

Slab Region Localization for Text Extraction using SIFT Features

최종현* · 최성후** · 윤종필*** · 구근휘§ · 김상우†
 (Jong Hyun Choi · Sung Hoo Choi · Jong Pil Yun · Keunhwi Koo · Sang Woo Kim)

Abstract - In steel making production line, steel slabs are given a unique identification number. This identification number, Slab management number (SMN), gives information about the use of the slab. Identification of SMN has been done by humans for several years, but this is expensive and not accurate and it has been a heavy burden on the workers. Consequently, to improve efficiency, automatic recognition system is desirable. Generally, a recognition system consists of text localization, text extraction, character segmentation, and character recognition. For exact SMN identification, all the stage of the recognition system must be successful. In particular, the text localization is great important stage and difficult to process. However, because of many text-like patterns in a complex background and high fuzziness between the slab and background, directly extracting text region is difficult to process. If the slab region including SMN can be detected precisely, text localization algorithm will be able to be developed on the more simple method and the processing time of the overall recognition system will be reduced. This paper describes about the slab region localization using SIFT (Scale Invariant Feature Transform) features in the image. First, SIFT algorithm is applied the captured background and slab image, then features of two images are matched by Nearest Neighbor (NN) algorithm. However, correct matching rate can be low when two images are matched. Thus, to remove incorrect match between the features of two images, geometric locations of the matched two feature points are used. Finally, search rectangle method is performed in correct matching features, and then the top boundary and side boundaries of the slab region are determined. For this processes, we can reduce search region for extraction of SMN from the slab image. Most cases, to extract text region, search region is heuristically fixed [1][2]. However, the proposed algorithm is more analytic than other algorithms, because the search region is not fixed and the slab region is searched in the whole image. Experimental results show that the proposed algorithm has a good performance.

Key Words : Slab Region Extraction, Search Rectangle Method, SIFT Features, Steel Image, Management Number

1. 서론

제철소에서 생산된 슬라브(Slab)는 넓적하고 기다란 판 모양의 철강제품으로서, 선박이나 자동차 등의 재료로 쓰인다. 슬라브에는 고유의 관리번호(Slab Management Number : SMN)가 기재되어 있으며, 이것은 슬라브 각각의 사용을 결정하는 중요한 정보이다. SMN의 인식은 현장의 근로자에 의하여 진행 되어왔으나, 하루에도 수백 장씩 장입되는 SMN를 사람의 눈으로 일일이 확인하는 것은 현장의 근로자들에게 큰 부담이고, 정확성도 다소 떨어진다. SMN의 인식이 잘못된다면, 현장에서 심각한 상황이 초래될 수 있다.

즉, 사용용도에 따라 슬라브의 재질과 특성이 각각 다르기 때문에, SMN의 인식이 잘못되어 다른 용도로 사용된다면 공정에 심각한 문제가 발생한다. 그러므로 영상처리기법이 도입된 SMN의 자동 인식 시스템이 개발되어야 한다.

현재, 공장에는 호스트 컴퓨터에서 보내온 SMN와 장입된 슬라브의 SMN의 일치 여부를 검사하기 위해 자동 인식 시스템이 설치되어 있다. 그러나 설치된 자동 인식 시스템의 인식률이 낮기 때문에, 이를 높이는 것이 필요하다. 자동 인식 시스템은, 복잡한 배경에서 문자열을 추출하고, 추출한 문자열을 한 글자씩 분리하고, 각각의 개별문자를 인식하는 과정으로 구성된다. SMN는 한 개의 알파벳과 8개의 숫자로 구성되어 있는데, 이 과정 중에 가장 어렵고 중요한 부분이 문자열을 정확히 추출하는 것이다. 최근에, 임의의 배경 안에 존재하는 임의의 문자열 추출에 대한 연구가 많이 수행되고 있다. 이런 알고리즘은 복잡한 배경을 가진 영상에서 수행되었다 하더라도, 제철소 현장의 인식 시스템에 바로 적용하기에는 많은 문제점을 가지고 있다[5]-[9].

제철소의 슬라브 생산 공장라인에 설치된 시스템은 메인 서버컴퓨터와 네트워크 카메라들로 이루어져 있다. 생산라

* 준 회원 : 포항공고 전자전기공학과 박사과정
 ** 준 회원 : 포항공고 전자전기공학과 통합박사과정
 *** 준 회원 : 포항공고 전자전기공학과 통합박사과정
 § 준 회원 : 포항공고 전자전기공학과 박사과정
 † 교신저자, 정회원 : 포항공고 전자전기공학과 부교수 · 공박
 E-mail : swkim@postech.ac.kr
 접수일자 : 2009년 2월 13일
 최종완료 : 2009년 3월 11일

인의 끝부분에 설치된 카메라에서, 제조된 슬라브가 도착하기 전의 영상을 촬영하고 슬라브가 도착한 후의 영상을 촬영하여, 이를 가지고 실시간으로 슬라브 관리번호를 인식하는 방법으로, 메인컴퓨터에 들어온 SMN 데이터와 자동인식 시스템에서 인식한 SMN 데이터가 다를 경우 관리자에게 알람을 주는 형식으로 구성된다. 슬라브가 도착하기 전의 영상을 배경영상이라고 하고, 슬라브가 도착한 후의 영상을 슬라브영상이라고 표현할 것이다(그림 1). 복잡하고 글자특성과 비슷한 배경을 가지는 슬라브영상에서 문자열을 바로 추출해 내는 것은 매우 어렵다. 따라서 문자열을 포함하는 슬라브 영역을 먼저 찾는다면, 문자열 추출 알고리즘이 좀더 정확하고 간단하며 단순해질 수 있다. 본 논문에서는, SIFT (Scale Invariant Feature Transform) 특징점을 이용하여 문자열을 추출하기 위한 탐색영역을 축소하는 알고리즘을 제안한다[3][4]. 배경영상과 슬라브 영상에 SIFT 알고리즘을 적용한 후, 얻은 특징점들을 이용하여 문자열을 포함하는 슬라브 영역을 추정하는 것이다. SIFT 알고리즘은 다소 많은 시간을 요구하는 알고리즘으로 실시간으로 적용하기 위해서는 성능 개선이 필요하다. 배경영상과 슬라브 영상의 특징점을 정합(matching)할 때, 보통은 Nearest Neighbor (NN) 알고리즘[10]을 사용하는데, 본 논문의 문제영상에서 나타나는 SIFT 특징점이 많기 때문에 이 또한 실시간으로 적용하는데 다소 무리가 있다. 본 논문에서 제안하는 알고리즘은 배경영상과 슬라브영상의 크기가 같다는 사실을 이용하여 이러한 문제점을 해결할 것이다. 두 영상의 특징점들이 정합될 때, 잘못 정합된 특징점들이 상당히 많이 발생하는데, 잘못 정합된 특징점들을 제거하기 위해서 일반적으로 RANSAC[11]이 사용된다. RANSAC은 확률적인 모델을 가지고 올바르게 정합된 특징점들을 찾는데, 배경영상과 슬라브영상의 SIFT 특징점들은 고르게 분포하기 때문에 어떠한 모델(Model)을 세우기가 어렵다. 즉, 올바르게 정합된 특징점, 즉 inlier라고 판단할 수 있는 기준이 애매모호하다. 그러므로 잘못 정합된 특징점들을 제거하기 위해서 정합된 특징점의 기하학적 위치(geometric location)를 이용할 것이다. 올바르게 정합된 특징점들은 슬라브 영역에는 나타나지 않는데, 이 점들은 Hough Transform[12]을 사용하여 무리(cluster)지을 수 있다. 이렇게 무리 지어진 영역을 제거함으로써 슬라브 영역을 얻을 수 있지만, Hough Transform을 이용해, 바르게 정합된 특징점들을 무리 지을 수 있는 기준이나 모델이 명확하지 않다. 때문에 독창적인 방법이 요구되는데, 슬라브 영역의 경계부분을 찾기 위해, 슬라브가 사각형이라는 특징을 이용한, Search Rectangle (SR)을 최종 정합된 특징점의 위치를 포함한 슬라브영상에 삽입한다. 이것을 가지고 슬라브의 상단 경계(Top boundary)와 좌우 경계(Side boundaries)를 찾아, 최종 슬라브 영역을 얻는다. 이전의 연구에서는, 문자열 찾는 영역을 경험적(heuristic)으로 고정시켜놓고 SMN를 추출하였지만, 본 논문에서 제안하는 알고리즘은 이런 경험적인(heuristic) 요소를 줄이고, 영역을 고정시키지 않고 영상 전체에서 슬라브 영역을 찾는 것이기 때문에 좀더 해석적이라고 할 수 있다[1][2]. 또한 SIFT 특징점을 사용함으로써 공정에서 발생할 수 있는 예측하지 못한 여

