 있는 자연영상과 달리 텍스트, 로고, 아이콘 및 하위 영상과 같은 여러 가지 요소들을 포함하고 있고, 각 요소들은 서로 다른 형식의 정보를 사용자에게 전달한다. 그러나 텍스트와 영상은 정보가 다른 형태로 제공되기 때문에, 엄연히 다른 특성을 가지고 있는 요소들을 지역적으로 분리할 필요성이 있다. 본 논문에서는 윈도우 영상을 지역적인 특성에 따라 다수의 블록으로 분할한 후, 분할된 각 영역을 배경, 텍스트, 하위영상으로 분류하였다. 이러한 분류기법을 통해 분류된 하위 영상은 3D입체영상 변환, 영상 검색, 영상 브라우징등과 같은 응용을 가질 수 있다. 영상을 분류하는 방법에는 여러 가지가 존재할 수 있으나, 본 논문에서는 기계학습 기반의 알고리즘이 하위 영상 검출에도 좋은 접근법이 될 수 있음을 증명하기 위해 AdaBoost를 이용하였고, 실험결과로부터 93.4%의 검출률, 13%의 거짓 긍정률을 보임으로서, 이를 입증하였다.

Abstract

Window image is displayed through a monitor screen when we execute the application programs on the computer. This includes webpage, video player and a number of applications. The webpage delivers a variety of information by various types in comparison with other application. Unlike a natural image captured from a camera, the window image like a webpage includes diverse components such as text, logo, icon, subimage and so on. Each component delivers various types of information to users. However, the components with different characteristic need to be divided locally, because text and image are served by various type. In this paper, we divide window images into many sub blocks, and classify each divided region into background, text and subimage. The detected subimages can be applied into 2D-to-3D conversion, image retrieval, image browsing and so forth. There are many subimage classification methods. In this paper, we utilize AdaBoost for verifying that the machine learning-based algorithm can be efficient for subimage detection. In the experiment, we showed that the subimage detection ratio is 93.4 % and false alarm is 13 %.

Keyword : Subimage detection, Projection Profile Cutting, Local Binary Patterns, AdaBoost

a) 강원대학교 IT대학 컴퓨터정보통신공학과(Dept. of Computer and Communications Engineering, Kangwon National University)

† Corresponding Author : 김만배 (Manbae Kim)

E-mail: manbae@kangwon.ac.kr

Tel: +82-33-250-6395

※ 본 연구는 중소기업청의 투자연계 기술혁신개발사업 (S2056930), 지식경제부 및 산업기술평가관리원의 시스템반도체 상용화기술사업 [10041082] 및 2014년도 강원대학교 학술연구조성비 (과제번호-No. C1010783-01-01)의 지원을 받았음.

※이 논문의 연구결과 중 일부는 “3D Systems and Applications (3DSA 2014)”에서 발표한 바 있음.

· Manuscript received July 13, 2014 Revised August 18, 2014 Accepted August 18, 2014

I. 서 론

자연영상(Natural image)은 카메라로부터 촬영하여 획득할 수 있는 영상을 뜻하며, 대부분의 영상처리 및 컴퓨터비전 분야의 연구는 이러한 자연영상의 처리 및 변환 등에서부터 출발한다. 반면, 윈도우 영상(Window image)은 자연영상과 달리 컴퓨터에서 수행되는 응용 프로그램으로부터 생성된 정보를 사용자에게 이해하기 쉽게 전달하기 위해 정보를 여러 가지 형태로 생성 및 변환하여 모니터에서 출력하였을 때 나타나는 영상이다. 그럼 1에서 볼 수 있듯이, 이러한 윈도우 영상은 최대한 다양한 정보를 하나의 화면에서 출력해야 하므로, 구역을 나누어 표현해야 한다. 만일 사용자가 영상으로부터 다른 정보 및 요소를 포함하고 있는 구역을 정확히 구별할 수 있다면, 각 구역의 목적에 맞도록 적절히 차별화된 영상처리를 적용할 수 있을 것이다.

이러한 윈도우 영상의 영역 분할이 적용될 수 있는 분야로써 2D-to-3D 변환이 있다. 이 변환 기술은 2D로 제작된 영상을 실시간으로 3D로 변환할 수 있도록 고안된 기술로써 높은 관심을 받고 있으며, 최근 이러한 기술을 채택한 3DTV가 출시되어 판매되고 있다^[1-4]. 그러나 이러한 2D-to-3D 변환기술은 자연영상에 초점이 맞추어져 있다. 2D 영상을 3D로 변환하기 위해 자연영상으로부터 깊이정보를 획득하는 여러 가지 연구가 진행되어 왔는데, 주로 소실점 기반, 모션 검출 기반, 전역 깊이 추정 기반 등의 깊이정보 추출 알고리즘이 연구되어왔다. 이러한 2D-to-3D 변환기술

은 꾸준히 발전하여 신뢰할 수 있을만한 결과를 볼 수 있게 되었으며, 실제로 이러한 기술을 통해 변환된 입체영상은 3DTV를 통해 올바른 입체감을 느낄 수 있다. 그러나, 이러한 변환기술은 자연영상에 초점이 맞추어져 있으므로, 윈도우 영상을 변환하였을 경우, 깊이왜곡의 원인이 된다. 깊이왜곡을 발생시키는 주된 원인으로 텍스트를 예로 들 수 있는데, 이외에도 작업 표시줄, 제목 표시줄 및 스크롤바와 같이 주관적인 사람의 눈으로도 깊이를 예측하기 어려운 경우, 부적절한 깊이가 할당되면 심각한 깊이왜곡이 발생하게 된다. 이러한 깊이 왜곡은 시각적 피로도를 유발하여 입체영상 시청시에 큰 불편함을 초래하게 된다. 이를 위해 가장 좋은 방법은 입체변환이 불필요한 부분은 변환하지 않는 것이다. 실제로 배경 및 텍스트를 강제로 변환하게 되면 많은 왜곡이 발생하게 되므로, 3D로 변환하지 않는 것이 좋다.

