

화상처리 기법을 이용한 디버링 시스템에 관한 연구

배준영* 주윤명* 최상균** 이상룡**

A Study of Deburring System Using The Image Processing Technique

Joon-Young Bac*, Youn-Myung Joo*, Sang-Kyun Choi**, Sang-Ryong Lee**

ABSTRACT

Burr is a projected part of finished workpiece. It is unavoidable and undesirable by-product of most metal cutting or shearing process. Also, it must be removed to improve the fit of machined parts, safety of workers, and the effectiveness of finishing operation. But deburring process is one of manufacturing processes that have not been successfully automated, so deburring automation is strongly needed.

This paper focused on developing a basic algorithm to find edge of workpiece and match two different image data for deburring automation which includes automatic recognition of parts, generation of deburring tool paths and edge/corner finding ability by analyzing the DXF drawing file which contains information of part geometry. As an algorithm for corner finding, SUSAN method was chosen. It makes good performance in finding edge and corner in suitable time. And this paper suggested a simple algorithm to find matching point between CCD image and drawing file.

Key Words : Image processing (화상처리), Deburring (디버링), SUSAN method (수산법), Point matching (점매칭), Edge finding (가장자리 검출), DXF (데이터 변환파일)

기호설명

\vec{r}_0 = position of the nucleus in image
 \vec{r} = position of any other point within the mask
 $I(\vec{r})$ = brightness of pixel
 n_{\max} = number of pixel in the mask
 t = brightness difference threshold
 c = output of the comparison

n = the number of pixels in the USAN
 R = initial edge response
 P = characteristic point in drawing file
 Q = characteristic point in image file
 Q_k = base point of matching

.. 2002년 1월 2일 접수
* 경북대학교 대학원 기계공학과
** 경북대학교 공과대 기계공학부

- $\Delta\theta$ = rotation angle
- s = scale factor
- d_n = relative distance from exact point
- d_e = total distance error

1. 서론

1.1 연구배경

일반적인 버(burr)의 정의는 가공 후 공작물의 모서리에 발생하는 원치 않는 돌출부이다(Fig. 1). 이는 소성 변형의 결과로서 나타나기도 하고 주조 가공의 경우에도 발생한다. 절삭에 의해 생기는 버는 여러 가지 문제의 원인이 되고, 생산성을 저해시킨다.

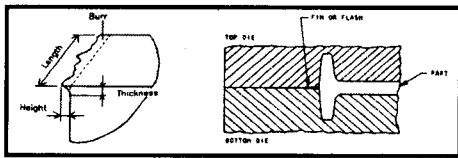


Fig. 1 General feature of burr

디버링 작업은 매우 복잡할 뿐 아니라 비용 또한 큰 비중을 차지하는데 기계가공 부품의 경우 디버링 비용이 총 가공비의 50%를 차지하기도 한다.⁽¹⁾ 그러므로 발생한 버는 반드시 제거되어야 하며, 가능하면 버의 발생을 방지하거나 허용 가능한 최소한의 크기로 생성되도록 유도하여야 한다.⁽²⁾ 주된 버제거 방법으로는 다음과 같은 것들이 있다.⁽³⁾⁽¹³⁾

- 연마제를 이용한 버제거
- 기계적 공구를 이용한 버제거
- 열을 이용한 버제거
- 전기를 이용한 버제거
- 화학 물질에 의한 버제거

기계적 버제거 작업으로는 수작업, 브러쉬를 이용한 작업, 그리고 기계적 장비를 이용한 작업이 있다. 이런 작업들은 모서리를 따라 움직이기 때문에 기계장치에 의한 모서리부의 추적 능력이 가장 큰 문제로 제기 된다.⁽⁵⁾

1.2 기존연구 및 연구목적

기존의 연구들이 접촉식 센서를 사용하여 복잡한 버의 형상을 측정한다. 그 측정된 크기를 제어 입력으로 사용할 때 지연이 생기기 때문에 공작물의 파손이 야기될 수 있다.⁽⁸⁾⁽¹¹⁾ 이 때문에 국부적인 버의 측정보다는 모서리부의 인식과 공작물의 형상에 따른 적합한 방법들을 찾아야 한다. 로봇을 이용한 버제거의 경우에도 역시 모서리 형상의 인식이 중요한 문제가 된다.⁽⁷⁾ 이에 본 연구에서는 모서리부의 추적을 위한 새로운 영상 처리 기법인 SUSAN(Smallest Univalve Segment Assimilating Nucleus) 영상처리 기법을 소개하고 이에 필요한 모듈을 개발하여 디버링 공정에 적용하고 실제 산업현장에서 사용가능한가를 검증해 보고자 한다.