러 상황에 대해 알고리즘에 강인한 성격을 부여할 수 있다. 본 논문의 구성은 다음과 같다. 2장에서는 슬라브 영상과 슬라브 영역의 특징에 대해 기술하고, 3장에서는 SIFT 특징점을 추출하여 슬라브 영역을 추정하는 알고리즘을 제안한다. 4장에서는 3장의 내용을 바탕으로 제안된 방법의 유용성을 확인하기 위한 실험결과를 수록하고, 5장에서는 제안된 방법의 결과에 대하여 논의한다.

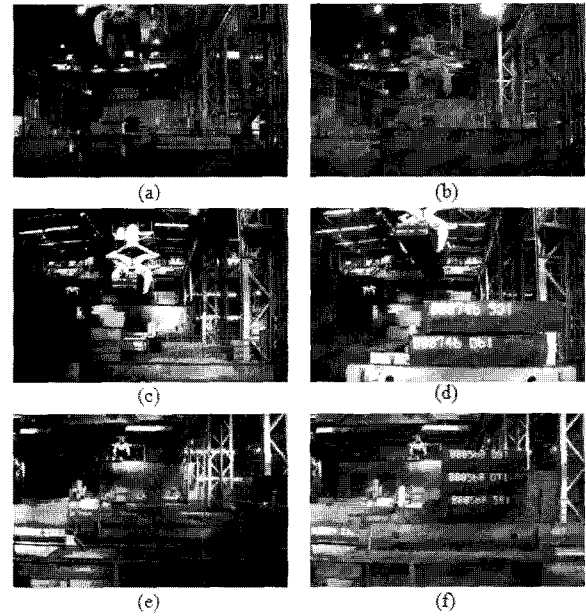


그림 1 슬라브영상(1920×1200)의 특징 : (a),(c)는 1대차에 대한 배경영상, (b),(d)는 1대차에 대한 슬라브영상, (e)는 2대차에 대한 배경영상, (f)는 2대차에 대한 슬라브영상 수

Fig. 1 background and slab images

2. 슬라브 영역 및 영상의 특징

2.1 슬라브 영상 및 슬라브 영역의 특징

제철소 슬라브 야적장에는 대차(Cart)와 크레인(Crane)이 생산된 슬라브를 운반하고 있고, 각 대차가 이동하는 라인의 끝부분에는 네트워크 카메라가 설치되어 있으며, 메인서버컴퓨터가 이들을 통제하도록 구성되어 있다. 현장에 설치된 SMN 자동인식시스템은 대차 라인의 끝부분에 설치된 카메라에서, 대차에 실린 제조된 슬라브가 도착하기 전의 배경영상을 촬영하고 대차에 실린 슬라브가 도착한 후의 슬라브영상을 촬영하여, 이를 가지고 실시간으로 SMN을 인식한다. 슬라브영상에서 문자열 검출을 위해 슬라브 영역을 추정하는 것에는 많은 어려움이 뒤따르는데, 본 논문의 대상인 슬라브영상은 다음과 같은 특징을 가지고 있다(그림 1)

- 복잡한 배경 - Corners, lines
- 많은 noise
 - 거친 슬라브 표면, 카메라의 Stochastic noise

- 슬라브 영역의 크기, 위치, 색깔 등
- 조명 변화 (Illumination Changes)
- 영상 내의 움직이는 물체
 - 크레인, 현장의 조업자 등

슬라브영상에서 슬라브의 크기와 위치, 색깔 등이 슬라브마다 가지각색이다. 크레인(Crane)이 슬라브의 무게중심을 집어 슬라브를 대차로 나르기 때문에, 대차에 놓이는 슬라브의 위치가 매우 유동적이다. 즉, 슬라브의 무게중심에 따라, 슬라브를 찍는 카메라 위치에 대하여 앞 혹은 뒤쪽으로 놓일 수 있기 때문에, 슬라브영상에서 슬라브가 차지하는 비중이 클 수도 있고 작을 수도 있다. 또한, 슬라브 자체적인 크기 또한 슬라브마다 다르고, 슬라브가 대차 위에 한 장에서 세 장까지 쌓일 수 있다. 때문에 영상에서 간단한 방법으로 슬라브가 존재하는 부분을 추출해내는 것은 상당히 어려운 일이다. 게다가, 슬라브 온도에 따라, 슬라브 색깔이 바뀌기 때문에 Color segmentation 알고리즘도 적용하기에 다소 무리가 있다. 또한 슬라브 주위의 배경부분이 매우 복잡하고, 영상에서 슬라브 특성과 비슷한 코너 점(corner point)들이나 선(line)들이 많이 존재하기 때문에 Corner detection 알고리즘[13]이나 line detection 알고리즘은 좋은 성능을 발휘하지 못한다.

