

# 영상처리를 이용한 반복적 작업의 측정에 관한 연구

이정철<sup>1</sup> · 심익수<sup>2</sup> · 김남주<sup>3</sup> · 박관권<sup>4</sup> · 박진우<sup>2</sup>

<sup>1</sup>동부제강 / <sup>2</sup>서울대학교 산업공학과 / <sup>3</sup>대우정보시스템 / <sup>4</sup>영산대학교 정보경영학부

## A Study on Measurement of Repetitive Work using Digital Image Processing

Jeong-Cheol Lee<sup>1</sup> · Eok-Su Sim<sup>2</sup> · Nam-Joo Kim<sup>3</sup> · Chan-Kwon Park<sup>4</sup> · Jin-Woo Park<sup>2</sup>

Previous work measurement methods need much time and effort of time study analysts because they have to measure required time through direct observations. In this study, we propose a method which efficiently measures standard times without involvement of human analysts using digital image processing techniques. This method consists of two main steps: motion representation step and cycle segmentation step. In motion representation step, we first detect the motion of any object distinct from its background by differencing two consecutive images separated by a constant time interval. The images thus obtained then pass through an edge detector filter. Finally, the mean values of coordinates of significant pixels of the edge image are obtained. Through these processes, the motions of the observed worker are represented by two time series data of worker location in horizontal and vertical axes. In the second step, called the cycle segmentation step, we extract the frames which have maximum or minimum coordinates in one cycle and store them in a stack, and calculate each cycle time using these frames. In this step we also consider methods on how to detect work delays due to unexpected events such as operator's escapement from the work area, or interruptions. To conclude, the experimental results show that the proposed method is very cost-effective and useful for measuring time standards for various work environment.

### 1. 서론

작업측정(work measurement)을 통하여 세워진 표준시간은 생산 계획과 공정계획을 수립하는 데 기초가 될 뿐만 아니라 생산 실적을 평가하는 척도가 된다. 지금까지 이러한 작업측정 방법으로 스톱워치(stopwatch)를 이용하여 직접 측정하는 방법, 기본 요소 동작에 대한 기정시간 자료(predetermined time data)를 이용하여 산출하는 방법, 워크 샘플링 법, 여러 작업에 대해 표준화된 데이터(standard data)를 이용하는 방법 등이 사용되었다. 그러나 기존의 방법들은 관측자가 작업시간을 직접 측정함으로써 많은 시간과 비용을 수반하거나, 기존의 자료를 이용하여 간접적으로 추정함으로써 실제 작업과의 차이가 발생할 수 있는 단점들을 가지고 있다.

최근 영상처리 기술을 산업 현장에 활용하는 연구가 여러 분야에서 활발히 진행되고 있다. 손의 자세와 동작을 이용한 전자기기와 컴퓨터 제어에 관한 연구(Christopher, 1996; Freeman,

1995; Toshiba 1998), 몸에 부착된 센서를 통해 전달되는 데이터를 이용하여 사람의 움직임을 3차원으로 인식하여 의료분야나 컴퓨터 애니메이션 등에 활용하는 연구(정순기, 1998), 카메라를 통해 입력되는 이미지 시퀀스를 분석하여 사람의 동작을 인식하거나 추적하는 것과 관련된 연구(양윤모, 1998; Bobick, 1996; Davis, 1999; Fejes, 1999), 그리고 생산현장에서 부품의 번호를 인식하거나 제품의 상태를 감시하여 불량을 감지하는 것과 관련된 연구(홍기상, 1996) 등이 그것이다.

사람의 동작 분석에 대한 기존의 연구들은 작업측정의 영역에 적용될 때 다음과 같은 한계점들을 가지고 있다. 첫째, 움직임의 정보를 얻기 위해 각종 센서와 장비를 사용하는 경우 작업동작에 영향을 미치게 되므로 실제의 작업시간과 다른 측정 결과를 얻을 수 있다. 둘째, 움직임을 인식하기 위해 연속된 프레임에서 신체 부분에 대한 특징추출(feature extraction)과 매칭(matching) 과정을 수반하는 경우 많은 계산 과정이 필요하며, 또한 추출한 특징 벡터(feature vector)에서 작업의 패턴을 인식하는 과정이 매우 어렵다. 셋째, 동작 인식과 관련된 연구들의

경우, 정해진 동작의 집합에 대해서만 인식을 수행할 수 있으므로 다양한 작업형태에 대해 적용하는 데는 한계가 있다.

본 연구에서는 영상처리 기술을 이용하여 보다 효율적으로 작업시간을 측정할 수 있는 접근방법을 제시하고자 한다. 이러한 접근방법은 크게 동작표현(motion representation) 단계와 사이클 분리(cycle segmentation) 단계의 두 단계로 구성된다. 동작 표현 단계에서는 먼저 인접한 이미지 사이의 차연산(image subtraction)을 통해 고정된 배경과 움직이는 대상을 분리한다. 이렇게 얻어진 동작 이미지에서 윤곽선을 추출(edge extraction)하고 한계값(threshold)을 넘는 픽셀의 위치 평균을 계산함으로써 잡음의 효과를 줄이면서 이미지 내에서 움직이는 작업자의 수평방향과 수직방향의 위치변동을 추정한다. 이러한 과정을 통해 연속된 작업자의 동작은 각 축에 대해 일정한 패턴이 반복되는 시계열(time series) 데이터로 표현된다. 사이클 분리단계에서는 이렇게 표현된 동작의 위치를 추적하면서 사이클을 분리하는 데 사용되는 프레임을 추출하고, 이 프레임을 이용하여 사이클 시간을 계산한다. 정상적인 작업 상황에서도 발생할 수 있는 카메라 범위 이탈(camera area escape) 상황과 일시적인 작업 지연(work delay) 상황을 처리하기 위하여 프레임 타임(timer)과 카운터(counter)를 이용한다.

본 논문은 다음과 같이 구성되어 있다. 2절에서는 본 연구에서 제시하는 접근방법인 작업 동작의 표현과 사이클 분리과정을 설명하였고, 3절에서는 이를 바탕으로 설계한 작업측정 시스템의 구조와 작동 원리에 대하여 설명한 뒤, 4절에서는 구현된 시스템의 검증에 대해 여러 작업을 대상으로 수행했던 실험과 그 결과를 요약하였다. 끝으로 5절에서 본 연구의 결과를 정리하고 후속 연구과제를 제안하였다.

## 2. 접근방법

### 2.1 작업 동작의 표현 (Motion Representation)

만약 관측을 통해 반복적 수행 작업의 수행 시간을 측정하는 경우를 생각하여 보자. 그 경우 우선 반복되는 작업의 패턴을 인식한 후, 사이클을 분리하기 위해 기준이 되는 동작이나 위치 등을 선정한 다음, 작업을 관찰하면서 선정한 기준까지의 시간을 기록한다. 드릴 작업의 경우, 사이클을 분리하는 기준으로 부품을 상자에서 집는 동작이나, 부품을 기계에 올려놓는 동작 등을 사용할 수 있다. 이러한 동작들은 모두 관측하기에 용이한 특징을 가지고 있다.

영상처리를 이용하여 작업 동작의 패턴을 인식하기 위해서는 먼저 작업 동작에 대한 표현방식이 정의되어야 한다. 이러한 표현방식을 위하여 첫째, 동작을 전체로서 인식할 것인가, 또는 신체 각 부분과 관절의 동작으로 인식할 것인가를 결정해야 한다. 둘째, 각각의 동작을 표현하는 특징인자들을 결정해야 한다. 이러한 인자들은 동작의 특징을 잘 반영해야 하며,

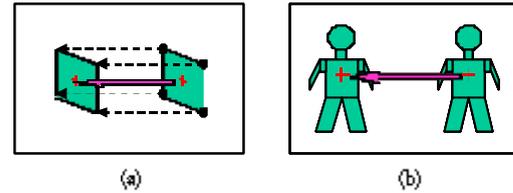


그림 2. 중심의 이동에 의한 움직임 표현.

작업 동작의 패턴을 쉽게 감지할 수 있어야 한다. 셋째, 이러한 인자들을 획득하기 위해 센서나 다른 장비들을 이용할 것인가, 아니면 영상처리를 이용할 것인가를 결정해야 한다.

