



Journal of Knowledge Information Technology and Systems

ISSN 1975-7700

<http://www.kkits.or.kr>

High Reliability Fire Detection Systems Using Smoke Detection and Image Processing in Rack Type Warehouse

Hyung-Dae Lee, Jae-Woong Kim*

Computer Science and Engineering, Kongju National University

ABSTRACT

The fire management in rack type warehouses is a very important and essential element. Usually, a fire is first produced smoke, then a flame is produced, leading to a fire. Generally, smoke detection systems have smoke detectors installed in all of the rack spaces and detect smoke early on and deliver the correct location. However, if there is not enough space in the rack to install smoke detectors, or if all racks do not have smoke detectors installed, one smoke detector is installed at the top of the rack to manage the surrounding racks. If the smoke detector detects smoke in this configuration, it is not possible to determine which of the grouped racks is generating smoke. Therefore, this study proposes a system to efficiently locate the position of the smoke-producing racks. For this purpose, the camera is installed on a stacker crane that loads and leaves products in a rack warehouse. Image frames are transmitted in real time from fixed cameras so that the difference images are calculated reliably. The images obtained detect the direction of smoke using a technique called dense optical flow, an algorithm that detects the movement of an object. The purpose of this paper is to develop a highly reliable fire detection system that detects the location of the racks in which smoke occurs by moving the stacker cranes in the detected direction and repeatedly.

© 2020 KKITS All rights reserved

KEYWORDS : Rack type warehouse, Smoke detector, Stacker crane, Image processing, Optical flow, Smoke direction

ARTICLE INFO: Received 20 December 2019, Revised 31 January 2020, Accepted 7 February 2020.

*Corresponding author is with the Computer Science and Engineering, Kongju National University, 1223-24, Cheonan-daero, Seobuk-gu, Cheonan-si,

Chungcheongnam-do, 31080, KOREA.
E-mail address: jwkim@kongju.ac.kr

1. 서론

최근 rack형 창고에서 화재가 발생하면 적재되어 있는 물품이 모두 소실되어 큰 피해를 입는 경우가 대부분이다. 하지만 rack형 창고는 선반형 구조로 공간을 합리적으로 활용하며 저장 용량을 최대한으로 늘릴 수 있다. 또한 rack에 적재된 제품은 서로 압착되지 않고 제품의 기능을 완전히 보장하므로 제품의 손실을 없앨 수 있는 장점이 있다[1].

rack형 자동화 창고 특성상 상주 인원이 비교적 적은 편으로 화재 발생 시 화재 사실을 직접 발견할 수 있는 확률이 낮아진다[2]. 그래서 일반적인 rack형 자동화 창고에 있는 화재 감지 시스템은 모든 rack 위치에 연기감지기를 설치하여 화재의 초기 현상인 연기를 감지하게 되면 연기감지기의 정확한 위치를 알려주게 되어 있다. 그러나 rack 안에 연기감지기를 설치할 공간이 부족하거나 금액적인 문제로 모든 rack 공간에 연기감지기를 설치하지 못한 경우에는 연기가 발생하는 정확한 위치를 찾을 수 없다. 그러므로 rack을 2x2 (4개 중에 1개) 또는 2x3 (6개 중에 1개)으로 묶어 1개의 연기감지기가 설치되어 해당 rack을 관리한다. 1개의 연기감지기만으로는 연기의 발생 위치를 정확하게 발견할 수 없어 화재를 조기에 막을 수 없다.

본 논문에서는 2x2, 2x3 영역에 1개의 연기감지기가 설치되어 있는 환경에서는 연기감지기가 연기를 감지하여도 정확한 위치를 알 수 없는 문제를 해결하기 위해 연기를 감지한 감지기의 위치에서부터 카메라로 영상을 획득하면서 이미지 프로세싱을 이용하여 최초의 연기 발생지로 이동하는 방법을 연구 개발하고자 한다.

