



M Y T H

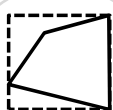
| M a k i n g Y o u r T h i n g s H a p t i c |

컴퓨터 공학부
미네르바
이광현 최준환 전세미 최여진 정하엽

•
c o n t e n t s
•



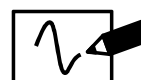
01 클린 코딩



02 왜곡 보정



03 Calibration



04 곡선 보정

Contents 1

1 . 클 린 코 딩

1 . 클 린 코 딩

함수의 세분화 및 데이터의 클래스화

KinectUI

- ▶ Properties
- ▶ 참조
- ▶ images
- ▶ App.xaml
 - ▶ App.xaml.cs
 - ▶ App
- ▶ MainWindow.xaml
 - ▶ MainWindow.xaml.cs
 - ▶ MouseEvent
 - ▶ ToolBar
- ▶ PenTool.xaml
 - ▶ PenTool.xaml.cs
 - ▶ PenTool
- ▶ SetUp.xaml
 - ▶ SetUp.xaml.cs
 - ▶ SetUp

MYTH

- ▶ Properties
- ▶ 참조
- ▶ images
- ▶ App.xaml
 - ▶ App.xaml.cs
- ▶ Control.cs
- ▶ ToolBar
- ▶ Data.cs
- ▶ ToolBar
- ▶ MainWindow.xaml
 - ▶ MainWindow.xaml.cs
 - ▶ MouseEvent
 - ▶ ToolBar
- ▶ PenTool.xaml
 - ▶ PenTool.xaml.cs
 - ▶ PenTool
- ▶ SetUp.xaml
 - ▶ SetUp.xaml.cs
- ▶ TouchPoint.cs
- ▶ TouchPoint

기존
클린 코딩 이후

1 . 클 린 코 딩

불필요한 작업의 최소화 및 변수 선언

상수들의 변수 선언

```
// 좌우 반전으로 Color 값 출력
for (int i = 0; i < 480; i++)
{
    for (int j = 639; j >= 0; j--)
    {
        short currentDepth = depthPixel[i, j];
        short initDepth = initDepthPixel[i, j];
        short subDepth;
        if (currentDepth > 2000 || initDepth > 2000)
            subDepth = 0;
        else
            subDepth = (short)(initDepth - currentDepth);

        if (subDepth < 0)
            subDepth = 0;
        if (0 <= subDepth && subDepth <= 7)
        {
            colorPixels[colorPixelIndex++] = 255;
            colorPixels[colorPixelIndex++] = 0;
            colorPixels[colorPixelIndex++] = 0;
        }
        else if (7 < subDepth && subDepth <= 14)
        {
            colorPixels[colorPixelIndex++] = 0;
            colorPixels[colorPixelIndex++] = 0;
            colorPixels[colorPixelIndex++] = 255;
        }
        else
        {
            colorPixels[colorPixelIndex++] = 0;
        }
    }
}
```

조건문의 최소화
상황 연산자로 반복되는 코드 처리

```
// 좌우 반전으로 Color 값 출력
for (int i = 0; i < ScreenPixelHeight; i++)
{
    for (int j = (int)(ScreenPixelWidth - 1); j >= 0; j--)
    {
        short currentDepth = depthPixel[i, j];
        short initDepth = initDepthPixel[i, j];
        short subDepth;

        if (currentDepth > depthMaxLimit || initDepth > depthMaxLimit)
            subDepth = 0;
        else
            subDepth = (short)((initDepth - currentDepth < 0 ? 0 : (initDepth - currentDepth));

        colorPixels[colorPixelIndex++] = (byte)(subDepth <= blueColorLimit ? 255 : 0);
        colorPixels[colorPixelIndex++] = 0;
        colorPixels[colorPixelIndex++] = (byte)(blueColorLimit < subDepth && subDepth <= redColorLimit ? 255 : 0);

        ++colorPixelIndex;
    }
}
```