전술한 바와 같이 2D-to-3D 변환기술은 주로 자연영상에 초점이 맞추어져 있다. 반면 윈도우 영상의 변환기술은 연구된 바 없으므로 새로운 접근법이 필요하다. 이를 위해 먼저 윈도우 영상으로부터 하위 영상(Subimage)을 탐지해야 한다. 기존에 문서, 영화, 표지판 등으로부터 텍스트를 검출하는 연구는 많이 진행되어 왔다. M. Khodadadi^[5]는 비디오로부터 텍스트를 검출하는 연구를 진행하였다. 이를 위해 영상으로부터 그래디언트를 검출한 후, stroke filter, density filter, dilation 등을 차례로 적용하여 초기 텍스트 영역을 검출하였다. 이후, 분할과 병합을 반복하여 최적의 텍스트 영역을 찾는 방법을 제안하였다. Y. Song^[6]의 연구에서

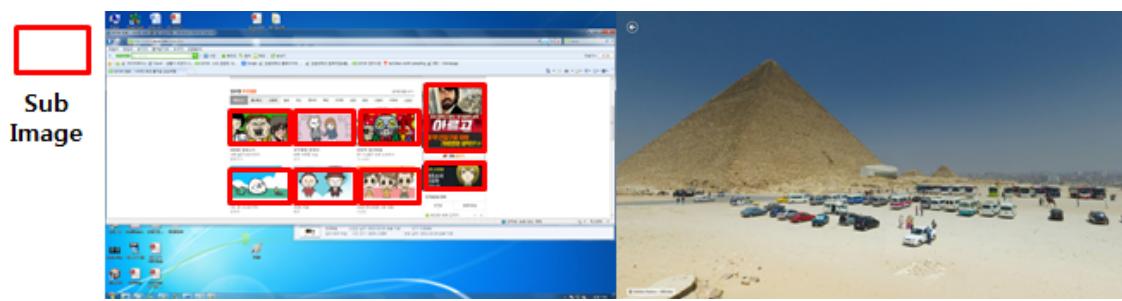


그림 1. 윈도우 영상과 자연 영상의 비교. (a) 윈도우 영상 및 (b) 자연 영상
Fig. 1. The comparison between window and natural images. (a) window image and (b) natural image

는 역시 비디오로부터 자막을 검출하기 위해 에지를 이용하였다. 그러나 자막의 크기는 각각 다르기 때문에, 에지 영상을 각각 업샘플링과 다운샘플링하여 처리한 후, 병합하는 방법으로 접근하였다. 이후에 텍스트 영역을 개선하기 위해 에지 영상으로부터 프로젝션 프로파일을 측정한 후, 프로파일로부터 valley/peak 점을 찾아서 교정하는 방법을 사용하였다. 그러나 M. Khodadadi와 Y. Song의 연구는 영상으로부터 텍스트를 검출하는 것이 목적인 반면, 본 연구에서는 윈도우 영상으로부터 하위 영상을 탐지하는 것이 목적이이다. 따라서 본 논문에서는 윈도우 영상만이 가지는 고유한 특성을 이용하여 하위 영상을 검출하는 방법을 제안한다. Y. Song에서 사용한 프로젝션 프로파일은 텍스트 검출을 위해 사용되었으나, 여기서는 하위 블록 분할을 위해 사용되었다.

윈도우 영상에 존재하는 하위 영상은 몇 가지의 특성을 갖는다. 먼저 하위 영상은 대부분 사각형의 형태를 갖는다. 사각형의 경계를 갖지 않거나 모호한 경계를 갖는 하위 영상에 대해서는 자동으로 사각형의 형태를 갖는 경계 상자(Bounding box)를 생성한다. 이를 위해 윈도우 영상을 다수의 블록(block)으로 분할한다. 정확한 경계를 갖는 하위 영상들로부터 획득한 정보를 이용하여 경계가 정확하지 않은 하위 영상까지 경계를 설정할 수 있게 해준다. 분할된 각 블록은 크게 배경, 텍스트, 하위 영상으로 나눌 수 있다. 또한 하위 영상은 복잡한 텍스쳐(Texture)를 갖는다. 이러한 특성을 이용하면 쉽게 판별이 가능하나, 복잡하지 않는 텍스쳐를 갖는 하위 영상이 존재할 수 있고, 블로그와 같은 웹페이지의 경우 배경에서 텍스쳐를 가지고 있는 경우도 존재한다. 따라서 윈도우 영상을 다수의 블록으로 분할한 후, 각 블록을 AdaBoost를 이용하여 분류함으로써 해결하고자 하였다. 이때, 배경은 큰 의미를 갖지 않고 손쉽게 탐지가 가능하므로 본 논문에서는 고려하지 않았다.

본 논문의 구성은 다음과 같다. 먼저 II장에서는 전체 흐름도를 통해 제안하는 시스템을 자세히 소개한다. III장에서는 윈도우 영상을 여러 개의 하위 블록으로 분할하는 방법을 제안하고, 분할된 각 하위 블록을 텍스트 또는 하위 영상으로 분류하는 방법을 IV장에서 설명한다. 이후, V장에서는 II, III, IV장에서 제안한 방법을 토대로 수행한 실험 결

과를 보여주고, 마지막으로 VI장에서 결론을 맺는다.

II. 제안 방법

본 논문에서 제안하는 방법은 3가지 단계로 구성되어 있다. 첫째로 윈도우 영상으로부터 그래디언트 영상(Gradient map)을 생성한다. 둘째로 PPC(Projection Profile Cutting) 기법을 이용하여 다중 블록(Multi block) 분할을 수행한다. 세 번째 단계에서는 AdaBoost를 이용하여 영상 분류를 수행한다. 다음 그림 2는 제안 방법의 전체 흐름도를 보여준다. 마지막에는 검출된 하위 영상을 입체영상으로 변환해야 한다. 그러나 하위 영상의 입체변환은 기존의 다양한 변환 기술들을 적용할 수 있기 때문에 본 논문에서는 다루지 않는다.

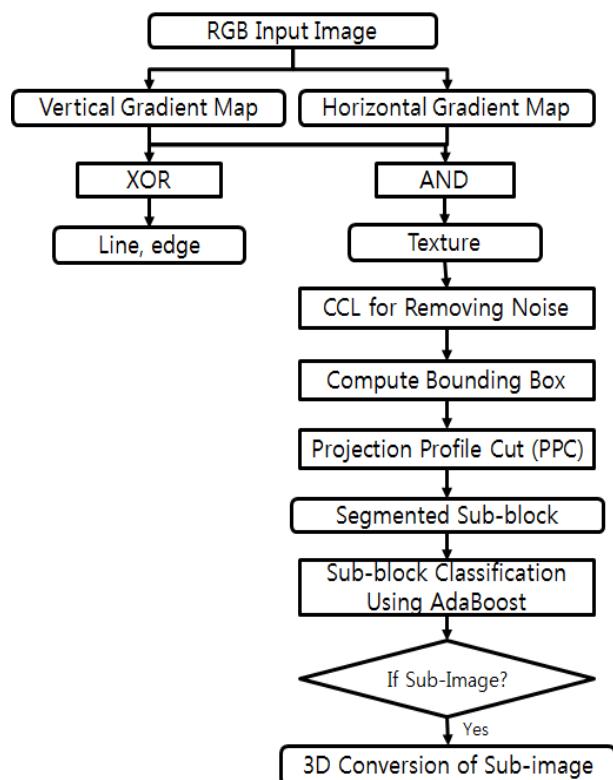


그림 2. 제안 방법의 전체 흐름도

Fig. 2. The overall block diagram of proposed method

III. 하위 블록 분할

윈도우 영상으로부터 하위 영상을 탐지하기 위해서, 전체 영상을 여러 개의 하위 블록으로 분할해야만 한다. 윈도우 영상을 적절히 분할하기 위해서는 먼저 윈도우 영상만의 고유한 특성을 알아야 할 필요가 있다. 윈도우 영상은 자연 영상과 달리 수평 및 수직선을 많이 포함하는 특징을 가지고 있다. 그림 3에서 보는 바와 같이 윈도우 영상은 각 공간의 역할을 구분하기 위해 선으로 경계를 나누고 있다. 또한 영상 내부에 존재하는 하위 영상도 사각형의 형태를 가지고 있기 때문에, 수평 및 수직선을 많이 포함하고 있다는 가정은 타당하다. 따라서, 이러한 특성을 활용한다면 효율적인 하위 블록 분할이 가능하다.

