Design Homework Report

Subject : 허프 원형 변환과 HSV 색상 모델을 이용한 동전 계산기

|  |  |  |
| --- | --- | --- |
| **Lecture** | **HCI** | |
| **Lecturer** | **Prof. Jin-Woo Jung** | |
| **Student** | **Student ID Number** | **Student Name** |
| 2014112093 | 양준영 |
| 2015112169 | 김진형 |
| 2016112143 | 김우용 |

Abstract

|  |  |
| --- | --- |
| **Subject** | 허프 원형 변환과 HSV 색상 모델을 이용한 동전 계산기 |
| **Keywords**  **(2~7 words)** | **Hough Circle Transform, HSV Color Model,** |
| 1. Research Purpose  본 시스템은 디지털 영상처리를 이용하여 자동으로 동전을 금액별로 분류하고 그 총액을 계산하는 시스템이다. 동전의 개수가 많은 경우 은행에서 찾아볼 수 있는 동전 계수기를 사용할 수 있지만, 일상생활에서 사용되는 소량의 동전의 경우 그 액수를 파악하기 위해서 이와 같은 기계를 사용하는 것은 비현실적이다. 물론 사람이 기계의 사용없이 동전의 액수를 계산하는 방법도 존재하지만, 오류를 범할 가능성이 다소 크다는 문제점이 존재한다. 따라서, 소량의 동전을 계산할 때 일일이 세는 번거로움을 줄이고 액수 계산의 정확도를 높이기 위해 디지털 영상처리르 이용하여 동전의 총액을 계산하는 방법을 구상하게 되었다.  2. Research Method   |  |  |  | | --- | --- | --- | | OpenCV Method | cvtColor() | 이미지의 색상 영역을 변환한다. | | threshold | 회색조 영상을 주어진 임계 값에 따라 이진화한다. | | GaussianBlur | Gaussian Blur를 사용하여 잡음을 제거한다. | | getStructuringElement | 매개변수에 맞는 구조요소를 생성한다. | | morphologyEx | Closing 연산을 수행한다. | | Canny | Canny Edge Detector로 Edge를 검출한다. | | HoughCircles | 영상 내에서 원을 검출한다. |   개체이(가) 표시된 사진  자동 생성된 설명3. Research Result  (a) 달러화 결과 영상  **검은색, 사진, 전면, 작은이(가) 표시된 사진  자동 생성된 설명**  (b) 원화 결과 영상  4. Discussion  본 시스템에서 설정한 실험 장비와 환경에서 실행한 결과 달러화의 1센트, 5센트, 1다임, 25센트, 1달러를 모두 검출하여 총 금액을 계산한 결과 영상을 확인하였고, 원화의 구 10원, 10원, 50원, 100원, 500원도 동일하게 결과 영상을 확인하였다.  단, 동전 여러 개가 맞닿아 있는 경우 정확한 검출이 어려웠다.  (up to 2 pages) | |

**1. Introduction**

**1.1 Background**

최근 컴퓨터 기술과 이를 지원하는 주변장비의 발전으로 디지털 영상처리를 사용하여 활용 목적에 따라 여러 모양의 객체를 검출하는 알고리즘과 시스템을 제안하는 다양한 연구가 활발히 이루어지고 있다. 얼굴 인식을 기반으로 한 회사나 공공기관의 출입 관리 시스템, 주차장 차량 혼잡도 분석 시스템, 무인 매장의 상품 인식 알고리즘 등이 예시에 해당된다.

이러한 컴퓨터 비전 기술의 발전은 기계에 시각을 부여했다는 것에 큰 의미가 있으며 인공지능 기술과 결합될 경우 실생활에서 높은 활용도를 기대할 수 있다1).

**1.2 Previous study**

**1.2.1 유사난수 발생기 메르센 트위스터 알고리즘2)**

마츠모토 마코토의 유사난수 발생기 메르센 트위스터 알고리즘은 난수의 발생 주기를 메르센 소수에서 착안하였다. 메르센 소수는 M(n) = 2n -1 의 형태를 지니는 메르센 수 중 소수를 일컫는다. 메르센 트위스터 알고리즘은 주기를 219937–1 인 MT19937을 사용하며, 동일인이 발표한 TT800에 비하여 더 좋은 성능을 지닌다.

C++ Random 라이브러리의 Random 함수는 232-1 의 주기를 가지고 있으며, rand() 함수에 의해 생성된 난수는 표본값이 작을수록 그 분포가 고르지 못한 특성을 지닌다. 메르센 트위스터에 의해 생성된 난수의 분포는 이에 비해 고른 형태를 지닌다. 또한, 비트연산만으로 구현되어 있어 일반 rand() 함수에 비해 4배 이상 빠른 속도를 지니고 있다.

**1.2.2 허프 변환을 이용한 원과 타원 검출 알고리즘 구현3)**

본 연구의 검출법은 원과 타원을 검출하는데 있어서 누적값을 이용하지 않고, 타원의 특징을 결정하는 p와 theta를 이용하여 타원 내부의 잡음에 영향을 받지 않는 타원에 대한 접선의 접점을 택하였으며, 오차를 줄이기 위해 최소 자승법으로 타원을 결정하여 연산 회수를 감소시켰다. 이 방법은 Aoki 알고리즘의 변형으로 모든 각에 대하여 허프 변환하지 않고 특정각에서만 허프 변환하여 수선의 길이가 최소, 최대가 되는 타원의 접점 8개를 택하여 타원을 검출하고, 같은 방법으로 원을 검출한다.

허프 변환은 2차원 함수이기 때문에 곡선의 접선에 따른 밀도 함수가 형성된다. 따라서 곡선 형상에 따른 최대 피크 값은 검출되지 않는다. 그러므로 허프 평면에 직선을 이루고 있는 부분을 검출하여 다시 한 번 허프 변환하면 theta 축에 대칭인 2개의 최대 피크 값을 검출할 수 있다.

