

# **[EE474] Introduction to Multimedia**

## **Term Project Final Report**

### **“Real Time Laundry”**

Project 기간 : 2016. 5. 19 ~ 2016. 6. 8

< Group 1 >	
20100874	조 정 열
20130388	양 자 운
20130402	유 기 중
20154920	송 우 태

## 1. Purpose & Motivation

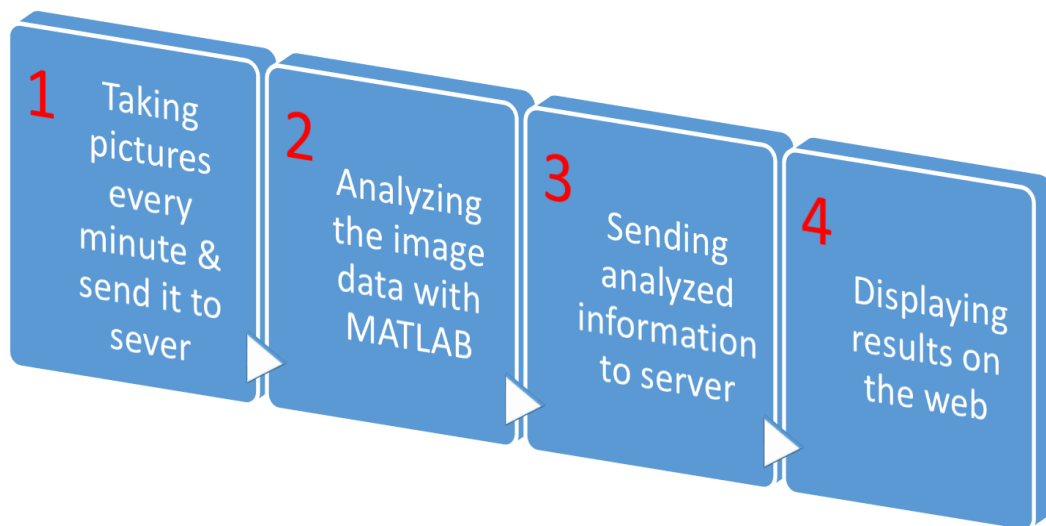
- Project 'Real Time Laundry'는 쉽게 말해 세탁기를 사용할 수 있는지의 여부를 실시간으로 확인할 수 있도록 하는 것을 목적으로 한다. KAIST는 학생의 대부분이 기숙사에서 생활하고 있다. 그들은 세탁물을 세탁소나 빨래방에 맡기기도 하지만, 각 기숙사에 있는 세탁실에서 세탁기를 이용해 빨래를 하는 경우가 대부분이다. 그러나 기숙사에 사는 사람 수에 비해서 세탁기가 적거나, 사람들끼리 빨래 하는 시간이 겹치는 경우가 생겨 세탁기를 이용하지 못하는 문제가 종종 생긴다. 세탁기를 더 많이 구입하는 것도 한 가지 방법일 수 있겠지만, 구입에는 경제적인 비용이 필요하며 세탁기가 사용되지 않는 시간이 충분히 많다는 것을 보면 비효율적이다. 세탁기가 모두 사용 중이라면 다른 세탁기의 동작이 끝날 때까지 기다릴 수도 있겠지만 바쁜 학생들은 무작정 기다릴 수만도 없다. 게다가 설상가상으로 세탁물이 많다면 세탁물을 갖고 다시 방으로 돌아가기도 힘들 것이다. 세탁을 하기 위해 세탁물을 갖고 방을 나서기 전에 사용할 수 있는 세탁기가 있는지 미리 확인할 수 있다면 문제를 해결할 수 있을 것이다.



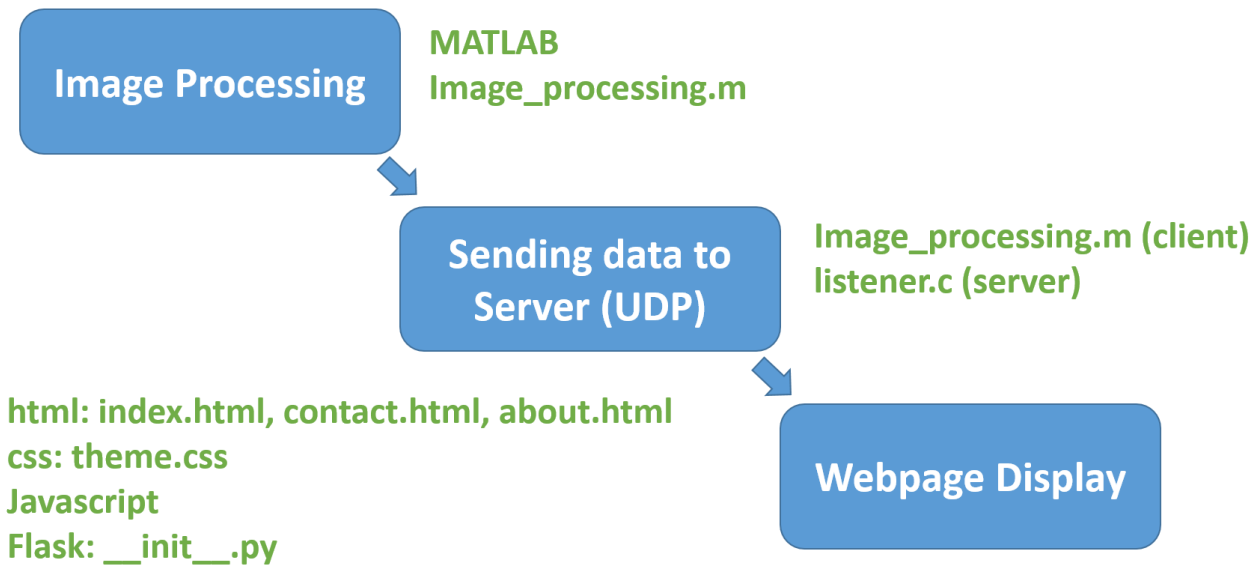
- 이 해결책은 기숙사에서 사용되는 경우 외에, 외국에도 적용할 수 있을 것이다. 유럽이나 미국의 도시에는 세탁기를 모아놓은 세탁실이 따로 있다. 그래서 집에 세탁기가 없더라도 빨래를 하려는 사람은 빨래를 갖고 세탁실을 방문한다. 큰 세탁실에는 세탁기가 많아서 임의의 시간에 세탁실을 방문하더라도 높은 확률로 바로 세탁기를 사용할 수 있다. 그러나 작은 세탁실의 경우에는 그렇지 않다. 그리고 세탁실까지의 이동거리도 기숙사의 경우 보다 훨씬 길다. 여기에 이 system을 적용한다면 빨래로 인해서 낭비하게 될 수 있는 시간을 많이 절약할 수 있을 것이다.

## 2. Scheme & process

- Project의 필요성과 목적에 대해서는 위에서 충분히 설명하였다. 사용 가능한 세탁기가 있는지를 확인하기 위해서는 여러 과정이 필요하다. Webcam으로 찍은 세탁실의 세탁기 사진들이 가장 먼저 필요하며 이 사진을 image processing 한다. Image processing을 통해서 각 세탁기가 현재 동작하고 있는지를 판단할 수 있을 것이다. 그리고 이 정보를 서버 통신으로 실시간 전송한다. 결과적으로 client는 세탁실에는 총 몇 대의 세탁기가 있고 사용할 수 있는지 확인할 수 있다.