사람의 동작을 각 신체 부분이 아닌 전체로서 파악하게 되면, 동작 표현의 정확도는 감소하지만, 계산과정이 훨씬 빨라지며 동작의 전체적인 특성을 쉽게 파악할 수 있다. 사람의 동작을 전체로서 표현하기 위해서는 영상 내에 사람의 존재영역을 찾아낸 뒤, 이 영역의 무게중심(center of gravity)의 위치를 계산할 수 있다. 이러한 표현방식은 영역에 기반한 형상 표현(region-based shape representation) 연구의 범주에 속하며(Koenderink, 1986) 등이 이러한 접근방법을 사용하였다. <그림 1>은 물체의 움직임을 중심의 이동으로 표현하는 방식을 나타낸다. (a)에서는 기하학에서의 도형의 평행이동을 나타낸다. 도형이 평행이동할 때 도형 위에 존재하는 모든 점이 이동하지만 무게중심의 이동만으로 이러한 이동을 효율적으로 나타낼 수 있다. 마찬가지로 (b)에서처럼 이미지 내에서 사람의 움직임을 사람이 차지하는 영역 중심의 이동으로 표현할 수 있다. 물론 이러한 표현방식에서 회전과 같은 운동을 나타내기 어렵다.

영상처리를 통한 동작분석을 용이하게 하는 반복 작업의 특징으로는 다음과 같은 것이 있다. 첫째, 일정한 동작의 패턴이 반복된다. 이것은 작업자의 동작을 전체로서 파악하더라도 그것의 패턴을 쉽게 발견할 수 있음을 나타낸다. 즉, 작업자의 작업동작은 그 속도와 위치가 매우 유사한 패턴을 따르기 때문에 영상에 포함된 움직임 정보도 이와 유사한 패턴을 따르게 된다. 그리고 비록 이상 동작이 발생할 경우에도 정상 상태와 비교함으로써 이를 감지할 수 있는 가능성이 높아진다. 둘째, 작업자의 동작의 범위가 제한되어 있다. 대체로 작업자는 일정한 범위에서 작업을 수행하기 때문에 작업자가 나타나는 위치를 예측할 수 있다. 이것은 하나의 고정된 카메라를 이용하여 작업의 영상을 획득하는 것이 용이함을 나타낸다. 물론 작업 범위가 매우 큰 작업이나, 복수의 작업자가 함께 수행하는 작업은 하나의 카메라로 영상을 얻기가 쉽지 않다. 끝으로 많은 작업들이 무게 중심의 변화가 크게 일어나는 동작을 하나의 작업 사이클 내에서 하나 이상 포함한다. 이는 사이클을 분리하는 데 중요한 단서가 된다. 팔을 뻗었다 접는 동작이나 전체 몸의 움직임의 방향이 왼쪽에서 오른쪽으로 바뀌는 동작들에 의하여 작업자의 무게 중심의 변화율의 방향이 바뀌게 된다. 이러한 순간은 수학적으로 말하면 무게 중심의 일차 미분계수가 0이 되는 지점을 나타낸다. 양윤모(1997)는 이러한 프레임을 키 프레임(key frame)이라 정의하였으며 수화인식을 위한 기

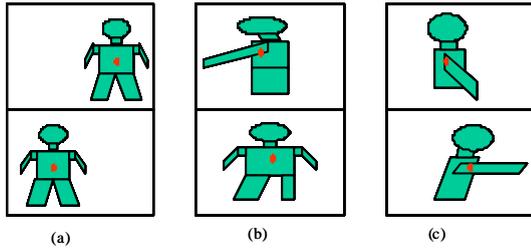


그림 2. 작업자의 중심이 변하는 작업의 주요 동작.

본 프레임으로 사용하였다.

<그림 2>는 작업자의 중심의 위치가 변하는 대표적인 작업의 동작을 나타낸다.

(a)는 작업자가 수평으로 이동하는 동작을 나타낸다. 작업자가 수평으로 이동할 때 무게 중심도 수평으로 이동하게 된다. 따라서 이미지 내에서 무게 중심의 x좌표는 증가 또는 감소한다. (b)는 허리를 중심으로 한 회전운동을 나타낸다. 이러한 동작은 물건을 집고 나르는 상황에서 많이 발생하는데, 이미지 내에서 팔 등의 위치의 변화로 인하여 무게중심의 위치가 변하게 된다. (c)는 팔을 펴는 동작으로서 기계에 부품을 올려놓거나 공이 끝난 후 내려놓을 때 흔히 발생하는 동작이다. 이러한 동작은 몸통이 굽혀지면서 팔이 앞으로 뻗어지기 때문에 이미지 내에서 사람의 무게 중심의 x좌표는 증가하게 된다.

<그림 3>은 문서를 정렬하는 작업의 주요 프레임들을 나타낸다. 일정한 시간 간격으로 얻은 것은 아니며, 작업의 특징을 잘 나타내는 프레임들을 포착(catch)하여 시간 순서에 따라 나열한 것이다. 이 작업은 4종류의 문서를 한 장씩 모아 하나의 묶음으로 만드는 작업이며, 소요시간은 약 8초가 소요된다.

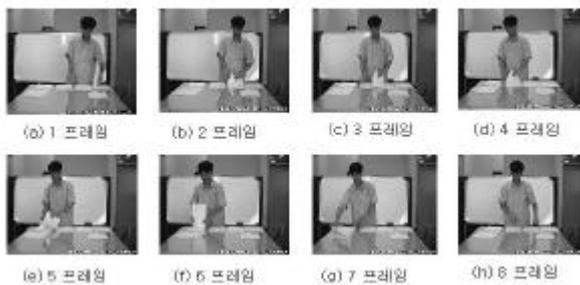


그림 3. 작업의 주요 프레임.

위 그림에서 작업자를 보면, 팔과 머리의 움직임이 많이 발생하나, 전체적으로 작업자의 위치가 오른쪽에서 왼쪽으로 이동했다가 다시 오른쪽으로 이동하는 동작이 반복되고 있음을 알 수 있다. 이러한 작업자의 동작을 분석하기 위해서는 영상에서 작업자의 동작을 분리시켜야 한다. 고정된 배경과 움직이는 대상의 분리는 인접한 이미지 사이의 차연산(image subtraction)을 통해 가능하다. <그림 4>는 <그림 3>의 이미지에 대해 차연산을 수행한 결과를 나타낸다.

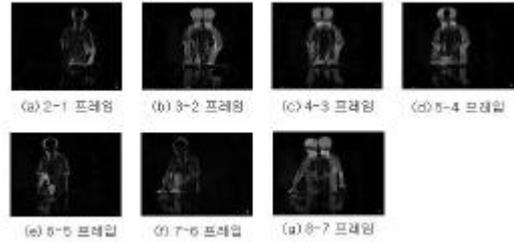


그림 4. 차연산 수행 결과.

위 그림을 보면, 차연산을 통하여 인접한 프레임 사이에서 변화가 없는 픽셀들은 검게 나타나고 변화가 발생한 픽셀들은 비교적 희게 나타나는 것을 알 수 있다. 즉, 고정된 배경은 사라지고 움직이는 대상이 분명하게 나타나게 된다. 물론 이러한 차 이미지(difference image)에는 영상을 획득할 때 여러 가지 원인으로 인하여 잡음이 포함되어 있으며, 이러한 잡음을 최소화시키는 것이 필요하다. 또한 차 이미지에는 동작의 존재여부에 대한 정보만 가지고 있지 동작의 방향에 대한 정보는 가지고 있지 않다. 그러므로 작업자의 동작에 대한 분석을 위해서 동작의 방향에 대한 정보를 얻어야 한다. 이러한 정보를 얻는 방법으로는 크게 cumulative difference image를 이용하는 방법과 optical flow를 이용하는 방법이 있다(Sonka, 1999).