원은 3차원 파라미터를 가지고 있는 기하학적 성질을 가지고 있으며, 원을 표현하는 방정식은 다음과 같다. 여기서 ( x0 , y0) 은 원의 중심이고, r은 반지름이다.

( x – x0 )2 + ( y – y0 )2 = r2

**1.3 Research purpose**

본 시스템은 디지털 영상처리를 이용하여 자동으로 동전을 금액별로 분류하고 그 총액을 계산하는 시스템이다. 동전의 개수가 많은 경우 은행에서 찾아볼 수 있는 동전 계수기를 사용할 수 있지만, 일상생활에서 사용되는 소량의 동전의 경우 그 액수를 파악하기 위해서 이와 같은 기계를 사용하는 것은 비현실적이다. 물론 사람이 기계의 사용없이 동전의 액수를 계산하는 방법도 존재하지만, 오류를 범할 가능성이 다소 크다는 문제점이 존재한다. 따라서, 소량의 동전을 계산할 때 일일이 세는 번거로움을 줄이고 액수 계산의 정확도를 높이기 위해 디지털 영상처리를 이용하여 동전의 총액을 계산하는 방법을 구상하게 되었다.

추가적으로 본 시스템은 원화 뿐만 아니라 달러화도 계산할 수 있도록 고안되었다. 해외 여행도중 익숙하지 않은 외화 구성 체계로 인해 동전 사용에 어려움을 겪은 적이 있을 것이다. 또한, 몇몇 통화의 경우(특히 미화), 동전에 그 동전이 가지는 가치가 명확히 숫자로 표기되어 있지 않아 사용자가 느끼는 불편함을 가중시킨다. 따라서 원화 동전 가치 계산과 더불어 세계적으로 가장 많이 통용되고 있는 미화 동전 가치 계산 기능을 추가하게 되었다.