- 위의 그림에서 설명하고 있는 대략적인 scheme을 설명하면 다음과 같다.
  - 1) 세탁기의 사진을 일정한 시간마다 촬영(캡처)하고 서버로 보낸다.
  - 2) 받은 사진 정보를 MATLAB(image\_process.m)으로 분석한다.
  - 3) 분석한 정보를 다시 서버로 보낸다.
  - 4) 서버에서는 웹 페이지에 결과를 출력해낸다. (MM Characteristics : Information sharing via Network)
- 각 단계에서 사용된 프로그램과 code의 구성은 아래에서 확인할 수 있다.



### 1) Image Processing

위 scheme의 2번 과정이라고 할 수 있는 부분이다. Matlab을 이용해서 구현한 code인 'image\_processing.m' 으로 받아온 세탁기의 image들을 processing한다. 여기서는 다른 프로그램은 사용되지 않으며 오직 matlab에만 의존한다. Matlab 내부에는 여러 함수가 구현되어 있기 때문에 filter를 씌우는 작업도 손쉽게 할 수 있었다.

### 2) Sending data to Server

UDP를 이용해서 세탁기의 state를 전송한다. 전송하는 것은 matlab에서 가능하며 image\_processing.m 내부에 구현되어 있다. 전송된 정보는 listener.c code에서 읽어 들인다.

### 3) Webpage Display

html, css, javascript, Flask 등으로 웹 페이지를 제작하였다. 받아온 정보를 분석해서 세탁기가 총 몇 대인지, 사용 가능한 개수는 몇 개인지 등을 알아낸다. 이 분석된 정보를 보기 좋게 webpage에 display 한다. 이는 Multimedia의 특징 중 하나인 Information sharing via network를 충족한다고 볼 수 있다.

### 3. Method

#### 1) Image Processing

Image processing 은 세탁기의 Timer를 Filter를 통해 포착하여 사진 안에 세탁기가 몇 개 있는지, Timer 내부에 Light가 켜져 있는지 확인을 하여 결과를 도출하기 위해 필요하다. 사진 안의 세탁기 개수를 detect 하기 위해서는 필터를 적용하여 분석하거나 object를 detect 할 수 있는 다른 알고리즘이 필요하다. 우리가 시도해 본 case를 모두 살펴볼 것이다. 우선 세탁기가 2개인 경우를 sample로 하여 적용해보았다.



이 원본 사진을 갖고 필터와 알고리즘을 적용해볼 것이다.

#### - detectMSERFeatures()

detectMSERFeatures는 matlab에서 구현되어 있는 함수이다. 이 때 MSER은 Maximally Stable Extremal Regions의 약자이다. 이 함수를 사용하면 MSERRegions object를 반환하는데, 세탁기에 대해 필터를 사용했을 경우의 결과는 아래의 사진과 같았다. 이 알고리즘은 간단하게 다음과 같이 사용할 수 있다.

```
img = rgb2gray(imread('image.jpg'));
```

```
[regions, cc] = detectMSERFeatures(img, 'ThresholdDelta',19,'MaxAreaVariation',0.6);
```

```
imshow(img); hold on;
```

```
plot(regions);
```



여기서 ThresholdDelta의 경우 Region이 검출되기 위해서 gray scale의 최대값과 최소값의 차이가 이 범위 안에 있어야 함을 의미한다. MaxAreaVariation의 경우 만약 ThresholdDelta의 범위에 gray scale 값이 모두 포함된다 하더라도 이 Maximum Variation값을 벗어나는 Variation값을 벗어날 경우 그것은 Region으로 검출이 되지 않음을 의미한다. 따라서 위의 코드의 경우는 gray scale의 최대와 최소 차이가 19 이하이고, Variation이 0.6보다 작아야 함을 의미한다.

이 함수의 문제는 햇빛을 반사하는 부분과 같이 전혀 의미 없는 직사각형의 경우가 Region으로 검출되기 좋다는 것과, Timer위에 존재하는 손잡이 부분이 Timer와 크기, 변수 값들이 매우 흡사하여서 Timer만을 검출하는 것이 굉장히 어렵다는 점이였다. 이런 이유로 인해 detectMSERFeatures는 사용하지 못하였다.

#### - Laplacian Filter

Edge를 검출함에 있어서 미분 방정식을 사용할 때, 1차 미분과 2차미분을 사용할 수 있다. 라플라시안 연산자의 경우 2차 미분의 대표적 연산자이다. 라플라시안 함수는 아래와 같다.

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

1차 미분 filter의 경우 edge를 검출하기 위해서는 가로와 세로 각 방향에 대해서 따로 mask를 제작하여 적용하지만, 2차 미분 filter의 경우는 2차원 그림에 대해서 방향성을 따져주지 않아도 사용할 수 있다는 장점이 있다. 라플라시안 필터의 mask의 예시는 다음과 같다.

0	1	0
1	-4	1
0	1	0

0	-1	0
-1	4	-1
0	-1	0

1	1	1
1	-8	1
1	1	1

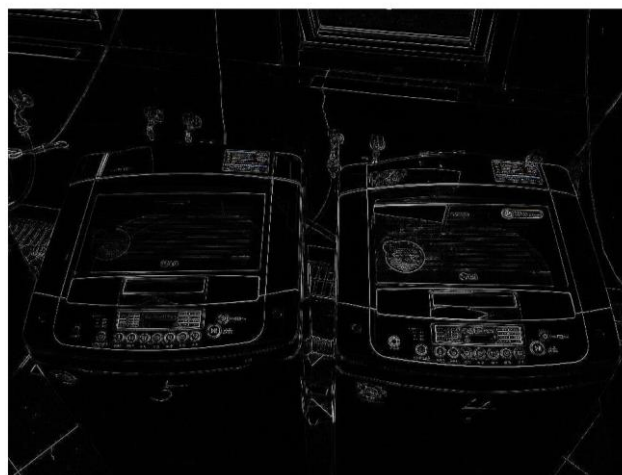
-1	-1	-1
-1	8	-1
-1	-1	-1

라플라시안 필터의 경우는 결국 grey scale 사진에서 명암도의 불연속점을 mask 연산을 통해 더욱 두드러지게 나타내어 edge를 detect하는 원리이다.

$$\nabla^2 = \frac{4}{(\alpha+1)} \begin{bmatrix} \frac{\alpha}{4} & \frac{1-\alpha}{4} & \frac{\alpha}{4} \\ \frac{1-\alpha}{4} & -1 & \frac{1-\alpha}{4} \\ \frac{\alpha}{4} & \frac{1-\alpha}{4} & \frac{\alpha}{4} \end{bmatrix}$$

Matlab에서 라플라시안 필터의 mask는 위의 행렬을 의미한다. Alpha의 값을 달리하여 mask를 변화시킬 수 있다. 사용한 코드와 결과 사진은 다음과 같다.

```
filter2 = fspecial('laplacian', 0.1);
```

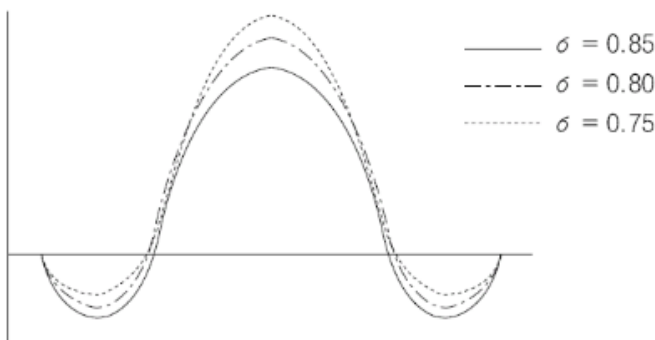


이 필터를 적용한 경우 edge 는 굉장히 잘 detect가 됨을 알 수 있다. 하지만 Timer의 edge만의 특이사항이 전혀 존재하지 않기 때문에 Laplacian filter를 사용하지 못하였다.