본 논문의 구성은 2장에서 연기감지기와 이미지 프로세싱 관련 연구를 분석하며, 3장에서는 이미지 프로세싱을 이용해 연기의 발생지로 찾아가는 방

법 및 기법을 제시한다. 4장에서는 설계에 따라 개발된 프로그램으로 이미지 프로세싱의 결과를 나타내고, 5장에서는 결론 및 향후 연구 방향에 대해 기술한다.

2. 관련 연구

2.1 연기 감지기의 종류 및 특징

연기감지기의 종류에는 <표 1>과 같이 크게 이온화식 감지기와 광전식으로 분류된다.

표 1. 연기감지기의 종류
Table 1. Kinds of Smoke detection sensor

구분	종류	분류	특징
연기 감지기	광전식	축적형, 비축적형	연기로 인한 센서의 전류 흐름에 이상이 발생하였을 때 동작
	이온화식	축적형, 비축적형	
		아날로그식	

이온화식 감지기는 검지부에 연기가 들어가면 전류가 변하는 것을 이용하여 연기를 감지하는 방식이고, 광전식 감지기는 검지부에 연기가 들어가는데 따라 광전소자의 입사 광량이 변화하는 것을 이용하여 연기를 감지하는 방식이다. 광전식 감지기는 감지기 속의 밀폐된 공간에 공기가 깨끗할 때에는 빛의 산란이 발생하지 않는 특징이 있으므로 주변 환경이 깨끗한 rack형 자동화 창고에서의 사용이 적합하다[3].

2.2 이미지 프로세싱

2.2.1 이미지 프로세싱의 정의

이미지 프로세싱은 사진이나 영상을 처리하는 디지털 이미지 프로세싱을 의미한다. 디지털 이미

지 프로세싱은 컴퓨터를 사용하여 입력받은 이미지를 처리하는 것이다. 그중에 컴퓨터 비전은 이미지나 영상에서 물체(Object), 전경(Foreground), 배경(Background) 등 물체와 주변 환경에 대한 데이터를 분석해서 유의미한 정보를 생성하는 기술이다[4].

2.2.2 컴퓨터 비전의 특징

컴퓨터 비전은 입력된 영상의 객체나 패턴을 인식하여 필요한 정보를 이끌어 내는 것을 말한다. 컴퓨터 비전 기술을 응용하여 제품의 결함을 검사하거나, 문자 인식, 얼굴인식, 움직이는 물체의 검출 및 추적하는 분야에 사용된다[5]. <그림 1>과 같은 절차를 통해 컴퓨터가 사람처럼 인지하고 이해할 수 있다[6].

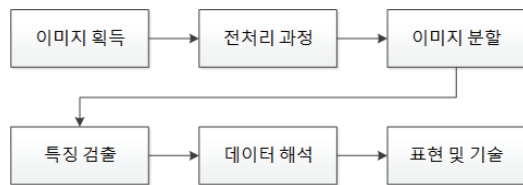


그림 1. 컴퓨터 비전의 절차도
Figure 1. Computer Vision Process

2.2.3 객체 탐지 알고리즘

객체의 형태가 일정하지 않고 색의 구분도 명확하지 않는 연기라는 객체의 방향을 감지하기 위해서는 <그림 2> 와 같이 다양한 객체 탐지 알고리즘이 있다[7]. 그중에 Object Detection의 기술 중에 하나인 Optical Flow 기술을 사용한다[8]. 이 기술은 차영상(difference image)의 진화 버전으로 이전 프레임에서의 특정 포인트를 추출하여 현재 프레임에서 같은 특정 포인트를 찾아 얼마만큼의 위치 변화가 있었는지를 분석하는 방식이다[9]. 특정 포

인트의 분포가 움직이는 물체 부분에 밀집되게 되어 있어 객체 위치 인식 및 추적용으로 사용된다. 그러므로 물체가 어떻게 움직이고 있는지를 판단하기에는 가장 좋은 알고리즘이라고 할 수 있다.