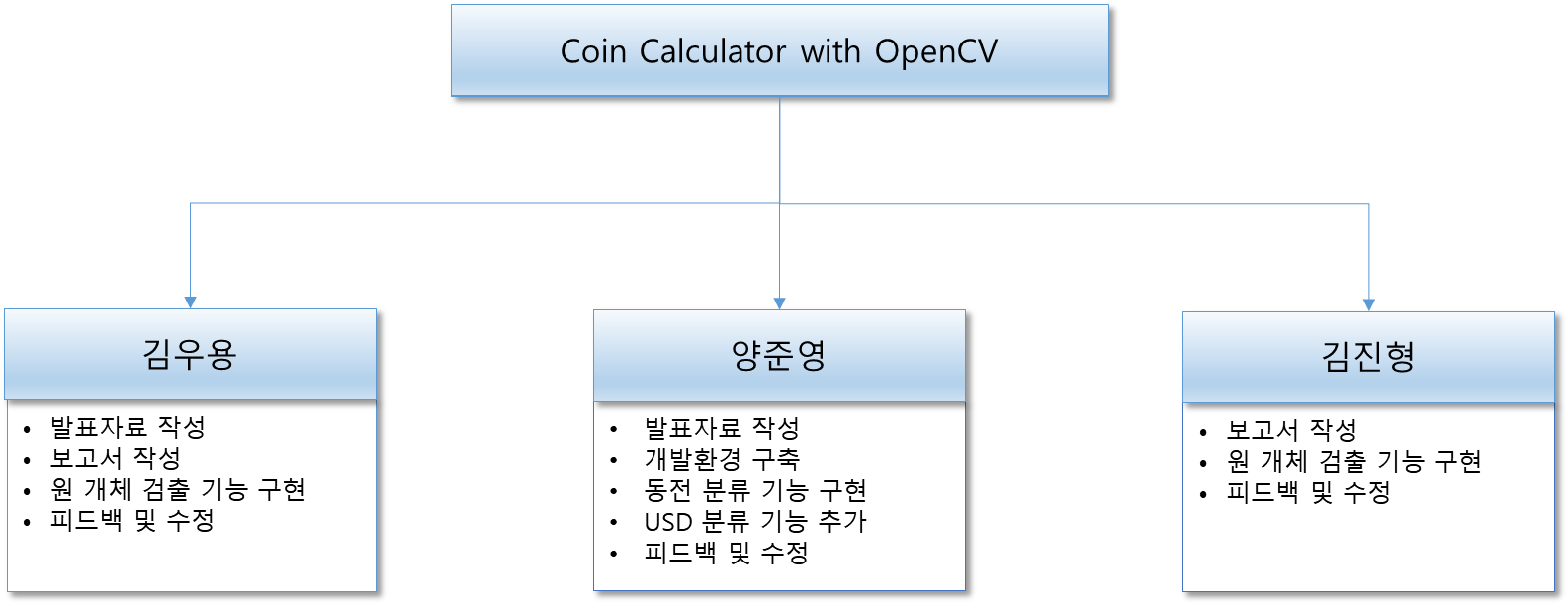
**1.4 Research plan**

Fig 2 프로젝트 업무 분담

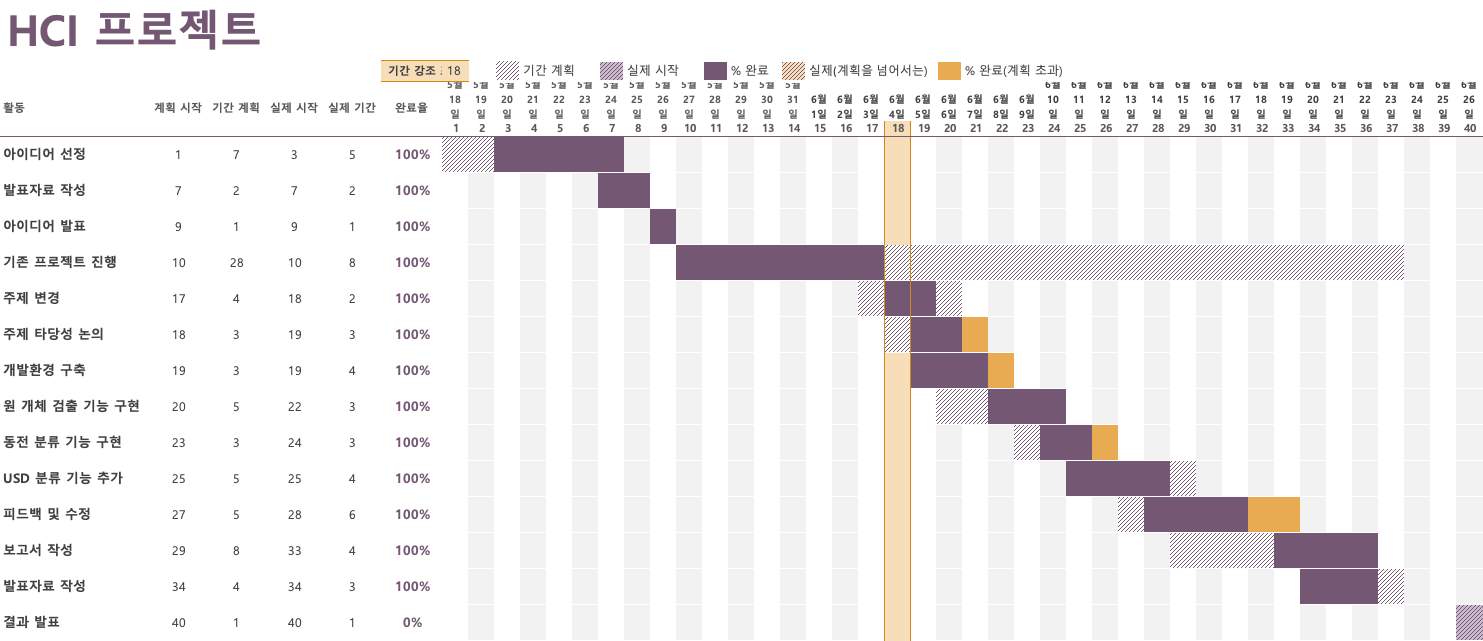


Fig. 3 프로젝트 간트 차트

**1.5 주제 변경 사유**

기존 프로젝트 주제로 얼굴 인식 및 선택적 로우패스필터 적용을 통해 영상에서 지정된 사용자를 제외한 타인의 얼굴을 자동으로 블러 처리하는 프로그램을 선정하고 template matching 방식을 활용하는 방안과 기존 머신러닝 학습 데이터를 활용하는 방안 두 가지를 구상하였으나 다음과 같은 문제가 발생하였다.

첫번째로 template matching을 활용하는 방안은 실시간으로 움직이는 영상에 scaling, rotation 등을 적용하는데 어려움이 발생하였고 SURF, SIFT 알고리즘의 사용을 검토하였으나, 성능상의 문제가 존재하였다.

두번째로 기존의 머신러닝 학습 데이터를 활용하는 방안의 경우 OpenCV에서 제공하는 Haar Cascade는 얼굴의 감지(detection)는 가능하나 인식(recognition)의 어려움이 있었고, 라이브러리 Dlib의 기본 제공 학습 데이터의 경우, DNN을 사용하여 얼굴 감지 및 인식이 가능하였으나 정면이 아닌 옆모습은 성능에 제한이 있어 실시간 영상을 처리하는 데 적합하지 않았다. 이를 해결하기 위한 별도의 새로운 학습 데이터 없이는 프로젝트 진행이 어렵다고 판단하여 순수 영상처리 기법만을 활용한 주제로 변경하는 것을 논의한 결과 본 주제를 선정하게 되었다.

**2. Constraints**

**2.1 Constraints for development process (개발환경에 대한 제한요소)**

본 시스템은 Windows 10 OS 환경의 PC (CPU는 AMD Ryzen 5 3600x 6-Core Processor (~4.1 GHz), 메모리는 16GB RAM)에서 Visual Studio 2019로 OpenCV 3.4.10을 이용하여 구현하였으며, C++언어를 사용하였다.

**2.2 Constraints for execution environment (실행환경에 대한 제한요소)**

본 시스템은 실행 시 사용자가 두 가지의 모드 중 하나를 선택하도록 한다. 이는 mode 변수를 통하여 사용자로부터 ‘1’ 또는 ‘2’를 입력하도록 한다. 또한 하나의 모드를 실행 중 다른 모드로 변경하기 위해서 ‘D’와 ‘W’를 입력할 수 있다. 각각 D는 달러화 W는 원화 모드에 해당된다. 해당 입력을 받으면 다른 모드로 변경이 가능하다. 또한, 본 시스템의 실행 중 하나의 통화 모드에서는 해당 통화만 처리 가능하기에 다른 통화의 동전을 배치하는 경우는 제외한다. 즉, 달러화 모드에서 원화 동전을 배치하는 경우는 고려하지 않는다.

본 시스템에서 동전을 판별하기 위해 실시간으로 처리하는 영상의 경계에 물체가 걸릴 경우 ROI 생성이 제대로 이루어지지 않아 판별이 원활하게 수행되지 않으며, 그렇기에 이러한 상황은 try-catch문을 통해 예외처리 한다. 즉, 동전을 영상의 경계부에 걸치지 않고 실행 화면에 보이는 프레임 안에 동전이 모두 보이도록 배치하여야 한다. 더불어, 원활한 판별을 위하여 동전을 겹쳐서 배치하는 경우도 제외한다.

본 시스템은 동전만 판별하기 때문에 동전 이외의 물체를 판별하려 하는 경우는 제외한다. 즉, 동전 이외의 물체와 섞어서 배치하는 경우는 배제한다.