본 연구에서는 차 이미지(difference image)를 이용하는 것이지만 cumulative difference image를 이용하는 방법처럼 참조 이미지(reference image)를 따로 두거나 차연산 결과를 하나의 이미지에 축적시키지는 않는다. 그 대신 <그림 4>와 같이 인접한 이미지 사이의 차연산 결과로 얻은 이미지에서 바로 동작의 위치에 대한 정보를 추정하였다. 참조 이미지 대신 인접 이미지와의 차연산을 수행한 이유는 다음과 같다. 첫째, 실제 작업 현장에서는 참조 이미지를 얻기 어려운 경우가 많다. 둘째, 참조 이미지를 사용하게 되면 영상에 포함될 수 있는 잡음에 약하다는 단점이 있다. 인접한 이미지 사이의 차 이미지에는 일정한 시간 간격의 작업자의 위치가 함께 포함되어 있기 때문에, 이 이미지에서 위치를 추정하는 것은 두 시점에서의 위치의 평균을 구하는 것이 된다. 따라서, 이러한 방법을 사용하면 추정한 작업자의 위치를 평활(smoothing)하는 효과를 가져오며, 시간 간격이 짧아질수록 이러한 효과는 줄어들는다. 그러므로 이 방법은 작업 동작의 패턴을 표현하는 데는 큰 영향을 주지 않는 대신, 영상에 포함되는 잡음을 줄이는 효과를 가져온다.

동작의 위치 정보를 추출하기 위해 차 이미지의 윤곽선을 추출한 뒤 정해진 임계값을 초과하는 픽셀에 대해서만 위치의 평균을 구한다. 이 값은 각 픽셀의 가중치를 1로 가정하였을 때, 무게 중심(center of gravity)을 나타내게 된다. 실제로는 프레임 사이의 시간간격이  $\Delta t$ 라면  $\Delta t$  동안 움직이는 모든 물체의 평균적인 위치를 나타낸다. 차연산을 수행하고 윤곽선을 추출하였을 때에도 결과 이미지에는 여러 잡음이 포함되어 있을 수 있다. 이러한 잡음에는 조명의 변화나 필름, 카메라나 보드에서 이미지 취득시 포함된 잡음 등이 있다. 이러한 잡음 효과

표 1. 중심 추정 결과

이미지	픽셀의 수																			중심		
	수평방향																			X	Y	
(a)	0	8	3	0	0	0	0	11	58	68	150	101	213	150	43	0	0	0	37	409	232	
(b)	0	10	0	0	0	1	0	68	95	136	152	151	220	118	40	0	0	0	0	350	236	
(c)	0	18	4	0	0	0	32	168	218	192	121	149	81	13	0	0	0	0	16	306	200	
(d)	1	6	4	0	0	145	240	239	220	115	97	80	2	0	0	0	2	0	1	38	263	208
(e)	1	0	0	1	17	158	240	168	185	25	68	0	0	0	0	0	2	0	0	39	267	206
(f)	0	6	20	55	69	152	134	130	75	29	101	0	0	0	0	0	0	0	0	37	252	240
(g)	0	0	19	60	69	60	184	167	145	107	175	109	57	1	0	0	0	0	1	30	270	200

를 줄이기 위하여 일정한 임계값을 초과하는 픽셀에 대해서만 고려하기로 하였다. 이렇게 한 이유는 실험결과 차 이미지에 포함된 잡음에 해당하는 픽셀이 대체로 희미하게 나타나기 때문이다.

<표 1>은 <그림 4>에서 얻어진 차 이미지에 대하여 중심을 추정한 결과를 나타낸다. 추정에 사용한 식은 3장에서 설명하였다. 중심을 추정하기 위하여 각각의 차 이미지에 대해 윤곽선을 추출한 뒤, 임계값보다 큰 값을 갖는 픽셀의 수를 계산한다. 이때 각 위치의 변화를 알아보기 위하여 이미지를 수직 방향으로 20개의 구간으로 나누어 각 구간에 대하여 임계값보다 큰 값을 갖는 픽셀의 수를 계산하였다. 이론적으로 움직임이 발생하지 않은 구간에서의 픽셀의 수는 0이 되어야 하나, 영상 획득 시 포함된 잡음으로 반드시 0이 되지는 않는다. 이러한 문제는 분석구간을 설정함으로 해결할 수 있으며 3절에서 보다 자세히 다루었다. 픽셀의 수의 분포를 보면 작업자가 오른쪽에서 왼쪽으로 이동할수록 이러한 픽셀이 나타나는 구간도 같은 방향으로 이동하는 것을 알 수 있다.

추정된 중심의 좌표는 이미지 상에서 작업자의 위치를 나타낸다. <그림 5>는 각 축에 대하여 추정한 중심의 변화를 나타낸다. 사용된 이미지의 크기가 640\*480이므로 중심의 좌표는 0에서 640까지의 값을 취할 수 있다. 중심의 x좌표의 값을 보면 작업자의 움직임을 잘 추적하고 있음을 알 수 있다. 수직방향에 대해서는 큰 움직임이 없기 때문에 수평방향에 비해 중심의 변화가 작다. 그러므로 이 작업의 동작은 수평방향에 대해 분석하는 것이 더 적절함을 알 수 있다. 반복적으로 수행하는

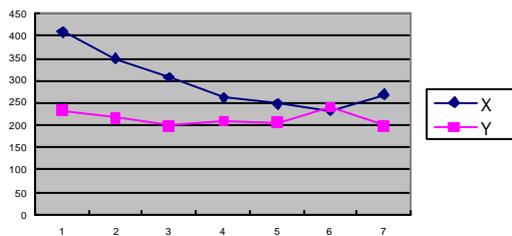


그림 5. 추정한 중심의 그래프.

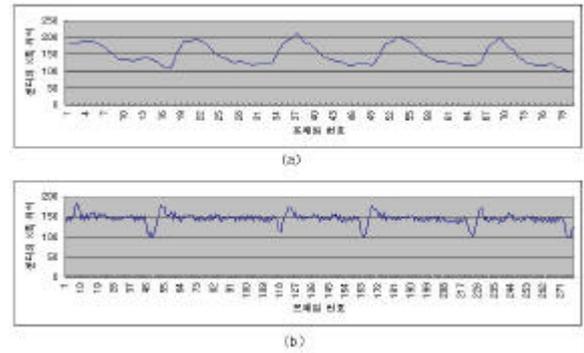


그림 6. 작업 그래프.

작업에 대해서는 이러한 중심을 보다 짧은 시간 간격으로 추적하면 더 정밀하게 패턴을 파악할 수 있다. 반복 사이클의 분리의 문제는 2.2절에서 다루도록 하겠다.

### 2.2 작업 사이클의 분리(Cycle Segmentation)

작업을 반복 수행할 때, 이 작업의 연속적 프레임에 대해 센터의 위치를 그래프로 나타내면 일정한 패턴이 발견되어진다. <그림 6>은 두 종류의 반복작업에 대하여 작업자의 센터를 추정하여 그래프로 나타낸 것의 일부분이다. 이러한 작업 그래프는 다음과 같은 특징을 가지고 있다.

- 유사한 패턴이 반복된다.
- 하나의 사이클 내에 최소값과 최대값이 존재하며, 이 점을 이용하여 사이클을 분리할 수 있다.
- 사이클 시간은 반복횟수마다 다르며 대체로 통계적 분포를 따른다.

이 작업 그래프는 일정한 시간 간격으로 사이클이 나타나지 않기 때문에 autocorrelation과 같은 시계열 분석의 기법들을 적용하기가 어렵다. 또한, 수리적 모델을 사용하여 추정하는 것도 기본적으로 사이클 분리과정이 선행되어야 한다.

본 연구에서 사용한 방법은 작업 그래프에서 반복적으로 나타나는 특징점들을 사용하여 사이클을 분리하는 것이다. 이러한 특징점은 작업 그래프에서 최대값 혹은 최소값을 갖는 점이다. 간략하게 특징점을 찾는 과정을 설명하면, 연속적으로 들어오는 이미지 프레임에서 최대값(혹은 최소값)을 갖는 프레임인지 판별하고, 최대값(혹은 최소값)을 갖는 프레임일 경우 현재의 프레임들 사이클 분리 프레임으로 선정한다. 그리고 나서 이전 사이클 분리 프레임과의 시간 간격을 계산하면 현재 사이클에 대한 수행시간을 계산할 수 있다.