절삭가공에서 발생한 버의 가공정화에 의한 특성과 버형성의 불규칙성을 고려할 때, 주로 용접로봇이나 도장용 로봇과 비교하여 다음과 같은 특성을 갖추어야 한다.⁽¹³⁾

- 버 제거시의 저항력과 대응하는 로봇 팔의 강성(stiffness)이 있어야 한다.
- 균일한 가장자리(edge) 상태를 유지하기 위한 가장자리 형상에 따른 추적 능력이 있어야 한다.
- 설치 오차나 불규칙한 버형상을 보정할 수 있는 공구나 로봇의 기능이 요구된다.

이에 본 연구에서는 개발하고자 하는 요소 기술은 다음과 같다. 첫 번째는 로봇 교시에 필요한 DXF 기반의 CAD/CAM 모듈로서 복잡한 공구경로를 생성하고, 공작물의 정확한 형상을 인식하기 위해 필요하다. 두 번째는 CCD 영상중 공작물의 형상을 인식하는 기술이다. 이를 구현하기 위해 영상의 Edge와 Corner의 정보를 찾아낼 수 있는 SUSAN 영상처리방법을 사용하였다. 세 번째는 로봇의 경로 제어 기술이다. 디버링 숙련자라면 버의 크기와 재질에 따라 브러싱 작업을 여러 번 반복한다. 이런 데이터들을 가지고 있다면 디버링 전문가 시스템을 구성할 수 있다.

2. 이론해석

2.1 DXF 기반의 CAD/CAM 모듈

CAD 도면으로부터 공구경로를 자동 생성할 수

있다. 이 방법을 이용하면 교시 도중 발생하는 실수를 미리 예방할 수 있고, 설계 도면과 실제 가공물 사이의 오차와 시간을 줄일 수 있다.⁽⁶⁾ 본 연구에서는 CAD 파일의 형식중 하나인 DXF(Drawing Interchange File) 파일을 해석하고 공작물의 형상을 인식하는 방법을 사용하였다.

도면 정보 중에 ENTITES에 관한 섹션을 해석하면 도면의 형상을 인식할 수 있다. LINE 객체의 끝점들을 형상을 나타내는 특징점으로 선정하게 되는데 이렇게 선정된 특징점들은 대부분 공작물의 모서리 부분을 나타내게 된다.

2.2 SUSAN Method를 이용한 형상인식

시각에 의존하지 않은 디버깅은 센서의 응답에 따라 가공물에 파손을 주기도 한다. 정확한 가공을 위해서는 CCD 카메라 등의 시각정보를 처리할 수 있는 방법들이 필요하다. 특히, SUSAN 영상처리 기법은 기존의 미분형 필터와는 달리 적분형 필터에 가깝다. 따라서, 다른 방법들에 비해 속도가 빠르며, 노이즈 제거 등의 효과를 부가적으로 얻을 수 있다.

2.2.1 SUSAN Principle

SUSAN 영상 처리기법은 다른 Edge 검출 필터들이 3×3나 5×5 등의 정방형 마스크를 사용하는데 반해 원형의 마스크를 사용한다. 원형의 마스크를 사용함으로써 프로그래밍이나 기타 계산과정이 복잡해지지만 처리과정의 방향성 문제를 해결할 수 있다.⁽⁴⁾ 먼저 SUSAN 방법에 사용되는 용어를 정리하면 마스크의 중앙을 Nucleus라고 하고 마스크 내에 Nucleus를 둘러싼 주변 픽셀들 중에 Nucleus와 같은 밝기를 가지는 픽셀을 USAN (Univalue Segment Assimilating Nucleus)이라고 한다. 이를 이용하여 영상의 정보를 얻어내기 위한 기본적인 2가지 원리는 다음과 같다.

