

영상처리 기반 건축 구조물의 동적 변위 계측 및 증강현실을 통한 변위 분포 시각화 시스템에 대한 프로토타입 개발

Development of Real-Time Dynamic Displacement Measurement and AR-visualization
Prototype of Construction Structures Using Image Processing Techniques

백승현* 백찬우** 송기수*** 인성우**** 최성훈*****

This prototype can measure dynamic displacement of a structure using image process and visualize load distribution with augmented reality. It improves the limited range of displacement measurement with image process based on markers and provides more intuitive visualization of load distribution on entire structure using augmented reality.

키워드 : 영상처리, 증강현실, 변위 계측, 시각화

1 서론

1.1 연구의 필요성 및 목적

최근 건축공학 기술이 발달함에 따라 신축 건축물들의 크기와 형상이 다양해지고 있다. 특히 비정형 건축물은 종래의 건축물에 비해 형상이 매우 복잡하여, 구조적 안전성을 파악하기가 쉽지 않다. 건축물의 구조적 안전 여부를 파악하기 위해서는 건축 구조를 이루고 있는 부재에 대한 변위 계측 실험이 필수적이다. 따라서 최근에는 이전보다 다양한 변위 계측시스템이 필요한 실정이다.

종래의 변위 계측시스템은 크게 접촉식 센서(Sensor)와 비접촉식 센서가 있다. 변형률 게이지(Strain gauge) 및 LVDT(Linear Variable Differential Transformer)는 대표적인 접촉식 센서로 구조물에 직접 접촉시켜 변위를 측정하기 때문에 정확도가 높아 가장 많이 사용되고 있다. 이러한 접촉식 센서는 크게 3가지의 문제점이 있다. **첫째로 접촉식 센서를 사용할 경우 센서를 부착한 위치 외의 변위를 알 수 없다.** 이 때문에 실험 전에 구조 해석 프로그램으로 실험체를 모델링(Modeling) 하여 가장 큰 변위가 발생할 것으로 예상 되는 부분에 센서를 부착하게 된다. 실험체의 최대 변위를 예측하는 과정에서 오차가 발생할 경우 의미 없는 계측값을 얻을 수 도 있다. **둘째, 측정 범위가 증가할수록 추가적으로 센서를 구입해야 하기 때문에 많은 비용이 발생한다.** LVDT의 경우 개당 가격이 높기 때문에 상당한 추가 비용이 발생할 수 있다. **셋째,**

접촉식 센서는 측정 방향성이 존재하기 때문에 불량 부착 시 측정 오차가 발생할 수 있다. 또 이때문에 실험 준비기간이 많이 소요되며 센서 고정을 위한 추가 설비가 필요하다. 이러한 접촉식 센서의 문제점을 개선하기 위해 비접촉식 레이저 센서인 LDV(Laser Doppler Vibrometry)가 개발되었으나 매우 고가이기 때문에 일반적인 실험환경에서 적용하기에는 한계가 있다.

최근에는 위와 같은 종래 변위계측 시스템의 단점들을 보완하기 위해 영상처리를 이용한 변위 계측 시스템이 개발되었다. 일반적인 영상처리를 이용한 변위계측 시스템은, 실험체에 마커(Marker)를 부착하고 영상내에서 마커를 추적하여 구조물의 변위를 측정하는 방식이다. 마커는 측정 방향성이 없기 때문에 비교적 부착이 용이하며, 불량 부착으로 인한 측정 오차가 발생할 확률이 적다. 또 센서와는 달리 카메라는 반 영구적인 장비로 추가 장비 구입에 따른 비용이 발생하지 않는다.

하지만 일반적인 영상처리 변위계측 시스템으로는 앞서 기술한 측정 범위가 제한적이라는 단점은 보완할 수 없다. 센서를 이용한 계측 방식과 마찬가지로 일반적인 영상처리 변위 계측 시스템은 마커를 부착한 부분 외의 변위 분포는 계측 할 수 없기 때문에 최대 변위 지점 예측과정에서 오차가 발생할 경우 실험이 실패할 가능성이 있다. 일반적으로 건축 공학 분야에서의 실험은 규모가 크고, 주로 실험체를 가력 하여 훼손시키는 경우가 많다. 따라서 실험이 실패할 경우 새로 실험체를 제작해야 하기 때문에 큰 경제적 손실이 발생한다.