#### - LOG (Laplacian of Gaussian) Filter

어떤 필터를 사용하여도 Timer만의 특이사항이 전혀 존재하지 않았기 때문에 검출하지 못하였다. 그래서 세탁기에 영향을 주지 않는 선에서 assumption을 하기로 결정했다. 세탁기의 Timer 부분 좌우에 Pink Sticker를 부착하는 것이다. Sticker를 검출할 수 있다면 Timer의 범위를 Sticker를 통해 쉽게 구할 수 있게 되었다. 이후 세탁기가 사용 중일 때 Timer의 범위 내에 존재하는 Red Light를 선명하고 확실하게 검출하는 것이다. 우리는 이를 LOG filter를 사용하여 해결하였다. LOG filter는 Laplacian filter를 적용하기 전에 Gaussian Smoothing을 먼저 적용하여서, Laplacian filter가 굉장히 취약한 부분인 잡음을 제거하여 더욱 효과를 높인 filter이다. 함수에 Gaussian 연산자를 먼저 적용한 후 Laplacian 연산자를 적용한 결과는 다음과 같다.

$$\text{LoG}(x, y) = \frac{1}{\pi\sigma^4} \left[ 1 - \frac{(x^2 + y^2)}{2\sigma^2} \right] - e^{-\frac{(x^2 + y^2)}{2\sigma^2}}$$



0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0



matlab에서 LOG filter의 코드의 경우 다음과 같이 사용할 수 있다.

```
filter2 = fspecial('log',[50 50], 0.45);
```

여기서 변수는 각각 Mask의 크기와 Sigma 의 값이다. Mask의 크기의 경우는 크기가 커질 수록 색이 잘 검출되는 것을 확인할 수 있었다. 다음 사진들은 Sigma 변수 값의 차이에 따른 결과 차이이다.



좌측의 경우는 Sigma value가 0.5이고 우측의 경우는 Sigma value가 0.45이다. Sigma value는 Gaussian Smoothing의 정도에 관한 변수인 만큼 값의 크기가 커질 수록 Smoothing을 통한 잡음이 사라지는 반면에 과한 Smoothing으로 인해 필요한 색들 조차도 연해지는 현상이 나타난다. 값의 크기가 작을 수록 잡음을 제거하는 효과는 줄어드는 반면에 필요한 색이 훨씬 선명해지는 것을 확인할 수 있다.

즉, log filter를 사용해서 detection에서 여러 장점을 얻었다고 정리할 수 있으며 그 장점은 다음과 같다. 첫째로 Timer의 불빛을 육안으로 확인하기 어려운 경우에 filter를 적용하면 정확하게 구별할 수 있었다. 둘째로 적절한 parameter로 filter를 적용한 경우, sticker와 비슷한 색을 갖는 성분들은 모두 gray 성분을 높게 가지게 되므로 sticker의 detection 정확성이 증가했다. 셋째로 주변에 빛이 많이 있는 적게 있는 상관없이 detection 할 수 있다.

지금까지의 Trial and Error를 겪으며, 결론적으로 Pink Sticker를 통해 Timer의 위치를 확인하고, LoG filter를 통해서 Timer 내부에 있는 Red Light를 선명하게 검출하는 방법을 사용하기로 했다.

## 2) Flask

Flask는 작은 코어와 easy-to-extend의 성질을 가진 Python기반 웹 프레임워크이다.



Flask 웹 적용 코드는 더 명시적이기 때문에 가장 Python과 가깝다고 볼 수 있다. 간단한 어플리케이션을 만들고 돌리는 데에는 진부한 반복 코드가 없는 Flask가 초보자들이 사용하기에 매우 좋다.

## 3) HTML

HTML은 하이퍼텍스트 마크업 언어라는 의미의 웹 페이지를 위한 지배적인 마크업 언어다. HTML은 제목, 단락, 목록 등과 같은 본문을 위한 구조적 의미를 나타내는 것뿐만 아니라 링크, 인용과 그 밖의 항목으로 구조적 문서를 만들 수 있는 방법을 제공한다. 그리고 이미지와 객체를 내장하고 대화형 양식을 생성하는 데 사용될 수 있다. HTML은 웹 페이지 콘텐츠 안의 꺾쇠 괄호(<,>)에 둘러싸인 "태그"로 되어있는 HTML 요소 형태로 작성한다. HTML은 웹 브라우저와 같은 HTML 처리 장치의 행동에 영향을 주는 자바스크립트와 본문과 그 밖의 항목의 외관과 배치를 정의하는 CSS 같은 스크립트를 포함하거나 불러올 수 있다. HTML과 CSS 표준의 공동 책임자인 W3C는 명확하고 표상적인 마크업을 위하여 CSS의 사용을 권장한다.

## 4) CSS

종속형 시트 또는 캐스케이딩 스타일 시트(Cascading Style Sheets, CSS)는 마크업 언어가 실제 표시되는 방법을 기술하는 언어로, HTML과 XHTML에 주로 쓰이며, XML에서도 사용할 수 있다. W3C의 표준이며, 레이아웃과 스타일을 정의할 때의 자유도가 높다. 마크

업 언어가 웹사이트의 몸체를 담당한다면 CSS는 옷과 액세서리 같은 꾸미는 역할을 담당한다고 할 수 있다. 즉, HTML 구조는 그대로 두고 CSS 파일만 변경해도 전혀 다른 웹사이트처럼 꾸밀 수 있다. CSS Zen Garden은 하나의 HTML(필요에 따라 최소한의 수정이 있다.)을 CSS만으로 얼마나 다양하게 표현할 수 있는지 알 수 있는 웹사이트이다. 현재 개발 중인 CSS3의 경우 그림자 효과, 그라데이션, 변형 등 그래픽 편집 프로그램으로 제작한 이미지를 대체할 수 있는 기능이 추가되었다. 또한 다양한 애니메이션 기능이 추가되어 어도비 플래시를 어느 정도 대체하고 있다.