- 이미지의 작은 영역에서 각 포인트는 주변 영역과 비슷한 밝기를 가진다.
- 따라서 USAN의 크기, 무게중심에서 영상의 정보를 찾을 수 있다.

다른 영상 처리와의 차이점으로는 미분 연산이 사용되지 않는다는 것이며, 적분형 연산의 특징 때문에 추가적인 노이즈 제거가 필요 없다.

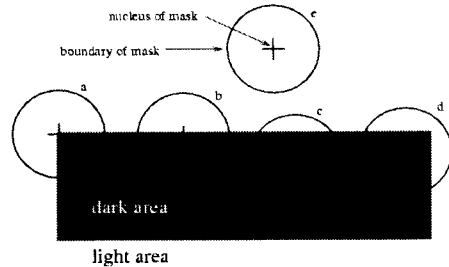


Fig. 2 Mask and nucleus

Fig 2를 보면 알 수 있듯이 USAN의 면적이 최대 되면 화상에 큰 특징이 없다는 것이고 USAN의 면적이 마스크의 절반 정도가 되면 직선을 나타내며, 1/4 정도이면 90°정도의 모서리가 존재한다는 것을 쉽게 알 수 있다. 따라서 현재, Nucleus가 어떤 위치에 있다는 것을 USAN의 면적을 통하여 알아낼 수 있다. 각 픽셀에서의 USAN 영역의 크기를 구하면 영상의 정보를 정량적인 값으로 나타낼 수 있는데, USAN의 면적이 작을수록 더 민감한 Corner를 나타낸다는 의미이다. 만약, 입력 영상에 약간의 노이즈가 섞여 있다 하더라도 USAN 영역 계산시 무시된다.

2.2.2 SUSAN Corner Finder

SUSAN 기본원리를 이용하여 Edge 검출기와 Corner 검출기를 설계할 수 있다. 찾아지는 특징점의 품질과 연산속도를 고려하여 마스크의 크기는 $\Phi=6.8$ 픽셀인 원형마스크를 사용한다. 이를 구현하면 영역 내 픽셀의 갯수는 37개가 된다.

실제 구현을 위한 첫 번째 과정은 Nucleus와 임의의 픽셀의 밝기 차를 구하는 것이다. 임의의 포인트 \vec{r} 에서의 밝기를 $I(\vec{r})$ 이라고 정의하고 Nucleus와 임의의 점의 밝기가 비슷하다면 1을 다르다면 0을 돌려주는 함수를 정의한다.

$$c(\vec{r}, \vec{r}_0) = \begin{cases} 1 & \text{if } |I(\vec{r}) - I(\vec{r}_0)| \leq t \\ 0 & \text{if } |I(\vec{r}) - I(\vec{r}_0)| > t \end{cases} \quad (1)$$

이제 마스크 내의 모든 점의 c 를 구했다면 그 합으로 USAN의 크기 n 을 구할 수 있다.

$$n(\vec{r}_0) = \sum_{\vec{r}} c(\vec{r}, \vec{r}_0) \quad (2)$$

Edge Finder의 경우 영상에 노이즈가 전혀 없다면 n 자체만으로도 형상정보를 구할 수 있다. 하지만 실제 영상에서는 노이즈가 존재하기 마련이며 이를 제거하기 위해서 형상 임계 게인(Geometric thresholding value) g 가 도입되었다. 만약 n 값이 g 보다 크다면 노이즈에 의해 영향으로 보고 무시하면 된다. 최적 노이즈 제거를 위해 일반적으로

$\frac{3}{4} n_{\max}$ 의 값이 사용된다. g 를 적용한 초기 응답 Initial edge response R 은 다음과 같이 정의된다.

$$R(\vec{r}, \vec{r}_0) = \begin{cases} g - n(\vec{r}_0) & \text{if } n(\vec{r}_0) < g \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

이제 식(3)의 응답 R 에서 최대 값들을 살펴보면 Edge의 형상을 알 수 있다. 만약 식(1)의 c 대신에

$$c(\vec{r}, \vec{r}_0) = e^{-\left(\frac{R(\vec{r}) - R(\vec{r}_0)}{t}\right)} \quad (4)$$

식(4)를 사용한다면 훨씬 안정적이고 민감한 응답을 구할 수 있다.