본 연구는 건축 구조물의 실시간 변위(Displacement)를 영상 처리를 통해 측정하여, 부

* wonca14@ajou.ac.kr 건축공학과 ICT 융합전공 4학년

** qorcksdn2@gmail.com 건축공학과 4학년

*** thdrln92@ajou.ac.kr 건축공학과 ICT 융합전공 4학년

**** nerizell@ajou.ac.kr 건축공학과 4학년

***** chjis1118ajou.ac.kr 건축공학과 4학년

제의 요소(Element)별 변위 분포를 증강현실(Augmented Reality, AR)을 통해 출력 하는 시스템 개발을 목표로 한다. 본 연구를 통해 개발된 프로토타입(Prototype)의 기능은 다음과 같다.

- 1) 영상처리기법으로 구조물의 동적 변위를 측정
- 2) 측정된 변위를 기반으로 부재의 요소(element)별 변위(displacement) 산출
- 3) 증강현실(Augmented Reality, AR)을 통하여 GUI(Graphic User Interface)에 변위 분포를 실시간으로 출력

본 연구의 결과물은 영상처리 계측 방식을 사용함으로써 기존 센서의 단점을 보완하고, 마커를 부착한 부분 뿐만 아니라 전체 실험체의 변위 분포를 출력하기 때문에 측정 범위가 제한적이지 않다는 장점이 있다. 이에 따라 예상되는 기대효과는 1) 센서 구입으로 인한 비용 절감 2) 센서 부착으로 인한 실험 준비 기간 감소 3) 불량 부착으로 인한 측정오차 발생 방지 4) 실험 실패에 따른 비용 절감 등이 있다.

1.2 기술 동향

최근 영상처리 기술이 발달함에 따라 많은 연구기관에서 영상처리를 이용한 변위 계측 시스템을 연구해왔다. 이중재 외 1(2014)은 다수의 카메라를 이용하여 구조물의 3차원 변위 및 회전각 측정에 대한 연구를 진행했다. 이는 무선 라우터를 이용한 환경에서 한번에 다양한 위치에서 변위계측이 가능한 시스템으로, 다수의 카메라로 영상을 측정후 측정 정보를 조합하여 3차원 계측을 실시하는 방식이다. 특별한 데이터 처리 방식이 없이 다수의 카메라만 이용하면 적용할 수 있는 방안이기 때문에 이후 연구에서 적용할 수 있을 것으로 보인다. 강일민 외 2(2017)는 영상처리 기법으로 변위를 측정하고, 실시간 모니터링을 통해 구조물의 붕괴를 감지하는 시스템을 제안했다. 이는 구조물의 변위 발생시 수직 수평방향의 변위 가속도를 기반으로 위험도를 판단하는 것으로 이후 본 연구에서도 충분히 적용 가능하다고 판단된다.

영상처리 변위 계측 시스템의 인식 정확도 향상 방안면에서도 많은 연구가 진행되고 있다. 한정현 외 3(2012)은 가속도와 변위 센서를 융합하여 정밀한 동적 변위 및 속도를 산정하는 칼만필터 알고리즘을 연구했다. 손기성 외 3(2013)은 카메라 해상도의 한계로 인한 측정 오차 발생을 해결하기 위해 윤곽선 성분의 확률 분포 기대값 산출을 통해 실수형 윤곽선을 검출했다. 이를 통해 측정 오차를 0.002mm 이하로 줄였다. 하지만 이는 영상의 주파수를 이용하여 영상처리를 하는 방식으로 구현 난이도 측면에서 본 연구에는 적용하기는 어려워 보인다. 인식 정확도 측정을 위해서는 추후 다른 시스템을 참고해야 할 것이다.

기존의 영상처리기법은 마커를 부착한 부분만 변위계측이 가능하므로 계측 범위가 제한적이다. 본 연구는 실시간 영상처리를 통해 구조물의 동적 변위를 측정하고, 이를 기반으로 마커가 부착된 위치

외에도 부재 요소별 변위를 산출하여 증강현실로 구조물의 변형을 시각화 한다는 점에서 기존 연구와 차별점이 있다.