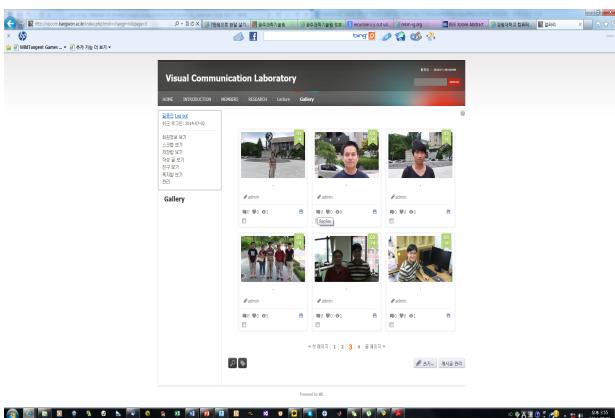


그림 3. 하위 영상 및 텍스트 영역은 수평 및 수직선으로 구분되어 있는 윈도우 영상의 예

Fig. 3. The example of window image containing the subimage and text regions divided by vertical and horizontal lines

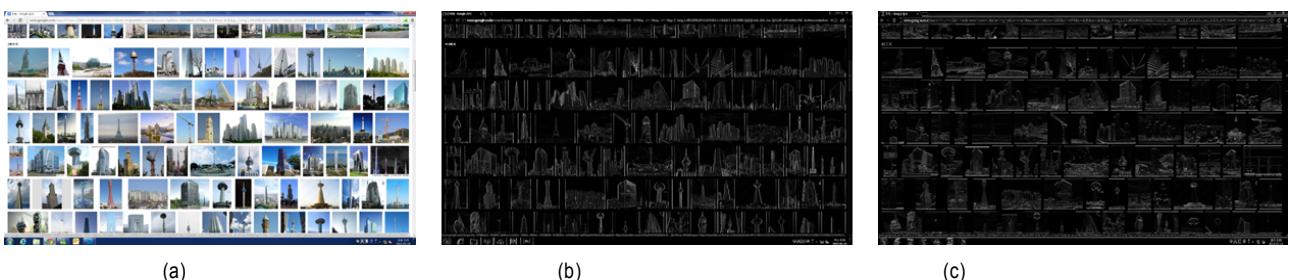


그림 4. 입력 컬러 영상과 생성된 그래디언트 영상. (a) 윈도우 영상, (b) 수평 그래디언트 영상 f_H 및 (c) 수직 그래디언트 영상 f_V

Fig. 4. Input color image and generated gradient maps. (a) window image, (b) horizontal gradient image f_H and (c) vertical gradient image f_V

1. 그래디언트 검출

그래디언트는 각 영역의 경계를 검출하기 위한 좋은 방법이 될 수 있다. 입력 영상은 컬러 영상이므로, R, G, B의 세 채널로 구성되어 있다. 먼저 다음 식 (1)을 이용하여 각 채널에 대해 수평 및 수직 방향의 그래디언트를 측정한다. 그 후 세 채널에서 가장 큰 값을 갖는 그래디언트를 대푯값으로 선택한다.

$$\begin{aligned} f_H(x,y) &= I^c(x+1,y) - I^c(x-1,y) \\ f_V(x,y) &= I^c(x,y+1) - I^c(x,y-1) \end{aligned} \quad (1)$$

위 식에서 (x, y) 는 영상의 2차원 좌표, $c \in \{R, G, B\}$ 는 컬러 채널을 의미한다. 위 식을 통해 2개의 그래디언트 영상을 획득할 수 있다. 다음 그림 4는 생성된 두 그래디언트 결과 영상을 보여준다.

하위 영상은 내부에 복잡한 텍스쳐를 가지고 있는 반면, 배경 및 텍스트 영역은 상대적으로 적은 양의 텍스쳐를 포함한다. 따라서 위의 그래디언트 영상을 이용하면 텍스쳐 픽셀을 추출할 수 있다. 이를 위해 생성된 두 그래디언트 영상을 하나로 병합한다. 이때, 고려할 수 있는 방법은 두 가지가 존재한다. 첫째는 AND 연산이고 둘째는 XOR 연산이다. AND 연산자를 적용하게 되면 수평/수직 방향으로 모두 그래디언트를 가지고 있어야 하므로 이러한 텍스쳐를 모두 추출할 수 있다. 반면에, XOR 연산자는 수평 혹은 수직 방향 중 하나의 방향으로만 그래디언트를 가져야 한다. 이러한 픽셀은 하위 영상의 경계일 가능성이 크다. 다음 그림 5는 XOR과 AND 연산자를 이용하여 병합한 마스크 영

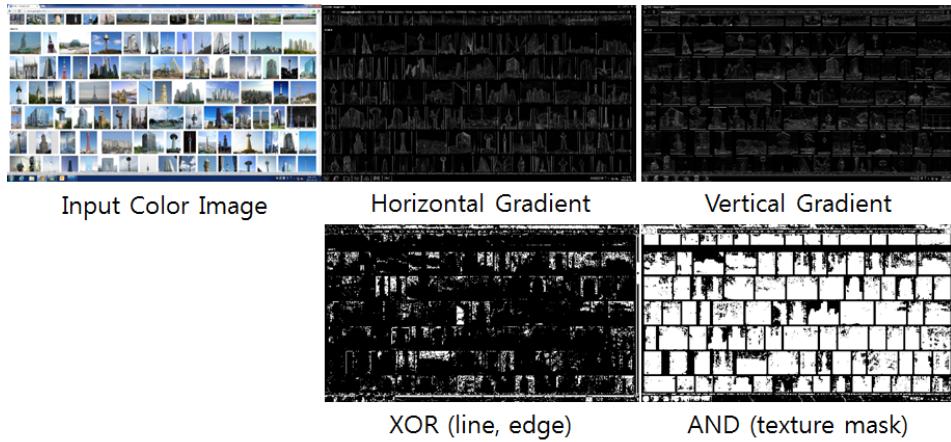


그림 5. 두 그래디언트 영상과 XOR, AND 연산자를 이용하여 병합한 이진 영상

Fig. 5. Two gradient maps and merged gradient mask using XOR and AND operators

상을 보여준다. 식(2)에서 \otimes 는 AND 연산을 의미한다.