## 5) JavaScript

자바스크립트(Javascript)는 객체 기반의 스크립트 프로그래밍 언어이다. 이 언어는 웹 브라우저 내에서 주로 사용하며, 다른 응용 프로그램의 내장 객체에도 접근할 수 있는 기능을 가지고 있다. 또한 Node.js와 같은 런타임 환경과 같이 서버 사이드 네트워크 프로그래밍에도 사용되고 있다. 자바스크립트는 본래 넷스케이프 커뮤니케이션즈 코퍼레이션의 브랜던 에이크가 처음에는 모카(Mocha)라는 이름으로, 나중에는 라이브스크립트라는 이름으로 개발하였으며, 최종적으로 자바스크립트가 되었다. 자바스크립트가 썬 마이크로시스템즈의 자바와 구문(syntax)이 유사한 점도 있지만, 이는 사실 두 언어 모두 C 언어의 기본 구문을 바탕 했기 때문이고, 자바와 자바스크립트는 직접적인 관련성이 없다. 네트워크 프로그래밍에 이용되기 때문에 이번 프로젝트에서 소켓 프로그래밍 부분을 Javascript를 사용하여 코딩 하였다.

## 4. Source Code

### 1) Image\_processing.m

- Matlab을 통해서 sticker를 detection하고 각 세탁기가 사용 가능한지 image processing으로 알아보는 code의 전문이다. 알고리즘이 핵심 내용이라고 할 수 있다.

image_processing.m	Explanation
<pre>count = 0; first = 1; tog = 0; diameter = 0; diameter2 = 0;</pre>	아래 알고리즘에서 사용될 parameter들이 정의되는 부분이다. loop 안에서 index 될 때 사용되는 등 여러 곳에 사용된다.

<pre> quit = 0; on = 0; on2 = 0; count_tog = 0;  sticker_array = zeros(4,3)+1; img = imread('2 (4).jpg');  filter2 = fspecial('log',[50 50], 0.47); img2 = imfilter(img,filter2,'replicate'); i = 1; j = 1;  [height, width, z1] = size(img); count2 = 0;  % get sticker info while j &lt; width     while i &lt; height         if i &gt;= height    j &gt;= width             break;         end         count_tog = 1;         first = 1;         while img(i,j,1) &gt; 180 &amp;&amp; img(i,j,3) &gt; 180 &amp;&amp; img (i, j, 2) &lt; 30             if first == 1                 sticker_array(count+1, 1) = j;                 sticker_array(count+1, 2) = i;                 if count_tog == 1                     count = count + 1;                     count_tog = 0;                 end             end             first = 0;             img(i, j, :) = 0;              j = j + 1;             tog = 1;             diameter = diameter + 1; </pre>	<p>sticker_array에 세탁기에 대한 정보가 들어간다. 세탁기의 개수가 최대 2개라고 먼저 가정하였다. img에는 세탁기 그림 정보를 불러온다.</p> <p>log filter를 제작해서 씌우는 작업이다. [50 50] 사이즈를 사용한 것과 std를 0.47으로 사용한 것은 실험적으로 얻어낸 값들이다. 가장 detect에 유용하게 사용될 수 있을 value이다.</p> <p>들어온 사진의 size를 측정한다. loop가 돌 때 유용하게 사용될 수 있다.</p> <p>j와 i를 사용해서 불러온 사진의 전 구간을 돌아다니며 알고리즘을 적용할 것이다. 세탁기가 일반적으로 옆으로 나란히 있으므로 위에서 아래로, 즉 동일한 width 자리에서 height 부터 살펴본다. 그리고 처음에는 terminate 조건을 걸어준다.</p> <p>sticker를 찾기 위해서 적절한 r,g,b 정보가 있는지 탐색한다. 그렇다면 그 곳의 위치를 sticker_array의 첫, 둘째 자리에 저장한다. 그리고 멈추지 않은 상태로 적당한 diameter를 찾는다. 그리고 sticker_array의 셋째 자리에 값을 넣는다. count의 값이 sticker를 찾은 경우에 증가한다.</p>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

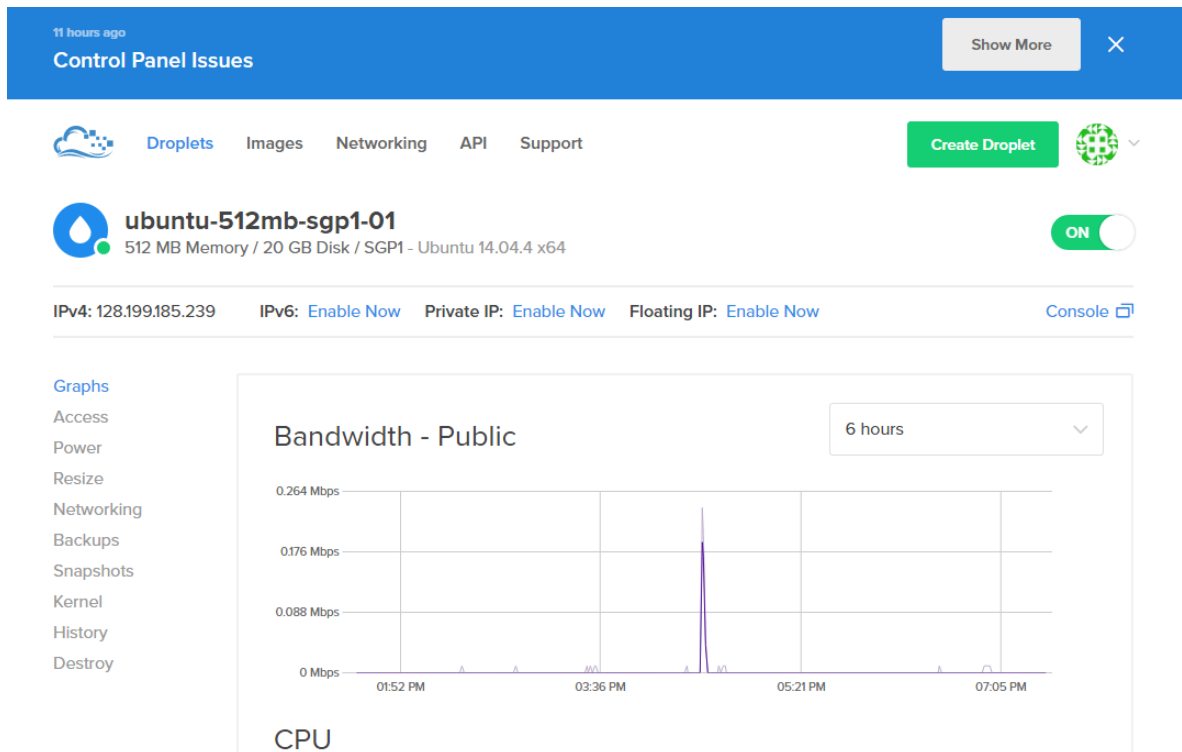
<pre> end  if tog == 1     j = j + 10;     sticker_array(count,3) = diameter;     tog = 0; end  i = i + 1; end j = j + 1; i = 1; end  if count == 2     for j = sticker_array(1,1)+sticker_array(1,3)+30:sticker_array(2,1)-15         for i = sticker_array(1,2)-round(sticker_array(1,3)/2)-15:sticker_array(1,2)+round(sticker_array(1,3)/2)+15             if img2(i, j, 1) &gt; 240 &amp;&amp; img2(i, j, 2) &lt; 200 &amp;&amp; img2(i, j, 3) &lt; 200                 on = 1;             end         end     end end  if count == 4     for j = sticker_array(1,1)+sticker_array(1,3)+30:sticker_array(2,1)-15         for i = sticker_array(1,2)-round(sticker_array(1,3)/2)-15:sticker_array(1,2)+round(sticker_array(1,3)/2)+15             if img2(i, j, 1) &gt; 130 &amp;&amp; img2(i, j, 2) &lt; 110 &amp;&amp; img2(i, j, 3) &lt; 110 </pre>	<p>그리고 이어서 나머지 sticker도 검출한다. 아래의 부분들은 모두 비슷한 방법이다. count가 늘어나는 것을 볼 수 있다. 그리고 on이 1으로 증가하는데, 이는 세탁기가 켜졌는지를 알려주는 variable이다.</p> <p>( 이하 중략 )</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------