2 본론

2.1 시스템 구성요소

프로토타입의 시스템은 카메라, 노트북으로 구성되어 있다. 그림 1 과 같이 실험체에 마커를 부착하고, 가력 영상을 카메라로 촬영한다. 촬영된 영상을 노트북에 전송하여, 영상 데이터를 처리한다. 이후 증강현실을 통해 변위 분포를 출력한다.

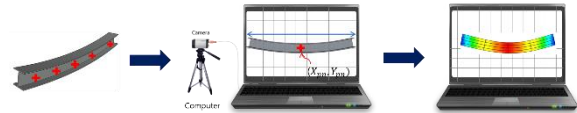


그림 1 시스템 구성도

2.2 개발 환경 및 제한 조건

개발 환경으로는 매트랩(MATLAB)을 사용했다. 매트랩은 주로 공학 수치해석에 사용되는 개발환경으로 코드가 직관적이고, 영상처리에 사용할 수 있는 다양한 라이브러리를 제공한다. 따라서 여러가지 영상 처리 알고리즘의 성능을 테스트하기에 적합하며, 프로토타입 개발기간을 단축할 수 있다. 하지만 다른 언어에 비해 반복문 처리속도가 느리기 때문에 실제 프로그램 개발시에는 C++을 사용할 예정이다. C++을 사용해서 개발하는 이유는 영상처리 오픈소스(Open Source)인 OpenCV가 대부분 C++로 구현이 되어 있고 MFC(Microsoft Foundation Class)를 사용하여 비교적 쉽게 GUI를 구현할 수 있기 때문이다. 영상처리는 약 60fps(Frame Per Second) 정도로 프레임을 받아와서 영상 데이터를 처리하기 때문에 일반적으로 고성능의 컴퓨터가 필요하다. 더욱이 본 시스템은 프레임당 영상 데이터를 받아서 많은 연산을 하기 때문에 다소 고성능의 노트북이 필요하다고 판단된다. 실제로 프로토타입을 개발하면서 i5-6500 CPU, 16G RAM, GTX 750Ti Graphic Card 성능의 데스크탑으로 테스트를 진행했는데, 프로그램 구동 시 화면이 끊기는 현상이 발생했다.

2.3 시스템 동작 시나리오 및 주요 알고리즘

시스템 주요 알고리즘은 크게 1) 마커 추적 과정, 2) 실시간 변위 산출 과정, 3) 변위 분포 출력 과정으로 이루어 진다. 그림 2는 위 3가지 과정을 포함한 시스템 전체 알고리즘이다.