작업을 수행하는 동안에 여러 가지 원인으로 인하여 작업 그래프에 변동이 발생할 수 있으며, 이로 인하여 사이클의 분리가 어려워질 수 있다. 이러한 원인으로서는 작업의 지연, 작업자의 카메라 범위 이탈, 작업자가 가림, 작업중지, 생리적인 이유 등에 의한 특이한 동작 등이 있다. <그림 7>은 이러한 변동은

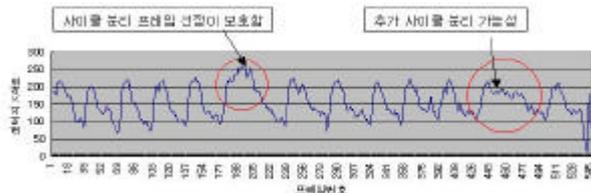


그림 7. 작업지연을 포함한 작업 그래프.

포함한 작업 그래프를 나타낸다.

6번째 사이클에서 작업의 방향이 오른쪽에서 왼쪽으로 바뀌는 시점에서 작업 지연이 발생하였다. 최대값을 갖는 프레임들을 사이클 분리 프레임으로 선정하였을 경우 관측자가 분할하는 시점보다 늦어지는 경향이 있다. 이보다 더 중요한 문제는 12번째 사이클에서 발생한다. 평균적인 사이클 시간에 비해 크게 길어질 경우 사이클 중간을 분할할 수 있는 가능성이 있다. 3절에서 이러한 문제들을 해결하기 위한 접근방법에 대하여 좀더 자세히 설명하였다.

### 3. 시스템의 설계 및 구현

본 절에서는 본 연구에서 설계한 작업측정 시스템의 전체 구성과 시스템을 구성하는 각 하부 모듈들의 작동 원리 및 구현에 대하여 설명하였다. <그림 8>은 시스템의 전체 구성을 나타낸다.

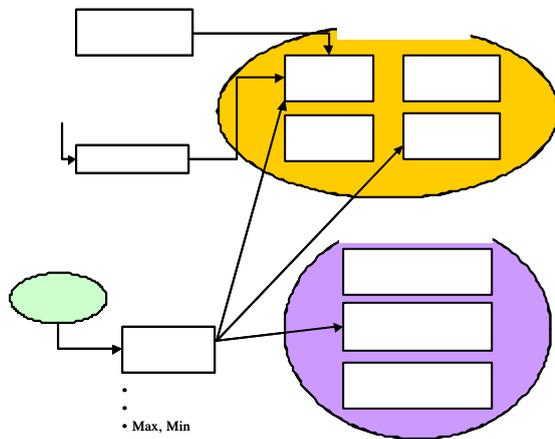


그림 8. 시스템의 전체 구성.

#### 3.1 전체적인 구성과 흐름

본 연구에서 설계한 작업측정 시스템의 전체적인 흐름은 <그림 9>와 같다. 전체적인 흐름은 다음의 네 가지 하부 단계로 구성된다.

- 차연산(Difference Operation) 단계
- 윤곽선 추출(Edge Filtering) 단계

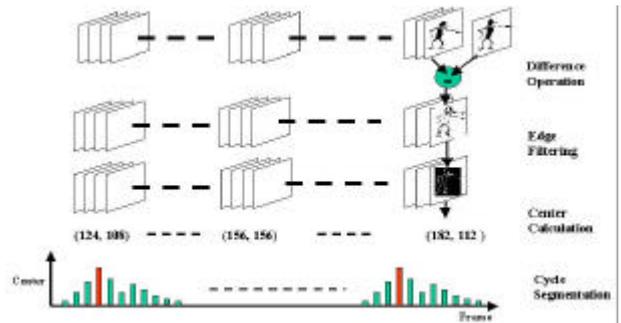


그림 9. 전체적인 흐름.

- 센터 추정(Center Estimation) 단계
- 사이클 분할(Cycle Segmentation) 단계

차연산 단계에서는 고정된 배경과 움직이는 대상을 분리한다. 연속적인 프레임으로 구성된 동영상 데이터가 입력되면 차연산 단계에서는 움직이는 물체와 움직이지 않는 배경을 분리하기 위하여 연속된 두 프레임의 차 이미지를 구하게 된다. 여기서 차연산은 픽셀 단위로 이루어지고 각 픽셀은 0에서 255 사이의 값을 갖기 때문에 0보다 작은 결과가 나올 경우 -1을 곱하여 준다. 차연산 단계에서 현재 가능한 입력 데이터로는 CCD 카메라를 통해 들어오는 영상 신호와 AVI(Audio Video Interleave) 동영상 파일이다. 프레임 포착기(Frame Grabber)를 통하여 일정 시간 간격으로 획득된 영상은 차연산 단계로 입력된다. 반면에 AVI 동영상 파일의 경우는 영상의 획득과정 없이 바로 파일로부터 읽어 버퍼에 저장된 프레임이 입력된다. 0보다 작은 값을 갖는 픽셀에 대하여서는 움직임 정보를 담고 있기 때문에 -1을 곱하였다.

윤곽선 추출 단계에서는 이렇게 얻어진 이미지의 윤곽선을 추출한다. 윤곽선은 물체의 형상 특징을 잘 나타내면서도 다루기에 효과적이다. 차연산 단계의 결과로 나온 이미지는 움직임의 존재를 나타낸다. 그러나 방향에 대한 정보는 없다. 이러한 방향에 대한 정보를 얻기 위하여 움직이는 물체의 위치를 추정해야 한다. 움직이는 물체의 위치 추정은 센터 추정 단계에서 수행되는데 여기에 사용하기 위해서 간편하면서도 움직임 정보를 잘 보존하고 있으며 포함된 잡음이 어느 정도 제거된 이미지를 얻는 단계가 윤곽선 추출 단계이다. 윤곽선의 추출은 이차 미분 연산자인 라플라시안 필터를 이용하여 컨볼루션(Convolution)을 수행함으로써 얻어진다. 물론 각 픽셀이 가질 수 있는 값은 0에서 255 사이이며 구간을 초과하는 값들은 경계값(0혹은 255)으로 설정하여 준다.

센터 추정 단계는 윤곽선 추출 단계에서 얻어진 윤곽선 이미지를 이용하여 하나의 대표값을 생성하는 단계이다. 여기서 사용하는 대표값은 윤곽선 이미지의 센터, 즉 무게중심이다. 식 (1)은 윤곽선을 구성하는 픽셀들의 센터를 추정하는 식이다. 이 식에서 센터의 X좌표와 Y좌표는 임계값보다 큰 값을 갖는 픽셀의 X좌표와 Y좌표의 평균을 나타낸다.

$$X_{center} = \frac{\sum X_{frame}}{(I_{frame} > T_{임계값})인 픽셀의 개수} \quad (1)$$

$$Y_{center} = \frac{\sum Y_{frame}}{(I_{frame} > T_{임계값})인 픽셀의 개수}$$

사이클 분할 단계는 이렇게 얻어진 센터의 좌표를 추적하여 사이클을 분할하는 단계이다. 사이클을 분할하는 데 사용되는 프레임은 분할 프레임이라고 정의한다. 이 프레임은 사이클을 분할하기에 적절한 프레임으로 설정한다. 현재의 프레임이 분할 프레임으로 판정되면 분할 프레임의 스택에 저장한다. 모든 프레임에 대한 처리가 끝난 후 이 스택에 저장되어 있는 프레임의 총 개수와 각 프레임의 번호를 이용하여 사이클의 개수와 사이클 시간을 계산하게 된다.

반복 작업에서 작업자의 움직임은 거의 유사한 형태를 띄게 된다. 그러므로 연속된 프레임에는 이러한 움직임의 패턴이 나타나게 된다. 이전 단계에서 추정된 센터의 위치의 그래프를 그려보면 일정한 패턴을 갖게 되는데 <그림 10>과 같은 형태를 가질 수 있다. 이러한 그래프는 수평방향(혹은 수직방향)의 센터의 위치의 변화를 나타낸 것이다. 여기서 작업은 왼쪽에서 오른쪽으로, 혹은 아래에서 위로 움직이는 형태의 작업이라 추측할 수 있다. 센터의 위치의 극대값 혹은 극소값을 갖는 프레임을 사이클 분리 프레임으로 사용한다.