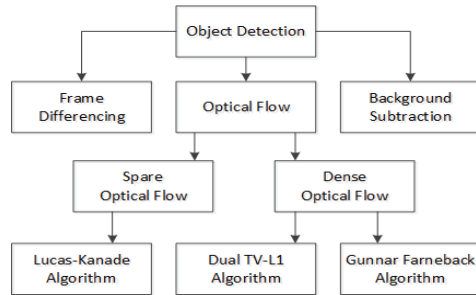


그림 2. 객체 탐지의 흐름도
Figure 2. Flow Chart of Object Detection

optical flow 기술 중에는 밀집 광학 흐름(dense optical flow) 알고리즘과 희소 광학 흐름(sparse optical flow) 알고리즘이 있다. 그중에 밀집 광학 흐름 알고리즘인 파네백(Gunnar Farneback)이 개발한 다항식 확장 알고리즘을 활용하면 이미지에 있는 모든 픽셀을 모션 벡터로 할당하고 밀도가 높은 광학 흐름을 찾을 수 있다. 특정 포인트를 검출하는 희소 광학 흐름 알고리즘의 검출 방식보다 객체 검출에 대한 오류가 적고 움직임 데이터의 정확도가 높아진다[10].

희소 광학 흐름에서 가장 중요한 특정 포인트를 제대로 검출하지 못하면 객체의 움직임에 대한 정확성이 떨어진다. 하지만 밀집 광학 흐름은 모든 픽셀에 대해 밀도가 높은 광학 흐름을 찾기 때문에 정확도가 높다[11].

2.2.4 파네백 다항식 확장 알고리즘

Gunnar Farneback[12] 알고리즘의 기본 개념은 모든 포인트에서 다항식을 이미지에 일치되도록

하여 이미지를 함수로 계산하는 것이다. 이미지를 이차 다항식과 각 포인트를 연관시키는 표현으로 변환한 후 다항식을 픽셀 주변 영역의 중심에 좀더 가까운 포인트에 가중치가 적용되도록 계산한다. 이미지 일부분의 작은 변위는 같은 포인트에서 다항식 확장 계수에 해석적으로 계산될 수 있어 해당 변위의 크기를 구할 수 있다.

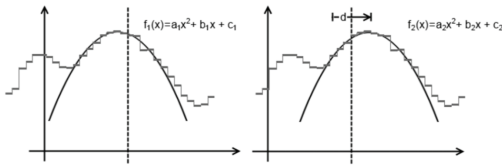


그림 3. 변위의 크기
Figure 3. Size of Change in Position

<그림 3>과 같이 1차원의 경우 이미지는 작은 변위 d 의 왼쪽과 오른쪽의 그레이 히스토그램으로 표현된다. 임의의 점선은 포물선 곡선에서 근처의 강도 값에 맞출 수 있다. 평활 함수 $I(x)$ 와 작은 변위의 근사 결과로 얻은 2차 다항식 f_1 과 f_2 는 수식 (1)과 같은 해석 공식의 관계를 갖는다.

$$\begin{aligned} a_2 &= a_1, \\ b_2 &= b_1 - 2a_1d, \\ c_2 &= a_1d^2 - b_1d + c_1. \end{aligned} \quad (1)$$

그리고 피팅 전과 후의 계수 값이 주어지면 두 번째 관계식인 f_2 를 사용해 수식 (2)로 정의된다.

$$d = -\frac{1}{2}a_1^{-1}(b_2 - b_1) \quad (2)$$

a 는 대칭 행렬, b 는 벡터, c 는 스칼라 이다[13]. 이러한 방법을 통해 이미지를 연속적인 표면에 근사하는 해석학적 기법으로 광학 흐름을 계산한다.

2.3 스택커 크레인의 정의 및 특징

스태커 크레인은 rack형 자동화 창고에서 제품을 보관하고 꺼내는 역할을 하는 장비이다[14]. <그림 4>와 같은 구조를 가지고 있으며 스택커 크레인은 하부의 레일을 따라 이동한다[15].