$$f(x, y) = f_H(x, y) \otimes f_V(x, y) \quad (2)$$

XOR 마스크 영상은 수직/수평 방향을 갖는 그래디언트 픽셀을 나타낸다. 이러한 특성은 윈도우 영역 분할을 위해 좋은 특징이 될 것으로 보인다. 그러나 실제로 명확한 경계를 갖지 않는 하위 영상이 존재할 수 있고, 하위 영상의 경계가 아닌 내부 및 외부에서도 발견될 수 있으므로, 신뢰할 만한 정보가 될 수 없다. 반면, AND 연산자는 온전히 하위 영상 내부의 텍스쳐를 검출해 낼 수 있다. 이러한 텍스쳐 마스크를 이용하면 XOR 마스크 영상의 단점이었던 명확한 경계를 갖지 않는 하위 영상도 검출이 가능하다. 또한, AND 연산자는 텍스트 영역에서 최대한 검출을 억제함으로써, 이후의 단계에 큰 영향을 미치지 못한다. 따라서 본 논문에서는 AND 연산자를 이용하여 획득한 텍스쳐 마스크를 이후의 단계에서 사용하였다.

2. Projection Profile Cutting (PPC)^[7,8]

PPC (Projection Profile Cutting) 기법은 주로 문서 영상에서 텍스트 영역 검출을 위해 사용되었던 방법이다. 이미 앞의 과정을 통해 배경 영역이 제거되고, 하위 영상이 될 수 있는 후보 영역만 검출되었다. 이로부터 다음 식 (3)을

이용하여 텍스쳐 마스크 영상의 수평 및 수직 프로파일을 측정한다. 그림 6은 식 (3)으로부터 측정된 텍스쳐 마스크 영상의 수직/수평 프로파일을 보여준다.

$$\begin{aligned} p_H(y) &= \sum_x f(x, y) \\ p_V(x) &= \sum_y f(x, y) \end{aligned} \quad (3)$$

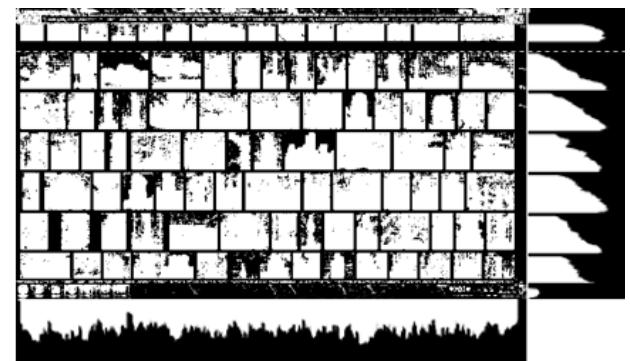


그림 6. 텍스쳐 마스크 영상의 수직 및 수평 프로파일

Fig. 6. Vertical and horizontal profiles of texture mask

식 (3)에서 f 는 AND 연산자를 이용하여 병합된 텍스쳐 마스크 영상이고, p_H , p_V 는 각각 수평 및 수직 프로파일을 의미한다. 그림 6에서 볼 수 있듯이, 하위 영상의 영향으로 인해 불분명한 수직 및 수평 프로파일을 획득하였다. 따라

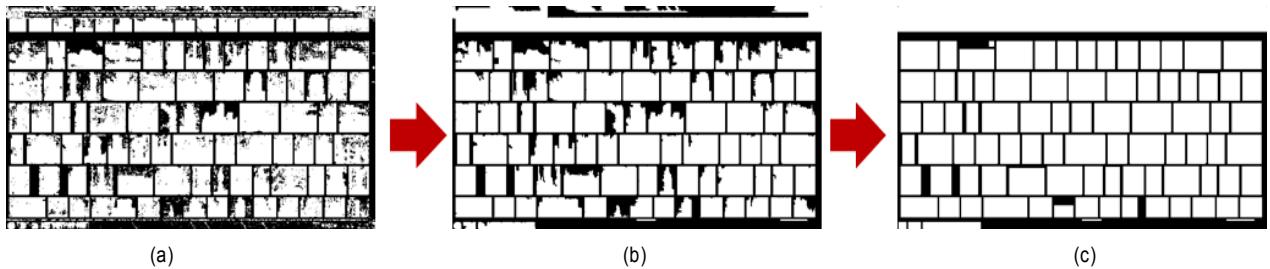


그림 7. 텍스쳐 마스크로부터 경계상자를 획득과정. (a) 텍스쳐 마스크, (b) 노이즈 제거 및 (c) CCL을 이용한 경계상자 생성
 Fig. 7. The process flow obtaining bounding box from texture mask. (a) texture mask, (b) noise removal and (c) bounding box from CCL

서 조금 더 명확한 프로파일을 생성해야 할 필요가 있다. 먼저 텍스쳐 마스크 영상에서 레이블링을 수행한다. 레이블링 알고리즘 역시 여러 가지 존재하지만 본 논문에서는 CCL(Connected Component Labeling)을 사용하여 레이블링하였다^[9]. 이진 영상에서 서로 연결되어 있는 픽셀의 집합을 연결 성분(Connected Component)이라 한다. CCL은 같은 연결 성분에 속하는 픽셀에 같은 레이블을 할당하고 다른 연결 성분에는 서로 다른 레이블을 할당하는 과정이다. CCL을 이용하여 이진 영상에서 레이블링을 수행하게 되면, 특정 레이블에 속한 픽셀의 수를 알 수 있다. 만일 특정 레이블에 할당된 픽셀의 수가 임계치 이하이면 제거하는 방식으로 노이즈 성분으로 판단되는 픽셀들을 제거하였다. CCL을 통해 불필요한 픽셀들을 제거한 결과를 그림 7(b)에서 보여주고 있다. 제거되지 않고 남겨진 레이블에 대해서는 경계 상자 (Bounding box)를 생성하고, 생성된 경계 상자의 내부를 채운다. 그 결과를 그림 7(c)에서 보여주고 있다. 그림 8에서는 교정된 텍스쳐 마스크 영상을 이용하여 프로파일을 측정한 결과를 보여준다.