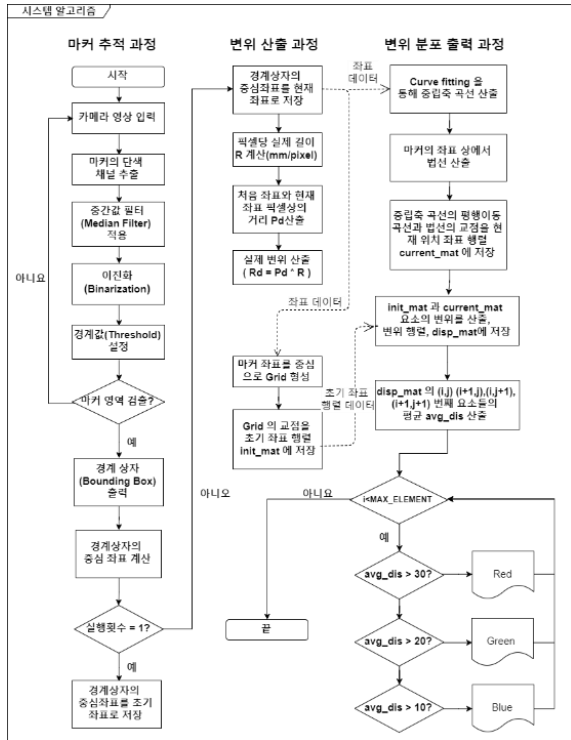


그림 2 시스템 알고리즘

2.3.1 마커 추적 단계

그림 3, 그림 4는 MATLAB을 이용하여 영상 내에서 마커를 추적하는 과정을 구현한 스크립트다. 마커를 추적하여 좌표를 산출하는 과정은 다음과 같다, 먼저 카메라로부터 받은 컬러 영상의 단색 채널을 추출한다. 이후 중앙값 필터(median filter)를 적용하여 잡음(noise)을 제거하면 단색 마커가 적당한 밝기값을 가진 회색조(gray scale) 영상을 얻을 수 있다. 적절한 픽셀 경계값(threshold)을 결정하고 이를 기준으로 영상을 이진화(binarization) 처리하면 마커 영역의 픽셀값은 0, 이외의 모든 영상의 픽셀값은 225가 되면서 영상 내에서 마커를 추적할 수 있다. 픽셀값이 0인 지역에 직사각형의 구속 상자(bounding box)를 형성하고, 구속 상자의 도심을 마커의 좌표로 저장한다. 영상의 프레임마다 위 과정을 반복하면서 좌표값을 저장하면 마커를 추적할 수 있다. 그림 5는 실험체에 붉은색 마커를 부착했을 때 각 영상 처리 단계를 거치면서 변화하는 영상을 나타낸다.

```
%% 마커 식별
data = getsnapshot(vide); % frame 받아오기
diff_im = imsubtract(data(:,:,1),rgb2gray(data)); % gray scale 영상 제거 1:red
diff_im = medfilt2(diff_im,[3,3]); % 중간값 필터
diff_im = im2bw(diff_im,0.18); % 흑백처리
diff_im = bwareaopen(diff_im,300); % 픽셀값 300 이하 0으로 처리 (필터)
bw = bwlabel(diff_im,8); % 행렬값 리턴 - 8 connected object
stats = regionprops(bw,'BoundingBox','Centroid'); % bounding box 출력
axes(handles.axes3);
imshow(data); % 데이터 출력
```

그림 3 마커 인식 과정 스크립트

```
for object = 1:length(data)
    bb = stat(object).BoundingBox;
    bc = stat(object).Centroid;
    rectangle('Position',bb,'LineStyle','r','LineWidth',2);
    plot(handles.axes3,bb(1),bb(2),'-r');
    a = text(bb(1),bb(2),1 ( num2str(round(bc(1))) ) , num2str(round(bc(2))) ); % Bounding Box의 Centroid 좌표 출력
    set(a, 'FontName', 'Arial', 'FontSize', 12, 'Color', 'yellow');
    marker(object).init_x = round(bc(1));
    marker(object).init_y = round(bc(2));
    x = num2str(marker(object).init_x);
    y = num2str(marker(object).init_y);
    str = sprintf('%d %d',x,y);
    set(handles.init_coord_txt,'String',str);
end
```

그림 4 마커 좌표 산출 및 저장 과정 스크립트

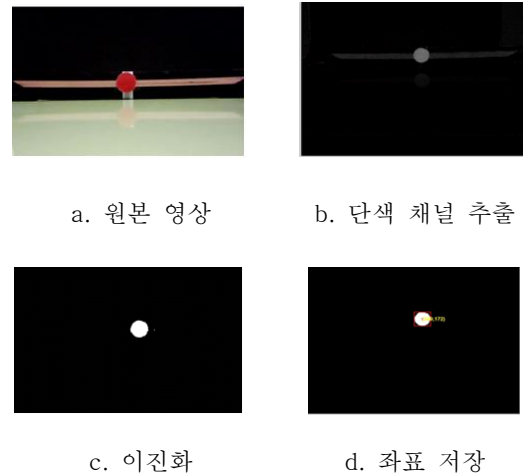


그림 5 색상 기반 마커 인식 영상처리 과정

2.3.2 실시간 변위 산출 단계

마커가 움직인 실제 변위는 픽셀상에서 마커가 움직인 거리와 단위 픽셀당 실제 거리를 곱하여 산출한다. 마커가 움직인 픽셀 거리를 산출하기 위해서 먼저 마커의 초기 좌표를 저장한다. 이후 영상 프레임마다 업데이트되는 마커의 좌표와 식 (1)을 통해 2차원 거리 연산을 하면 영상에서 마커가 움직인 픽셀 거리를 알 수 있다. 단위 픽셀당 실제 거리는 화면상에 표현된 실험체의 픽셀 길이와 이미 알고 있는 실험체의 길이의 비와 같다. 화면상에 표현된 실험체의 길이를 측정하기 위해 그림 8의 스크립트와 같이 화면상의 특정 지점을 클릭하고, 클릭한 지점 간의 길이를 저장하도록 프로그램을 설계했다. 마지막으로 식 (2)를 통해 마커가 움직인 픽셀 거리와 단위 픽셀당 실제 거리를 곱하면 마커의 실제 변위를 산출할 수 있다. 실시간 변위 측정 실험을 위해 그림 6과 같이 모눈종이에 50mm, 100mm, 150mm 지점에 표시를 하고 붉은색 마커를 표시 지점에 따라 이동 했다. 단위 픽셀당 실제 거리를 구하기 위해서는 150mm 길이의 선분을 표시 했다. 그림 7은 이를 기반으로 실험체의 변위 측정 시스템을 구현한 것이다. 실험 결과 실제 변위와의 오차는 약 0.016으로 만족할 만한 측정 정확도라고 판단된다.