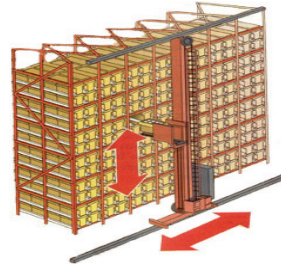


그림 4. 스택커 크레인의 구조
Figure 4. Structure Of stacker crane

스태커 크레인은 연기가 발생하는 rack의 위치로 이동하고 설치되어 있는 카메라에서 연기 탐지를 위한 영상을 전달하기에 적합하다.

3. 연기 방향을 탐색하기 위한 시스템 설계

본 논문에서는 서버 PC와 연기 감지기의 연결, 서버 PC와 스택커 크레인의 연결, 서버 PC와 카메라의 연결로 설계하여 서버 PC에서 실시간으로 연기가 감지된 연기감지기의 위치 정보와 스택커 크레인의 제어 및 카메라의 영상 데이터를 받을 수 있도록 구성한다.

<그림 5>와 같이 SDBOX(Smoke Detect Box)에서 연기 감지기로부터 받은 데이터를 취합하여 서버 PC로 전달한다. 서버 PC는 연기가 감지된 위치로 스택커 크레인을 이동시킨다. 스택커 크레인에 설치되어 있는 카메라 영상을 서버 PC에서 획득한다. 획득한 영상에 대해 이미지 프로세싱을 진행하여

서버 PC에서 스택어 크레인에 위치정보를 전달할 수 있도록 <그림 7>과 같이 스택어 크레인을 작동시키는 PLC와의 Ethernet 통신으로 연결한다.

<그림 8>은 연결된 PLC에 목적지 rack의 위치정보를 write 하고 스택어 크레인의 이동 명령을 전송하면 해당 rack의 위치로 이동한다.

```
// 스택어 crane 이동 신호값
short[] Ipscheck = new short[1];
Ipscheck[0] = 1;
// 이동할 rack의 목적지 정보 (열,연,단)
short[] Ipsdata = new short[3];
Ipsdata[0] = Convert.ToInt16(AgingColumn);
Ipsdata[1] = Convert.ToInt16(RackColumn);
Ipsdata[2] = Convert.ToInt16(RackRow);
// 스택어 crane에 목적지로 열연단 정보를 write ( DataWriteaddr="W210" )
WriteFireDataCheck = plcEtherDriver.WRITE_WORD_BLOCK(this.mDataWriteaddr, 3, ref Ipsdata[0]);
// 스택어 crane의 이동 신호를 write ( FireRequest="W200" )
WriteCheck = plcEtherDriver.WRITE_WORD_BLOCK(this.mFireRequest, 1, ref Ipscheck[0]);
```

그림 8. 스택어 크레인의 이동 명령
Figure 8. Movement Command of stacker crane

3.4 서버 PC와 카메라의 연결

서버 PC와 카메라는 Rtsip 및 http 프로토콜을 이용하여 아래의 함수를 이용하여 접속하여 전송되는 영상을 객체 추적 알고리즘을 적용하여 연기의 방향을 분석할 수 있게 된다.

```
string str = string.Format("http://{0}&rtsp={1}", nzdata.Http, nzdata.Rtsp);
base.NZCommand(NZ_CMD.NZ_CMD_CONNECT, nzdata.Ip, str, "admin", "admin");
```

그림 9. 카메라의 연결 함수
Figure 9. Connect Function of Camera

<그림 9>과 같이 카메라에 연결하기 위한 함수를 설계하였고, <그림 10>에서는 카메라와 연결하기 위한 clsCCTV 클래스를 정의하였다.

```
public class clsCCTV
{
    public frmCCTVMain frm;
    public int MyPort = 8080;
    public string Ip = "192.168.0.10";
    public int Rtsp = 554;
    public int Http = 8080;
}
```