SUSAN corner 검출기는 Edge 검출기와 매우 유사하다. 하지만 Edge 검출기에서 g 의 값이 노이즈 제거를 위해서만 사용되는 것에 비해 Corner 검출기에서는 Corner를 찾아내는 중요한 파라미터로 사용된다. 영상에서 Corner를 찾기 위해서는 USAN 영역의 값 n 이 n_{\max} 의 1/2 이하가 되는 곳을 찾아야 한다. 이렇게 구한 초기 응답의 Local minimum을 구하면 Corner의 위치를 구할 수 있다.

Corner Finder의 t 게인을 조절하여 찾을 Corner의 갯수를 조절할 수 있다. t 는 영상의 contrast 등에 영향을 받고, 또 찾을 특징점의 갯수에 영향을 받는다. 이를 이용하여 적절한 값을 선택한다. g 는 찾을 Corner의 형상에 영향을 준다. 즉, g 값이 작아질수록 큰 각도의 Corner는 무시되며, 더 날카로운 Corner만 검출된다. t 는 Nucleus와 임의 픽셀의 밝기를 비교하는 파라미터이다. 찾을 Corner의 품질에는 영향을 미치지 않지만 찾을 갯수에 영향을 미친다. 영상의 히스토그램이 어느 한쪽으로 쏠려있어도 평활화 등의 추가적인 처리가 필요 없고 게인의 조절만으로 해결할 수 있다.

이 방법을 사용하면 이상적인 영상의 경우 정확한 Corner를 찾을 수 있다. Corner를 잘못 인식하는

경우를 False positives라고 하는데 크게 두 가지 이유로 발생한다. 하나는 카메라의 포커스가 맞지 않아 영상의 경계가 명확하지 않고 간섭되는 현상이며, 또 다른 이유는 노이즈이다. 이를 해결하기 위해 도입된 개념이 CG(Center of Gravity), 즉, 무게 중심점이다. CG는 식(5)와 같이 정의된다.

$$\vec{r}(\vec{r}_0) = \frac{\sum_r \vec{r} c(\vec{r}, \vec{r}_0)}{\sum_r c(\vec{r}, \vec{r}_0)} \quad (5)$$

먼저 정상적인 corner의 경우는 Fig. 3 (a)와 같다. +는 Nucleus를 의미하고 \circ 는 CG를 의미한다. 경계에 의해 간섭이 발생한 영상은 Fig. 3 (b)에 나타나 있다. 정상적인 경우와 비교하면 Nucleus와 CG의 거리가 너무 가깝다. 두 번째로 노이즈에 의한 false positive는 (c)에 나타나 있다. 노이즈에 의해 화상의 일부분의 정보가 소실되는데, CG도 Nucleus와 어느 정도 떨어져 있고 USAN의 갯수가 적기 때문에 Corner로 오인하기가 쉽다.

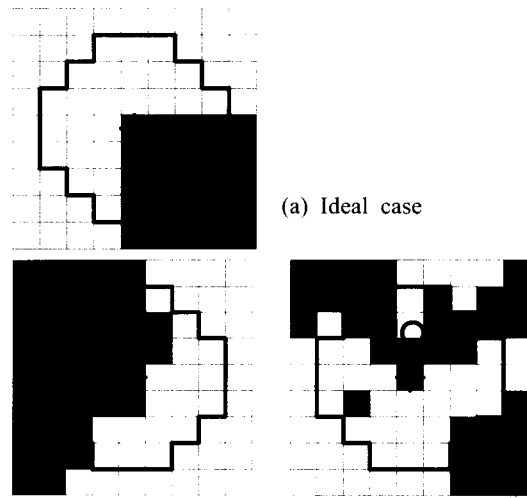


Fig. 3 Cases of false positives

이러한 False positive을 제거하고 정확한 Corner 판정을 위해 두 가지 검사를 추가한다.

- CG를 계산하여 Nucleus와의 거리가 너무

가까우면 False positive 이다.