**2.3 Constraints for performance (시스템 성능에 대한 제한요소)**

본 시스템에서 고해상도의 USB 카메라를 사용하여 해상도가 지나치게 큰 경우 영상처리를 위한 연산 수행하는 속도가 현저히 느려 지고 전체적인 시스템 성능에 악영향을 미친다. 또한, 너무 작은 해상도로 설정할 경우 동전을 판별하는데 어려움이 발생한다. 따라서 동전을 성공적으로 판별하고 시스템의 영상 처리 속도가 크게 저하되지 않도록 고정 해상도 640x480를 가지는 USB 카메라를 사용하였다.

또한, 일정 수준 이상의 해상력을 확보하여 동전의 색상을 인식할 수 있도록 하기위해 카메라를 지면으로부터 20cm 거리에 고정해 놓았기 때문에 일정 개수 이하의 동전만 계산할 수 있다. 본 시스템에서 사용중인 USB 카메라는 수동으로 화각, 초점, 화이트밸런스를 조절할 수 없으며, 따라서 직사광선에 의한 동전 표면의 반사광, 광원의 색온도의 변화와 같은 변수에 민감하다.

**2.4 Additional constraints**

본 시스템에서 판별할 수 있는 동전의 통화는 원화와 달러화로 지정하였다. 동전의 경우에는 원화(구 10원, 10원, 50원, 100원, 500원), 달러화(1센트, 5센트, 1다임, 25센트, 1달러)를 사용하였으며, 원화의 경우 1원과 5원은 시중에 잘 유통되지 않기 때문에 제외하였으며 달러화의 50센트 역시 시중에서 잘 찾아볼 수 없는 동전이므로 제외하였다. 또한, 동전의 상태가 심하게 훼손되어 육안으로도 확인이 불가능한 경우는 제외하였다.

**3. Proposed System**

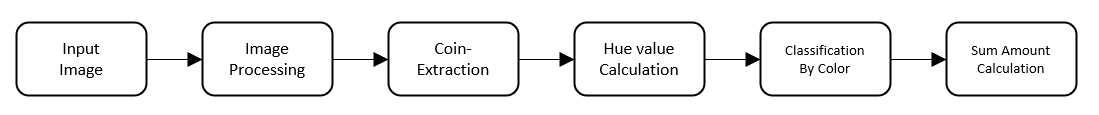
 **3.1 Overall system description**

Fig. 3 시스템의 전체 처리 과정

본 시스템의 전체 처리과정은 **Fig. 3**과 같다. 시스템을 실행하면 계산할 동전의 통화(원화, 달러화)를 선택한다. 통화를 선택한 후 USB 카메라를 통해 실시간 동전 영상을 입력 받으면 입력 영상의 전처리 과정을 거친 후 각 동전 객체를 검출한다. 그리고 분류 과정에서 각 동전 객체의 Hue 값을 사용하기 위해 색상 공간을 RGB에서 HSV로 전환한다. Hue 값을 통해 동전 객체들을 먼저 색상으로 이진분류하고 그 다음 객체의 크기 정보로 동전을 금액별로 최종 분류한다. 마지막으로 분류된 모든 동전의 전체 금액을 계산하여 영상에 표시한다.

**3.2 Detailed explanation of each system module**

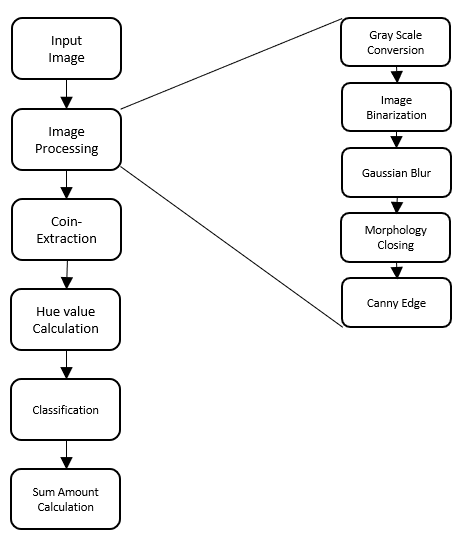
**3.2.1 전처리 과정**

Fig. 4 전처리 과정

USB 카메라에서 입력된 실시간 컬러 영상에 전처리를 수행하여 동전 객체의 검출을 용이하게 한다. 입력 영상의 전처리 과정은 **Fig. 4**와 같다