그림 8에서 볼 수 있듯이 실험에 사용된 테스트 영상은 수직 방향으로 정렬이 되어있다. 따라서, 수평 프로파일을 이용하면 영상을 손쉽게 수직방향으로 분할할 수 있다. 그러나 수직 프로파일의 경우 정렬이 되어 있지 않기 때문에 프로파일만을 이용하여 분할하기 어렵다. 그러므로, 이러한 영상에 대해서는 먼저 수직방향으로 분할하는 것이 옳다. 수직방향으로 분할된 후, 각 분할된 모든 영역에 대해 수직 프로파일을 측정한 후, 수평방향으로 분할해 나가면 된다. 그러나 모든 경우에 대해 수평방향으로 먼저 분할하는 것은 타당하지 않으므로, 먼저 두 방향에 대해 모두 프로

파일을 측정한다. 그리고 각 프로파일에 대해 다음 식 (4), (5)와 같은 변화량을 측정한다.

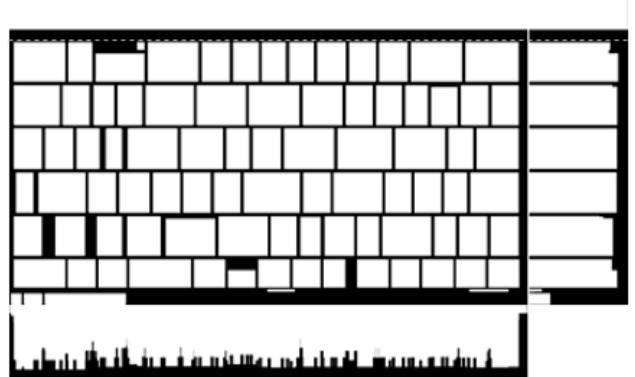


그림 8. 교정된 텍스쳐 마스크 영상의 수직 및 수평 프로파일
 Fig. 8. Vertical and horizontal profiles of rectified texture mask

$$g_H = \begin{cases} 1 & \text{if } p_H(i) - p_H(i-1) > \text{Width} \times T \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$$g_V = \begin{cases} 1 & \text{if } p_V(j) - p_V(j-1) > \text{Width} \times T \\ 0 & \text{otherwise} \end{cases}$$

$$c_H = \sum_{i=1}^{\text{Height}} g_H(i) \quad (5)$$

$$c_V = \sum_{j=1}^{\text{Width}} g_V(j)$$

식 (4), (5)에서 p_H, p_V 는 식 (3)에서 계산된 프로파일이며, 임계값 T 는 $[0, 1]$ 의 값을 갖는다. c_H, c_V 는 각각 프로파일의 수직/수평 변화량을 의미하며, 큰 값을 갖는 방향부터 분할하면 된다. 즉, c_H 가 크면 수평방향으로 먼저 분할한 뒤, 수직방향으로 분할한다.

IV. 영상 분류

분할된 각 하위 블록을 배경, 텍스트 및 하위 영상으로 분류해야 한다. 본 논문에서는 AdaBoost + LBP를 이용한 기계학습 기반의 분류를 사용하였다. 기존에 HOG, SIFT, SURF, Harr, ICF등의 여러 가지 영상 특징 검출 방법이 개발되어있으나 이러한 영상 특징들은 대부분 템플릿 매칭 (Template matching) 방법에 기반하고 있으며, 이는 영상의 그래디언트의 방향 및 크기와 같은 정보를 이용하여 기하학적인 정보를 유지함으로써 원하는 영상을 검출하게 된다. 이는 주로 객체 검출 및 추적을 위해 사용되며 여러 문헌을 통해 성능이 검증되었다. 그러나, 본 연구에서는 특정 객체의 검출이 목적이 아니라 일반 영상과 텍스트 영상의 분류가 목적이이다. LBP는 텍스처 분석을 위해 개발된 특징으로써, 텍스처 정보는 객체 인식에 유용하게 사용될 수 있다. 따라서 LBP는 본 연구와 가장 적합한 특징 기술자라고 할 수 있다.

1. Local Binary Patterns (LBP)^[10]

AdaBoost 분류기 학습을 위해 LBP를 특징값으로 이용한다. LBP는 텍스처 분석을 위해 주로 사용되는 기법으로서, 계산적으로 효율적인 텍스처 기술자이다. LBP 연산자는 영상의 각 픽셀에 레이블을 할당한다. LBP는 다음 식 (6)을 이용하여 획득할 수 있다.

$$LBP(x,y) = \sum_{p=0}^{P-1} s[f(x,y) - f(x_p, y_p)] \cdot 2^p \quad (6)$$

이때, P 는 이웃한 픽셀의 개수이고, 3x3의 window를 이

용한다면, $P=8$ 이 된다. $f(x,y)$ 는 중심 픽셀, $f(x_p, y_p)$ 는 인접한 이웃 픽셀의 밝기값을 의미하며, $s(z)$ 는 임계치 함수로써 다음 식 (7)과 같이 정의된다.

$$s(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases} \quad (7)$$

위의 식 (6), (7)을 조금 더 직관적으로 설명하면 다음과 같다. 레이블은 이웃한 3x3 픽셀에 의해 결정된다. 만일 이웃한 픽셀이 중심 픽셀보다 크면 “1”的 레이블을 할당하고, 이웃한 픽셀이 중심 픽셀보다 작으면 “0”的 레이블을 할당 한다. 이웃한 8개의 픽셀에 대해 8개의 이진부호를 획득한 후, 이를 시계 방향으로 나열하여 8비트로 표현한다, 그리고 이를 [0, 255]의 정수값으로 변환하면 변환된 값이 중심 픽셀의 LBP값이 된다. 다음 그림 9에서 보는바와 같이, 이웃한 8개의 픽셀에 대해 이진부호를 계산하면 10001111되고 이는 143의 정수값으로 변환할 수 있다.

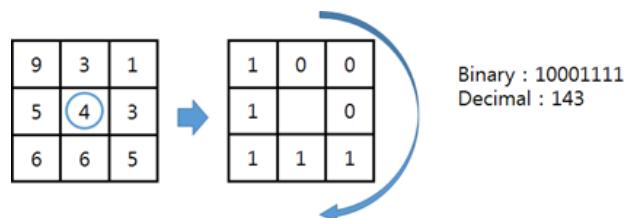


그림 9. LBP로 변환된 밝기값
Fig. 9. Transformation from intensity to LBP

분할된 각 하위 블록에 속한 모든 픽셀을 LBP값으로 변환하면 다음 그림 10과 같은 결과를 획득할 수 있다. 이러한 LBP값을 이용하여 구성된 히스토그램이 해당 하위 블록의 특징이 된다. 따라서 각 하위 블록은 256개의 특징을

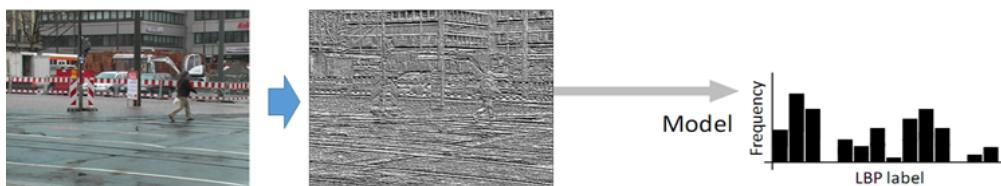


그림 10. 입력 영상으로부터 획득한 LBP 영상과 히스토그램
Fig. 10. The LBP image and histogram obtained from input image

갖게 된다.