센터의 극대값을 갖는 프레임을 사이클 분리 프레임으로 사용할 경우 문제가 되는 것은 하나의 사이클 내에 여러 개가 존재할 수도 있는 local maximum 값을 갖는 프레임이다. 이들은 사이클을 찾는 데 방해가 되므로 이러한 프레임들이 사이클 분리 프레임으로 선정되지 않도록 하는 방법이 필요하다.

이러한 local maximum 값을 갖는 프레임들이 사이클 분리 프레임으로 선정되는 것을 피하기 위하여 임계값을 사용하였다. 이 임계값은 현재까지 센터의 최대값을 갖는 프레임은 사이클 분리 프레임으로 선정하기 위한 지속시간을 나타낸다. 이 임계값이 너무 작게 되면 local maximum에 빠지기 쉽고, 반면에 너무 크게 되면 사이클을 분리하지 못하는 경우가 발생하게 된다. 작업 그래프에 따라 이 값을 적절히 조절해 주어야 하는데, 대체로 평균 사이클 시간의 1/2의 값을 이용하면 된다.

<그림 11>은 작업의 시퀀스에 대하여 작업 사이클을 분리

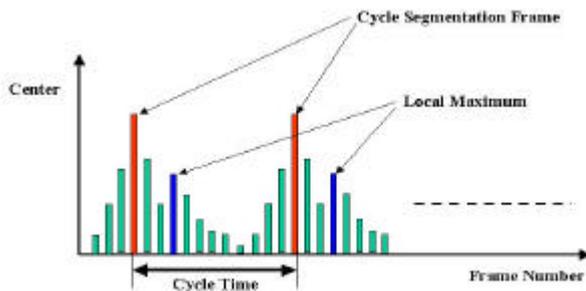


그림 10. 반복 작업에서 센터의 변화.

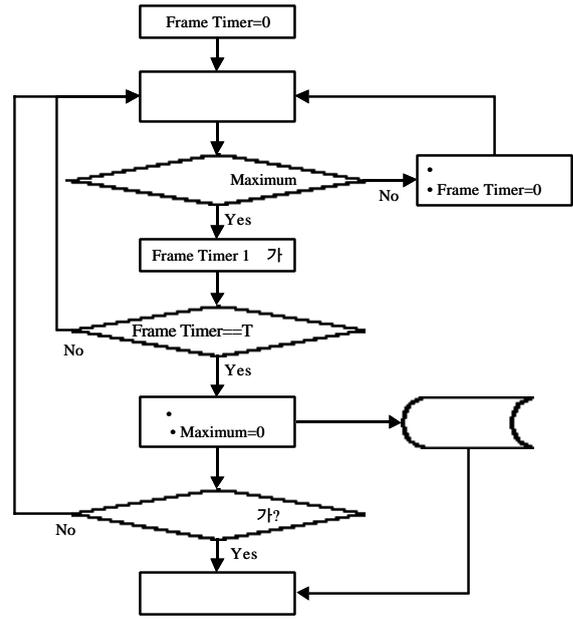


그림 11. 사이클 분리 알고리즘.

하는 알고리즘이다. 우선 프레임의 센터의 좌표가 maximum값으로 지속되는 시간을 나타내는 timer 변수를 0으로 설정한다. 카메라를 통해 들어오는 프레임을 하나 포착한 뒤 여러 과정을 거쳐 센터를 추정한다. 추정된 센터가 현재의 maximum 값보다 클 경우 현재의 센터 값으로 maximum 값을 교체하고 timer를 0으로 reset한 뒤 새로운 이미지를 다시 얻어 같은 과정을 반복한다. 만약 추정된 센터가 현재의 maximum 값보다 작을 경우 timer를 하나 증가시킨 뒤 임계값과 비교한다. 이 임계값은 사이클 분리 프레임으로 선정되기 위한 timer의 임계값이다. 만약 현재의 timer의 값이 이 값과 같아질 경우 현재의 프레임은 사이클 분리 프레임 스택에 저장한다. 이때 다음 분리 프레임 선정을 위하여 maximum 값을 0으로 설정한다. 이와 같은 과정은 마지막 이미지까지 계속된다. 마지막 이미지에 대한 분석을 마친 후 스택에 들어있는 분리 프레임을 이용하여 사이클을 분리하고 사이클 시간을 계산하게 된다.

그러나 실제 작업 그래프는 2.2절에서 제시하였던 여러 원인으로 인하여 불규칙한 특성을 띄게 된다. 정상적인 작업상황에서도 작업지연과 카메라 범위 이탈은 빈번하게 발생할 수 있으며, 사이클의 분리를 어렵게 한다. 3.2.3.3절에서는 이러한 문제들을 해결하기 위한 접근 방법을 설명하였다.

### 3.2 작업지연

작업지연은 평균적인 작업수행시간을 크게 초과하는 경우를 말한다. <그림 7>에서 나타나는 것처럼 시험지 정렬 작업에서는 종이를 2장 겹치는 경우, 또는 복사가 안된 종이 삽입된 경우와 같은 환경적인 요인과 재채기를 하는 경우와 같은 작업자 요인이 있다. 이러한 요인들은 평상시와는 다른 움직임

의 패턴을 생성하여 작업 그래프에 영향을 미친다. 그 결과 사이클 시간의 증가, 불규칙한 움직임 패턴 생성 등이 발생할 수 있다.

작업지연이 발생한 사이클을 분리하는 문제는 두 가지로 분류할 수 있다. 첫째는 사이클 분리 프레임으로 선정된 작업자의 주된 움직임의 방향이 바뀌는 지점에서의 작업지연의 경우와 사이클 중간에서의 작업지연의 경우이다.

### 3.2.1 사이클의 양끝에서의 작업지연

주된 움직임의 방향이 바뀌는 지점에서의 작업지연은 사이클 분리 프레임의 선정을 변화시키므로 인접한 두 사이클 시간을 계산하는 데 영향을 미치게 된다. 그러나 사이클을 분리하는 데는 영향을 미치지 않으므로 개별 사이클 시간에는 영향을 미치지나 평균 사이클 시간에는 큰 영향을 미치지 않는다.

### 3.2.2 사이클의 중간에서의 작업지연

사이클 중간에서 작업지연이 발생하게 되면 작업수행시간이 길어지게 되며, 그로 인하여 <그림 11>의 알고리즘으로는 중간에 포함되었을 local maximum(혹은 local minimum)을 갖는 프레임들이 사이클 분리 프레임으로 선정될 가능성이 증가하게 된다.

<그림 12>는 작업지연을 고려한 사이클 분리 알고리즘의 순서도를 나타낸다. 전체적인 흐름은 <그림 11>의 알고리즘과 유사하나 키 프레임을 선정할 때 maximum과 minimum을 동

시에 고려하는 부분을 추가하였다. 작업지연이 있더라도 작업의 움직임이 한쪽 방향으로의 특성을 유지하고 있다고 가정하면, maximum과 minimum을 동시에 고려함으로써 작업지연이 포함된 사이클을 효과적으로 분리할 수 있다. 알고리즘에 대하여는 maximum 값으로 사이클을 분리하는 경우를 가지고 설명하고자 한다. minimum 값으로 사이클을 분리하는 경우는 거의 동일한 과정을 거친다.

우선 2종류의 변수에 대하여 0으로 초기화를 한다. 그 하나는 프레임의 센터의 좌표가 maximum과 minimum 값으로 지속되는 시간을 나타내는 timer 변수이고, 다른 하나는 최근 maximum과 minimum 값을 갖는 프레임을 통과했는지를 나타내는 tag 변수이다. 두 변수는 maximum과 minimum에 대해 각각 2개씩 존재한다.

카메라를 통해 들어오는 프레임을 하나 포착한 뒤 여러 과정을 거쳐 센터를 추정한다. 추정된 센터가 현재의 maximum 값보다 클 경우 현재의 센터 값으로 maximum 값을 교체하고 timer를 0으로 reset한 뒤 새로운 이미지를 다시 얻어 같은 과정을 반복한다. Minimum에 대해서도 같은 과정을 따른다. 만약 추정된 센터가 현재의 maximum 값보다 작을 경우 timer를 하나 증가시킨 뒤 임계값과 비교한다. 이 임계값은 사이클 분리 프레임의 후보로 선정되기 위한 timer의 임계값이다. 만약 현재 timer의 값이 이 값과 같아질 경우 현재의 프레임을 사이클 분리 프레임의 후보로 선정하며, 반대의 경우 새로운 이미지를 얻는 과정으로 돌아간다.