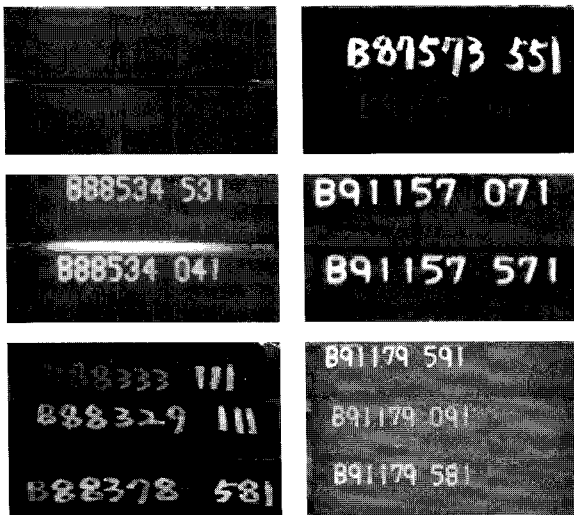


그림 2 슬라브 영역의 특징
Fig. 2 properties of slab region

2.2 SIFT 특징점 사용의 이유

영상처리(Image Processing)에서는 조명(Illumination)의 영향을 많이 받을 수밖에 없는데, 슬라브 야적장의 실내, 실외 조명이 수시로 바뀌는 문제점이 있다. 에너지 절약 차원에서 낮에는 실외 조명, 즉 태양광을 주로 이용하는데 그 날의 날씨 상황에 따라 빛의 변동(light variation)이 심하고, 실내조명 또한 변동이 심하기 때문에 많은 어려움이 뒤따른다. 보통의 경우, 배경영상과 슬

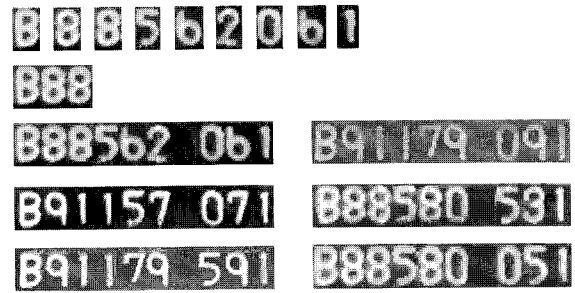


그림 3 슬라브에 포함되어 있는 문자열의 특징
Fig. 3 properties of characters embedded in slab

라브영상 사이에도 영상 간의 조명의 변화가 존재하게 된다. 게다가, 영상에서 슬라브 위에 움직이는 크레인이 자주 나타나고, 가끔 현장의 조업자가 등장하기 때문에 단순히 두 영상의 pixel값의 차이나 경계(edge) 영상을 가지고 슬라브 영역을 찾기가 쉽지 않다.

이러한 이유로 인하여, 일반적인 방법으로 슬라브 영역을 추정하기에는 다소 무리가 있다. 따라서 본 논문에서는 크기(Scale), 이동(Translation), 회전(Rotation)이나 조명의 변화에도 강인한 SIFT(Scale Invariant Feature Transform) 알고리즘을 사용함으로써 이러한 문제점을 해결하고자 한다. SIFT는 물체 인식(Object recognition)에 효과적인 알고리즘으로 SIFT를 통해 얻어진 특징점(feature)은 외부의 변화에 매우 강력한 특징을 가진다 [3][4]. 본 논문에서는 SIFT 알고리즘 사용의 일반적인 경우인, 슬라브 영역을 찾기 위해 슬라브에 SIFT를 적용시키는 것이 아니라, 슬라브 주위의 배경을 인식하여 배경부분을 제거함으로써 슬라브 영역을 얻는다. 슬라브를 인식하려면 슬라브 데이터(data)를 기준 데이터(reference data)로 가지고 있어야 하는데, 슬라브마다 특징이 매우 다양하고, 슬라브에 쓰여 있는 문자열 또한 다양하기 때문에, 이 방대한 양의 데이터를 기준 데이터로 저장하기 어려울 뿐만 아니라, 다른 특징을 가지는 새로운 슬라브가 장입되었을 때, 슬라브 인식이 매우 힘들 것이다(그림 2*3). 즉, 철강공장에서 막 생산된 슬라브이기 때문에, 슬라브 온도에 따라 색깔이 매우 다양하고, 슬라브 표면의 성질도 다양하다. 또한, 슬라브영상에 포함되어 있는 문자열은 열악한 환경에서 생성되기 때문에 일반적인 문자열과는 매우 다른 특성을 갖는다.

하지만, 슬라브 주위의 부분은 배경영상과 슬라브영상의 비교를 통해서 인식을 할 수 있기 때문에, 조명의 변화나 크기의 변화에 강인한 SIFT를 슬라브 이외의 배경부분에 적용시켜 SIFT 특징점(feature)들을 얻을 수 있다. 이 특징점들을 바탕으로 슬라브의 경계를 추정함으로써 슬라브 영역을 얻을 수 있다.

3. 강건한 특징점 추출 및 슬라브 영역 추정

본 논문은 SIFT 특징점(feature)을 이용하여 슬라브영상에서 슬라브 영역을 추정하는 알고리즘을 제안하는데, 슬라브 영역 추정 알고리즘의 전반적인 흐름은 다음과

같다. 배경영상과 슬라브영상에 I-SIFT(Improved SIFT)를 통해 특징점(feature) 후보를 추출하고, 배경영상과 슬라브영상의 특징점들을 Local Nearest Neighbor (L-NN) 알고리즘을 이용하여 정합(matching)한 후, 잘못 정합된 특징점들을 제거한다. 슬라브영상에서 최종 추출된 특징점들을 바탕으로 Search Rectangle 기법을 사용하여 슬라브의 상단 경계(Top boundary)와 좌우 경계 (Side boundaries)를 찾아 슬라브 영역을 추정한다.