ULBP는 LBP의 변형으로써, LBP의 256개의 특징 중에서 특별한 의미를 갖는 패턴만을 선택하고 이외의 무의미한 패턴을 고려하지 않음으로써, 특징의 차원을 감소시킨 기술자이다^[11]. ULBP는 8비트로 표현된 LBP값에서 0→1 혹은 1→0으로 변하는 구간의 수를 측정하여 2를 넘지 않는 패턴만을 선택한다. 256개의 패턴 중 58개의 패턴이 선택될 수 있으며, 나머지 198개의 패턴은 동일한 패턴으로 취급되어 UBL는 총 59개의 패턴을 갖게 된다.

2. AdaBoost

AdaBoost는 다수의 약 분류기를 결합하여 강 분류기를 생성하는 방법이다. 각 약 분류기는 LBP로부터 추출한 특징을 이용하여 학습된다. 그림 11은 이러한 AdaBoost의 개념을 도식화하였다.

학습 단계에서는 다수의 자연 영상 샘플과 텍스트 영상 샘플로부터 추출한 LBP 특징을 입력으로 받아 약 분류기를 학습한다. 이때, 정확히 인식된 샘플에 대해서는 가중치를 감소시키고, 오인식된 샘플에 대해서는 가중치를 증가시키는 과정을 반복함으로써 학습하게 된다. N개의 약 분

류기를 조합하여 최종적으로 하나의 강 분류기를 생성하면 훈련 단계는 종료된다. 테스트 단계에서는 앞의 단계에서 분할한 각 하위 블록을 입력으로 받게 되고, 하위 블록으로부터 LBP 특징을 추출한다. 강 분류기는 입력된 하위 블록의 LBP 특징을 분석하여 자연 영상인지 텍스트 영상인지지를 판단하게 된다. AdaBoost는 P. Viola^[12,13]의 논문에서 자세히 설명하고 있다.

V. 실험 결과

테스트를 위해 사용된 영상은 1920x1080의 해상도를 갖는 PC에서 웹브라우저를 실행한 상태에서 수집하였으며. 그림 12는 이러한 과정을 통해 분할된 윈도우 영상의 결과이다.

AdaBoost의 훈련을 위한 텍스트 및 자연 영상 샘플은 웹페이지 및 포토샵과 같은 툴을 이용하여 수집 또는 제작하였으며, 각 훈련 샘플은 모두 다른 해상도를 갖는다. 본 연구에서 텍스트 샘플은 533개, 영상 샘플은 총 648개가 사용되었다. 일반적으로 훈련 샘플의 수가 많을수록 AdaBoost의 성능은 향상된다고 알려져 있으나, 텍스트 샘플의 경우,

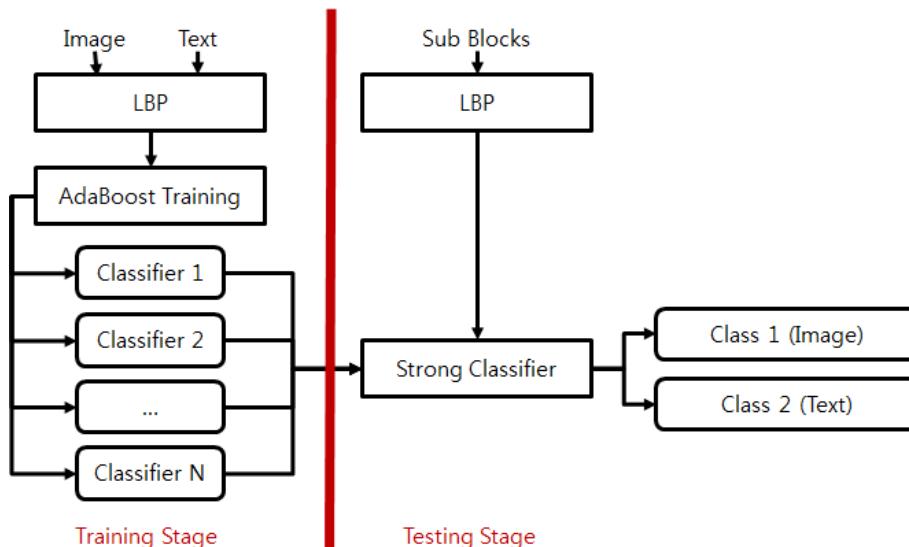


그림 11. AdaBoost 처리의 개념도
 Fig. 11. Conceptual scheme of AdaBoost

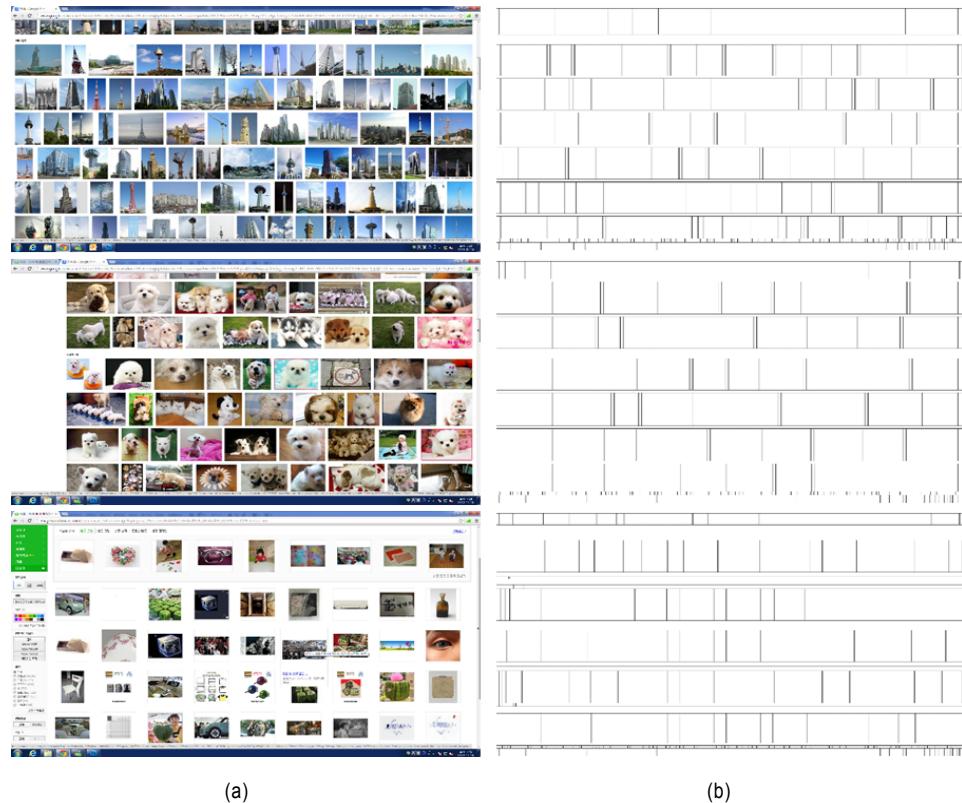


그림 12. 입력 윈도우 영상과 영역 분할 결과. (a) 입력 영상, 및 (b) 분할 영상
Fig. 12. Input window images and the result of decomposition. (a) input image and (b) decomposed image