현재의 프레임이 사이클 분리 프레임의 후보로 선정되면 최근에 minimum 값을 통과했는지 여부를 알기 위해 min\_pass\_tag 변수의 값을 비교한다. Tag 변수의 값이 1일 경우(즉, minimum 값을 통과했을 경우) 사이클 분리 프레임의 조건을 모두 충족시키기 때문에 현재의 프레임을 분리 프레임 스택에 저장한다. 이때 max\_pass\_tag 변수를 1로 설정한 뒤, 다음 분리 프레임 선정을 위하여 min\_pass\_tag 변수의 값을 0으로 설정한다. 이와 같은 과정은 마지막 이미지까지 계속된다. 마지막 이미지에 대한 분석을 마친 후 스택에 들어있는 분리 프레임을 이용하여 사이클을 분리하고 사이클 시간을 계산하게 된다.

### 3.3 카메라 범위 이탈

<그림 13>은 카메라 범위를 이탈한 경우를 포함한 작업의 그래프이다. 카메라의 범위를 이탈하게 되면 움직임이 발생하는 픽셀의 수가 거의 없게 된다. 그러므로 이러한 픽셀의 수가 정계값 이하를 갖는 프레임이 일정시간 계속되는 경우 카메라의 범위를 이탈한 것으로 고려할 수 있다.

<그림 14>는 작업자가 카메라 범위를 이탈하는 경우를 감지할 수 있는 알고리즘을 나타낸다. 영상 내에 움직임이 발생하지 않는 경우는 카메라 범위를 이탈할 때도 있지만, 일시적으로 작업자가 동작을 멈출 때에도 발생한다. 이 두 가지 경우는 변화가 없는 프레임의 지속시간을 이용하여 분리될 수 있

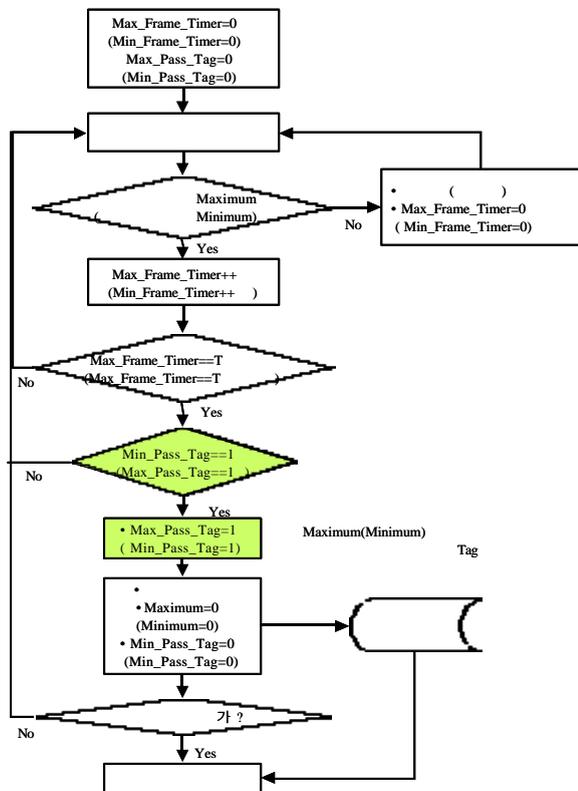


그림 12. 작업지연을 고려한 사이클 분리 알고리즘.

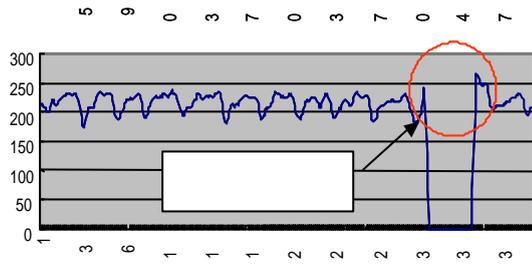


그림 13. 카메라 범위를 이탈한 경우.

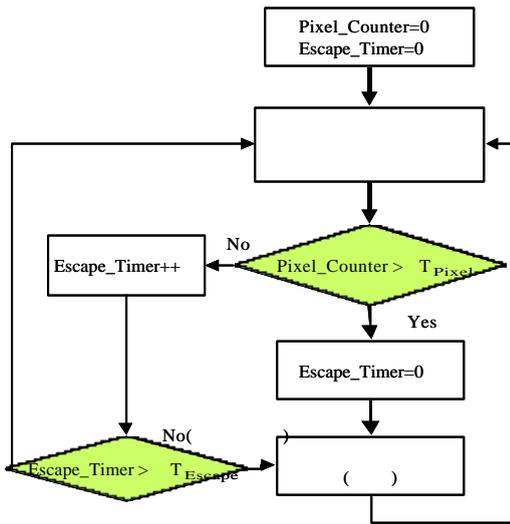


그림 14. 카메라 범위 이탈 감지 알고리즘.

다. 픽셀 카운터는 센터 추정 단계에서 고려되는 픽셀들의 개수를 나타내며 움직임의 발생 여부에 대한 정보를 담고 있다. 타이머는 움직임이 발생하지 않은 채로 지속된 시간을 나타낸다. 픽셀 카운터의 값이 임계값 미만일 경우, 현재 타이머의 값이 또 다른 임계값과 비교된다. 현재 타이머가 이 임계값보다 크면 카메라 범위 이탈로 간주되고 이때의 시간이 저장된다. 반대의 경우는 아직 카메라 범위 이탈로 간주될 수 없기 때문에, 영상을 획득하는 단계부터 다시 시작된다.

3.4 작업에 대한 사전 지식의 활용

작업을 측정할 때 작업에 대한 사전 지식은 효과적인 분석을 가능하게 한다. 이러한 지식은 파라미터로 표현되어 시스템에 입력된다. <그림 15>는 본 연구에서 개발한 작업측정 시스템의 파라미터 입력 대화상자를 나타낸다.

3.4.1 처리/저장 프레임 수(frames/sec)

초당 저장 프레임 수는 분석의 정밀도를 나타내는 값으로 시스템 하드웨어 성능과 분석 알고리즘의 최적화 정도에 크게 영향을 받는다. 본 연구에서 사용된 시스템의 하드웨어 구성은 <표 2>와 같다. 연속된 이미지는 초당 30프레임으로 입력되거나 연속된 프레임을 분석하는 데 소요되는 시간으로 인하여

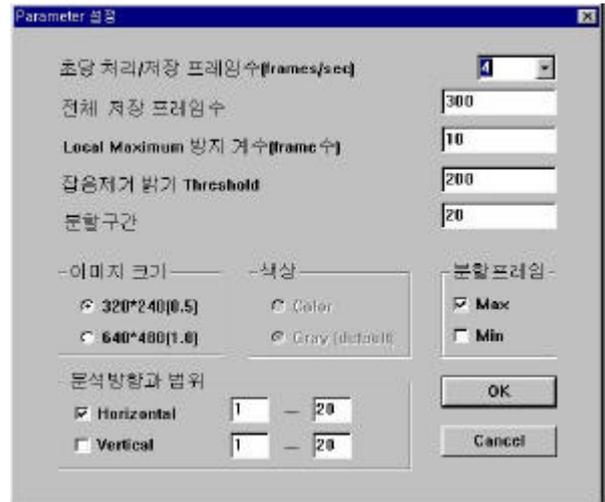


그림 15. 파라미터 입력 대화상자.

표 2. 시스템의 하드웨어 구성

CPU	Pentium II 350
RAM	64M
Frame Grabber	Matrox Meteor2-MC/4 Board
Camera	Samsung Color CCD Camera SAC-410ND
Camera Recorder	Samsung 8MM Video Camcorder MagicV

현재 시스템의 사양에서는 초당 8장까지 처리가 가능하다. 정밀한 분석이 요구되는 환경에서는 최적화된 이미지 라이브러리와 하드웨어 보강으로 인하여 분석의 정밀도를 초당 30프레임까지 끌어올릴 수 있다.

작업의 속도와 분석의 정밀도를 고려하여 이 값을 적절히 선정할 수 있다. 빠른 속도의 작업인 경우 움직임의 변화가 크기 때문에 프레임 수를 크게 설정하고, 느린 속도의 작업인 경우 프레임 수를 작게 설정한다.

