VISION - TME 4
Disparity Map Estimation Using Graph Cuts

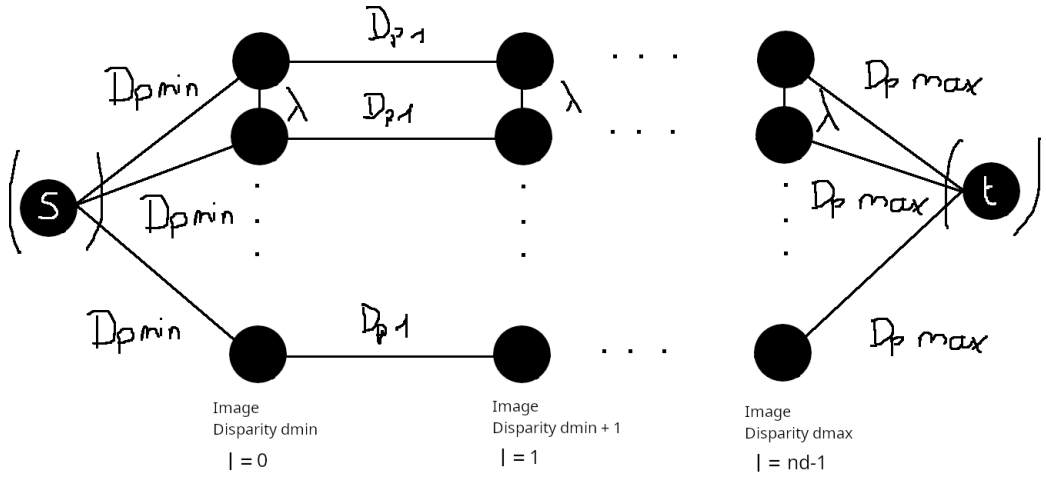
Marie Diez

Contents

1 1. Problem statement

1.1 Compute a disparity map using graph cut

1.1.1 Graph representation



Dpmin = (int)(wcc*rho(zncc(I1, I1M, I2, I2M, x, y, x+dmin, y, n)) + Kp)
Dpmax = (int)(wcc*rho(zncc(I1, I1M, I2, I2M, x, y, x+dmax, y, n)) + Kp)
Dl = (int)(wcc*rho(zncc(I1, I1M, I2, I2M, x, y, x+dmin+l, y, n)) + Kp)

Figure 1: Graph representation

The differents layer represents the image at differents disparity, from dmin to dmax, so there are nd layers. On each layer the image is represented as a vector of size $nx * ny$ and we can access each coordinates as $j * nx + i$.
To create the graph we first assign weight to the first layer and the last layer, then we iterate over layers to assign weight between them. In each layer we assign a weight λ between neighboors pixels of the image $j * nx + (i + 1)$ and $(j + 1) * nx + i$.

Now that our graph is constructed, we can compute the min cut (max flow) on our graph and each element of the sink part will be used to compute the disparity. For each pixel i,j the disparity correspond at the first node of this pixel in the sink for a given layer and will be $D(i, j) = dmin + l$.