글자의 모양, 크기 및 색상 등에 따라 여러 가지 형태를 가질 수 있지만, 과도하게 많은 샘플은 중복을 발생하게 한다.

따라서 대략 500개의 텍스트 샘플로 글자의 모양 및 색상을 대부분 표현할 수 있다. 또한 텍스트 샘플과 이미지 샘플의



(a)



(b)

그림 13. AdaBoost의 훈련 집합의 예. (a) 텍스트 집합 및 (b) 영상 집합

Fig. 13. The examples of training samples for AdaBoost. (a) text sample and (b) image sample

개수를 비슷하게 사용한다면 분류 결과를 신뢰할 수 있다.
 수집된 훈련 샘플을 그림 13에서 보여주고 있다.

다음 그림 14에서는 본 논문에서 제안한 방법을 통해 검출한 하위 영상의 결과를 보여주고 있다. 파란색으로 표시

된 영역이 검출된 하위 영상이며, 녹색으로 표시된 영역은 배경 및 텍스트 영역을 의미한다.

표 1에서는 실험 결과로부터 획득한 하위 영상 검출률을 나타내고 있다. 총 20장의 실험영상에 대해 제안 방법을 적



그림 14. 실험 결과 (a),(c) 입력 영상, (b),(d) 탐지된 하위 영상 (파랑: 하위 영상 / 녹색: 배경 및 텍스트)
 Fig. 14. Experimental results (a),(c) input images, (b),(d) detected subimage (blue: subimage / green: background and text)

표 1. 하위 영상 검출률

Table 1. Subimage detection ratio

Test Image	No. of Blocks	No. of Sub Images	No. of Non-Sub Images	Detected Images	False Alarm	False Positive Rate (%)	Detection Rate (%)
1	140	10	130	9	1	10% (1/10)	90% (9/10)
2	97	19	78	19	0	0% (0/19)	100% (19/19)
3	138	9	129	9	7	43.8% (7/16)	100% (9/9)
4	33	2	31	2	0	0% (0/2)	100% (2/2)
5	133	53	80	53	0	0% (0/53)	100% (53/53)
6	194	89	105	88	0	0% (0/88)	99.9% (88/89)
7	358	30	328	19	0	0% (0/19)	63.3% (19/30)
8	132	41	91	40	8	16.7% (8/48)	90.2% (37/41)
9	142	6	136	6	4	40% (4/10)	100% (6/6)
10	139	10	129	7	1	12.5% (1/8)	70% (7/10)
11	66	2	64	2	1	33.3% (1/3)	100% (2/2)
12	83	17	67	15	2	11.8% (2/17)	88.2% (15/17)
13	83	7	76	6	5	45.5% (5/11)	85.7% (6/7)
14	90	2	88	2	0	0% (0/2)	100% (2/2)
15	128	13	115	13	5	27.8% (5/18)	100% (13/13)
16	38	2	36	1	4	80% (4/5)	50% (1/2)
17	138	7	131	6	0	0% (0/6)	85.7% (6/7)
18	87	2	85	2	0	0% (0/2)	100% (2/2)
19	54	12	42	12	9	42.9% (9/21)	100% (12/12)
20	77	1	76	1	0	0% (0/1)	100% (1/1)
Sum	2,351	334	2,017	312	47	13% (47/359)	93.4% (312/334)

용하였고, ground-truth는 사람의 눈으로 직접 판단하여 생성하였다. 표 1에서 2, 3, 4열은 ground-truth를 의미한다. 2열의 No. of Blocks는 윈도우 영상에 존재하는 총 하위 블록의 수를 의미하고, 3열의 No. of Subimages는 하위 영상의 수, 4열의 No. of Non-Subimages는 비 하위 영상의 수를 나타내므로, 3열과 4열의 값을 더하면 2열의 총 하위 블록의 수와 일치한다.

5열의 Detected Images는 제안방법을 통해 검출된 하위 영상의 수를 나타내고, 6열의 False Alarm은 제안방법을 통해 검출되었지만 실제 하위영상이 아닌 블록의 수를 나타낸다. 7열의 거짓 긍정률(FPR)과 8열의 검출률(DR)은 다음 식 (8)을 이용하여 측정하였다.

$$\begin{aligned} FPR &= \frac{\text{False Alarm}}{\text{Detected Images} + \text{False Alarm}} \\ DR &= \frac{\text{Detected images}}{\text{No. of images}} \end{aligned} \quad (8)$$

표 1의 하위 영상 검출률에서 보는바와 같이 제안하는 방법의 하위 영상 검출률은 93.4%를 달성하였고, 거짓 긍정률은 13%가 발생하였다. 일부 영역에서는 대부분 텍스트를 포함하지만 아이콘 혹은 작은 이미지가 포함되어 있을 수 있다. 이와 같이, 그림과 텍스트가 혼합된 경우가 발생할 수 있지만, AdaBoost에서 사용된 학습 샘플에서 텍스트와 이미지가 혼합된 샘플은 존재하지 않는다. 앞서 언급한 바와 같이 ground-truth는 사람의 눈으로 직접 판단하여 생성하였으므로, 이러한 경우는 하위 영상이 아니라고 판단하는 것이 타당하나, AdaBoost가 이를 하위 영상으로 판단하는 것은 불가피하다. 다시 말해, 텍스트와 하위 영상의 판단이 모호한 경우, AdaBoost는 하위 영상으로 판단하는 반면, 사람의 눈으로는 텍스트로 판별하였으므로 거짓 긍정이 발생하게 된다.