<pre> on = 1; end end end  for j = sticker_array(3,1)+sticker_array(1,3)+30:sticker_array(4,1)- 15 for i = sticker_array(3,2)- round(sticker_array(1,3)/2)- 15:sticker_array(3,2)+round(sticker_array(1,3)/2)+15 if img2(i, j, 1) &gt; 130 &amp;&amp; img2(i, j, 2) &lt; 110 &amp;&amp; img2(i, j, 3) &lt; 110 on2 = 1; end end end  end  for i=sticker_array(1,2):sticker_array(1,2)+10 for j=sticker_array(1,1):sticker_array(1,1)+10 img(i,j,:) = 0; end end % for i=sticker_array(2,2):sticker_array(2,2)+10 for j=sticker_array(2,1):sticker_array(2,1)+10 img(i,j,:) = 0; end end  for i=sticker_array(3,2):sticker_array(3,2)+10 for j=sticker_array(3,1):sticker_array(3,1)+10 img(i,j,:) = 0; end end </pre>	<p>( 중 략 )</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------

<pre> for i=sticker_array(4,2):sticker_array(4,2)+10     for j=sticker_array(4,1):sticker_array(4,1)+10         img(i,j,:) = 0;     end end  figure(1); imshow(img) figure(2); imshow(img2)  laundry_count = count / 2;  if laundry_count == 1     status = sprintf('%d %d ', laundry_count, on); end  if laundry_count == 2     status = sprintf('%d %d %d ', laundry_count, on, on2); end  % send data ip = '128.199.185.239'; port = 4950; my_port = 33833;  udpA = udp(ip, port, 'LocalPort', my_port); fopen(udpA); fwrite(udpA, status); fclose(udpA); </pre>	<p>laundry count는 세탁기의 개수이다. 위에서 sticker의 개수를 count에서 세었다. 그런데 세탁기 1개 당 sticker가 2개이기 때문에 count/2 한 값이 곧 세탁기의 개수가 된다.</p> <p>이렇게 가공한 data는 udp 통신을 통해서 server로 보내져야 한다. 이 또한 matlab에서 간단하게 작성할 수 있다. ip address, port number 등을 정의하고 udp를 open, write, close 할 수 있다.</p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 2) Server

In order to satisfy the multimedia characteristic of data shared via network, I purchased a server which costs 5 dollars per month, which is located in Singapore. The server is based on Ubuntu and has IP address of 128.199.185.239. After connecting to the server using putty, I installed apache2.



(digital ocean)

```
login as: root
root@128.199.185.239's password:
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 3.13.0-85-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

System information as of Tue Jun 21 23:29:25 EDT 2016

System load:  0.0               Processes:    83
Usage of /:   12.7% of 19.56GB  Users logged in:  1
Memory usage: 35%              IP address for eth0: 128.199.185.239
Swap usage:   0%

Graph this data and manage this system at:
https://landscape.canonical.com/

18 packages can be updated.
13 updates are security updates.

*** System restart required ***
Last login: Wed Jun  8 21:18:24 2016 from 192.249.21.87
root@ubuntu-512mb-sgp1-01:~#
```

(Connecting to server using putty)

I made our website using python based programming language called Flask. I configured the Flask by making FlaskApp.conf file. In this file I included the IP adress, wsgi directory, and others.



FlaskApp.conf:

```
<VirtualHost *:80>

    ServerName 128.199.185.239

    ServerAdmin youemail@email.com

    WSGIScriptAlias / /var/www/FlaskApp/flaskapp.wsgi

    <Directory /var/www/FlaskApp/FlaskApp/>

        Order allow,deny

        Allow from all

    </Directory>

    Alias /static /var/www/FlaskApp/FlaskApp/static

    <Directory /var/www/FlaskApp/FlaskApp/static/>

        Order allow,deny

        Allow from all

    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log

    LogLevel warn

    CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
```

Basically what this file does is that, when you enter the servername, 128.199.185.239, on browser, you get the website that I am going to make with Flask program.

After, I made `__init__.py`, which is the program that controls the website. The source code for `__init__.py`:

```
from flask import Flask, render_template
```

```
app = Flask(__name__)
```

```
@app.route('/')
```

```
def homepage():
```

```
with app.open_resource('count.txt') as f:  
    laundry_count = f.read()
```

```
with app.open_resource('ava.txt') as f2:  
    ava = f2.read()
```

```
with app.open_resource('left.txt') as f3:  
    left = f3.read()
```

```
with app.open_resource('right.txt') as f4:  
    right = f4.read()
```

```
return render_template('index.html', laundry_count = laundry_count, num = ava, left_stat = left,  
right_stat = right)
```

```
@app.route('/index')
```

```
def index():
```

```
with app.open_resource('count.txt') as f:  
    laundry_count = f.read()
```

```
with app.open_resource('ava.txt') as f2:  
    ava = f2.read()
```

```
with app.open_resource('left.txt') as f3:  
    left = f3.read()
```

```
with app.open_resource('right.txt') as f4:  
    right = f4.read()
```

```
return render_template('index.html', laundry_count = laundry_count, num = ava, left_stat = left,  
right_stat = right)
```

```
@app.route('/news')
```

```
def news():
```

```
    return render_template('news.html')
```

```
    return('hello')
```

```
@app.route('/contact')
```

```
def contact():
```

```
return render_template('contact.html')
```

```
@app.route('/about')
```

```
def about():
```


```
    return render_template('about.html')
```

```
if __name__ == "__main__":
```

```
    app.run()
```

In this python code, @app.route() function deals with directories. So for example, if we type <http://128.199.185.239/index>, the code under @app.route('/index') is executed. When <http://128.199.185.239/news> is typed, the code under @app.route('/news') is executed, and so on. So when <http://128.199.185.239/news> is typed, the function render\_template('news.html') is executed. The function render\_template renders the html file and displays on the website. As a result, when <http://128.199.185.239/news> is typed on a browser's link, the file called news.html will be displayed on the website. I have four html files for our website: index.html, news.html, contact.html, and about.html. The four html files are shown below:





Home	News	Contact	About
------	------	---------	-------



# Real Time Laundry

---

**Right now, out of 2 washing machine(s)**

**0 laundry machine(s) is/are available**

Time: 12:20:01

[Refresh](#)