1.2 Display the resulting disparity map

(a) images

(b) disparity map

(c) blur disparity map

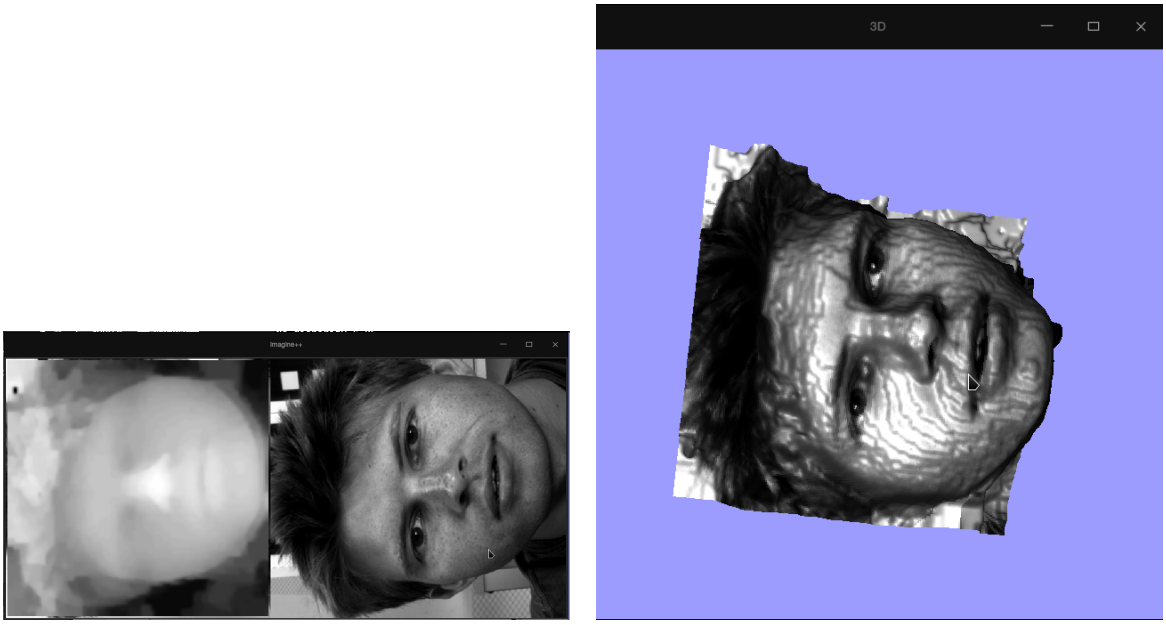
(d) 3D

Figure 2: Results for $\lambda_f = 0.1$, $\sigma = 3$ and $n = 3$

Computation time : 20 sec

1.3 Test of blur effect

1.3.1 $\sigma = 1$

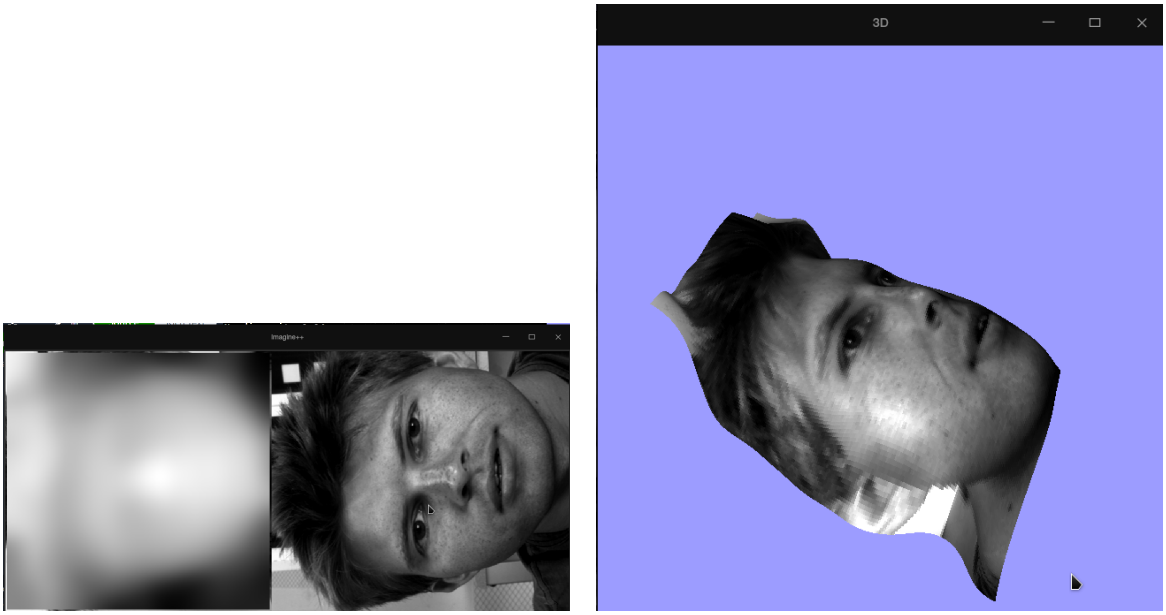


(a) blur disparity map (b) 3D

Figure 3: Results with $\sigma = 1$, $\lambda_f = 0.1$ and $n = 3$

Computation time : 20 sec

1.3.2 $\sigma = 10$



(a) blur disparity map (b) 3D

Figure 4: Results with $\sigma = 10$, $\lambda_f = 0.1$ and $n = 3$

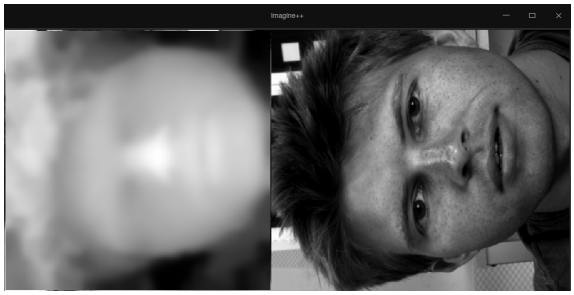
Computation time : 20 sec

We can see that the blur effect enable to smooth the 3D result, a too small σ leads to too sharp result and a too large value leads to oversmoothing. The result where much better with previous value of $\sigma = 3$.