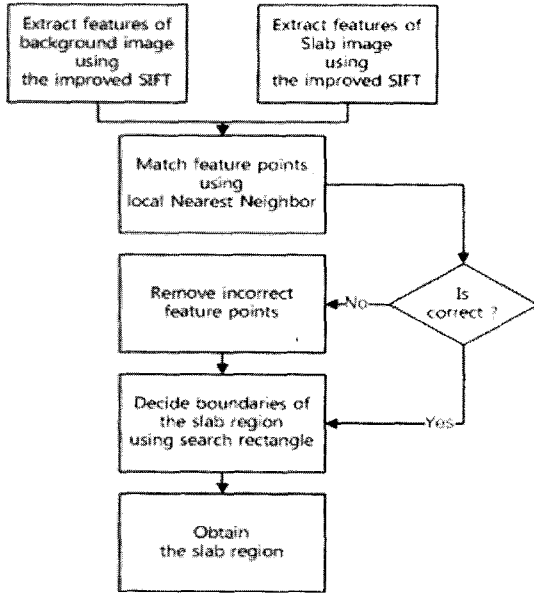


그림 4 슬라브 영역 추정 알고리즘
Fig. 4 Slab region localization algorithm

3.1 강건한 특징점 추출

3.1.1 I-SIFT를 이용한 특징점 추출

일반적으로 사용되는 SIFT는 다소 무거운 알고리즘으로 처리 시간(processing time)이 많이 걸리는 것이 단점이다. 때문에 자동인식시스템에서 실시간으로 적용하기 위해서는 알고리즘을 변형을 시켜야 한다. conventional SIFT는 크기(scale)에 강한 특징을 부과하기 위해, 가우시안 영상피라미드를 가지고 이후 과정을 진행하게 되는데, 일반적으로 입력영상을 2배 확대(up sampling)한 후, 영상의 크기가 매우 작아질 때까지 축소(down sampling)하면서, 각각의 octave에서 특징점 후보를 추출한다[3]. 하지만, 슬라브영상에서 크기(scale)가 변화하는 물체(object)는 슬라브 주위에서 움직이는 크레인들과 간혹 영상에 등장하는 현장의 조업자들인데, 이것들은 슬라브 경계를 찾기 위한 특징점을 얻는데 큰 방해요소가 아니므로, 크기(scale)에 강한 특성은 크게 필요하지 않다. 때문에 입력영상을 down sampling을 하지 않고, 2배 up sampling한 영상과 원본(original)영상만 가지고 Difference of Gaussian(DOG)에서 특징점 후보를 추출하고 특징벡터(descriptor)를 생성하여, 특징점을 얻음으로써 시간을 단축할 수 있다.

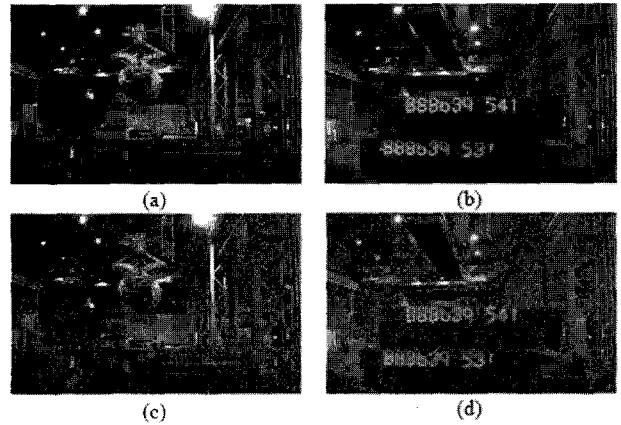


그림 5 배경영상과 슬라브영상의 특징점 추출
Fig. 5 SIFT features extraction of background and slab image

3.1.2 특징점 정합 (matching)

SIFT에서 특징점을 정합(matching)하기 위해 일반적으로 Nearest Neighbor (NN)알고리즘을 사용한다[10]. NN 알고리즘은 현재영상에서 특징점의 특징벡터(descriptor)와 가장 비슷한 특징벡터를 갖는 특징점을 비교영상에서 찾는 것인데, 특징점들의 특징벡터 사이의 norm이 가장 작은 것을 정합되었다고 판단하는 방법이다. 이 때, 가장 작은 norm과 두 번째로 작은 norm 사이의 비율을 통해 정합 여부를 판단한다. 하나의 특징점을 정합하기 위해, 비교영상의 모든 특징점들과 비교하게 되는데, 이 방법을 본 논문의 적용대상에 그대로 적용한다면, 시간이 많이 걸릴 뿐만 아니라 정합 정확도도 또한 떨어진다.

배경영상과 슬라브영상 사이에서 조명의 변화는 심하지만 슬라브 주위에 고정 되어있는 물체(object)들이 존재하고, 배경영상과 슬라브영상의 크기(size)가 같기 때문에 전체 영역에 대해 NN 알고리즘을 적용하는 것이 아니라, 지역별 영역(local region)에 대해 NN 알고리즘을 적용함으로써, 시간을 대폭 단축하고, 정합(matching) 정확도 역시 향상 시킬 수 있다. 즉, 배경영상의 특징점과 정합되는 특징점을 슬라브영상에서 찾을 때, 슬라브영상에서 배경영상의 특징점 위치(location)를 중심으로 주변의 일부 영역에 대해서만 검색을 하는 것이다. 이 때, 벡터의 norm이 문턱치(threshold)보다 크면 정합된 특징점으로 판단하지 않는다. 그림 6은 특징점들을 정합하기 위해 전체영역에 대해 NN을 적용한 것과 주변의 일부(local) 영역에 NN을 적용한 것을 비교한 것으로, 배경영상과 슬라브영상이 겹쳐(overlap)진 영상에 정합되었다고 판단된 특징점들을 나타낸 것이다. 선으로 이어진 특징점들은 잘못 정합된 것이고, 점들로 나타내어진 것들은 올바르게 정합된 특징점들이다. 주변의 일부(local) 영역에 NN을 적용하여 정합했을 때, 전체영역에 대해 적용한 결과보다 더 좋은 성능을 나타내는 것을 확인할 수 있다.



(a)



(b)

그림 6 배경영상과 슬라브영상의 특징점 정합: (a) NN을 이용한 정합, (b) Local-NN을 이용한 정합

Fig. 6 Matching SIFT features of background and slab image

3.1.2 특징점 정합 (matching)

주변의 일부(local) 영역에 NN을 적용하여 정합된 결과를 얻었다 하더라도, 잘못 정합된 특징점들이 발생하는데, 이렇게 잘못 정합된 점들을 제거하기 위해서, 보통은 RANSAC이라는 확률적인 모델을 가진 알고리즘으로 inlier와 outlier를 판단하여 outlier, 즉 잘못 정합된 특징점들을 제거한다[11]. 하지만 본 논문의 대상에서 나타나는 특징점들은 전 영역에 걸쳐 고루 분포하기 때문에 확률적인 모델을 세우기가 힘들다. 때문에 conventional 방법으로 잘못 정합된 점들을 제거하는 것은 어려움이 뒤따른다. 이러한 문제점을 해결하기 위해, 두 영상의 특징점들 간의 기하학적 위치(geometric location) 정보를 이용한다. 배경영상과 슬라브영상의 크기가 같기 때문에, 정합되는 두 특징점들의 영상 내에서의 기하학적 위치를 가지고 Euclidean 거리 정보를 이용하여, 잘못 정합된 것들을 제거한다.

$$dist_1 = \sqrt{(x_i^s - x_j^b)^2 + (y_i^s - y_j^b)^2} < threshold$$

(x^s, y^s) 는 슬라브영상에서 특징점들의 좌표이고, (x^b, y^b) 는 배경영상에서 특징점들의 좌표이다. 이 Euclidean 거리가 어떤 문턱치(threshold)보다 작으면 올바르게 정합되었다고 판단하고, 슬라브영상에서 최종 특징점을 추출한다.



