

1. Implementatieplan chin recognition

1.1. Namen en datum

Daniël van den Berg - 1636304

Hylco Uding - 1635936

Dinsdag 26 Mei 2015

1.2. Doel

Het doel van de "chin recognition" is het herkennen van kin contouren. Hierdoor kunnen twee verschillende foto's vergeleken worden.

1.3. Methoden

Voor het herkennen van de kin contouren is er 1 methode die het makkelijkste en het beste te implementeren is. bij deze methode pak je een punt net onder je mond en trekt hier 18 lijnen van 0 tot 180 graden. Bij deze methode is het eerste punt wat je tegen komt op deze lijn een stuk contour van de kin.

1.4. Keuze

Wij kiezen voor de methode die hier boven wordt beschreven.

1.5. Implementatie

De methode is als volgt geïmplementeerd:

```
for(int i = 0; i < 19; i++)
{
    bool pointFound = false;
    int x = MouthCenter.x;
    int y = MouthCenter.y;
    int j = 0;
    while (x > features.getFeature(FEATURE_HEAD_LEFT_SIDE).getX() && x
    < features.getFeature(FEATURE_HEAD_RIGHT_SIDE).getX() && y > 0 && y <
    features.getFeature(FEATURE_CHIN).getY() && !pointFound){
        x = (int) std::round(MouthCenter.x + (j * std::sin((270 + (10*i)) *
    (PI/180))));
        y = (int)std::round(MouthCenter.y + VerticalOffset + (j *
    std::cos((270 + (10 * i)) * (PI / 180))));
        if(image.getPixel(x,y) == 0){
            pointFound = true;
            chinCountor.addPoint(Point2D<double>(x, y));
        }
        j++;
    }
}
```

Er wordt dus voor elke lijn gekeken , tot die de zijkant of onderkant van het hoofd heeft geraakt, of er een andere zwart puntje wordt geraakt.

1.6. Evaluatie

Als evaluatie zal een visuele vergelijking tussen onze methode en de openCV methode. Daarnaast zal er worden gekeken naar de procentuele match bij een plaatje wat met de standaard methode is ingelezen en met onze methode wordt herkent.