그림 10. 카메라 속성 클래스
Figure 10. Camera Properties Class

3.5 연기 진행 방향의 알고리즘

<그림 11>과 같이 연기의 진행 방향을 결정하기 위해서 영상을 상하좌우 4방향으로 나누었다. 그리고 각각의 영역 안에서 움직이는 객체만을 방향 데이터로 사용하였다. 그래서 영역별 연기의 진행 방향을 계산하고 진행 방향에 대표 방향을 계산하였다. 이러한 진행 방향 데이터를 취합해 연기의 발생 방향과 방향에 대한 확률을 계산하였다.

```
// 상하좌우 4방향 영역 지정
Object UP[0] = Slice(Iplmage_original, new OpenCvSharp.CvPoint(SliceWidth, 0), new CvSize(SliceWidth * 2, iSliceHeight));
Object DOWN[0] = Slice(Iplmage_original, new OpenCvSharp.CvPoint(SliceWidth, Iplmage_original.Height - (iSliceHeight)), new CvSize(SliceWidth * 2, iSliceHeight));
Object LEFT[0] = Slice(Iplmage_original, new OpenCvSharp.CvPoint(0, iSliceHeight), new CvSize(SliceWidth, iSliceHeight * 2));
Object RIGHT[0] = Slice(Iplmage_original, new OpenCvSharp.CvPoint(Iplmage_original.Width - (iSliceWidth), iSliceHeight), new CvSize(SliceWidth, iSliceHeight * 2));

// 연기의 방향데이터로 진행방향 계산
public Object[] OpticalFlowFameback_ob(Object[] obj, RichTextBox rich, PictureBox pic) {
    Cv_CalcOpticalFlowFameback(prev, curr, flow, pyrScale, level, winSize, iterations, polyN, polySigma, I_KFlowFlag.PyrAReady);
    for (int i = 0; i < cols; i++) {
        for (int j = 0; j < rows; j++) {
            int dx = (int)flow[i, j][0];
            int dy = (int)flow[i, j][1];
            int64 dx64 = dx;
            int64 dy64 = dy;
            if (dx != 0 || dy != 0) {
                Cv_DrawLineOptical, Cv_Pointf(i, j), Cv_Pointf(i + dx, j + dy), Cv_Color.Blue, 1, LineType.AntiAlias, 0);
                double dx_ = Math.Abs(dx);
                double dy_ = Math.Abs(dy);
                if (dx_ > dy_) {
                    if (dx > 0) { iRight++; } else { iLeft++; }
                } else {
                    if (dy > 0) { iDown++; } else { iUp++; }
                }
            }
        }
    }
    sRtn += "Up : " + iUp.ToString() + "\r\n";
    sRtn += "Down : " + iDown.ToString() + "\r\n";
    sRtn += "Left : " + iLeft.ToString() + "\r\n";
    sRtn += "Right : " + iRight.ToString() + "\r\n";
    sRtn += "Detect Count : " + iSum.ToString();
    rich.Text = sRtn;
}

// n 개의 프레임마다 연기의 발생 방향과 확률 계산
switch (int_Sort_Hi(CurrUp, iCurrDown, iCurrLeft, iCurrRight))
{
    case "UP":
        pic_Arrow_FrameL.Image = Properties.Resources._6;
        double d1 = (dUP / dSum) * 100;
        d1 = Math.Round(d1, 2);
        lbl_Frame_PercentL.Text = d1.ToString() + "%";
        break;
    case "DOWN":
        pic_Arrow_FrameL.Image = Properties.Resources._12;
        double d2 = (dDOWN / dSum) * 100;
        d2 = Math.Round(d2, 2);
        lbl_Frame_PercentL.Text = d2.ToString() + "%";
        break;
    case "LEFT":
        pic_Arrow_FrameL.Image = Properties.Resources._3;
        double d3 = (dLEFT / dSum) * 100;
        d3 = Math.Round(d3, 2);
        lbl_Frame_PercentL.Text = d3.ToString() + "%";
        break;
    case "RIGHT":
        pic_Arrow_FrameL.Image = Properties.Resources._9;
        double d4 = (dRIGHT / dSum) * 100;
        d4 = Math.Round(d4, 2);
        lbl_Frame_PercentL.Text = d4.ToString() + "%";
        break;
}
```

그림 11. 연기 진행 방향의 알고리즘
Figure 11. Algorithms in the direction of smoke progression

4. 실험 및 평가

4.1 연기의 진행방향 검출

연기의 방향을 분석하기 위해 화면 전체를 분석하는 것이 아니라 각각의 면에 일정 크기만큼의 ROI(Region Of Interest)[16]를 기준으로 파네백 다항식 알고리즘을 적용하여 연기의 흐름을 검출되게 하였다.