- Nucleus와 CG를 직선으로 연결한 픽셀들의 밝기가 Nucleus와 차이가 있다면 False positive 이다.

실제 이미지에서 나타나는 여러 False positives 를 위의 2가지 Rule을 이용하여 대부분 제거 할 수 있다.

2.3 형상 매칭 알고리즘

본 연구에서는 형상을 표현하는 최소단위로 Corner를 정의하고 CCD에서 획득한 영상에서 찾은 Corner 정보와 DXF 도면 파일에서 찾은 Corner를 매칭시켜 위치, 회전, 크기변화를 모두 알아낼 수 있는 알고리즘을 연구하였다.

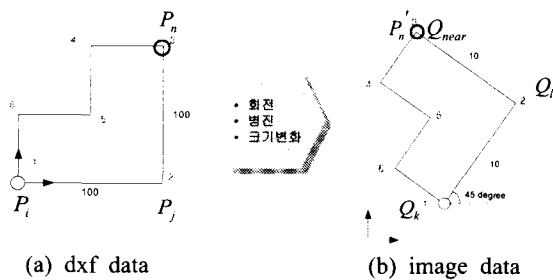


Fig. 4 Characteristic point and matching

Fig. 4에서 DXF 도면 파일에서 얻은 특징점을 Fig. 4 (a)에 P로 표시하고, CCD 영상의 데이터에 SUSAN 영상처리 기법을 사용하면 얻을 수 있는 특징점을 Fig. 4 (b)에 Q로 표시하였다. 도면에서 특징점을 찾는 방법으로는 각 라인 객체의 끝점을 특징점으로 추출한다. 이 P점들과 Q점들을 비교하면 각 모서리들이 어떤 특징점을 중심으로 회전, 병진, 크기변화가 일어난다는 것을 알 수 있다. 기준점과 회전, 병진, 크기변화의 값을 알아낸다면 도면상의 물체가 이미지 영역 중 어디에 위치해 있는 지를 알 수 있다. 이를 알아내기 위해 먼저 도면영역에서 임의의 두 점 P_i, P_j 를 선택하고, 영상영역에서도 임의의 두 점 Q_k, Q_l 을 선택한다. 각 두 점이 서로 같다고 가정하면 식 (6),(7),(8)에서 변화량을 계산할 수 있다.

$$\begin{aligned} & \text{기준점 } Q_k & (6) \\ & \text{각도변화} \end{aligned}$$

$$\Delta\theta = \tan^{-1}\left(\frac{Q_{iy} - Q_{ky}}{Q_{ix} - Q_{kx}}\right) - \tan^{-1}\left(\frac{P_{jy} - P_{iy}}{P_{jx} - P_{ix}}\right) \quad (7)$$

$$\text{크기변화 } s = \frac{\overline{Q_k Q_l}}{P_i P_j} \quad (8)$$

이 변화량을 이용하여 P영역의 나머지 점들을 Q영역으로 사상을 시킨다. 사상된 각 점 P'_n 과 가장 가까운 점을 Q_{near} 라고 한다. 이 사이의 거리를 거리 오차(식(9))라고 정의한다. s로 나누어준 이유는 잘못 찾아진 아주 가까운 두 특징점의 경우라면 오차도 작게 나타난다. 따라서 스케일 변화로 나누어주어 선택된 기준점에 대한 상대적인 거리 오차를 구한다.

$$d_n = \overline{P'_n Q_{near}}/s \quad (9)$$

이 거리오차를 모두 합한 것을 최종오차라 정의한다. 최종오차는 식(10)에 나타내었다.

$$d_e = \sum d_n \quad (10)$$

이 최종오차가 최소가 되는 점이 바로 일치점이며 이때의 각도변화, 기준점, 크기변화가 두 특징점 사이의 변환이 된다. (식11)

$$\min(d_e) \rightarrow \Delta\theta, Q_k, s \quad (11)$$

이 방법을 이용하면 영상데이터를 이용하여 찾은 특징점의 갯수가 도면 영역의 갯수보다 더 많거나 소실되더라도 가장 근사한 일치점의 추정이 가능하다. 다음으로 실제 가공 코드를 생성하는 부분은 매칭기법으로 얻은 이미지영역에서의 공작물 위치를 알면 실제 로봇 좌표계에서의 위치도 알 수 있다. Fig. 5를 보면 CCD영상은 로봇 작업영역중의 일부분을 나타낸다. 만약, 이미지중의 어떤 특징점이 실제 로봇 좌표계의 위치를 나타내는지를 알고 있다면 이를 통해 로봇작업영역의 공구 경로를 만들 수 있다.