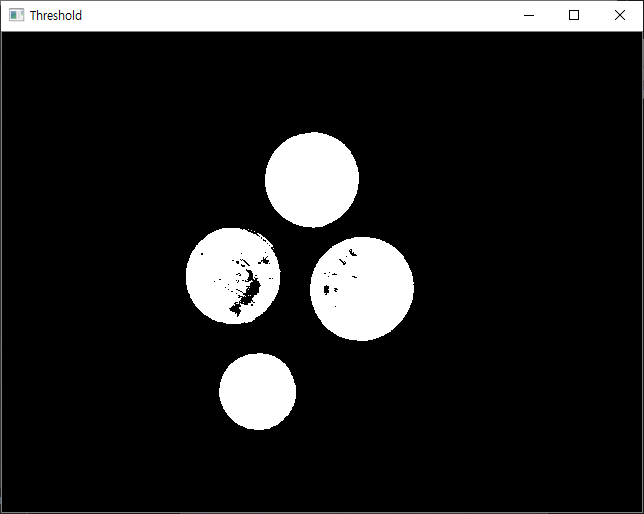


Fig. 5 그레이 스케일 영상

Fig. 6 이진화 영상

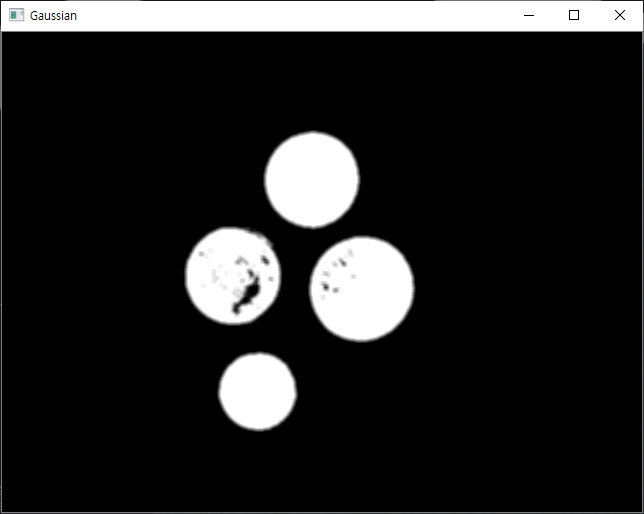
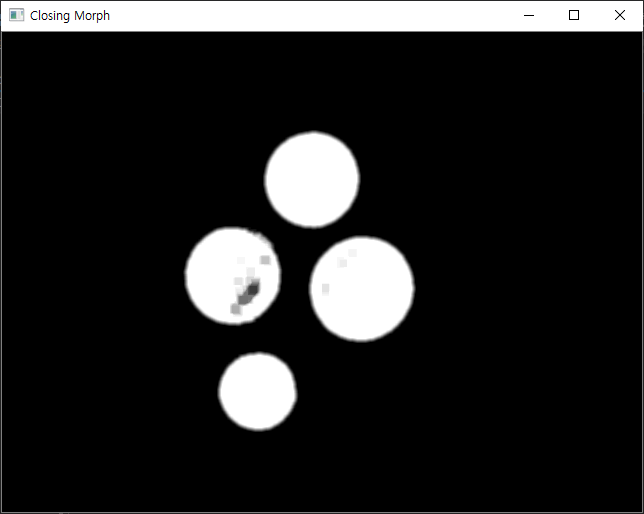
먼저 USB 카메라의 컬러 영상을 그레이 스케일 영상으로 변환한다(**Fig. 5**). 다음으로 임계 값 125를 설정하여 이보다 높은 밝기 값을 가지는 픽셀은 1로, 그렇지 않은 픽셀은 0으로 설정하는 이진화를 통해 그레이 스케일 영상을 이진 영상으로 만든다(**Fig. 6**). 여기서 Otsu의 방법을 사용해 이진화할 경우, 본 시스템에서는 필요하지 않은 동전의 내부 모습이 살아남고 또한 허프 원형 변환을 적용하는 과정에 악영향을 끼쳐 사용하지 않는다.

Fig. 8 모폴로지 연산(Closing)

Fig. 7 가우시안 블러 적용

이진 영상에 로우패스필터의 한 종류인 가우시안 블러를 적용하여 동전 내부 또는 배경의 잡음을 제거하고(**Fig. 7**) 모폴로지 연산 중 7x7 구조요소를 사용한 Closing 연산을 통해 동전 내부의 음각 부분을 메워준다(**Fig. 8**).

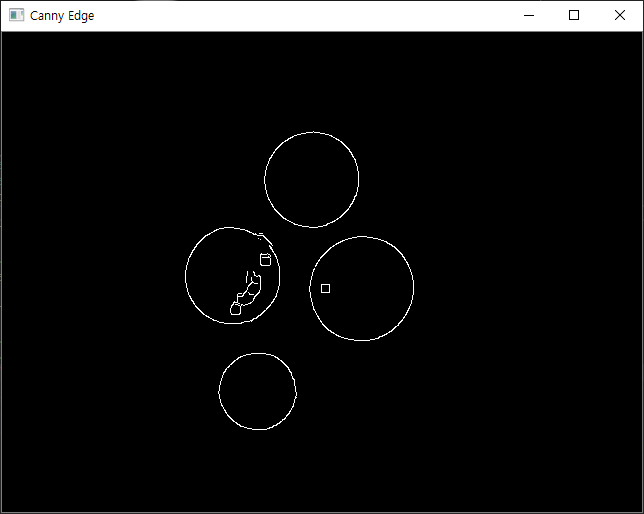


Fig. 9 Canny Edge Detector

마지막으로 Canny Edge Detector를 통해 영상에서 엣지를 검출한다(**Fig. 9**).

**3.2.2 동전 객체 검출 과정**

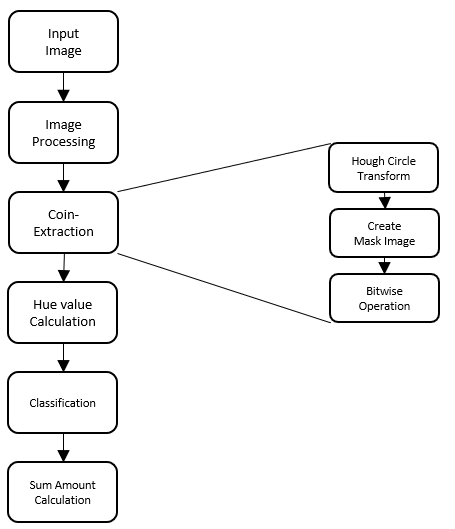


Fig. 10 동전 객체 검출 과정

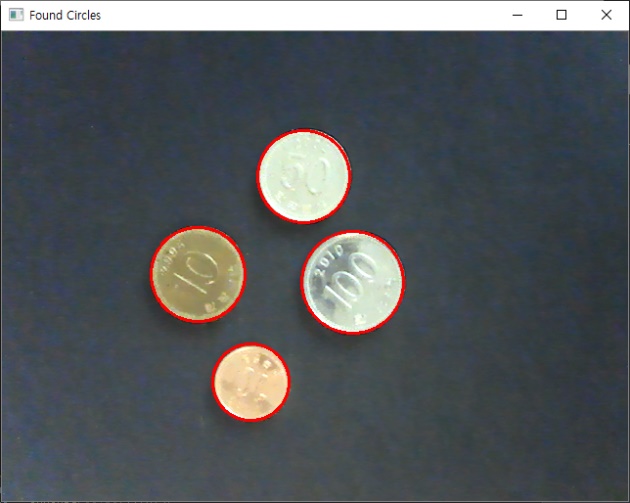
전처리를 통해 얻은 영상에서 동전 객체를 검출하는 과정으로 **Fig. 10**과 같다.