$$pl = \sqrt{(x_0 - x)^2 + (y_0 - y)^2} \quad (1)$$

$$rl = r \times pl \quad (2)$$

(x_0, y_0) : 마커의 초기 좌표

(x, y) : 실시간으로 업데이트 되는 마커의 좌표

pl : 마커의 픽셀 이동거리

r : 픽셀 하나당 실제 거리 (mm/pixel)

rl : 마커의 실제 이동거리



그림 6 실시간 변위 측정 실험 환경

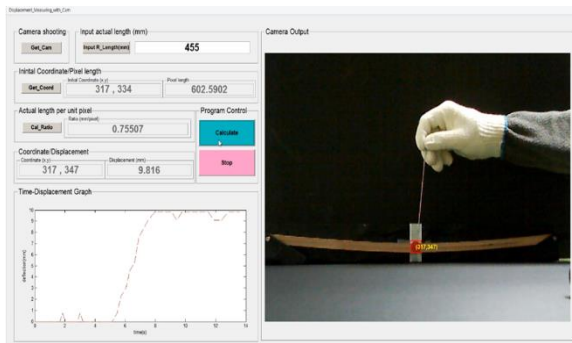


그림 7 실시간 변위 측정 실험

```

x_edge = [0:0];
y_edge = [0:0];
n = 0;
button = 1;

while n<2

    [xi, yi, button] = ginput(1, 'LineStyle', ':'); %마우스로 클릭한 좌표를 받는다.
    plot(xi, yi, 'ro'); %마우스로 클릭한 좌표에 o를 찍는다.
    n=n+1; %카운트 1증가 (카운트 = 2 일때 종료)
    x_edge(n,1)=xi;
    y_edge(n,1)=yi;

end

t = 1:n;
ts=1:0.1:n;
xs=spline(t,x_edge,ts);
ys=spline(t,y_edge,ts);

c1.x = x_edge(1,1);
c2.x = x_edge(2,1);
c1.y = y_edge(1,1);
c2.y = y_edge(2,1);

p_length = abs(marker(1).init_x-marker(5).init_x); % 보의 길이
p_height = abs(c2.y-c1.y); % 보의 높이
p_lang = num2str(p_length);
set(handles.P_length_text, 'String', p_lang);
    
```

그림 8 화면상에 표현된 실험체의 픽셀 길이 측정 과정 스크립트

2.3.3 변위 분포 출력 단계

그림 9는 변위 분포 출력을 위해 제작한 실험체다. 프로토타입 프로그램의 테스트를 위해 양단 지

점 조건(Support Condition)이 자유단(Pin)인 보에 붉은색 마커 5개를 설치했다, 변위분포 출력 과정은 다음과 같다. 먼저 그림 10과 같이 실험체를 붙인 마커 5개의 부착 위치를 중심으로 45개의 요소로 분절하고, 요소의 교점을 초기 좌표 행렬로 저장한다.