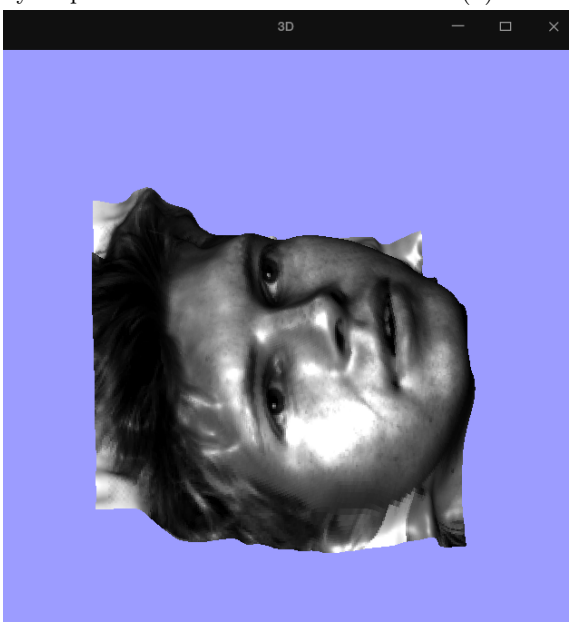
1.4 Test of NCC patch size

1.4.1 $n = 2$

(a) disparity map



(b) blur disparity map

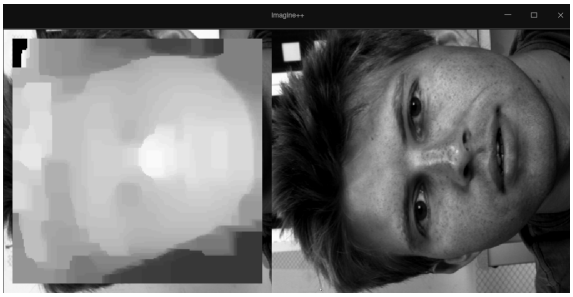


(c) 3D

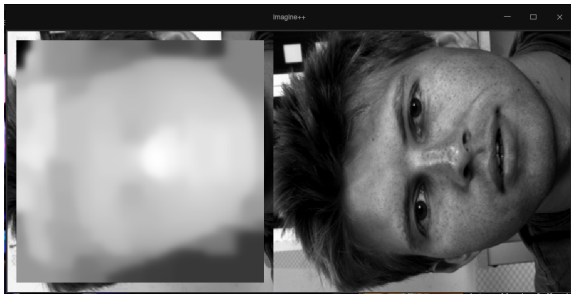
Figure 5: Results with $\sigma = 3$, $\lambda_f = 0.1$, $n = 2$

Computation time : 13 sec

1.4.2 $n = 15$



(a) disparity map



(b) blur disparity map

(c) 3D

Figure 6: Results with $\sigma = 3$, $\lambda_f = 0.1$, $n = 15$

Computation time : 4min40sec

The larger the patch size, the more simplified the disparity map. It can be noticed that the borders are not taken into account, the bigger the patch size the more visible it is, this comes from the calculation of the new coordinates in space $x = i * zoom + n$ and $y = j * zoom + n$.