Fig. 11 Hough Circle Transform

Hough Circle Transform을 통해 영상 내에서 원을 찾아 vector<Vec3f> 데이터형에 각 원의 중심 좌표 x, y와 반지름의 길이 r을 저장한다. **Fig. 11**은 Hough Circle Transform을 통해 찾은 원을 시각화하기 위해 영상에 붉은색 원을 그린 모습이다.

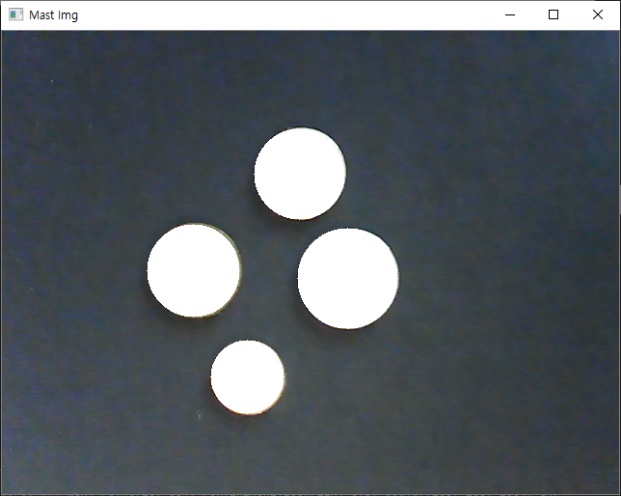
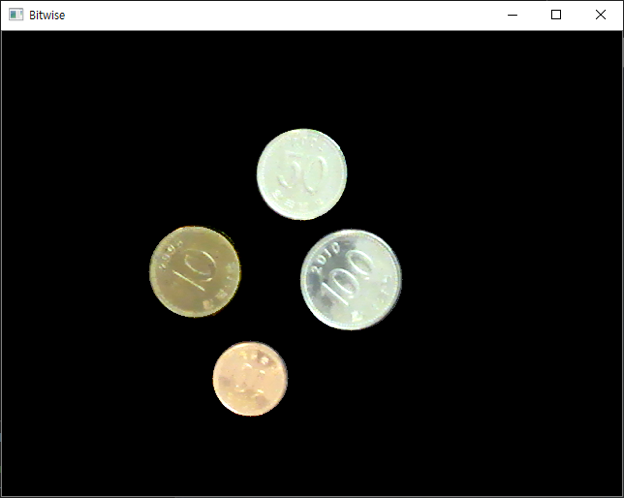


Fig. 13 입력영상 – (~Mask Imgae)

Fig. 12 Mask Image

검출한 원의 위치에 동일한 크기의 Mask 영상(**Fig. 12**)을 생성한 뒤 입력 영상과 반전시킨 Mask 영상의 차영상을 구한다(**Fig. 13**).

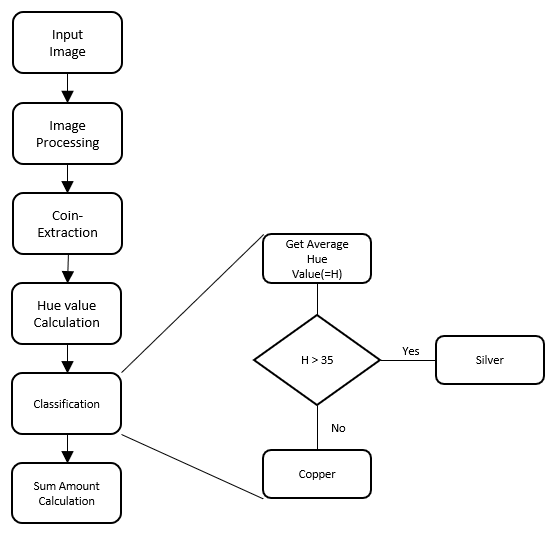
 **3.2.3 동전 분류 과정**

Fig. 14 동전 분류 과정(Hue 값)

본 시스템에서는 동전을 금액별로 분류하기 위해 동전의 Hue 값과 반지름 값 두 가지 정보를 이용한다.

동전의 Hue 값은 비교적 편차 범위가 적도록 하기 위해 동전의 지정한 범위(1/2 \* radius) 안에 메르센 트위스터 난수 발생기를 사용하여 20개의 임의의 점을 찍어 판별한 Hue 값의 평균값을 사용한다. 이를 통해 구한 Hue 값을 if 문을 통해 Silver Coin과 그 외 Coin으로 분류한다(**Fig. 14**).

Hue 값에 의해 분류된 동전은 반지름 값을 통해 분류를 진행한다. 앞서 Hough Circle Transform을 통하여 구한 반지름 값을 아래의 **Table. 1**과 **Table. 2**의 기준에 따라 분류한다. 분류 기준의 Radius 항목(동전의 반지름)의 범위는 아래 **Fig. 18**에서 진행한 외곽부 왜곡 현상으로 인해 발생하는 오차율 계산(**Table. 3**)에 따라 중심 동전의 이론값에서 오차율을 계산하여 범위를 지정하였다. 또한, 오차율을 적용한 범위에서 간극을 조금씩 넓혀줄수록 동전을 더욱 쉽게 판별한다. 따라서 다른 동전의 범위를 침범하지 않고 적절한 판별을 유도할 수 있는 간극을 일부 조정하여 범위표를 작성하였다.

|  |  |
| --- | --- |
| **Radius** | **Coin(달러화)** |
|  | 1 Dollar Coin |
|  | 25 Cent Coin |
|  | 1 Dime Coin |
|  | 5 Cent Coin |
|  | 1 Cent Coin |