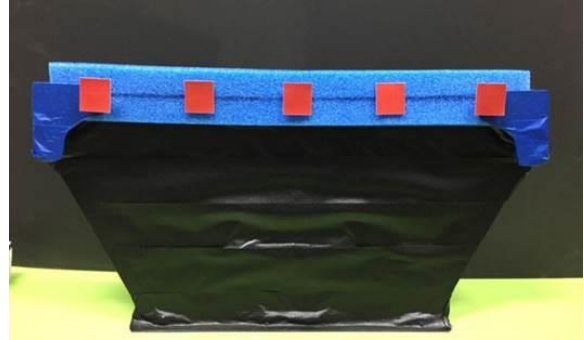


그림 9 실험체

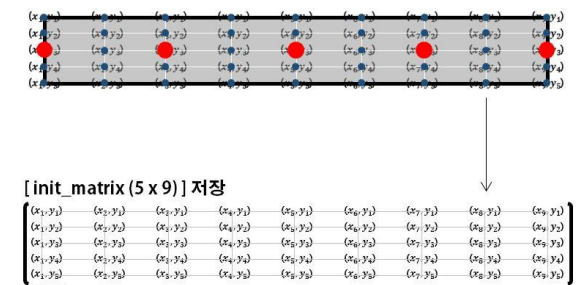


그림 10 초기좌표 행렬

실험체를 가력 하여 보의 형태가 변하게 되면 마커의 위치를 추적하고, 그림 11과같이 마커의 위치를 기반으로 커브 피팅(Curve fitting) 을 통해 구조체 중립축(Neutral Axis)의 함수를 산출한다. 프로토타입에서 커브 피팅은 4차함수를 기준으로 실행했다.

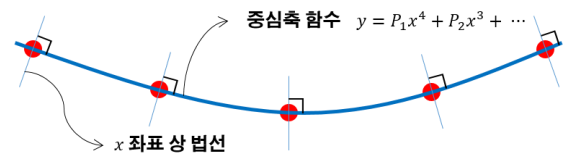


그림 11 중립축 함수 및 법선 산출과정

일반적인 보의 이론상 모든 수평축이 중립축과 동일하게 거동하고, 수직 단면은 평면을 유지하면서 중립축을 중심으로 회전한다. 이에 따라 그림 12와 같이 가력이 이후 변형된 실험체의 중립축을 평행 이동하고, 마커 좌표 상에서 법선의 함수를 산출한다. 법선과 평행이동한 곡선들의 교점을 실험체 가력 후 움직인 좌표라고 가정하고 현재 좌표 행렬로 저장한다.

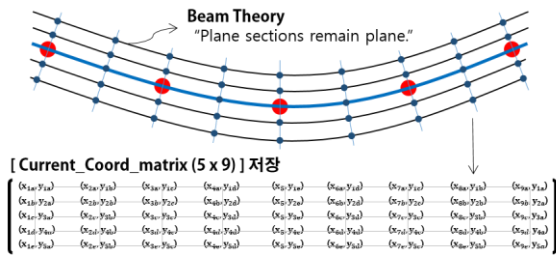


그림 12 현재 좌표 행렬

위의 과정을 거치면 처음 위치좌표 행렬 (init_matrix)와 실시간 위치 좌표 행렬 (Current_Coord_matrix)를 구할 수 있다. 그림 13과 같이 행렬 요소마다 유클리드 거리(Euclidean Distance)를 구하면 각 교점의 변위 행렬 (disp_matrix)을 구할 수 있다.

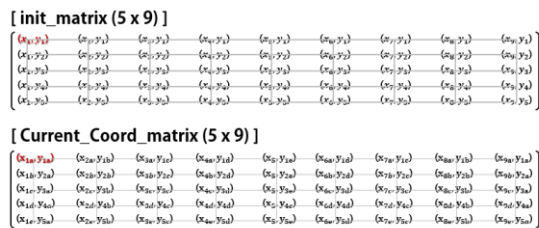


그림 13 변위 행렬

각 그리드의 평균 변위는 그림 14와 같이 그리드 상의 네 꼭지점 변위의 평균이라고 가정한다. 이와 같은 방식을 사용하면 분절한 요소의 수가 많아질 수록, 높은 계측 정확도를 확보할 수 있다.

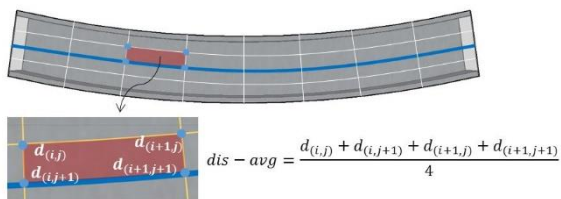


그림 14 요소별 평균 변위 산출

마지막으로 산출된 평균 변위에 따라 Grid의 색상을 출력한다. 본 프로토타입에서는 5개의 범위로 색상을 산출하도록 설계했다.