3.4.2 전체 저장 프레임 수

이 값은 입력되는 영상 신호를 AVI 파일로 저장할 때 사용하는 것으로 전체 저장 프레임 수를 나타낸다. 예를 들어 초당 4장씩 50초를 저장할 경우 200프레임을 입력하면 된다. 이 기능은 2장에서 설명한 필름분석법과 같이 작업 동작을 세밀하게 관찰할 수도 있고, 긴 시간의 작업을 짧은 시간에 관찰할 수 있는 장점이 있다.

3.4.3 Local Maximum 방지계수

이 값은 사이클을 분리하기 위해 기본적으로 입력해야 하는 값이다. 앞에서 설명한 것처럼 이 값은 사이클을 분리 프레임으로 선정되기 위해 센터의 좌표값이 maximum(혹은 minimum)을 유지해야할 시간을 프레임 단위로 나타낸 것이다. 여러 작업

에 대해 실험한 결과 대략 평균작업 시간의 1/2정도로 입력하면 좋은 결과를 보이는 것으로 나타났다. 예를 들어 평균작업 시간이 10초이고 초당 4프레임으로 처리를 할 경우 Local Maximum 방지계수로 20을 입력하면 된다.

3.4.4 이미지 크기와 색상

시스템에서 이미지 크기는 320\*240과 640\*480의 두 종류에 대해 처리가 가능하나 640\*480의 경우 320\*240에서와 같은 속도가 나오지 않아 실시간 분석이 어렵기 때문에 기본적으로 320\*240을 사용한다. 같은 이유로 흑백 이미지를 사용한다.

3.4.5 사이클 분리 프레임

사이클 분리 프레임으로는 중심 좌표의 maximum 혹은 minimum 값을 갖는 프레임을 선택할 수 있다. 작업 동작의 형태에 따라 적절히 선택할 때 개별 사이클 시간이 더 정확하게 계산될 수 있다.

3.4.6 분석범위와 분할구간

작업의 주된 동작은 수평방향으로 이루어지는 경우와 수직방향으로 이루어지는 경우로 나눌 수 있다. 그러므로 작업에 따라 분석방향을 적절히 선택해 주어야 한다. 또한 이미지에 포함된 잡음의 효과를 제거하거나, 관심이 있는 움직임이 발생하는 영역에만 분석을 할 수 있도록 영역을 선택할 수 있게 하였다. 분할구간은 이미지를 분할하였을 때의 구간의 개수이다. 예를 들어 320\*240이미지에 대해 분할구간을 20으로 하였을 때, 구간의 길이는 수평방향의 경우는 16프레임이 되며 수직방향의 경우는 12프레임이 된다.

4. 실험 및 결과분석

이 장에서는 본 연구에서 개발한 시스템의 성능을 검증하기 위해 실제 작업들을 대상으로 수행한 실험의 과정과 결과를 설명하였다.

4.1 시험지 정렬 작업

첫번째 실험은 <그림 4>와 같은 시험지 정렬작업을 대상으로 하였다. 녹화는 초당 4프레임으로 하였고 이미지 크기는 320\*240이다. <그림 16>은 이 작업의 움직임을 추적한 그래프의 한 예를 나타낸다. 가로축은 프레임 번호를 나타내고 세로축은 추정한 센터의 x좌표를 나타낸다. x값은 0에서 320까지의 값을 가질 수 있다. 이 그래프를 보면 6번째와 12번째 사이클을 제외하고는 대체로 작업의 형태가 비슷한 것을 알 수 있다. 6번째 사이클에서는 시작 시점에서 지연이 발생하였고, 12번째 사이클에서는 사이클의 중간에서 지연이 발생하여 사이클 시간이 평소보다 약 두 배 가까이 소요되었다.

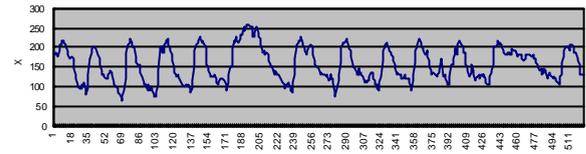


그림 16. 시험지 작업의 동작 그래프.

표 3. 사이클 분리 결과 비교 1

사이클	관측자	시스템 결과							
		1	2	3	4	5	평균	오차	오차율 (%)
1	7.95	7.25	8	7.75	8	7.75	7.75	-0.20	-2.5
2	8.89	9.25	8.75	9	8.75	9	8.95	0.06	0.7
3	8.19	8	9.5	9.25	9.5	9.25	9.10	0.91	11.1
4	8.73	9.25	7.25	8.25	7.25	8	8.00	-0.73	-8.4
5	9.1	11.75	13.25	12.5	12.5	11.75	12.35	3.25	35.7
6	15.78	13	12	12.25	12.5	13	12.55	-3.23	-20.5
7	11.07	11.25	11.5	11.25	11.75	11.25	11.40	0.33	3.0
8	10.47	9.75	9.75	9.75	9.75	10	9.80	-0.67	-6.4
9	8.06	8.25	8.25	8.25	8.25	8	8.20	0.14	1.7
10	9.26	9.75	10	10	9.75	9.5	9.80	0.54	5.8
11	10.21	9.75	9.5	9.5	9.5	9.75	9.60	-0.61	-6.0
12	17.13	18.25	17.5	17.5	17.75	18.5	17.90	0.77	4.5
평균	10.40	10.46	10.44	10.44	10.44	10.48	10.45	0.05	1.57

녹화된 영상에 대해 재생을 5번 반복하면서 시스템과 관측자가 측정한 결과를 <표 3>에 요약하였다.

두번째 열은 관측자가 녹화한 작업을 보면서 스톱워치(stopwatch)를 이용하여 측정한 결과의 평균을 나타내며, 나머지 열은 시스템의 결과를 나타낸다. 스톱워치는 100분의 1초까지 측정이 가능하며, 시스템의 경우는 초당 4프레임씩 영상을 얻었으므로 0.25초 단위로 측정이 가능하다. 같은 영상에 대해 시스템의 결과가 다른 이유는 분석을 시작하는 시점이 달라지면 연속된 영상 내에서 grab되어 분석에 사용되는 영상이 달라지기 때문이다.

매 반복 때마다 사이클의 수에 대해서 5번 모두 시스템을 통해 분리한 결과와 작업자가 분리한 결과가 같게 나왔다. 각 사이클 시간에 대해서는 관측자와 시스템이 분리하는 기준이 다르기 때문에 약간 차이가 있으나 대체적으로 큰 차이는 나지 않았다. 5, 6사이클에서 차이가 많이 난 이유는 사이클이 시작되는 시점에서 작업 지연이 발생하여 관측자의 사이클 분리 기준과 시스템이 분리하는 기준에 시간적 차이가 발생했기 때문이다. 이러한 결과는 권후 사이클에 대해 한쪽으로는 증가 효과를, 다른 한쪽으로는 감소 효과를 가져오기 때문에 평균을 구하면 이러한 영향이 상쇄된다.

4.2 계량기 번호 찍기 작업

<그림 17>은 기계를 사용하여 계량기 번호를 찍는 작업의

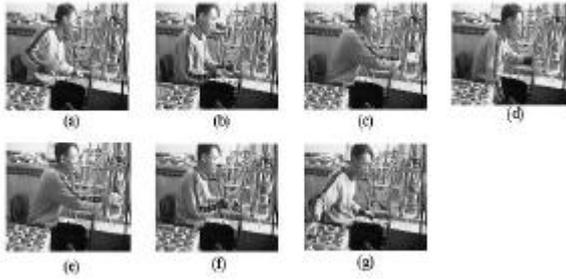


그림 17. 작업의 주요 프레임.

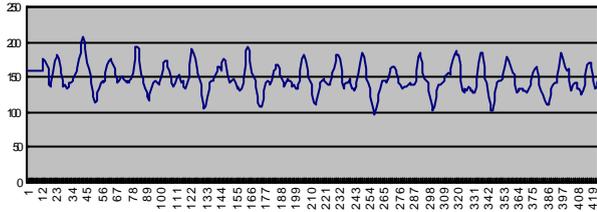


그림 18. 번호 찍기 작업의 동작 그래프.