기존

클린 코딩 이후

1 . 클 린 코 딩

기존에 구현되어 있던 Linked List로 변경

직접 구현한 Linked List 를

```
// 없어진 TouchPoint Linked List에서 삭제
public void removeNoneExist()
{
    TouchPoint T;
    for (T = head; T.next() != null; T = T.next())
    {
        TouchPoint next = T.next();
        if (!next.getExist())
        {
            if (next.revision > 5)
            {
                T.next().setClick(false);
                Up(MouseButton.Left);
                rightFlag = true;
                T.removeNext();
                T = head;
                pointCnt--;
                if (T.next() == null)
                    break;
                next.revision = 0;
            }
            else
                next.revision++;
        }
    }
    initExist();
}
```

Linked ListNode<TouchPoint> R;

```
// 없어진 TouchPoint Linked List에서 삭제
public void removeNoneExist()
{
    for (R = PointList.First; R != null; )
    {
        if (!R.Value.getExist())
        {
            if (R.Value.getrevision() > 5)
            {
                R.Value.setClick(false);
                if (!gestureFlag)
                    Up(MouseButton.Left);
                rightFlag = true;
                PointList.Remove(R);
                pointCnt--;
                R.Value.setrevision(0);
                R = PointList.First;
            }
            else
            {
                R.Value.revisionPlus();
                R = R.Next;
            }
        }
        else
            R = R.Next;
    }
    initExist();
}
```

기존

클린 코딩 이후

라이브러리를 이용하여 코드의 안정성을 상승시킴.

Contents 2

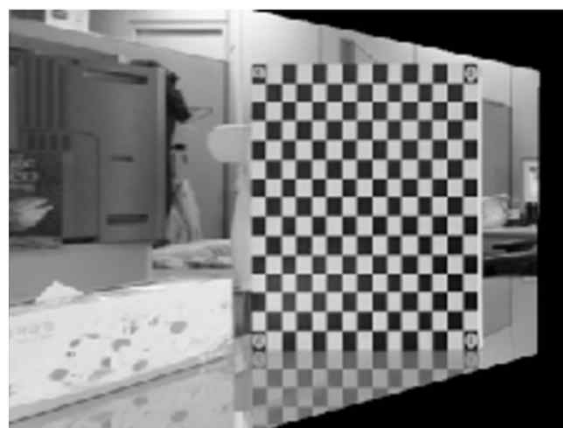
2 . 왜 곡 보 정

2 . 왜 곡 보 정

Perspective Projection의 이미지 변화 예시



보정 전



보정 후

위 방법을 좌표 보정에 사용.

2 . 왜 곡 보 정

좌표 보정 계산 방법

$$\begin{bmatrix} x_1' \\ y_1' \\ x_2' \\ y_2' \\ x_3' \\ y_3' \\ x_4' \\ y_4' \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x_1' & -x_1'y_1' \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y_1' & -y_1y_1' \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x_2' & -x_2'y_2' \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y_2' & -y_2y_2' \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3x_3' & -x_3'y_3' \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3y_3' & -y_3y_3' \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4x_4' & -x_4'y_4' \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x_4y_4' & -y_4y_4' \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix}$$

Projection Matrix

$$\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homogenous Coordinate

$$x' = \frac{ax + by + c}{gx + hy + 1}, \quad y' = \frac{dx + ey + f}{gx + hy + 1}$$

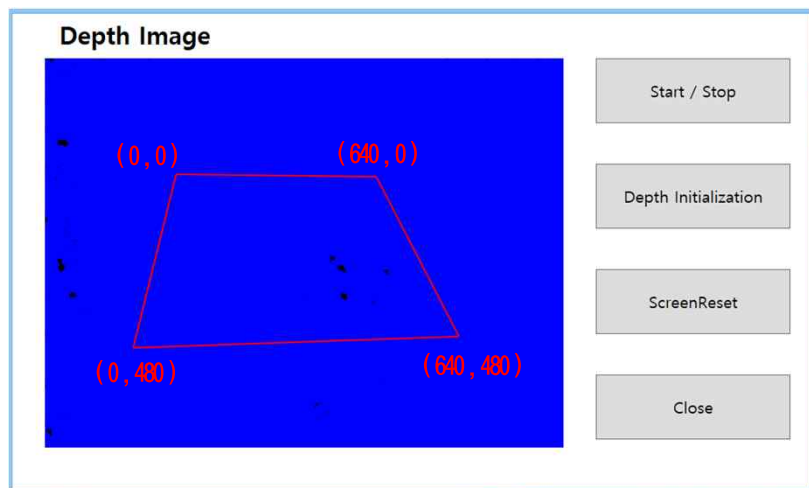
보정 후 좌표

Inverse Matrix를 양쪽 좌변에 곱해 a~h를 구함.