---

<EE474 Introduction to Multimedia>

**Project Name:** Real Time Laundry

**Project description:** Real Time Laundry saves precious time of people by providing the information about washing machine availability to users before they

<http://128.199.185.239/index> (index.html)



# Real Time Laundry

**This page is under construction**

<EE474 Introduction to Multimedia>

**Project Name:** Real Time Laundry

**Project description:** Real Time Laundry saves precious time of people by providing the information about washing machine availability to users before they head to the laundry place

<Group 1>

JeongYeul Cho Jaun Yang Kijung Ryu Wootae Song



School of  
Electrical Engineering

(<http://128.199.185.239/news>) (news.html)



# Real Time Laundry

## Contact Information

If you have any questions or suggestions please email me at

[alansong@kaist.ac.kr](mailto:alansong@kaist.ac.kr)

<EE474 Introduction to Multimedia>

**Project Name:** Real Time Laundry

**Project description:** Real Time Laundry saves precious time of people by providing the information about washing machine availability to users before they head to the laundry place

<Group 1>

JeongYeul Cho Jaun Yang Kijung Ryu Wootae Song



School of  
Electrical Engineering

(<http://128.199.185.239/contact>) (contact.html)



# Real Time Laundry

## Group 1 Members



Name: JeongYeul Cho  
Role: Presenter, Designer



Name: Jaun Yang  
Role: Report, Powerpoint, Research



Name: Kijung Ryu  
Role: Image processing, Research



Name: Wootae Song  
Role: Server, Image processing, Webpage coding

### <EE474 Introduction to Multimedia>

**Project Name:** Real Time Laundry

**Project description:** Real Time Laundry saves precious time of people by providing the information about washing machine availability to users before they head to the laundry place

### <Group 1>

JeongYeul Cho Jaun Yang Kijung Ryu Wootae Song

(<http://128.199.185.239/about>) (about.html)

Index.html is the main page that displays the availability of laundries and the current time. News.html is the page that displays news related to Real Time Laundry. Unfortunately, there is not news about it yet, so this page is under construction. Contact.html is the page that displays the contact information so that clients can contact to administrator if they have any problems or questions. About.html displays the group members and what they did in this project. Obviously, there are many designs and images in each pages. This is done by combination of html and CSS. The theme.css file is the CSS style sheet that decorates our website and the source code is shown below:

```
p#status {
```

```
}
```

```
button {
  border: solid 1px somecolor;
  color: #eee;
  border-radius:5px;
  -moz-border-radius:5px;
  background: -moz-linear-gradient(top, #5E5E5E 0%, #474747 51%, #0a0e0a 51%, #0a0809 100%);
  background: -webkit-gradient(linear, left top, left bottom, color-stop(0%,#5E5E5E), color-
stop(51%,#474747), color-stop(51%,#0a0e0a), color-stop(100%,#0a0809));
}
button:hover {
  color: white;
}

h2#main {
  font-size: 165%;
}

div.first {
  height: 30px;
}

img#laundry {
  height: 100px;
  width: 100px;
}

img#on {
  height:50px;
  width: 50px;
}

img#off {
  height:50px;
  width: 50px;
}

img#icon {
  height: 50px;
  width: 50px;
}
```

```
img#kaist {  
    width: 318px;  
    height: 46px;  
}
```

```
img#profile {  
    float:left;  
    width: 110px;  
    height: 110px;  
}
```

```
span#blue {  
    font-weight: bold;  
}
```

```
h1#firstHeader {  
    font-family: 'Malgun Gothic';  
    font-size: 250%;  
}
```

```
h1#about {  
    font-family: 'Malgun Gothic';  
    font-size: 150%;  
}
```

```
ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
    overflow: hidden;  
    background-color: #333;  
    width: 100%;  
    position: fixed;  
    top: 0;  
    width: 100%  
}
```

```
li {  
    text-align: center;
```

```
padding: 0% 1%;
border-right: 1px solid #bbb;
display: inline;
float: left;
}
```

```
li:last-child {
border-bottom: none;
}
```

```
li a {
display: block;
color: white;
text-align: center;
padding: 14px 16px;
text-decoration: none;
}
```

```
li a:hover {
background-color: #111;
}
```

```
.active {
background-color: #4CAF50;
color: white;
}
```

This file assigns text font, text color, text size, image size, back ground color, and so on.

As you can see in our webpage, news.html, contact.html, and about.html are static that is their contents do not change. However, the index.html changes constantly whenever there is an input. The purpose of index.html is to show the availability of laundry machines at the moment. Therefore, index.html should be able to display how many laundry machines are available at the moment dynamically. Html files are static and they cannot display webpage dynamically. Therefore, I added Javascript to index.html to display the webpage dynamically. The source code of index.html is shown below:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
```



```

<title> Real Time Laundry </title>
<link rel="icon" href= {{ url_for('static', filename='img/icon.png') }} >
<link rel="stylesheet" type="text/css" href=" {{ url_for('static', filename='css/theme.css') }} ">

<!-- timer -->
<script>
function startTime() {
    var today = new Date();
    var h = today.getHours();
    var m = today.getMinutes();
    var s = today.getSeconds();
    m = checkTime(m);
    s = checkTime(s);
    document.getElementById('txt').innerHTML =
    h + ":" + m + ":" + s;
    var t = setTimeout(startTime, 500);
}
function checkTime(i) {
    if (i < 10) {i = "0" + i}; // add zero in front of numbers < 10
    return i;
}
</script>

</head>
<body onload="startTime()">

<ul>
    <li><a href="/index">Home</a></li>
    <li><a href="/news">News</a></li>
    <li><a href="/contact">Contact</a></li>
    <li><a href="/about">About</a></li>
</ul>

<div class="first"> </div>

<center> <h1 id="firstHeader"> <img id="icon" src= "{{ url_for('static', filename='img/icon.png') }}" >
Real Time Laundry </h1> </center>
<hr>

```

```
<h2 id="main"> <center> Right now, out of {{ laundry_count }} washing machine(s) </center> </h2>
```

```
<br><br><br><br><br><br><br><br>
```

```
<center> <p id="status"> </p> </center>
```

```
<script>
```

```
var count = '{{ laundry_count }}';
```

```
var left = '{{ left_stat }}';
```

```
var right = '{{ right_stat }}';
```

```
var img;
```

```
if (count == "1") {
```

```
    if (left == "1") {
```

```
        img = "<img id='laund' src= '{{ url_for('static', filename='img/icon.png') }} ' "> <br>
```

```
<img id='on' src= '{{ url_for('static', filename='img/on.png') }} ' ">";
```

```
    } else {
```

```
        img = "<img id='laund' src= '{{ url_for('static', filename='img/icon.png') }} ' "> <br>
```

```
<img id='off' src= '{{ url_for('static', filename='img/off.png') }} ' ">";
```

```
    }
```

```
}
```

```
if (count == "2") {
```

```
    if (left == "0" && right == "1") {
```

```
        img = "<img id='laund' src= '{{ url_for('static', filename='img/icon.png') }} ' ">
```

