Université Nice SOPHIA ANTIPOLIS

Deformable models for Image Segmentation

UP - Master IMAMIS

August 2006

Diane Lingrand

lingrand@polytech.unice.fr

http://www.polytech.unice.fr/~lingrand

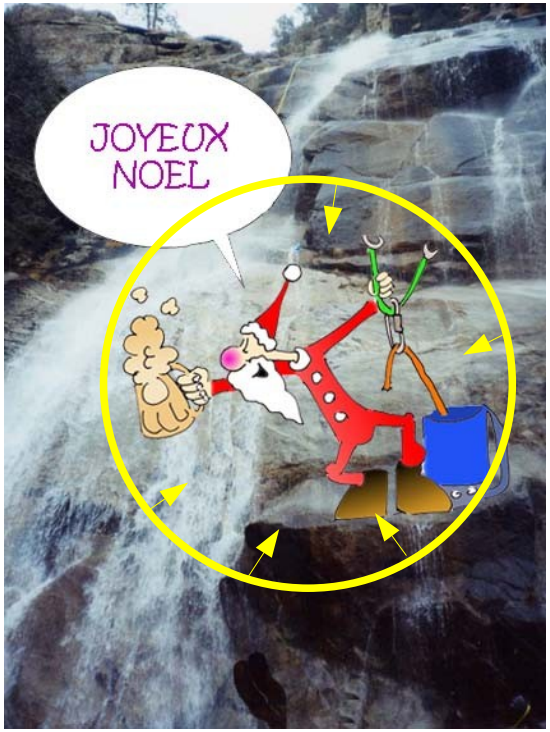
Deformable models for image segemntation

- Principle :
  - starting from a closed curve or closed surface (eg a circle or a sphere)
  - make it evolve regarding:
    - regularity criterions
    - the image data
    - evolution speed is computed at each iteration
  - how to represent such curve on the computer ?