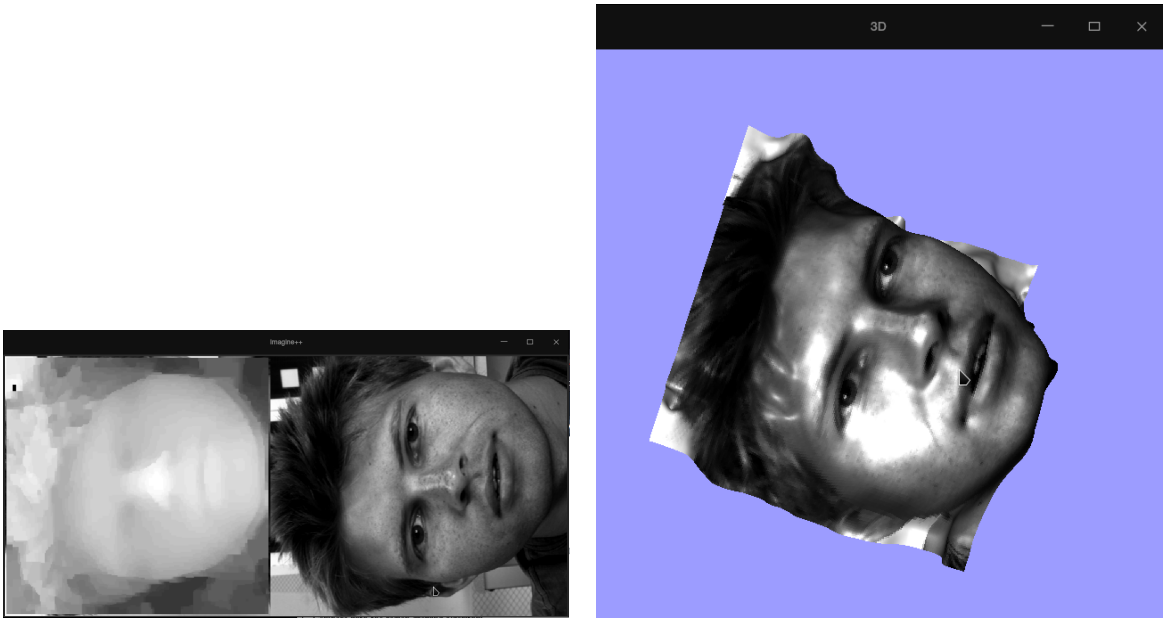
1.5 Test of the regularization

We search for a cut that minimize the energy :

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{(p,q) \in N} \lambda_{p,q} |f_p - f_q|$$

The first term is the data term and the second is a regularization term. The more λ is big the more regulazise will be the disparity :

1.5.1 $\lambda_f = 0.1$

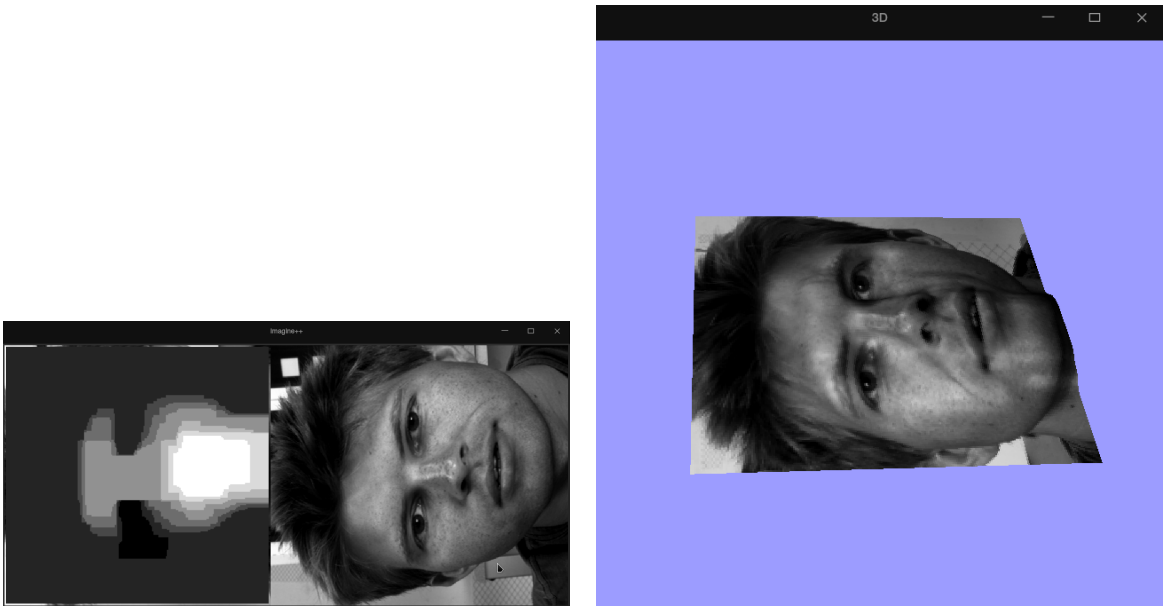


(a) disparity map (b) 3D

Figure 7: Results with $\lambda_f = 0.1$

Computation time : 20 sec

1.5.2 $\lambda_f = 1$



(a) disparity map (b) 3D

Figure 8: Results with $\lambda_f = 1$

Computation time : 25 sec
We can see that is the regularization term is high, the disparity is lot more regular than with a small λ_f.