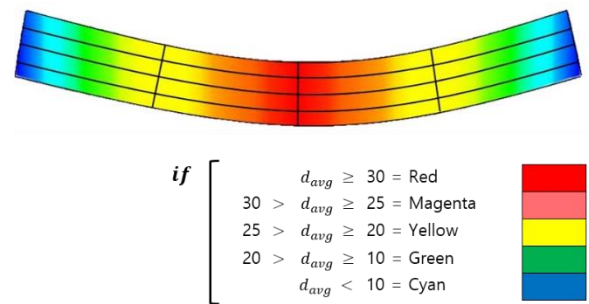


그림 15 평균변위별 색상 출력 개념도

그림 16은 위 과정을 통해 개발된 프로토타입의 GUI다. 그림에서 우하단에 보이는 그림이 커브 피팅을 통해 산출된 곡선이고 우상단에 보이는 그림은 평균 변위에 따라 증강현실로 출력된 변위 분포의 형태다. 실험결과와 변위 분포가 정상적으로 출력됨을 확인 할 수 있었다.

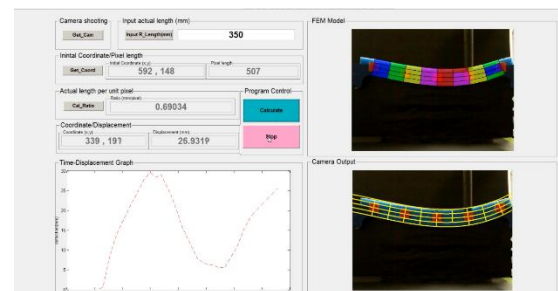


그림 16 변위분포 출력 실험

3 위험 요소 분석

본 연구의 위험요소는 1) 연산속도 2) 인식 정확도 3) 측정 정확도 4) 실험체 형태에 따른 데이터 처리 알고리즘으로 크게 4가지가 있다.

3.1 연산속도

매트랩을 사용하여 프로토타입을 구현해 본 결과 실시간 영상이 다소 끊기는 현상이 발생했다. 이는 카메라가 30fps로 프레임을 전송하는데 반해 프레임당 연산에 소요되는 시간이 많기 때문이다. 그림 18는 매트랩 환경에서 개발된 프로그램의 전체 프로세스를 50회 반복하고, 이에 따른 소요시간을 분석한 결과다. 분석결과 프로세스 1회당 소요시간은 최대 1.444초 최소 0.837초, 평균 0.921초로 프레임 처리 속도인 0.033초에 비해 약 27배 크다. 영상이 끊기지 않고 제대로 출력되기 위해서는 현재 소요시간의 약 3% 수준으로 속도를 향상시켜야 한다. 이는 매트랩의 느린 반복문 처리속도에 기인한다. 또한 프로그램 특성상 행렬에 데이터를 할당하기 위한 반복문의 수가 많고, 영상을 받아 올 때 큰 데이터를 처리 하기 때문에 처리속도가 느린 것 이라고 판단된다. 추후 본 시스템 개발 시 개발환경으로 C나 C++ 을 사용하고, 분산처리를 이용한다면 충분히 성능을 개선할 수 있을 것이라고 본다.

```

51 while(1)
52
53     tic %% 측정 시작
54
55     %% 마커 인식, 좌표 산출
56     TrackingMarker
57     %% 중심곡선 및 변위 산출
58     ShowFEMModel;
59     hold off
60     %% 변위 산출
61     CalDis;
62     %% 색상 출력
63     Color;
64     hold on
65
66     toc %% 측정 종료
67
68 end
69

```

그림 17 반복문 시간 측정 스크립트

반복 횟수(회)	측정시간(s)	MIN	MAX	AVG	GOAL
1	1.14114	0.837	1.444	0.921	0.033
2	1.44445				
3	0.849366	SUM	GOAL SUM	NEEDS(%)	
4	0.85281	46.062	0.833	3.618	
5	0.901797				
6	0.916385				
7	0.901505				
8	0.914104				
9	0.912182				
10	0.837407				
11	0.877286				
12	0.889038				
13	0.894805				
14	0.87185				
15	0.883364				
16	0.988964				
17	0.879307				
18	0.87018				
19	0.907907				
20	0.850861				
21	0.883064				
22	0.886182				