영상의 종류에 따라 큰 차이를 보여주는 경우도 있었는데, 그림이 수직/수평 방향으로 정렬된 영상일수록 높은 검출률을 보인 반면, 매우 작은 영상들이 혼재된 영상의 경우 검출률이 하락하였다. 이러한 영상의 경우, 실제 사람의 눈으로도 영상인지 아닌지의 여부를 판단하기 어려운 블록이 상당수 존재하였으며, 본 실험에서는 매우 작은 크기의 블

록은 입체영상으로 변환하여도 입체효과가 미미하므로, 배경으로 판단하도록 처리하였다. 또한 매우 큰 크기를 갖는 블록의 경우, 작은 크기의 이미지를 포함할 수 있다. 이는 블록의 크기에 비해 상대적으로 이미지의 크기가 작기 때문에 하위 영상이 아닌 것으로 판단하였다.

하위 영상 검출률은 하위 블록 분할에 크게 의존한다. 앞서 언급한 바와 같이, 하위 영상들이 정렬되어 있을 경우, 하위 블록 분할의 정확도는 향상되며, 이로 인해 하위 영상 검출률 역시 상승하게 된다. 그러나 하위 영상이 위치에 상관없이 불규칙하게 배열되어 있을 경우, 혹은 일관되지 않은 크기를 가지고 있는 경우 하위 영상 분할이 어렵게 된다. 본 논문에서는 수평/수직 방향으로 한번 씩 분할을 수행하였으나, 정확한 경계를 갖는 하위 영상을 탐지하기 위해 여러번 분할하는 방법이 사용될 수 있다. 즉, 수평방향과 수직 방향을 번갈아가며 재귀적으로 분할해 간다면 최적의 크기를 갖는 블록으로 분할할 수 있게 되며, 이로써 AdaBoost를 이용한 분류기의 성능 향상을 기대할 수 있다.

VI. 결론 및 향후 연구

본 논문에서는 윈도우 영상의 3D 변환을 위해 하위 영상을 탐지하는 방법을 제안하였다. 이를 위해 먼저 PPC을 이용하여 입력 윈도우 영상의 영역 분할을 수행하였다. 분할된 각 영역의 분류를 위해 ULBP를 이용하여 특징을 획득하였고, 이를 이용하여 AdaBoost 분류기를 학습하였다. 학습된 분류기를 이용하여 하위 영상을 탐지하였고, 탐지된 하위 영상은 3D로 변환된다. 검출된 하위 영상의 3D 변환 방법은 [14-16]에서 자세히 설명하고 있다. 실험 결과로부터 93.3%의 검출율을 달성하였고, 이로부터 AdaBoost를 이용한 기계학습 방법이 하위 영상 검출을 위한 효과적인 방법임을 입증하였다.

그러나 윈도우 영상에는 하단에 작업 표시줄 및 웹브라우저의 제목 표시줄 등이 존재한다. 이러한 영역들은 대부분 전체 영상의 경계에 위치하고 있으므로 처리가 용이하지 않다. 추후, 경계에 위치하는 작업 표시줄, 제목 표시줄 및 스크롤 바와 같은 요소들을 처리하기 위한 방법이 추가

로 연구되어야 할 것이다. 또한 윈도우 영상을 입체영상으로 변환하였을 때 입체감 효과에 대해서 증명되지 않았으므로, 이에 대한 연구를 진행할 계획이며, 비디오와 같은 연속적인 영상에 대한 처리 방법 또한 연구가 필요할 것이다.

참 고 문 헌 (References)

- [1] J. Kim, A. Baik, Y. Jung and D. Park, “2D-to-3D image/video conversion by using visual attention analysis,” Int’l Conf. on Image Processing (ICIP), 2009.
- [2] S. Battiato, A. Carpa, S. Curti and M. laCascia, “3D Stereoscopic Image Pairs by Depth-Map Generation,” Proceedings of 3DPVT, 2004.
- [3] W. Tam and L. Zhang, “3D-TV Content Generation: 2D-to-3D Conversion,” Proceedings of IEEE ICME, 2006.
- [4] L. Zhang and W. Tam, “Stereoscopic image generation based on depth image for 3DTV,” IEEE Trans. on Broadcasting, Vol. 51, Issue 2, June 2005.
- [5] M. Khodadadi and A. Behrad, “Text Localization, Extraction and Inpainting in Color Images,” 20th Iranian Conference on Electrical Engineering (ICEE), pp. 1035-1030, May 2012.
- [6] Y. Song, A. Liu, L. Pang, S. Lin, Y. Zhang and S. Tang, “A Novel Image Text Extraction Method Based on K-means Clustering,” 7th Int’l Conf. on Computer and Information Science, pp. 185-190, May 2008.
- [7] G. L. Nagy, S. Seth, and M. Viswanathan, “A prototype document image analysis system for technical journals,” Computer, 25(7):10-22, July 1992.
- [8] J. Ha, “Recursive X-Y cut using bounding boxes of connected components,” Document Analysis and Recognition, Volume: 2, pp. 952-955, 1995,
- [9] M. Sonka V. Hlavac and R. Boyle, “Image processing analysis and Machine Vision, 3rd Ed”, Thomson Press, 2008.
- [10] T. Ojala, M. Pietikainen and D. Harwood, “A comparative study of texture measures with classification based on feature distributions,” Pattern Recognition 29, pp. 51-59. 1996.
- [11] T. Ojala, M. Pietikainen and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” IEEE Trans. on Pattern Analysis and Machine Intelligence 24, pp. 971-987. 2002.
- [12] P. Viola and M. Jones, “Robust Real-Time Face Detection,” International Journal of Computer Vision, 57(2), 137-154, 2004.
- [13] P. Viola and M. Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features,” IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, 2001. pp. 511-518, 2001.
- [14] J. Gil and M. Kim, “3D Conversion of Webpages Using Image Region Detection,” 3DSA, June 2013.
- [15] J. Gil and M. Kim, “2D-to-3D Conversion of Window Images,” IWAIT, pp. 462-465, Jan, 2014.
- [16] J. Gil and M. Kim, “2D-to-3D Conversion of Pictures mixed with Text and Images,” ICIC Express Letters, Vol 8, No 1, pp. 303-319, Jan, 2014.

저 자 소 개

길 종 인

- 
- 2010년 8월 : 강원대학교 컴퓨터정보통신공학과 학사
 - 2012년 8월 : 강원대학교 컴퓨터정보통신공학과 석사
 - 2012년 9월 ~ 현재 : 강원대학교 컴퓨터정보통신공학과 박사과정
 - 주관심분야 : 3D영상처리, 휴면얼굴 검색, 객체 트래킹

김 만 배

- 
- 1983년 : 한양대학교 전자공학과 학사
 - 1986년 : University of Washington, Seattle 전기공학과 공학석사
 - 1992년 : University of Washington, Seattle 전기공학과 공학박사
 - 1992년 ~ 1998년 : 삼성종합기술원 수석연구원
 - 1998년 ~ 현재 : 강원대학교 컴퓨터정보통신공학과 교수
 - 주관심분야 : 3D영상처리, 입체변환, Human detection