1.6 Test of the penalisation Kp

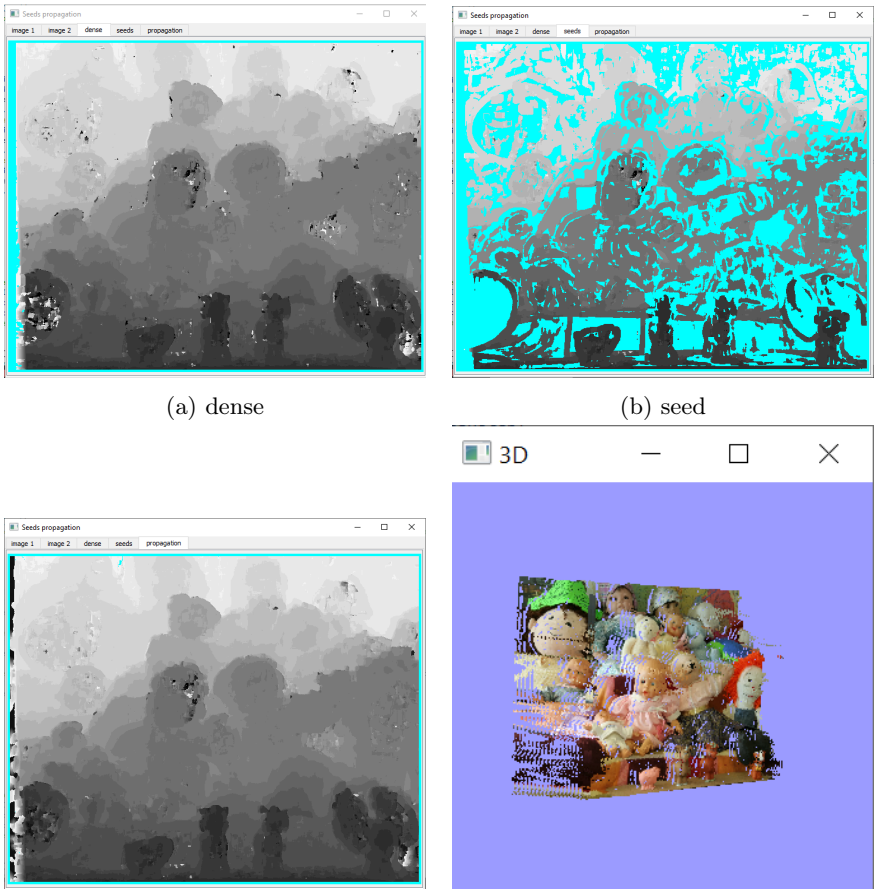
This penalisation avoid to have several cut locations on the same line, it penalize more cutting. It make some test with a small and large value of Kp, i obtain the same results as the first results of the report. I assume that in this case there are not sevral cut une the same line, so the regularization is not necessary in this case.

1.7 Compare with the region-growing algorithm

We first can try to compare both algorithm on the same image.

Toys images

- Region growing approach



(a) dense

(b) seed

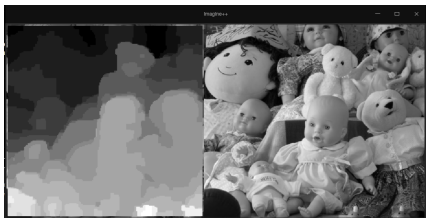
(c) seed propagation

(d) 3D

Figure 9

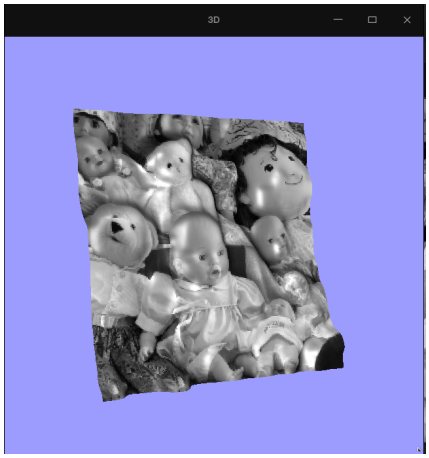
Computation time : 5min

- Graph cut approach
  With $dmin = 7$ and $dMax = 30$:



(a) Disparity map

(b) Blur disparity map



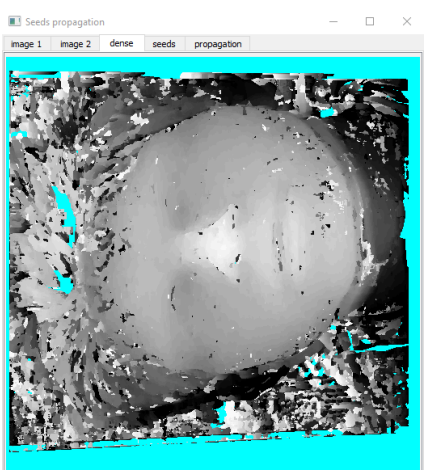
(c) 3D

Figure 10: Results with $\sigma = 3$, $\lambda_f = 0.1$, $n = 3$

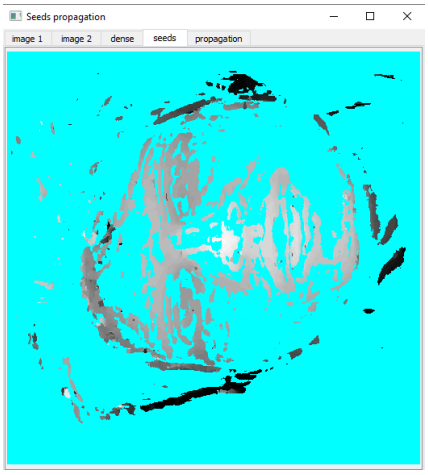
Computation time : 10 sec

Face images

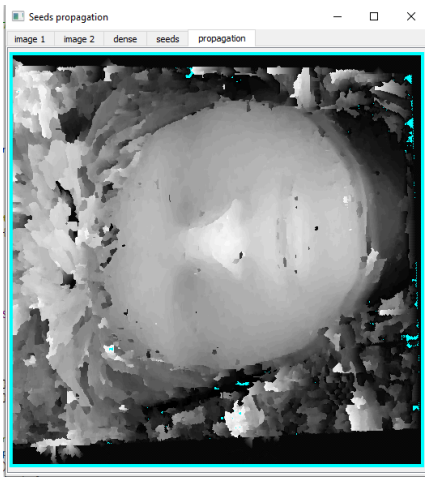
- Region growing approach



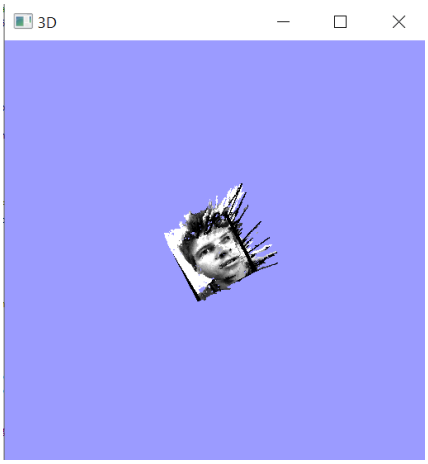
(a) dense



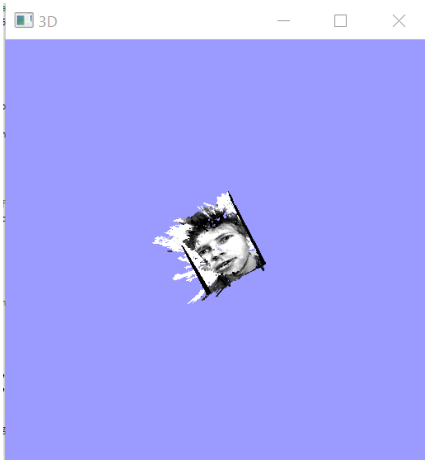
(b) seed



(c) seed propagation



(d) 3D



(e) 3D

Figure 11

Computation time : 7min40sec
The results are not very great, it will be necessary to find better paramaters and/or adapt $dMin$ and $dMax$.

1.7.1 Precision / Smoothness

What we can say is that overall the graphcut method provides a better precision and smoothness than region growing.

1.7.2 Computation time

I did all my computation time on a ryzen53500U processor.
The computation time for graph cut depends and the paramters, it can become very long when we increase NCC size. Otherwise with small ncc graph cut computation is lot faster than region growing in our case because it's a simple case where only one graph computation is done with a small NCC size. However in general global method (like graph cut) are often longer than local method (like region growing) but provides better results in term of smoothness and precision.