Homogenous Coordinate를 사용하여 이동한 좌표를 구함.

2 . 왜 곡 보 정

좌표 보정



유저가 설정한 스크린 좌표를 640X480 크기로 투영하여

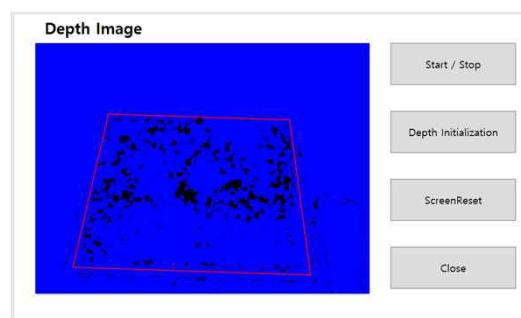
보정된 좌표를 구함 .

Contents 3

3 . C a l i b r a t i o n

3 . C a l i b r a t i o n

Calibration 방법 추가



기존

마우스 클릭 추가

기존의 Calibration 방법을 마우스 클릭을 추가.

3 . C a l i b r a t i o n

Depth Camera 와 Color Camera 시점 문제



기존의 문제



해결 시

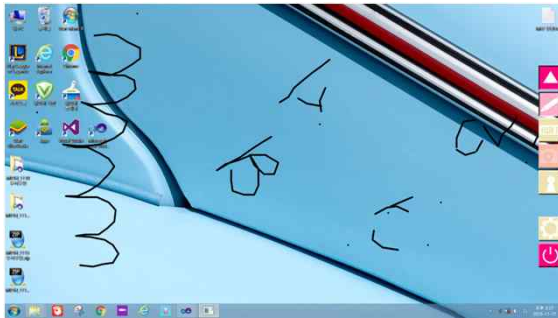
시점 차이로 인해 좌표 보정이 필요함.

Contents 4

4 . 곡 선 보 정

4 . 곡 선 보 정

펜 기능의 곡선 보정



- 곡선이 날카롭게 표현됨
- 선의 끊김 현상이 빈번함

보정 전



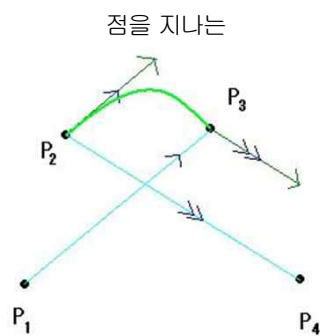
- 곡선이 비교적 부드럽게 표현됨
- 선의 끊김 현상이 줄어듦

보정 후

전반적인 펜 기능을 안정화시킴.

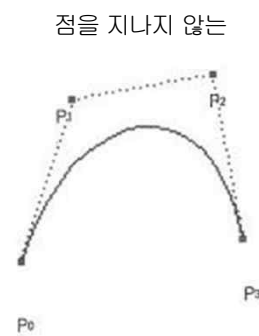
4 . 곡 선 보 정

곡선 알고리즘



Catmull-Rom Spline Curve

VS



Bezier Curve

Catmull-Rom Spline Curve 선택

Catmull-Rom 스플라인 곡선의 매개 변수 방정식

$$p(t) = 0.5 * ((2.0 * p_1) + (-p_0 + p_2) * t + (2.0 * p_0 - 5.0 * p_1 + 4.0 * p_2 - p_3) * t^2 + (-p_0 + 3.0 * p_1 - 3.0 * p_2 + p_3) * t^3)$$

| Q & A
질 의 응 답

| T h a n k Y o u
들 어 주 셔 서 감 사 합 니 다