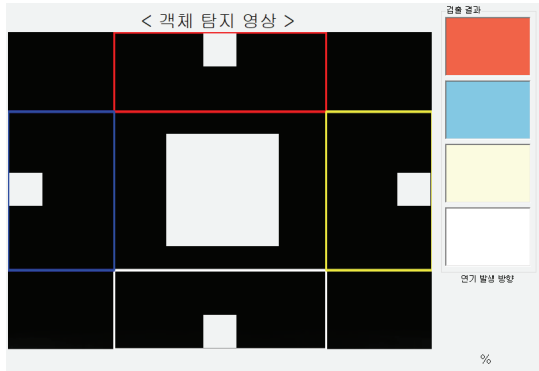


그림 12. 관심영역 설정
Figure 12. Set Region Of Interest(ROI)

<그림 12>와 같이 4개의 방향에 ROI가 설정된 영역만을 사용함으로써 연기의 방향을 결정하는 것에 방해되는 영역인 중심부의 데이터가 제외되어 연기 검출의 정확도를 높였으며, 불필요한 연산을 처리하지 않아 연산 시간이 개선되어 파네백 다항식 알고리즘의 단점을 보완했다. 이와 같은 알고리즘을 적용하여 획득한 영상을 기반으로 연기의 방향을 찾아내는 실험을 진행한다.

첫 번째 테스트 케이스는 연기의 방향이 오른쪽에서 왼쪽 위로 이동하는 영상을 사용한다. <그림 13>과 같이 ROI로 지정한 사각의 영역에 움직임이 있는 객체를 탐지하여 픽셀들이 움직이는 방향을 분류하였다. 객체들의 방향 검출 결과 중에 가장 많은 건수를 기록한 방향을 해당 ROI의 객체 탐지

방향으로 결정하였다. 그리고 4개의 ROI 중에서 가장 많은 객체가 검출된 ROI의 객체 탐지 방향이 해당 영상 프레임의 최종 방향으로 결정된다. 이러한 알고리즘으로 연기 방향 데이터를 취합하여 연기의 발생 위치를 확인했다.

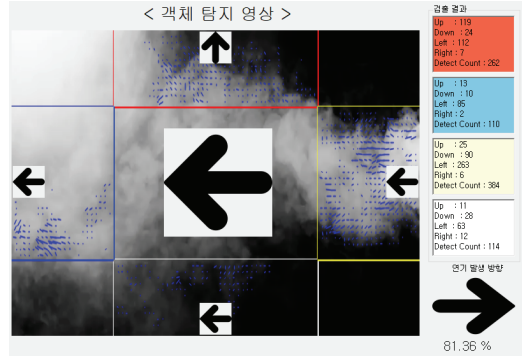


그림 13. 연기의 방향 검출
Figure 13. Directional Detection of Smoke

두 번째 케이스는 실제 현장에서 얻어진 영상을 기준으로 연기의 방향을 검출하였다.



그림 14. rack의 2x3 구조
Figure 14. 2x3 Structure of rack

<그림 14>와 같은 2x3 구조에 좌측 상단의 rack에 1개의 연기 감지기가 설치되어 있다. 연기 감지기가 연기를 감지한 위치에서 카메라를 통해 획득한 영상으로 아래와 같은 연기의 흐름이 검출되었다.