**Table. 1 Radius 크기에 따른 동전 분류 (달러화)**

|  |  |
| --- | --- |
| **Radius** | **Coin(원화)** |
|  | 500원 동전 |
|  | 100원 동전 |
|  | 50원 동전 |
|  | 10원 동전 |
|  | (구) 10원 동전 |

**Table. 2 Radius 크기에 따른 동전 분류 (원화)**

현재 실행되고 있는 모드에 따라 크기에 따른 분류에 적용하는 기준이 다르다. 각각의 실행모드에 맞는 분류 기준을 적용하여 분류를 진행하면 sumAmount 변수에 해당 동전의 가치(값)을 누적하여 더한다. 즉, 마지막 결과값에서 출력되는 전체 동전의 총합을 계산한다. 물론 이는 실시간 영상을 처리하기 때문에 총액 또한 실시간으로 연산이 수행되어 동전의 총액이 변할 때 마다 변동이 생기고 이를 출력한다.

**3.3 Implementation method and tool**

|  |  |  |
| --- | --- | --- |
| OpenCV version | OpenCV 3.4.10 | |
| OpenCV Method | cvtColor() | 이미지의 색상 영역을 변환한다. |
| threshold | 회색조 영상을 주어진 임계 값에 따라 이진화한다. |
| GaussianBlur | Gaussian Blur를 사용하여 잡음을 제거한다. |
| getStructuringElement | 매개변수에 맞는 구조요소를 생성한다. |
| morphologyEx | Closing 연산을 수행한다. |
| Canny | Canny Edge Detector로 Edge를 검출한다. |
| HoughCircles | 영상 내에서 원을 검출한다. |

시계, 테이블이(가) 표시된 사진

자동 생성된 설명실내, 개체, 검은색, 옅은이(가) 표시된 사진

자동 생성된 설명**4. Experimental Result**

Fig. 16 실험 환경(조명)

Fig. 15 실험 환경(시스템의 장치 구성)

**4.1 Experimental environment**

본 시스템의 실험 환경은 하드웨어적으로 USB 카메라, 동전 받침대, 카메라를 고정하는 거치대, PC, 조명으로 구성되며 **Fig. 15**와 **Fig. 16**에서 확인할 수 있다. PC는 Windows 10 OS 환경이다.(CPU: AMD Ryzen 5 3600x 6-Core Processor (~4.1 GHz). 메모리: 16GB DDR4 RAM)

USB 카메라는 Samsung SPC-A1200MB 제품을 사용하였으며 해상도는 640x480이다. 또한, 해당 카메라는 자동으로 White Balance 보정이 되는 제품이다. USB 카메라가 거치되어 있는 거치대의 경우 동전과 동전의 반지름 판별이 용이하도록 받침대에서 카메라까지의 거리가 약 20cm가 되도록 구성하였다. 또한, 구도에 따른 외곽부 왜곡을 최소화하기 위해서 직부감(지면과 평행) 환경으로 구성하였다(**Fig. 17**). 그러나 카메라에 따라 직부감으로 촬영을 하더라도 외곽부 왜곡으로 인해서 동전의 반지름의 값에 오차가 발생할 수 있으며 이에 따라 반지름의 오차범위를 (동전에 따른 차등 오차율)로 계산을 진행하였다(**Fig. 18**). A, B, D, E 동전은 외곽에 위치하고 있으며 C 동전은 중심에 위치하고 있다. **Table. 3**을 확인하면 외곽부 동전의 반지름은 C 동전과 차이가 있음을 알 수 있다. 이 값을 토대로 오차율을 계산하였다.

동전 받침대는 빛 반사가 제일 적도록 검은색을 사용하였으며 동전은 서로 겹치지 않도록 배치하였다.

조명의 경우에는 주광원으로는 커튼(**Fig. 19**)으로 걸러진 태양광과 IKEA의 트로드프리 조명을 사용하였다. 태양광의 경우 색온도와 시간에 따라 달라지므로 실험 시간을 오후 2시 ~ 4시로 지정하여 진행하였다. 또한, 날씨에 따라 색온도가 달라지므로 실험 당일 날씨가 비교적 균일한 날들로 실험일을 지정하였다(**Fig. 20**). 구름조금 날씨의 경우 오후 2시 ~ 4시 사이 태양광 색온도가 대략 4500K ~ 5400K인 것으로 알려져 있다. 해당 실험 공간은 남동향 공간에 지상 8층 높이이므로 해당 시간 일조량에 영향을 받는 외부적 요소가 적다. 보조 조명은 400루멘 밝기와 4000K 색온도로 설정하였다. 또한 동전에 비추는 빛의 광량이 과한 경우, 동전에서 반사광이 발생하고 이로 인해 실험에 사용한 카메라가 동전의 색상을 원활히 묘사하지 못하는 문제점을 해결하기 위해 전구의 직광을 사용하지 않고 디퓨저가 달려있는 전구를 사용하여 빛 반사가 적은 산란광 환경을 조성하였다. 이 둘의 조합을 통해 동전을 판별하는 환경의 색온도를 비교적 일정한 범위 안에서 유지되도록 하였다. 해당 과정은 노출계와 색온도 검출기를 활용하여 측정하였다.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **A** | 40 | 39 | 41 | 40 | 41 | 40 | 40 | 41 | 40 | 40 |
| **B** | 40 | 41 | 41 | 41 | 40 | 42 | 40 | 42 | 40 | 40 |
| **C** | 41 | 41 | 41 | 41 | 40 | 41 | 41 | 41 | 41 | 41 |
| **D** | 42 | 40 | 42 | 41 | 40 | 40 | 41 | 41 | 42 | 41 |
| **E** | 40 | 41 | 42 | 41 | 41 | 40 | 42 | 41 | 41 | 43 |
| 오차율(%) = | | | | | | | | | | |