그림 7 최종 특징점 추출

Fig. 7 final features extraction

3.2 슬라브 영역 추정

3.2.1 슬라브 영역의 상단 경계 결정

최종적으로 추출된 특징점으로부터 슬라브 영역을 추정하기 위해 슬라브가 사각형(rectangle)이라는 특징을 반영한 Search Rectangle (SR) method를 사용한다. 대부분의 경우에 슬라브는 영상에서 아래쪽 중앙(center) 부분에 위치하게 되는데, 이 아래쪽 중앙 부분에 Search Rectangle을 삽입한다. 이 때, 초기(initial) Search Rectangle의 크기는 슬라브의 최소 크기(minimum size)보다 좀 더 작게 하여, Search Rectangle의 폭(width)과 높이(height)를 확장시켜나감으로써 슬라브 영역을 추정한다. 슬라브 영역의 상단 경계(top boundary)를 결정하기 위해 초기 Search Rectangle의 가로(width line)가 특징점을 만날 때까지 Search Rectangle의 높이(height)를 증가시킨다. Search Rectangle의 높이가 결정되면, 이것이 바로 슬라브 영역의 상단 경계가 되고, 상단 경계 위쪽의 슬라브영상과 특징점들을 제거한다(그림 8).

표 1 슬라브 영역의 상단 경계 결정 알고리즘
Table 1 Top boundary determination algorithm

```

Initialization
Insert search rectangle at the lowest center
N : feature points above the search rectangle
w : width line of the search rectangle
Loop
for i=1:N
    if ( 'w' meet feature point ) then
        height = y coordinate of feature (i)
        break
    end if
end for
Remove image and features above the height
    
```

3.2.2 슬라브 영역의 좌우 경계 결정

상단 경계 위쪽 부분을 제거하고 남아 있는 특징점들은 슬라브 영역의 좌우에 분포하게 되는데, 슬라브 영역

의 좌우 경계(side boundaries)는 남아 있는 특징점들을 가지고 앞에서 이용한 방법과 마찬가지로 결정한다. 즉, Search Rectangle의 세로(height line)가 남아 있는 특징점들을 만날 때까지 Search Rectangle의 폭(width)을 확장시킨다. Search Rectangle의 폭이 결정되면 이것이 슬라브 영역의 좌우 경계가 되는데, 상단 경계와는 다르게 좌우 경계를 각각 하나씩만 가지고 결정한다면, 슬라브 영역을 침범하는 경계가 결정될 수도 있다. 즉, 두 장 이상의 슬라브가 쌓였을 때, 슬라브가 비스듬하게 쌓이거나 쌓여진 슬라브의 크기가 서로 다른 경우에 문제가 발생한다(그림 9).

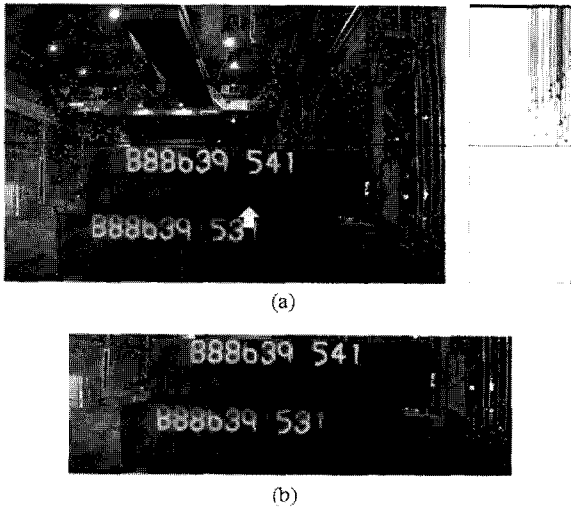


그림 8 슬라브 영역의 상단 경계 결정
Fig. 8 Determination of top boundary of slab region

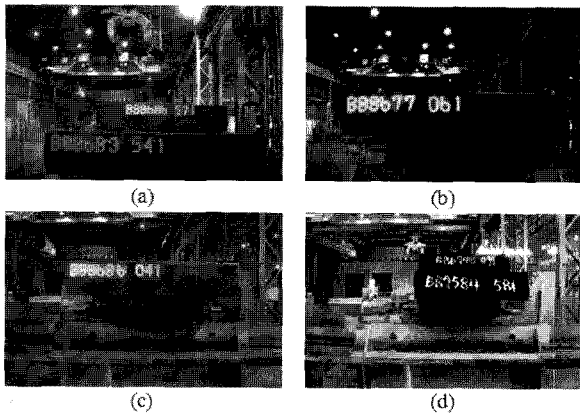


그림 9 슬라브가 비스듬하게 쌓이거나 크기가 서로 다른 경우
Fig. 9 Slabs are stacked up crossly or size of slabs is mutually unlike.

하지만 생산관리시스템으로부터 전송된 정보에는 슬라브의 장 수 정보가 포함되어 있는데, 이 정보를 이용하여, 슬라브의 장 수 만큼 좌우 경계(layer cutting points)를 각각을 찾는다. 상단 경계에 의해 제거되고 남은 부분에서, 쌓인 슬라브가 두 장일 경우에는 두 부분으로 나누고, 세 장일 경우에는 세 부분으로 나누어 좌우 경계를

각각 구한다(그림 10). 물론 슬라브마다 크기가 다르기 때문에, 좌우 경계를 추정할 구역을 분할할 수 있는 정확한 비율을 정하기가 애매하지만, 대략적(rough)으로는 그림 10과 같다.

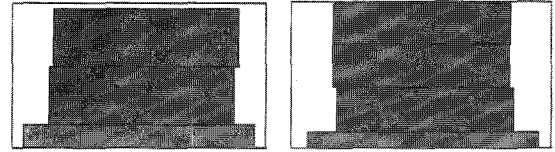


그림 10 좌우 경계를 결정하기 위한 구역 분할
Fig. 10 Region partition for deciding side boundaries

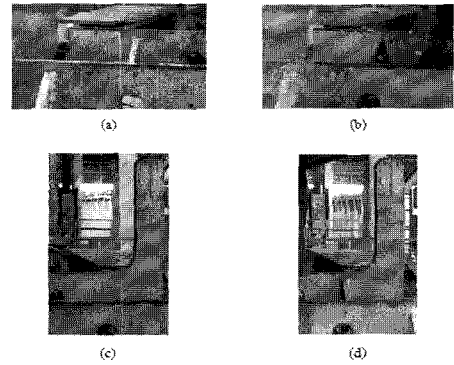


그림 11 대차의 특징점
Fig. 11 Features of carts

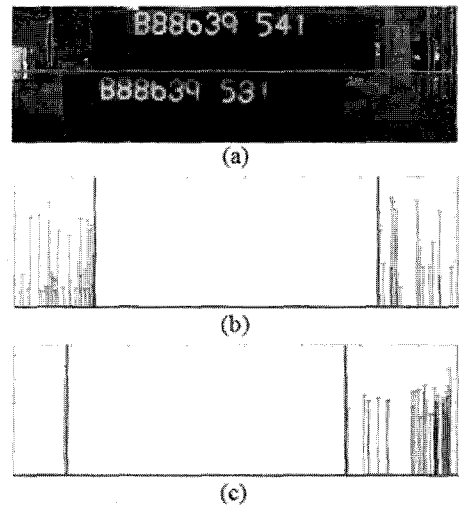


그림 12 슬라브 영역의 좌우 경계 결정
Fig. 12 Determination of side boundaries of slab region

또한, 슬라브는 대차를 통해서 운반되어지기 때문에, 슬라브영상에서 대차부분도 나타날 수도 있다. 즉, 대차에서 슬라브가 놓인 위치가 너무 앞쪽이면 슬라브에 의해 대차부분이 가려질 수 있지만, 그렇지 않는 대부분의 경우에 영상에서 대차가 등장하게 된다. 대차 역시 배경 영상에서는 나타나지 않기 때문에 특징점이 추출되지 않는다. 그렇기 때문에, 이 대차의 SIFT 특징점들을 데이터