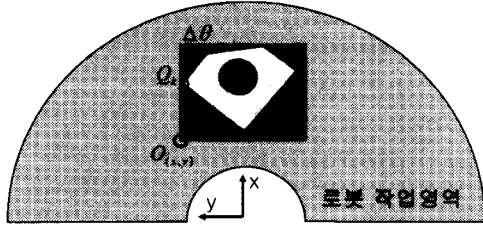


Fig. 5 Robot work space

실제 프로그램에서는 이미지의 원점의 로봇 좌표계 위치를 설정해주게 되어있다. 이를 설정해주면 식 (12)와 같이 로봇작업영역으로 변환 가능해진다.

$$R_{(x,y)} = O_{(x,y)} + Q_k/s \quad (12)$$

여기서 R은 로봇 작업영역을, O는 이미지 원점의 로봇좌표계, Q는 이미지 좌표상의 특징점 위치이다.

3. 실험

3.1 시스템 구성

본 연구에 사용된 디버링 로봇시스템은 삼성전자의 SM2 FARA 로봇으로 AC서보모터를 채용한 산업용 로봇이다(Fig.6 (a)).



(a) SM2 Robot (b) Camera and tool

Fig. 6 Deburring robot and tool

본 연구에서는 로봇의 운동 제어를 위해 PC와의 시리얼 인터페이스를 이용하여 제어하도록 한

다. 영상처리를 위해서는 일반 PC CAM 카메라를 사용하였으며, 로봇의 끝단에 전동 디버링 툴과 브러쉬를 이용하여 디버링 작업을 수행하도록 한다. 끝단에 (b)와 같이 전동 디버링 툴과 CCD 카메라를 설치 하였다. 표면처리에는 주로 와이어 브러쉬가 사용된다. 이상의 시스템으로 Fig. 7의 두 가지 공작물에 대하여 모서리 추적 및 디버링 실험을 실시하였다. 이 시편의 재질은 알루미늄합금 6061을 사용하였으며 원소재를 정면밀링(Face milling)한후 내부를 앤드밀 공구로 포켓가공 한 것이다. 가공 후에 버가 생성된 시편을 실험에 사용하였다.



Fig. 7 Workpieces

3.2 작업 순서

디버링 실험을 통합하여 운용할 소프트웨어는 Fig. 8과 같은 흐름을 갖는다.

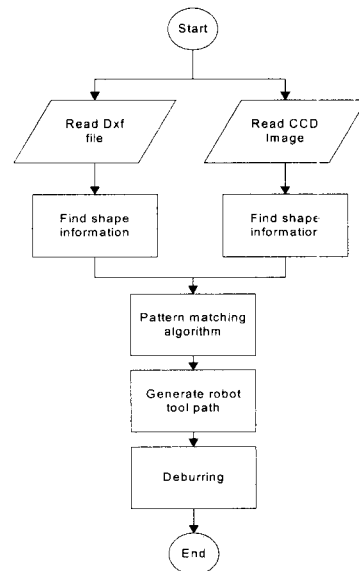


Fig. 8 Flowchart of S/W

3.3 형상 매칭 실험

형상 매칭 실험을 위해 작성한 소프트웨어를 이용하여 공작물의 위치를 찾고 그 오차를 계산해 보았다. Fig. 9에 실제 실행중인 화면을 볼 수 있다. 찾은 코너들이 왼쪽 화면에 빨간색 마크로 표시가 되어 있고, 매칭 된 기준점이 녹색 마크로 표시되도록 프로그램 되어있다.