주요 프레임을 나타낸다. <그림 18>은 시스템을 이용하여 작업의 위치를 추정한 결과의 한 예를 나타낸다. 이 작업 그래프에서 일반적인 기계작업 동작의 특징이라 할 수 있는 부품의 load 동작과 unload 동작으로 인하여 두 번의 peak가 발생하는 것을 알 수 있다.

<표 4>는 앞 절의 실험과 같은 방법으로 수행한 결과를 나타낸다. 작업을 분리하는 프레임으로 minimum값을 갖는 프레임임을 사용하였으며 초당 6프레임으로 영상을 처리하였다. 작업자의 위치 추정을 어렵게 하는 기계의 움직임과 영상 테두리 부분에 많이 포함된 잡음을 제거하면서 손의 움직임을 효과를 증가시키기 위해 아래 부분(11-19구간)에 대해서만 분석을 수행하였다.

실험 결과 관측자가 분리한 사이클 수와 같게 사이클을 분리하였으며, 평균작업시간의 경우 관측자가 분리한 결과와 1% 미만의 오차를 보였다.

표 4. 사이클 분리 결과 비교 2

사이클	관측자	시스템 결과							
		1	2	3	4	5	평균	오차	오차율 (%)
1	7.85	8.33	7.50	7.67	7.83	7.83	7.83	-0.02	-0.23
2	8.28	8.00	8.67	8.50	8.33	8.33	8.37	0.09	1.04
3	8.27	8.17	8.17	8.17	8.17	8.17	8.17	-0.10	-1.21
4	8.33	8.33	8.33	8.33	8.33	8.33	8.33	0.00	0.00
5	8.54	8.50	8.50	8.50	8.50	8.50	8.50	-0.04	-0.47
6	8.87	8.67	8.83	8.83	8.83	8.83	8.80	-0.07	-0.81
7	8.65	8.67	8.50	8.67	8.67	8.67	8.64	-0.01	-0.16
8	8.96	8.50	8.67	8.33	8.50	8.50	8.50	-0.46	-5.13
평균	8.47	8.40	8.40	8.38	8.40	8.40	8.39	-0.08	-0.87

### 5. 결론 및 추후 연구 과제

본 연구에서는 영상 처리 기술을 이용하여 반복적인 작업에 대한 수행시간을 측정할 수 있는 시스템을 제안하였다. 이러한 기능을 수행하기 위하여 시스템을 동작 표현 단계와 사이클 분리 단계로 구성하였다. 동작 표현 단계에서는 이미지 차연산과 임계값을 이용하여 잡음을 제거한 뒤 작업자의 움직임은 1차원의 시계열 데이터로 표현하였다. 사이클 분리 단계에서는 이렇게 표현된 작업의 패턴을 감지하기 위하여 작업지연과 카메라 이탈을 포함한 작업에 대해서도 사이클을 분리할 수 있는 알고리즘을 개발하였고, 작업에 대한 사전 지식을 활용할 수 있는 방법을 제시하였다. 끝으로 실험을 통하여 개발한 시스템의 성능을 보였다.

추후 연구과제로는 하나의 사이클이 복수의 요소작업으로 구성된 반복 작업의 사이클 분리를 들 수 있다. 또한 보다 다양한 작업의 사이클 분리를 위하여 인공지능 기법을 활용한 연구가 필요하다.

### 참고문헌

양윤모(1997)영상처리에 의한 수화 인식, ROSEF 과제 번호 951-0906-085-2, 한국과학재단, 4월

경순기, 설창환, 원광연(1998), 동작 포착을 이용한 인체 동작의 생성, 한국컴퓨터그래픽스학회 논문지, 4(1), 21-29, 3월

홍기상(1996) 슬라브 번호 인식 장치의 개발, 제2차 자동화시스템공학회지, 2(6), 11월.

Aggarwal, J. and Cai, Q. (1999), Human Motion Analysis: A Review, *Computer Vision and Image Understanding*, 73(3), 428-440, March.

Bobick, A. F. and Davis, J. (1996), Real-Time Recognition of Activity using Temporal Templates, in *Proc. of IEEE Computer Society Workshop Applications on Computer Vision*, Sarasota, FL, 39-42.

Boyd, J. and Little, J. (1997), Global versus Structured Interpretation of Motion: Moving Light Displays, *Processing of IEEE Nonrigid and Articulated Motion Workshop at CVPR 97*, 15-16, June.

Christopher Lee(1996), Yangsheng Xu, Online, Interactive Learning of Gestures for Human/Robot Interfaces, 1996 *IEEE International Conference on Robotics and Automation*, Minneapolis, MN, 4, 2962-2967.

Cui, Y. and Weng, J. J. (1997), Hand Segmentation using Learning-Based Prediction and Verification for Hand Sign Recognition, in *Proc. IEEE CS Conf. on CVPR*, Puerto Rico, 88-93.

Davis, J. (1999), Recognizing Movement using Motion Histograms, MIT Media Lab Perceptual Computing Group Technical Report No. 487, MIT

Fejes, S. and Davis, L. S. (1999), Detection of Independent Motion Using Directional Motion Estimation, *Computer Vision and Image Understanding*, 74(2), 101-120, May.

Freeman, W. and Roth, M. (1995), Orientation Histogram for Hand Gesture Recognition, In *Int'l Workshop on Automatic Face and Gesture Recognition*

Freeman, W. and Weisman, C. (1995), Television Control by Hand Gestures, *IEEE Int'l. Workshop on Automatic Face and Gesture Recognition*, Zurich, June

Jain, R. and Wakimoto, K. (1995), Multiple Perspective Interactive Video, in *Proc. of Int'l. Conf. on Multimedia Computing and Systems*, 201-211.

Koendsunk, J. J. and van Doorn, A. J. (1986), Dynamic Shape, *Technical report, Department of Medical and Physiological Physics, State University*,

Utrecht, The Netherlands

Niyogi, S. A. and Addison, E. H. (1994), Analyzing and Recognizing Walking Figures in XYT, in *Proc IEEE CS Conf. on CVPR*, Seattle, WA, 469-474.  
 Sanka, M., Hlavac, V. and Boyle, R. (1999), *Image Processing, Analysis, and Machine Vision*, PWS Publishing, CA, 2nd edition.  
 Toshiba Corp. (1998), Toshiba's Motion Processor Recognizes Gestures in Real Time Basis for Future Generation of Natural Interfaces between People

and Computers, <http://www.toshiba.com/news/980715.htm>, July  
 Wachter, S. and Nagel, H. (1999), Tracking Persons in Monocular Image Sequences, *Computer Vision and Image Understanding*, 74(3), 174-192, June.  
 Yamato, J., Chya, J. and Ishii, K. (1992), Recognizing Human Action in Time-Sequential Images using Hidden Markov Model, in *Proc IEEE Conf. on CVPR*, Champaign, IL, 375-385, June.



**이정결**

서울대학교 산업공학과 학사  
 서울대학교 산업공학과 석사  
 현재: 동부계강(주) 냉연사업부 공정관리  
 관심분야: 생산&공정관리, Motion & Time Study, Computer Vision, MRP, ERP, OR



**박관권**

서울대학교 산업공학과 학사  
 서울대학교 산업공학과 석사  
 서울대학교 산업공학과 박사  
 서울대학교 자동화 시스템 공동연구소 연구원  
 현재: 영산대학교 정보경영 학부 조교수  
 관심분야: ERP/BC, 생산정보시스템



**심익수**

서울대학교 산업공학과 학사  
 서울대학교 산업공학과 석사  
 현재: 서울대학교 산업공학과 박사과정  
 관심분야: 생산계획 및 일정계획, ERP/SCM



**박권우**

서울대학교 산업공학과 학사  
 KAIST 산업공학과 석사  
 미국 UC Berkeley 산업공학과 박사  
 현재: 서울대학교 산업공학과 정교수  
 관심분야: 제조시스템공학, ERP/MRP, EMS/CRM, 시뮬레이션/일정계획



**김남주**

서울대학교 산업공학과 학사  
 서울대학교 산업공학과 석사  
 현재: 퓨처텔리(주) 대표이사  
 관심분야: Network/Communication, Speech & Video Compression, Wavelet etc.