**Table. 3 외곽부 왜곡을 측정하여 오차율을 계산하기 위한 5영역 동전의 반지름**

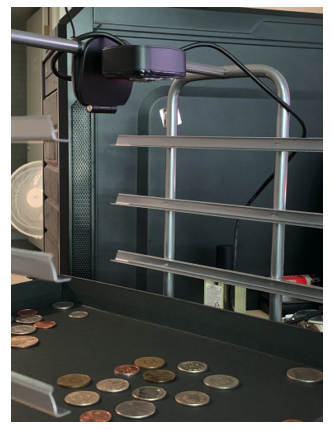
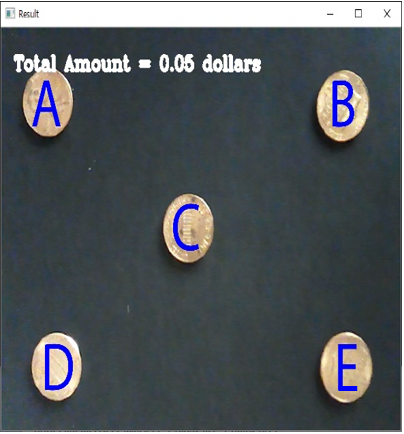


Fig. 18 외곽부 왜곡 실험

Fig. 17 카메라 세팅 모습

Fig. 20 과천 6월 날씨

Fig. 19 커튼 환경

**4.2 Performance measure**

동전 객체의 반지름 정보만을 활용하여 본 시스템을 구현해 동전을 판별할 경우 크기는 같으나 색과 가치가 다른 동전을 구별하는데 어려움이 있다 (특히 원화의 구 10원과 50원). 따라서, 해당 경우를 해결할 수 있도록 동전의 Hue 값을 추가 적용하여 판별하는 모델로 구현하였다.

이 경우, 원화 기준 한 번의 수행에 14개의 각기 다른 동전을 사용하고 같은 구성으로 총 100번의 수행을 통해 1,400개의 객체를 판별하였을 때, 41개의 객체가 오분류 되었다. 이를 통해 총 97.14%의 정확도를 지니고 있음을 계산할 수 있다. 또한, 달러화 기준 한 번의 수행에 20개의 각기 다른 동전을 사용하고 같은 구성으로 총 70번의 수행을 통해, 1,400개의 객체를 판별하였을 때, 18개의 객체가 오분류 되었다. 이를 통해 98.71%의 정확도를 지니고 있음을 계산할 수 있다. 따라서, 본 시스템의 정확도는 평균 97.93%로 나타난다.

**4.3 Experimental results**

위와 같은 실험 장비와 환경에서 실행한 결과는 **Fig. 21**과 같다. (a)는 달러화 모드를 선택하여 달러화 동전 객체를 검출하고 총 금액을 계산하여 영상의 상단 왼쪽에 나타낸 결과 영상이다. (b)는 원화 모드를 선택한 결과 영상이다.

개체이(가) 표시된 사진

자동 생성된 설명**검은색, 사진, 전면, 작은이(가) 표시된 사진

자동 생성된 설명**

(a) 달러화 결과 영상

(b) 원화 결과 영상

**Fig. 21 실행 결과 영상**

**4.4 Discussion**

컴퓨터, 노트북이(가) 표시된 사진

자동 생성된 설명실행 결과 달러화의 1센트, 5센트, 1다임, 25센트, 1달러를 모두 검출하여 총 금액을 계산한 결과 영상을 확인하였고, 원화의 구 10원, 10원, 50원, 100원, 500원도 동일하게 결과 영상을 확인하였다. 그러나 **Fig 22**와 같이 동전 여러 개가 맞닿아 있으면 정확한 검출이 어려워진다.

Fig. 22 부정확한 결과 영상

**5. Conclusion**

본 시스템은 실시간으로 주어지는 디지털 영상을 처리하여 허프 원형 변환과 HSV 색상모델을 활용해 원화와 달러화 두가지 통화의 소량의 동전을 판별하고 이 동전의 총액을 실시간으로 출력한다. 본 연구에서 기존의 허프 원형 변환으로 객체를 검출하여 동전을 인식하는 시도에서 크기가 같으나 색상이 다르고 가치가 다른 동전을 구별하고자 함에 있어 HSV 색상모델을 활용하고 나아가 원화 뿐만 아니라 달러화 통화에도 적용시켰다. 또한, USB 카메라가 지니고 있는 구조적 결함인 외곽부 왜곡을 해결하기 위하여 5개의 영역을 추출하여 측정된 반지름 값의 표본을 활용하여 오차율을 계산하고 좀 더 정확하고 쉽게 판별하기 위한 반지름의 범위를 기준으로 제작하였다. 마지막으로, 상수 반지름 값을 기반으로 동전을 판별하는 현 시스템 구조에서 카메라와 물체의 거리를 파악하고 이에 비례한 반지름 값 변화를 계산하는 방식으로 카메라와 물체 거리에 상관없이 사용할 수 있도록 개선하는 방법이 필요해 보인다. 이러한 개선을 통하여 최종적으로 별도의 동전 받침대 없이 본 시스템 사용할 수 있는 방법을 고안한다면, 개개인의 스마트폰에서 어플리케이션 형식으로 사용할 수 있도록 구현할 수 있을 것이고 이에 따라 장소와 환경에 구애 받지 않을 수 있도록 개선할 수 있을 것이다.

**Reference**

1) 디지털경제뉴스, 네이버, 네이버랩스, ‘CVPR 2020’에서 AI 비전 기술 발표, 이동진, 2020.06.17, http://www.denews.co.kr/news/articleView.html?idxno=13853

2) M. Matsumoto and T. Nishimura “Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator” : ACM Transactions on Modeling and Computer Simulation, Vol. 8, No. 1, January 1998

3) J. Cheong “Algorithm Design for Circles and Ellipses Detection using Hough Transform” : *조선대학교 전산통계확과*, pp.23 – 24, 1994