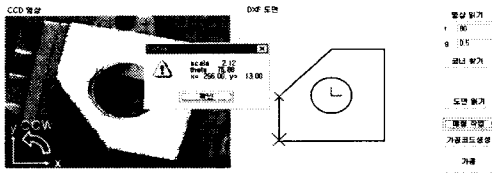


Fig. 9 Matching Process

오차 실험을 위해 실제로 작성한 프로그램을 이용하여 t를 50, 60, 70, 80으로 변화시키면서 위치를 찾아보았다. 각 경우의 실험결과는 Table 1.과 같다.

Table 1. Matching value

| | t=50 | t=60 | t=70 | t=80 |
|---------------|-------|-------|-------|-------|
| No. of Corner | 174 | 146 | 125 | 108 |
| s | 2.20 | 2.20 | 2.14 | 2.12 |
| theta | 76.34 | 76.34 | 76.69 | 76.88 |
| x | 266 | 266 | 267 | 266 |
| y | 13 | 13 | 15 | 13 |

t가 커지면 찾아지는 특징점의 갯수가 줄어들고 그에 따라 계산된 각도와 기준점의 위치가 조금씩 변화한다는 것을 알 수 있다. scale의 비가 2.2일 때는 픽셀당 오차가 0.4545 mm/pixel 이고 2.14일때는 0.4673 mm/pixel의 값을 가진다. 이를 이용하여 찾은 기준점의 위치오차와 회전오차를 합한 최대 오차값을 계산해 보면 약 1.17mm 정도의 오차가 발생하게 된다.

4. 결과 및 고찰

본 논문에서 중점을 둔 부분은 버의 정확한 형상 측정보다는 공작물의 모서리 위치 추적에 중점을 두었으며 이를 확인할 수 있도록 공작물을 특정 위치에서 0°에서 180°까지 45°씩 회전시키면서

위치를 찾는 실험을 수행하였다. t값을 변화시킬 때 실제 위치와의 오차를 매칭실험에 사용한 방법으로 산출하였다.

Table 2. Experiment result

| 실험각도(deg) | 0 | 45 | 90 | 135 | 180 |
|--------------|------|------|------|------|------|
| x축 오차(pixel) | 2 | 4 | 1 | 4 | 1 |
| y축 오차(pixel) | 1 | 1 | 0 | 3 | 0 |
| 각도오차(deg) | 0.31 | 0.36 | 0.14 | 0.35 | 0.16 |
| 처리시간(sec) | 4.5 | 5.6 | 3.8 | 4.8 | 3.2 |
| pixel 오차 | 2.23 | 4.12 | 1 | 5 | 1 |
| 최대오차(mm) | 0.78 | 1.36 | 0.32 | 1.51 | 0.36 |

실험 결과를 보면 각도에 따라 최대 1.5mm 정도의 오차 내에서 추적이 가능하며 처리시간 역시 5초 이내라는 것을 알 수 있다. 일반적인 영상처리 기법의 가장 큰 단점은 산업현장 적용하기에는 너무 많은 계산시간이 요구된다는 것이다. 이를 고려하면 5초 정도의 시간은 적용가능한 시간이다. 그리고, 오차의 정도를 보면 45도와 135도 각도인 경우 1.5mm 정도의 매우 큰 오차가 발생하였다. 이는 CCD 이미지의 특성상 수직선과 수평선은 표현하기 쉽지만 45도 각도의 대각선을 표현하기 어려우며, 조명의 단방향성에 기인한 것으로 사료된다. 1.5mm의 오차는 버의 크기가 1mm정도인 것을 고려하면 디버링 작업이 실패했다고 할 수 있다. 이를 위해 공작물을 고정할 때, 가능하면 수평이나 수직 방향으로 고정시키는 것이 필요하다.

디버링 작업 후, 버의 제거 정도는 수동으로 브러싱 작업을 한 것과 유사한 품질을 보였다. 1mm 정도의 미세한 버들은 거의 제거되었으나 표면의 거칠기가 조금 더 나빠졌다. 이는 브러싱 작업의 특징에 의한 것으로 만약 위치 추정의 정밀도를 향상시킬 수 있다면 브러싱 대신 밀링과 같은 절삭기법을 사용하여 표면 정밀도 역시 향상시킬 수 있을 것이다.