베이스(database)에 저장하여, 좌우 경계를 구할 때, 대차 부분의 특징점도 포함 시킨다(그림 11). 그렇게 함으로써, 좀 더 정확한 슬라브 영역의 좌우 경계를 얻을 수 있다(그림 12). 슬라브 상단 경계를 결정할 때와 비슷한 방법으로, Search Rectangle의 폭(width)을 확장시켜나가면서 좌우 경계를 결정하는데, 상단 경계는 하나의 경계만이 필요했지만, 좌우 경계를 슬라브 장수만큼의 경계가 필요하다는 점에서 다르다.

표 2 슬라브 영역의 좌우 경계 결정 알고리즘

Table 2 Side boundaries determination algorithm

```

Determine side boundaries in case of the two stacked slabs
Initialization
N1 : Remaining feature points in the upper side of the search rectangle
N2 : Remaining feature points in the lower side of the search rectangle
hL : Left height line of the search rectangle
hR : Right height line of the search rectangle
Loop
for i=1: N1
    if ( 'hL' meet upper left feature point ) then
        upper left side boundary = x coordinate of feature (i)
    else if ( 'hR' meet upper right feature point ) then
        upper right side boundary = x coordinate of feature (i)
    end if
end for

for i=1: N2
    if ( 'hL' meet lower left feature point ) then
        lower left side boundary = x coordinate of feature (i)
    else if ( 'hR' meet lower right feature point ) then
        lower right side boundary = x coordinate of feature (i)
    end if
end for
    
```

3.2.2 슬라브 영역의 좌우 경계 결정

3.2.1.에서 찾은 상단 경계(top boundary)는 초기(initial) Search Rectangle에서 얻은 것이기 때문에, 상단 경계를 최적화(optimization)하는 과정이 필요하다. I-SIFT(Improved SIFT)로 특징점을 얻을 때, 슬라브 위에서 움직이는 크레인을 배제시키기 때문에, 초기 Search Rectangle의 가로(width line) 위쪽에 크레인이 자리 잡는 경우, 초기 Search Rectangle 위쪽으로 특징점이 나타나지 않는 상황이 종종 발생한다. 또한, 간혹 초기 Search Rectangle 위쪽 부분이 너무 어두워서 정보가 없는 경우도 있는데, 이때에도 특징점이 나타나지 않을 수 있다. 이런 경우에 앞에서 찾은 슬라브 경계 안에 문자열 검출을 방해하는 요소가 포함될 수 있다. 그렇기 때문에 상단 경계를 최적화하는 과정이 반드시 필요하다. 3.2.2.에서 찾은 좌우 경계의 중간의 일부분을 배경영상과 슬라브영상에서 잘라내어 SIFT 특징점을 추가로 추출한 후, 좀 더 정확한 상단 경계를 얻는다(그림 13). 이 때, 배경영상과 슬라브영상에서 잘라낸 부분의 크기가 작기 때문에, 확대(up sampling)시킨 후 SIFT를 적용하여 좀 더 많은 특징점을 얻어낸다. 실험적으로 2³ 배만큼 확대(up sampling)시켰을 때 특징점의 수와 시간적인 면에서 좋은 결과를 얻을 수 있었다. 이렇게 얻은 특징점을 기반으로, 좀 더 정확한 슬라브 영역의 상단 경계를 추정할 수 있다.



(a)



(b)



(c)

그림 13 슬라브 영역의 상단 경계 최적화 : (a) 좌우 경계를 중심으로 배경영상과 슬라브영상에서 잘라낸 부분, (b) SIFT 특징점

Fig. 13 Top boundary optimization

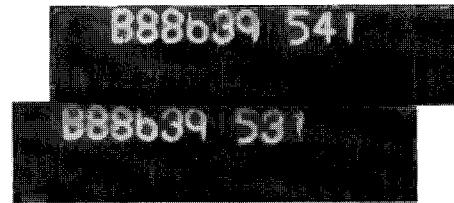


그림 14 최종 슬라브 영역 추정 결과

Fig. 14 Result of slab region localization

4. 실험 및 결과

본 논문에서는 배경영상과 슬라브영상에 I-SIFT(Improved SIFT)를 통해 특징점(feature)을 추출하였고, 배경영상과 슬라브영상의 특징점들을 Local Nearest Neighbor (L-NN) 알고리즘을 이용하여 정합(matching)한 후, 잘못 정합된 특징점들을 제거하였다. conventional SIFT를 사용했을 때보다, I-SIFT를 사용하여 특징점을 얻을 때, 처리 시간(process time)은 줄어들면서 추출된 특징점은 conventional SIFT를 사용하여 얻은 특징점과 비슷하게 분포하는 것을 볼 수 있었다(그림 15). 또한, 특징점을 정합(matching)할 때, Nearest Neighbor(NN)을 사용했을 때보다 L-NN을 사용하였을 때, 처리 시간(process time)은 1/10 이하로 줄어들면서 정합(matching) 정확도(accuracy)를 대폭 향상시켰음을 확인할 수 있었다(그림 16). 그림 15의 (a)는 conventional SIFT로 특징점을 얻고 Nearest Neighbor로 특징점을 정합(matching)하여 잘못 정합된 특징점을 제거한 결과이고, (b)는 I-SIFT로 특징점을 얻고 L-NN으로 특징점을 정합한 후, 잘못 정합된 특징점을 제거한 결과이다. 특징점들의 분포를 보면 거의 큰 차이가 없는 것을 볼 수 있다. 즉, 기존의 SIFT 방법과 비슷한 성능을 보이면서 처리 시간을 단축하는 효과가 있는 것을 확인할 수 있었다.