그림 18 반복문 소요시간 분석

3.2 인식 정확도

프로토타입 개발 시 색상 기반으로 마커를 인식하는 알고리즘을 사용했는데, 이는 조도가 높은 환경에서 잘 작동하지 않는 것으로 확인되었다. 추후 색상이 아닌 형태를 인식하는 특징점 추출(Feature Matching)알고리즘이나 비마커 방식인 윤곽선 검출(Edge detection)알고리즘을 사용한다면 이 부분을 개선할 수 있을 것으로 보인다. 현재 SIFT(Scale Invariant Feature Transform), SURF(Speeded Up Robust Features), Harris Corner 알고리즘 등 특징점을 찾아 물체를 인식하는 방식이 OpenCV를 통해 공개되어 있기 때문에 추후 무난하게 이를 개선할 수 있을 것으로 본다.

3.3 측정 정확도

건축공학 실험은 일반적으로 실험체의 미소한 변위(약 $10^{-3}mm$)를 측정하게 된다. 현재 개발된 프로토타입의 측정 오차는 약 1%미만으로 실험실에서 센서로 활용하기에는 측정오차가 크다. 프로토타입 테스트를 할 때 해상도가 640×480 인 다소 낮은 성능의 웹캠(WebCam)을 사용하여 실험했는데 추후 QHD (Quad HD/2K) 수준의 카메라를 이용한다면 이를 개선할 수 있을 것으로 보인다.

또 증강현실로 변위 분포를 출력할 때, 계측값이 아닌 건축 구조 이론을 기반으로 출력을 하기 때문에 실제 계측값과 상이한 변위 분포가 출력될 수 있다. 이는 윤곽선 검출 방식을 통해 구조체의 윤곽선을 인식하고, 선형(Linear) 마커를 사용하여 중심축을 실제로 계측하여 측정한다면 개선가능한 부분

이라고 판단된다.

3.4 실험체의 형태에 따른 데이터 처리 알고리즘

현재 개발된 프로토타입은 보(Beam)형태의 구조물에만 적용할 수 있다. 프레임(Frame)이나 막구조(Membrane Structure)등 다양한 형태의 구조물에 적용할 수 있는 알고리즘을 고안하기 위해서는 다양한 구조체의 거동에 대한 깊은 고찰을 필요로 한다. 혹은 유한 요소 해석 기법을 접목한다면 형태에 관계없이 유사한 알고리즘으로 비교적 정확하고 효율적으로 계측값을 얻을 수 있을 것이다.

4 결론

본 연구를 통해 마커 기반 영상처리 기법을 통해 실험체의 실시간 변위를 계측하고, 증강현실을 이용하여 구조물 전체의 변위 분포를 출력하는 시스템을 개발했다. 마커 기반 영상처리 기법을 통해 변위를 계측한 결과 실제 변위와의 오차는 약 1%미만으로, 실제 계측 센서로써 사용되기에는 다소 부족한 부분이 있다. 이를 통해 변위 분포를 출력해본 결과 비교적 정상적으로 출력되었다. 추후 변위 분포를 출력하는 알고리즘을 개선한다면 다양한 형태를 가진 구조물에도 적용할 수 있을 것으로 보인다.

5 참고문헌

- [1] 이종재 외 1 (2014),“영상기반 변위계측 기법”, 한국소음진동공학회 논문집, 24(4), pp. 18-24
- [2] 손기성 외 4 (2013),“카메라 영상을 이용한 진동변위 측정”, 한국소음진동공학회논문집, 23(9), pp. 789-796
- [3] 박성우 외 3 (2010),“모션 캡처를 이용한 구조물의 변위계측”, 대한건축학회 추계학술발표대회논문집, 30(1), pp. 117-118
- [4] 신진섭 외 1 (2013),“마커 인식과 회전 알고리즘을 이용한 수위측정 방법 연구”, 한국정보기술학회 논문지, 11(5), pp. 167-173
- [5] 강일민 외 1 (2017),“영상처리기법을 이용한 실시간 변위 측정 및 붕괴감지 시스템”, 대한건축학회 논문집-구조계, 33(1), pp. 27-32
- [6] 전형섭 외 3 (2014),“카메라 영상을 이용한 진동 변위 측정 시 카메라 흔들림 보정방법”, 한국소음진동공학회 논문집, 24(10), pp.817-824