5. 결론

본 논문에서는 디버링 공정의 자동화를 위한 요소 기술을 개발하였고, 다음과 같은 결론을 얻을

수 있었다.

- (1) 디버링 시스템에 SUSAN 영상처리 기법을 도입하여 공작물의 형상정보를 추출하는 시간을 5초 이내로 줄였다.
- (2) DXF 도면과 CCD영상의 특징점을 비교하여 공작물의 모서리 위치를 오차 범위 1.5mm 이내에 찾을 수 있었다.
- (3) 도면 데이터로부터 가공물의 형상을 인식하고 공구 경로를 자동 생성하였다.

실제 라인을 찾는데 주로 사용되는 Hough transform이나 영상매칭을 위한 GHT(General Hough Transform)같은 경우 표준 640×480 영상을 처리하는데 20~30초가 소요되었다. 하지만, SUSAN Method를 사용하면 이를 2~3초로 줄일 수 있다. 따라서, 실제 화상 처리 속도를 약 1/10 정도로 단축시켰다고 할 수 있다. 오차가 크게 나온 이유는 실험에 사용된 카메라가 오토 포커싱이 되지 않는 수동 웹 카메라를 사용하였기 때문에 잘못 판정된 특징점의 갯수가 많은 것으로 판단되며, 이를 실제 산업 현장에 사용되는 산업용 CCD 카메라로 바꾸어 작업을 한다면 보다 정밀하고 우수한 결과를 얻을 수 있을 것으로 판단된다. 추후작업으로는 디버링후, 공작물의 상태를 확인 할 수 있는 처리 방법과 아울러, 가공물의 재질이나 툴의 종류나, 도면 데이터를 분석하여 버의 생성위치를 예측하여 공구경로를 최적화하는 연구를 병행할 예정이며, 실제 전문가의 작업을 모사 하는 전문가 시스템의 개발이 가능할 것으로 전망된다.⁽⁹⁾⁽¹⁰⁾⁽¹²⁾

참고문헌

1. L. K. Gillespie, "Design to Reduce Deburring," American Machinist, Nov., pp. 125-129, 1985.
2. L. K. Gillespie, "Towards A Rational Approach to Deburring," SME Technical paper, MR74-996, 1994.
3. L. K. Gillespie, "Deburring Technology for Improved Manufacturing," SME, Michigan, 1981.
4. S. M. Smith & J. M. Brady, "SUSAN - A New Approach to Low Level Image Processing," Technical Report TR95SMS1c, 1995.
5. 조지승, 정병목, "로봇 비전을 이용한 대형 2차원 물체의 인식과 가공," 한국정밀공학회지, 제16권, 제2호, pp. 68-73, 1999.
6. 윤희중, 박동삼, "IGES 포맷을 이용한 디버링 경로의 자동 생성," 한국정밀공학회 춘계학술대회 논문집, pp. 769-772, 1999.
7. 신상운, 안두성, "비전을 이용한 디버링 기술을 로봇에 전달," 한국정밀공학회지 제15권, 제9호, pp. 93-100, 1998.
8. 최규중 외2, "산업용 로봇을 이용하는 지능 버 제거 시스템 개발에 관한 연구," 한국어업기술학회지, 제34권, 제1호, pp. 1-5, 1998.
9. 주진화 외2 "신경망을 이용한 ROBOT ARM의 디버링(Deburring) 작업에 관한 연구," 한국자동 제어학술회의논문집, pp. 139-142, 1996.
10. 채호철, 한창수, "디버링 작업을 위한 로봇 매니플레이터의 힘 제어에 관한 연구," 한국정밀공학회 춘계학술대회논문집, pp. 1169-1172, 1997.
11. Bone, G. M. "Sensing and Control for Automated Robotic Edge Deburring," IEEE IE, Vol. 41 No. 2, pp. 137-, 1994.
12. Shimokura, K., "Programming Deburring Robots Based on Human Demonstration with Direct Burr Size Measurement," Proceedings, No. 1, pp. 572-, 1994.
13. 고성림, "Burr의 처리와 초정밀 버(Burr) 형성원리의 이해와 적정 버 제거 방법의 선택," 정밀표면가공 및 디버링 기술 워크샵, 1996.