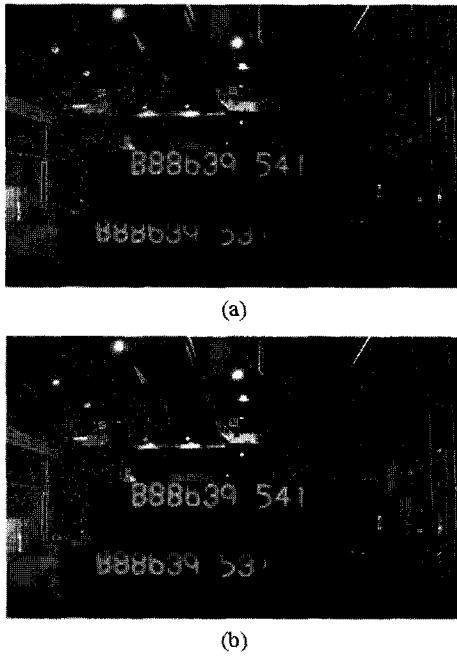


그림 15 특징점 추출 결과 비교 : (a) conventional SIFT + NN, (b) I-SIFT + L-NN

Fig. 15 Comparison of features extraction result

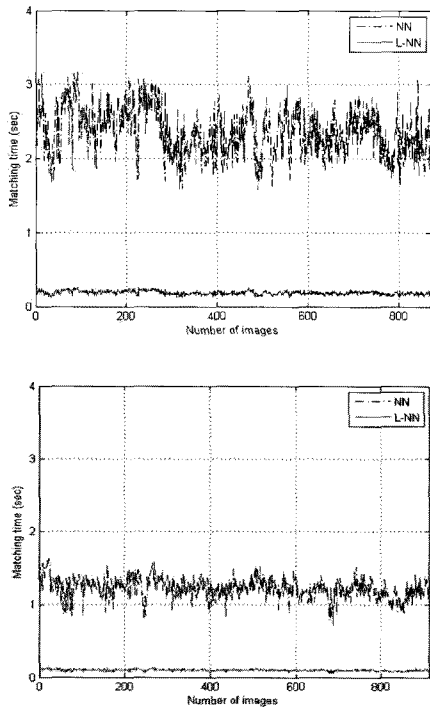


그림 16 정합 시간 비교 : (a) 1대차(870 images), (b) 2대차 (912 images)

Fig. 16 Comparison of matching time

본 논문에서 제안하는 알고리즘으로 슬라브 영역을 추정 한 결과는 그림 17과 같다. 쌓인 슬라브의 장수에 상관 없이, 슬라브가 비스듬하게 쌓이거나 그 크기가 서로 다른 경우에도 정확한 슬라브 경계를 추정하는 것을 확인

할 수 있었다. 추정된 슬라브의 경계가 슬라브를 침범하여 글자영역이 잘려져 나간 경우를 슬라브 영역 추정 실패로 보았는데, 1,2대차 총 1782장의 슬라브영상에 대하여 슬라브 영역의 추정 결과, 99.72%의 높은 성공률을 보였다. 이 수치는, 영상의 질(quality)에 상관없이 모든 경우를 포함하는 영상에 대한 것으로, 거의 모든 단계에서 높은 성공률을 보였는데, 간혹 SIFT를 통해 얻은 특징점이 슬라브 영역 위에 존재하는 경우가 있었다. 이럴 경우에는 슬라브 영역을 추정하는 것이 실패할 수 있지만, 실험 결과, 이런 경우는 매우 드물게 나타나기 때문에 본 논문에서 제안하는 알고리즘이 매우 좋은 성능을 보이는 것을 확인할 수 있었다.

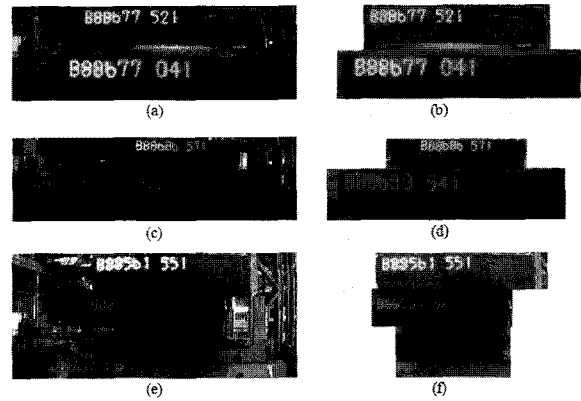


그림 17 슬라브 영역 추정 결과

Fig. 17 Result of slab region localization

표 3 슬라브 영역 추정 실험 결과

Table 3 Experimental result of slab region localization

algorithm state	성공	실패	성공률
상단 경계 결정	1780장	2장	99.89 %
좌우 경계 결정	1779장	1장	99.94 %
상단 경계 최적화	1777장	2장	99.89 %
결과	1777장	5장	99.72 %

본 논문에서 제안하는 슬라브 영역 추정 알고리즘으로 문자열을 검출하기 위한 Search 영역(region)을 슬라브 영역으로 축소함으로써, 문자열 검출 성공률을 높일 뿐만 아니라, 시간도 많이 단축되는 것을 확인할 수 있었다. 우리의 이전 연구에서 개발된 문자열 검출 알고리즘을 본 논문이 제안하는 방법이 적용된 결과에 적용했을 때, 총 2054개의 슬라브 개수에 대한 문자열 검출 성공률이 95.23%에서 98.01%로 증가함을 볼 수 있었다(표 4). 이 중 에서 슬라브 3개 정도가 본 논문에서 제안하는 알고리즘으로 찾은 경계가 슬라브 내의 문자열을 침범함으로써 인해 문자열 검출이 실패하였으나, 전반적으로 문자열 검출 성공률이 올라감을 확인할 수 있었다.

표 4 문자열 검출 결과

Table 4 Experimental result of text localization

	성공	실패	성공률
이전 연구에 의한 문자열 검출	1956	98	95.23 %
제안된 방법에 의한 문자열 검출	2013	41 (3 SIFT)	98.01 %

5. 결 론

본 논문은 제철소에서 생산되는 슬라브에 기재된 슬라브 관리번호(Slab Management Number : SMN)를 인식하기 위한 가장 중요한 단계로 슬라브 영역을 추정하는 알고리즘을 제안한다. 슬라브영상은 제철소 내의 공정 과정에서 찍은 영상이기 때문에, 영상에 노이즈가 많고 복잡할 뿐만 아니라, 글자특성과 비슷한 배경을 가지기 때문에, 슬라브영상에서 문자열을 바로 추출해 내는 것은 매우 어렵다. 그렇기 때문에 문자열을 포함하는 슬라브영역을 정확하게 찾는다면, 문자열 검출 알고리즘이 좀 더 간단하고 정확해질 수 있다.