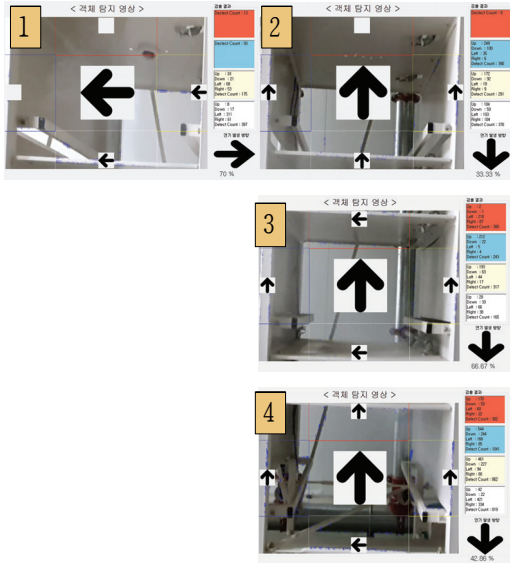


그림 15. 각 rack의 연기 추적 결과
Figure 15. Smoke Tracking Results For each rack

<그림 15>와 같이 1번 화면에 설치된 연기감지기에서 연기를 감지했고 연기의 진행 방향은 오른쪽에서 왼쪽으로 검출되어 스택어 크레인을 오른쪽으로 이동시켰다. 2번 화면에서는 연기의 진행 방향이 아래쪽에서 위쪽으로 검출되어 스택어 크레인을 아래쪽으로 이동시켰다. 이와 같은 방법으로 연기의 진행 방향을 역 추적한 결과 4번의 위치에서 연기가 시작되어 1번에 있는 연기감지기에서 연기가 감지된 것을 알 수 있다. 그래서 이미지 프로세싱으로 연기를 추적하여 연기가 발생하는 위치를 확인하였다.

5. 결론

본 논문은 객체 탐지 알고리즘으로 연기의 진행 방향을 검출하고 연기의 발생지를 확인했다. 소량의 연기감지기를 설치하고 영상을 이용한 이미지 프로세싱을 적용함으로써 모든 rack에 연기감지기를

설치할 필요가 없어졌다. 또한, 모든 rack에 연기감지기가 설치되지 않은 rack형 창고에서 추가적으로 연기감지기를 설치할 필요 없이 카메라를 설치하여 연기가 발생하는 위치를 찾을 수 있게 되었다.

향후 연구 과제로는 연기 감지기의 오작동 여부를 확인하기 위한 이미지 프로세싱 방안을 연구하고, 연기의 발생 위치를 판단하기 위한 방법으로 다양한 카메라들을 이용하여 연구해보면서 시스템의 기능을 향상시키는 연구가 필요하다.

References

- [1] Advantages of Warehouse Racks, <http://ko.chinatrolleybasket.com/info/what-are-the-advantages-of-warehouse-racks-com-26896595.html>, Oct. 2019.
- [2] U. H. Kim, *Fire risk analysis according to the nature of the warehouse*, Korean Institute Of Fire Science & Engineering, 2014.
- [3] T. Y. Lee, *Development of smart fire monitoring system for forest fire prevention*, A master's thesis of Korea University, 2017.
- [4] J. H. Kim, *A study on analyses of the Rey-O Test Image using image processing*, A master's thesis of Chung-Ang University, 2018.
- [5] D. G. Kim, *OpenCV programming to learn by python*, Kame publisher, 2018.
- [6] M. H. Kim, *Application of deep learning technique for detecting construction worker wearing safety helmet based on computer vision*, A master's thesis of Pukyong National University, 2019.
- [7] J. S. Lee, H. J. Lee, D. K. Lee, and S. J. Oh, *New scheme for smoker detection*, The Journal of Korean Institute of Communications and Information Sciences, Vol. 41, No. 09, pp. 1120-1131, 2016.