```
&nbsp;<img id='laund' src= '{{ url_for('static', filename='img/icon.png') }} ' "> <br> <img id='off' src= '{{ url_for('static', filename='img/off.png') }} ' "> &nbsp;&nbsp;&nbsp;<img id='on' src= '{{ url_for('static', filename='img/on.png') }} ' ">";
```

```
    } else if (left == "1" && right == "0") {
```

```
        img = "<img id='laund' src= '{{ url_for('static', filename='img/icon.png') }} ' ">
```

```
&nbsp;<img id='laund' src= '{{ url_for('static', filename='img/icon.png') }} ' "> <br> <img id='on' src= '{{ url_for('static', filename='img/on.png') }} ' "> &nbsp;&nbsp;&nbsp;<img id='off' src= '{{ url_for('static', filename='img/off.png') }} ' ">";
```

```
    } else if (left == "1" && right == "1") {
```

```
        img = "<img id='laund' src= '{{ url_for('static', filename='img/icon.png') }} ' ">
```

```
&nbsp;<img id='laund' src= '{{ url_for('static', filename='img/icon.png') }} ' "> <br> <img id='on' src= '{{ url_for('static', filename='img/on.png') }} ' "> &nbsp;&nbsp;&nbsp;<img id='on' src= '{{ url_for('static', filename='img/on.png') }} ' ">";
```

```
    } else {
```

```

        img = "<img id=W\"laundW\" src= W\"{{ url_for('static', filename='img/icon.png') }} W\">
&nbsp; <img id=W\"laundW\" src= W\"{{ url_for('static', filename='img/icon.png') }} W\"> <br> <img id=W\"offW\"
src= W\"{{ url_for('static', filename='img/off.png') }} W\"> &nbsp; &nbsp; &nbsp; <img id=W\"offW\" src=
W\"{{ url_for('static', filename='img/off.png') }} W\">";
    }
}

```

```
document.getElementById('status').innerHTML = img;
```

```
</script>
```

```
<br><br><br><br><br><br><br><br>
```

```
<h2 id="main"> <center> {{ num }} laundry machine(s) is/are available </center> </h2>
```

```
<div align="center"> <h2 style="font-size:125%"> Time: <span align="center" id="txt"> </span>
</h2> </div>
```

```
<br>
```

```
<center>
```

```
<form method="get" action="/index">
```

```
    <button type="submit"> Refresh </button>
```

```
</form>
```

```
</center>
```

```
<br>
```

```
<hr>
```

```

<p id="end"> <span id="blue"> &lt;EE474 Introduction to Multimedia&gt; </span> <br> <br> <span
style="font-weight:bold"> Project Name</span>: Real Time Laundry <br> <span style="font-
weight:bold">Project description</span>: Real Time Laundry saves precious time of people by
providing
the information about washing machine availability to users before they head to the laundry place
<br> <span id="blue"> <br> &lt;Group 1&gt; </span> <br> JeongYeul Cho Jaun Yang Kijung Ryu
Wootae Song </p>

```

```
<br> <br>
```

```
<img id="kaist" src= "{{ url_for('static', filename='img/kaist.png') }}">
```

```
</body>  
</html>
```

There are two things that are changing dynamically in index.html: timer that displays the current time and the information about laundry availability. I used open source for timer but made own Javascript code for the information about laundry availability, which is the most important thing in our webpage.

Right now, out of 2 washing machine(s)



0 laundry machine(s) is/are available

Time: 15:30:33

(the dynamic part in the webpage)

The Javascript code for the information about laundry availability is enclosed in `<script>` tag. There are three variables in the Javascript code: `var count`, `left`, and `right`. This variable is provided by the `__init__.py` code. If you look at `__init__.py` under `@app.route('/index')`, it uses the function `app.open_reoursce()` to open text files. After opening the text files, it saves the contents of the text file in variables `laundry_count`, `ava`, `left`, and `right`. Then, when it uses the `render_template` function to display the webpage, it also gives the html file the variables that it read with the `app.open_resource` as parameters. As a result, `index.html` gets the variables from `__init__.py`. After, obtaining the variables from `__init__.py`, it uses `if` statements to evaluate the variables. It first checks how many laundry machines there are with `count` variable. If `count` is 1, it displays one laundry icon; if `count` is two, it displays two laundry icons. After, it checks `left` and `right` variables. If the `count` is 1, it only checks for `left` variable; if the `count` is 2, it checks both `left` and `right` variables. `Left` and `right` variables contain information about whether a laundry machine is running or not. If `left` variable is 1, it indicates that left laundry machine is on; if the `right` variable is 0, it indicates the right laundry machine is off. According to the on/off information, it displays red circle below the corresponding machine if the machine is on and using; it displays green circle below the corresponding machine if the machine is off and available. This information is saved in variable `img` and displayed in the middle of webpage. This Javascript runs every time a client access the `index.html`. As a result, to get a new information, a client need refresh the page. Every other codes in the `index.html` and codes in `news.html`, `contact.html`, and `about.html` are straight forward. There are just text and images.

To summarize, `index.html` displays the information about the availability of laundry machines by getting

variables from \_\_init\_\_.py file, which gets the variables from text files. How are the text files are formed then? There is a program, listener.c that runs continuously in the server. This program is a UDP server program that receives the information about laundry information from MATLAB. The source code for listener.c is shown below:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>

#define MYPORT "4950"    // the port users will be connecting to

#define MAXBUFLEN 100

FILE *status;
FILE *status2;
FILE *status3;
FILE *status4;

// get sockaddr, IPv4 or IPv6:
void *get_in_addr(struct sockaddr *sa)
{
    if (sa->sa_family == AF_INET) {
        return &(((struct sockaddr_in*)sa)->sin_addr);
    }

    return &(((struct sockaddr_in6*)sa)->sin6_addr);
}

int main(void)
{
    int sockfd;
    struct addrinfo hints, *servinfo, *p;
    int rv;
```

```

int numbytes;
struct sockaddr_storage their_addr;
char buf[MAXBUFLEN];
socklen_t addr_len;
char s[INET6_ADDRSTRLEN];

while(1) {
    memset(&hints, 0, sizeof hints);
    hints.ai_family = AF_UNSPEC; // set to AF_INET to force IPv4
    hints.ai_socktype = SOCK_DGRAM;
    hints.ai_flags = AI_PASSIVE; // use my IP

    if ((rv = getaddrinfo(NULL, MYPORT, &hints, &servinfo)) != 0) {
        fprintf(stderr, "getaddrinfo: %s\n", gai_strerror(rv));
        return 1;
    }

    // loop through all the results and bind to the first we can
    for(p = servinfo; p != NULL; p = p->ai_next) {
        if ((sockfd = socket(p->ai_family, p->ai_socktype,
            p->ai_protocol)) == -1) {
            perror("listener: socket");
            continue;
        }

        if (bind(sockfd, p->ai_addr, p->ai_addrlen) == -1) {
            close(sockfd);
            perror("listener: bind");
            continue;
        }

        break;
    }

    if (p == NULL) {
        fprintf(stderr, "listener: failed to bind socket\n");
        return 2;
    }

    freeaddrinfo(servinfo);