이를 위해서 물체 인식에 많이 이용되는 물체의 크기(scale), 회전(rotation), 조명(illumination)의 변화에도 강인한 SIFT(Scale Invariant Feature Transform) 알고리즘을 사용한다. 그러나 SIFT 알고리즘은 다소 많은 시간을 요구하는 알고리즘으로 실시간으로 적용하기 위해서는 성능 개선이 필요하다. 시간을 단축하면서 원래의 SIFT와 비슷한 성능을 발휘하는 I-SIFT(Improved SIFT)로 배경영상과 슬라브영상에서 특징점을 추출한 후, Local Nearest Neighbor(L-NN)로 배경영상과 슬라브영상에서 추출한 특징점을 정합(matching)하고, 두 영상간의 기하학적(geometric) 특징을 이용하여 잘못된 정합된 특징점들을 제거하여, 슬라브영상에서 최종 특징점을 얻는다. 이 특징점을 이용하여 슬라브영역을 추정하기 위해 Search Rectangle method를 사용하였는데, Search Rectangle을 특징점을 포함한 슬라브영상에 삽입하여, Search Rectangle의 폭(width)과 높이(height)를 확장해 나가면서 슬라브 경계를 추정한다. 이 때, 슬라브의 좌우 경계(side boundaries)를 얻을 때, 슬라브 장수만큼의 좌우 경계를 찾아서, 슬라브가 비스듬하게 쌓이거나, 쌓인 슬라브의 크기가 다를 경우에도 정확하게 슬라브의 경계를 추정할 수 있도록 한다. 제안된 알고리즘으로 슬라브영역을 추정한 결과, 처리 시간을 기존의 SIFT 방법을 적용했을 때보다 단축함을 알 수 있었고, 총 1782장의 슬라브영상에 대해서 99.72%의 높은 성공률을 보였다. 이 성공률은 영상의 질(quality)에 상관없이 모든 경우를 포함한 영상에 적용한 것으로, 이 수치는, 영상의 질(quality)에 상관없이 모든 경우를 포함하는 영상에 대한 것으로, 공장에서 발생할 수 있는 예외상황에 대처할 수 있는 강인한 알고리즘이라는 사실을 증명한다. 즉, 제안한 알고리즘이 슬라브 영역 추정의 정확성뿐만 아니라 강인성까지 모두 만족함을 보여준다. 이렇게 슬라브영역을 추정한 결과에, 기존의 개발된 문자열 검출 알고리즘으로 성

능 실험을 한 결과, 총 2054개의 슬라브에 대해 문자열 검출 성공률이 95.23%에서 98.01%로 증가하는 것을 확인할 수 있었다. 이것은 제안된 알고리즘 개발 초기 목표였던 문자열 검출 성공률을 높이는데 기여했음을 보여준다. 정확도면에서는 목표치에 도달했지만, 슬라브영역을 추정하는 처리 시간을 좀 더 줄여야할 필요성이 있다. 물론, 갈수록 발달하는 기술로 인해, 하드웨어적인 요소로 이러한 문제점을 해결할 수 있지만, 알고리즘 자체적으로 좀 더 강인해 진다면 더 좋은 결과를 도출할 수 있기 때문이다.

감사의 글

본 연구는 교육과학기술부와 한국산업기술재단의 지역혁신인력양성사업으로 수행된 연구결과임.

참 고 문 헌

- [1] S.H. Choi, J.P.Yun, K.H.Koo, J.H.Choi, and S.W.Kim, "An Improved edge-based Text region Segmentation algorithm applied to Slab image data from Steel Plant.", Proceedings of the 10th IASTED International Conference COMPUTER GRAPHICS AND IMAGING, pp.70-75, 2008
- [2] S.H. Choi, J.P.Yun, K.H.Koo, J.H.Choi, and S.W.Kim, "Text Region Extraction Algorithm on Steel Making Process," 8th WSEAS International Conference on ROBOTICS, CONTROL and MANUFACTURING TECHNOLOGY, pp.24-28, 2008.
- [3] Lowe, D.G. "Distinctive Image Features from Scale Invariant Keypoints." International Journal of Computer Vision, pp.91-110,2004.
- [4] Lowe, D.G. "Object recognition from local scale-invariant features." Proceedings of International Conference on Computer Vision, pp.1150-1157, 1999.
- [5] J. Gao, J.Yang, "An Adaptive Algorithm for Text Detection from Natural Scenes.", Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition, 2001.
- [6] Xilin Chen, Jie Yang, Jing Zhang, Alex Waibel, "Automatic Detection and Recognition of Signs From Natural Scenes.", IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL.13, NO.1, 2004, pp.87-99.
- [7] K. Jung, K.I. Kim, and A.K. Jain, "Text information extraction in images and video: A survey", Pattern Recognition, vol.37, no.5, pp.977-997, May 2004.
- [8] Yingzi Du, Chein-I Chang, "Automated system for text detection in individual video images.", Journal of Electronic Imaging, 12(3), 2003, pp.410 - 422.
- [9] Q. Ye, Q. Huang, W. Gao, and D. Zhao, "Fast and robust text detection in images and video frames", Image and Vision Computing, Vol.23, No.6, 2005, pp.565-576.

- [10] Beis, Jeff, and David G.Lowe, "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces," Conference on Computer Vision and Pattern Recognition, PuertoRico(1997), pp.1000-1006.
- [11] M.A.Fischler and R.C.Bolles. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. Commun. Assoc. Comp. Mach. 24:381-395, 1981.
- [12] P.V.C. Hough. Method and means of recognizing complex patterns, December 1692, U.S. Patent 30695418.
- [13] C.J. Harris and M. Stephens, A combined corner and edge detector, In Proceeding of 4th Alvey Vision Conference, pp.147-151, Manchester, 1988.

저 자 소 개



구근휘 (具根輝)

1983년 4월 18일생. 2007년 부산대 전자전기공학과 졸업(공학사). 2009년 포항공대 대학원 전자전기공학과 졸업(공학석사). 2009년~현재 동 대학원 전자전기공학과 박사과정
 Tel : 054-279-5018
 Fax : 054-279-2903
 E-mail : khkoo@postech.ac.kr



김상우 (金相禹)

1962년 8월 14일생. 1983년 서울대 제어계측공학과 졸업(공학사). 1985년 동 대학원 제어계측공학과 졸업(공학석사). 1990년 동 대학원 제어계측공학과 졸업(공학박사).
 2001년~현재 포항공대 전자전기공학과 부교수
 Tel : 054-279-2237
 Fax : 054-279-2903
 E-mail : swkim@postech.ac.kr



최종현 (崔鍾賢)

1985년 1월 11일생. 2007년 아주대 전자공학부 졸업(공학사). 2009년 포항공대 대학원 전자전기공학과 졸업(공학석사). 2009년~현재 동 대학원 전자전기공학과 박사과정
 Tel : 054-279-5018
 Fax : 054-279-2903
 E-mail : y2kscore@postech.ac.kr



최성후 (崔聖後)

1978년 4월 25일생. 2004년 경북대 전자전기공학과 졸업(공학사). 2004년~현재 포항공대 대학원 전자전기공학과 통합박사과정
 Tel : 054-279-5018
 Fax : 054-279-2903
 E-mail : csh425@postech.ac.kr



윤종필 (尹鍾弼)

1977년 9월 21일생. 2003년 경북대 전자전기공학과 졸업(공학사). 2003년~현재 포항공대 대학원 전자전기공학과 통합박사과정
 Tel : 054-279-5018
 Fax : 054-279-2903
 E-mail : rebirth@postech.ac.kr