- [8] D. K. Panda, *Motion detection, object classification and tracking for visual surveillance application*, A master's thesis of National Institute of Technology Rourkela, 2012.
- [9] H. S. Parekh, D. G. Thakore, and U. K. Jaliya, *A survey on object detection and tracking methods*, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 2, No. 2, 2014.
- [10] S. H. Son, J. W. Jeon, I. J. Lee, J. H. Cha, and H. C. Choi, *Tiny drone tracking with a moving camera*, Journal of Broadcast Engineering, Vol. 24, No. 5, pp. 802-812, 2019.
- [11] D. H. Yun, *OpenCV4 programming using C# and Python*, Wikibook, 2019.
- [12] G. Farneback, *Two-frame motion estimation based on polynomial expansion*, Scandinavian Conference on Image Analysis, pp. 363-370, 2003.
- [13] A. Kaehler, *Learning OpenCV 3*, O'Reilly, 2016.
- [14] J. J. Kim, Y. S. Lee, S. R. Shin, H. T. Lee, and D. H. Jo, *Dynamic characteristics analysis of stacker crane for automatic warehouse*, The Korean Society of Mechanical Engineers, pp. 436-441, 2001.
- [15] Automatic warehouse (AS/RS), <https://scminsight.tistory.com/9>, Oct. 2016.
- [16] T. H. Kim, *A Study on the Detection of Smoke Emitted and Diffused from a Blast Furnace with CCTV Imagery*, A master's thesis of Incheon National University, 2017.

Rack형 창고에서의 연기감지기와 영상처리를 이용한 고 신뢰성 화재 감지 시스템 개발

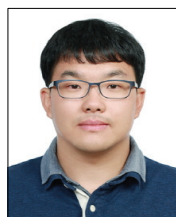
이형대¹, 김재웅²

¹공주대학교 컴퓨터공학과 석사과정

²공주대학교 컴퓨터공학과 교수

요약

rack형 창고에서의 화재관리는 굉장히 중요하고 필수적인 요소이다. 일반적으로 화재는 먼저 연기가 생성되고 그다음 화염이 발생하여 화재로 이어진다. 일반적으로 연기감지 시스템은 모든 rack 공간에 연기감지기가 설치되어 있으며 초기에 연기를 감지하여 정확한 위치를 전달한다. 그러나 rack 안에 연기감지기를 설치할 공간이 부족하거나 모든 rack에 연기감지기를 설치하지 않은 경우에는 최상단에 1개의 연기감지기를 설치하여 주변의 rack을 그룹으로 관리한다. 이와 같은 구성에서 연기감지기가 연기를 감지하면 그룹으로 묶인 rack들 중에서 어떤 rack에서 연기가 발생하고 있는지 판단할 수 없다. 따라서 본 연구에서는 연기가 발생하는 rack의 위치를 효율적으로 찾기 위한 시스템을 제안하고자 한다. 이를 위해 rack형 창고에 제품을 입고 및 출고 시키는 스택커 크레인에 카메라를 설치한다. 고정된 카메라에서 영상 프레임을 실시간으로 전송받아 차영상(difference image)이 안정적으로 계산되도록 한다. 얻어진 영상은 객체의 움직임을 탐지하는 알고리즘인 밀집 광학 흐름이라는 기술을 사용해 연기의 방향을 검출한다. 검출된 방향으로 스택커 크레인을 이동시켜 연기의 방향을 찾는 방법을 반복하여 연기가 발생하는 rack의 위치를 찾아내는 고 신뢰성 화재 감지 시스템의 개발을 목적으로 한다.



Hyoung-Dae Lee received the bachelor's degree in the Department of Computer Engineering from National Institute for Lifelong Education in 2007. He is working in IfonSystem Company. His current research interests include Sensor, Communication, Image Processing, Software Engineering. He is a member of the KKITS.

E-mail address: lhdkkang@naver.com

Jae-Woong Kim received the bachelor's degree and the M.S. degree in the Department of Computer Engineering from the Jungang



University in 1983 and 1988, respectively. He received the Ph.D. degree in the Department of Computer Engineering from Daejun University in 2000. He has been a professor in the Department of Computer Engineering at Kongju National University since 1992. His current research interests include software engineering. He is a life member of the KKITS.

E-mail address: jykim@kongju.ac.kr