```

```
printf("listener: waiting to recvfrom...\n");
```

```
addr_len = sizeof their_addr;
```

```
if ((numbytes = recvfrom(sockfd, buf, MAXBUFLen-1, 0,  
    (struct sockaddr *)&their_addr, &addr_len)) == -1) {  
    perror("recvfrom");  
    exit(1);  
}
```

```
printf("listener: got packet from %s\n",
```

```
    inet_ntop(their_addr.ss_family,  
        get_in_addr((struct sockaddr *)&their_addr),  
        s, sizeof s));
```

```
printf("listener: packet is %d bytes long\n", numbytes);
```

```
buf[numbytes] = '\0';
```

```
printf("listener: packet contains %s\n", buf);
```

```
char *count = malloc(sizeof(char));
```

```
char *left = malloc(sizeof(char));
```

```
char *right = malloc(sizeof(char));
```

```
count = strtok(buf, " ");
```

```
if (strcmp(count, "1") == 0) {  
    left = strtok(NULL, " ");  
} else if (strcmp(count, "2") == 0) {  
    left = strtok(NULL, " ");  
    right = strtok(NULL, " ");  
}
```

```
status = fopen("count.txt", "w");
```

```
if (status == NULL) {  
    fprintf(stderr, "Could not open status.txt\n");  
}
```

```
memset(status, 0, sizeof status);
```

```
fwrite(count, 1, 1, status);
```

```
status2 = fopen("left.txt", "w");  
if (status2 == NULL) {  
    fprintf(stderr, "Could not open status2.txt\n");  
}  
memset(status2, 0, sizeof status2);  
fwrite(left, 1, 1, status2);
```

```
if (strcmp(count, "2") == 0) {  
    status3 = fopen("right.txt", "w");  
    if (status3 == NULL) {  
        fprintf(stderr, "Could not open status3.txt\n");  
    }  
    memset(status3, 0, sizeof status3);  
    fwrite(right, 1, 1, status3);  
}
```

```
char ava[1];
```

```
if (strcmp(count, "1") == 0) {  
    if (strcmp(left, "1") == 0) {  
        ava[0] = '0';  
    } else {  
        ava[0] = '1';  
    }  
} else {  
    if (strcmp(left, "1") == 0 && strcmp(right, "0") == 0) {  
        ava[0] = '1';  
    } else if (strcmp(left, "0") == 0 && strcmp(right, "1") == 0) {  
        ava[0] = '1';  
    } else if (strcmp(left, "1") == 0 && strcmp(right, "1") == 0) {  
        ava[0] = '0';  
    } else {  
        ava[0] = '2';  
    }  
}
```



```

    }
}

status4 = fopen("ava.txt", "w");
if (status4 == NULL) {
    fprintf(stderr, "Could not open status4.txt\n");
}
memset(status4, 0, sizeof status4);
fwrite(ava, 1, 1, status4);

fclose(status);
fclose(status2);

if (strcmp(count, "2") == 0) {
    fclose(status3);
}
fclose(status4);

close(sockfd);
}

return 0;
}

```

This program creates a UDP socket and binds the port number of 4950 to it. As a result, MATLAB program can send the UDP packets to this server with destination IP address of 128.199.185.239 and port number 4950. After receiving the information about the availability of the laundry machines, it parses the data and saves the data into four textfiles: count.txt, left.txt, right.txt, and ava.txt. These are the text files that \_\_init.py\_\_ reads and sends to index.html.

To summarize, there is a UDP based server program called listener.c that is running all the time. This program receives the information about laundry availability from MATLAB. After, it parses the information and saves the information in four different text files. Then, the \_\_init\_\_.py Flask program reads these text file and sends

the information to index.html file. In index.html, it receives the information and uses Javascript to dynamically display the information about laundry availability.

The hierarchy of the server program is shown below:

static

css

theme.css

img

icon.png

kaist.png

off.png

on.png

profile.png

templates

about.html

contact.html

index.html

news.html

\_\_init\_\_.py

listener.c

## 5. Result

- 다음과 같이 4개의 사진에 대해서 code를 실행시킨 결과이다. 위에서부터 각각 원본, webpage display, listener의 모습이다. 4개의 사진은 각각 세탁기의 개수와 state가 다른 상태이다. 모두 잘 나오는 것을 확인할 수 있다.

example 1



example 2



Right now, out of 1 washing machine(s)



1 laundry machine(s) is/are available

Time: 15:52:54



Right now, out of 2 washing machine(s)



2 laundry machine(s) is/are available

Time: 15:55:19



```
root@ubuntu-512mb-sgpl-01:/var/www/FlaskApp/FlaskApp# ./listener
listener: waiting to recvfrom...
listener: got packet from 39.123.166.6
listener: packet is 4 bytes long
listener: packet contains "1 0 "
listener: waiting to recvfrom...
```

```
listener: got packet from 39.123.166.6
listener: packet is 6 bytes long
listener: packet contains "2 0 0 "
listener: waiting to recvfrom...
```

example 3



Right now, out of 1 washing machine(s)



0 laundry machine(s) is/are available

Time: 15:56:30

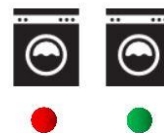


```
listener: got packet from 39.123.166.6  
listener: packet is 4 bytes long  
listener: packet contains "1 1 "  
listener: waiting to recvfrom...
```

example 4



Right now, out of 2 washing machine(s)



1 laundry machine(s) is/are available

Time: 15:57:14



```
listener: got packet from 39.123.166.6  
listener: packet is 6 bytes long  
listener: packet contains "2 1 0 "  
listener: waiting to recvfrom...
```

## 6. Discussion

- 실생활에 적용하기 위해서는 이 project를 단순히 확장하면 될 것이다. 세탁기가 최대 2개인 경우만 했지만, 동일 알고리즘을 적용시켜서 n개의 세탁기에 대해서 정상적으로 실행될 수 있도록 할 수 있다. 실제 고해상도의 webcam이 찍는 사진에는 2개 이상의 세탁기가 들어갈 수 있기 때문이다. 그리고 세탁실에는 1개나 2개의 세탁기만 있는 것이 아니다. 다수의 세탁기가 존재하기 때문에 한 세탁실에 대해서 여러 사진이 들어올 것이다. 이 여러 사진에서 분석한 정보를 합해서 가능한 세탁기가 몇 대나 되는지를 계산하는 것도 수정되어야 한다.
- 우리는 고가의 webcam을 구입하지 못해서 이미 찍혀있는 사진에 대해서 알고리즘을 적용하고 web display를 했다. 실제로는 webcam을 구입해서 적절한 위치에 달아두어야 세탁기를 분석하기 좋게 찍어줄 것이다. 이렇게 촬영된 사진을 전송하는 programming 또한 필요할 것으로 보인다. 이 외에는 general case에 적용시키더라도 수정할 부분이 크게 생기지 않으므로 우리 모두는 real time laundry의 핵심적인 알고리즘은 모두 구성했다고 평가하고 있다.

## ◎ References

1. detectMSERFeatures() : <http://kr.mathworks.com/help/vision/ref/detectmserfeatures.html>
2. Laplacian Filter : <http://kr.mathworks.com/help/images/ref/fspecial.html>
3. Laplacian Filter : <http://carstart.tistory.com/83>
4. LOG Filter : <http://kr.mathworks.com/help/images/ref/fspecial.html>