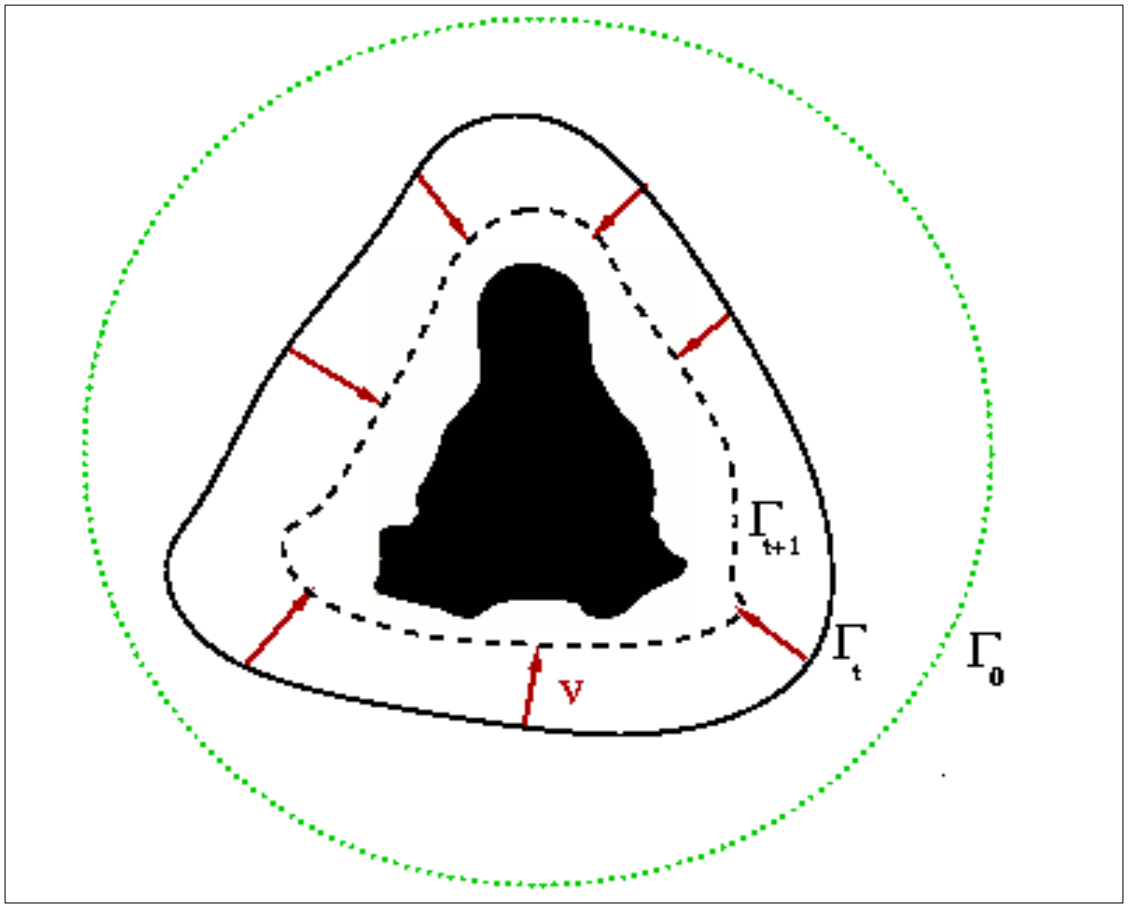
Principle

- Initial curve
- Iterative evolution of the curve
  - Forces
  - Stop criterion

Example



2D Deformable models



$$\frac{\partial \Gamma}{\partial \tau} = F\,\mathbf{n}$$

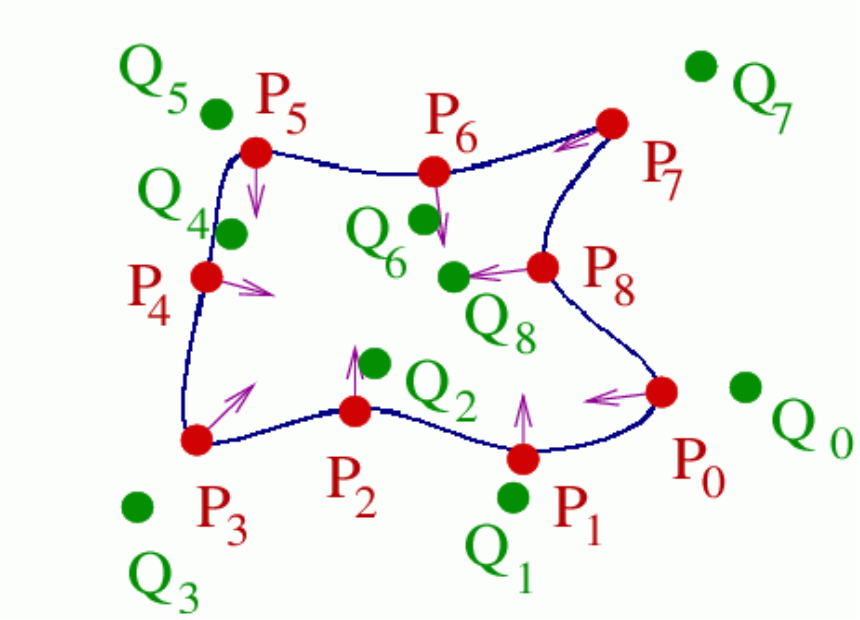
Polygonal representation

- Vertices P0, P1, ... Pn
- Normal vector at Pi defined as the normal vector to the line P(i-1) P(i+1)
- Curvature ?
- Advantages :
  - Very easy to manipulate

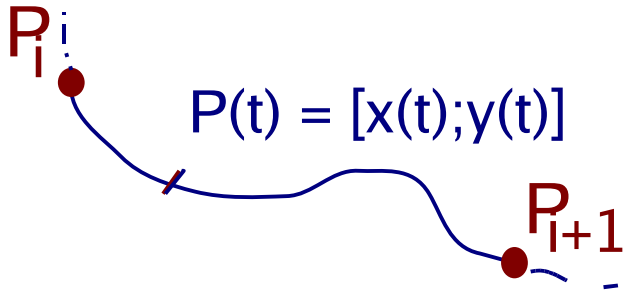
Parameterized curve

- Curve C parameterized by t
- Point P of C : $P(t) = (x(t), y(t))$
- Normal vector at P : $N(t) = (-y', x')/\mathrm{sqrt}(x'^2+y'^2)$
- Curvature : $k(t) = (x'y'' - x''y')/(x'^2+y'^2)^{3/2}$
- Examples : uniform Bsplines
  - $N(t) = (-b_1, a_1)/\mathrm{sqrt}(a_1^2+b_1^2)$
  - $k(t) = 2(a_1b_2 - a_2b_1)/(a_1^2+b_1^2)^{3/2}$

Uniform B-Spline representation



$$P_i = (Q_{i-1} + 4Q_i + Q_{i+1})/4$$



$$\begin{cases} x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \\ y(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3 \end{cases}$$

$$t \in [0\ 1]$$

$$[a_0\ ,\ b_0]^T = (Q_{i-1} + 4Q_i + Q_{i-1})/6$$
$$[a_1\ ,\ b_1]^T = (Q_{i+1} - Q_{i-1})/2$$
$$[a_2\ ,\ b_2]^T = (Q_{i+1} - 2Q_i + Q_{i-1})/2$$
$$[a_3\ ,\ b_3]^T = (Q_{i+2} - 3Q_{i+1} + 3Q_i - Q_{i-1})/6$$

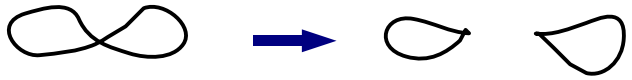
How to draw a Bspline ?

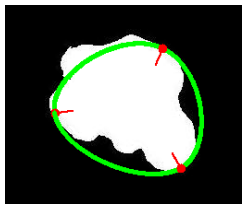
- For each point P_i , $0 \leq i \leq n-1$
  - draw the point P_i
  - for t from 0 to 1 : draw the point x(t),y(t)
    - $x(t) = a_0 + a_1\, t + a_2\, t^2 + a_3\, t^3$
    - $y(t) = b_0 + b_1\, t + b_2\, t^2 + b_3\, t^3$
    - step for t ?

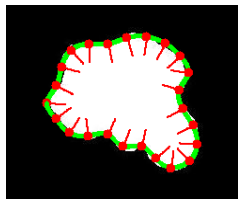
      1/max where max is the maximum value of $|x(P_i) - x(P_{i+1})|$ and $|y(P_i) - y(P_{i+1})|$
    - line thickness ?

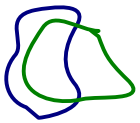
      fillOval(...)

Usual drawbacks and advantages

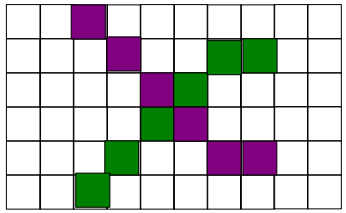
- Drawbacks:
  - Intersections 
  - Topology changes
  - Details depend on the number of P points





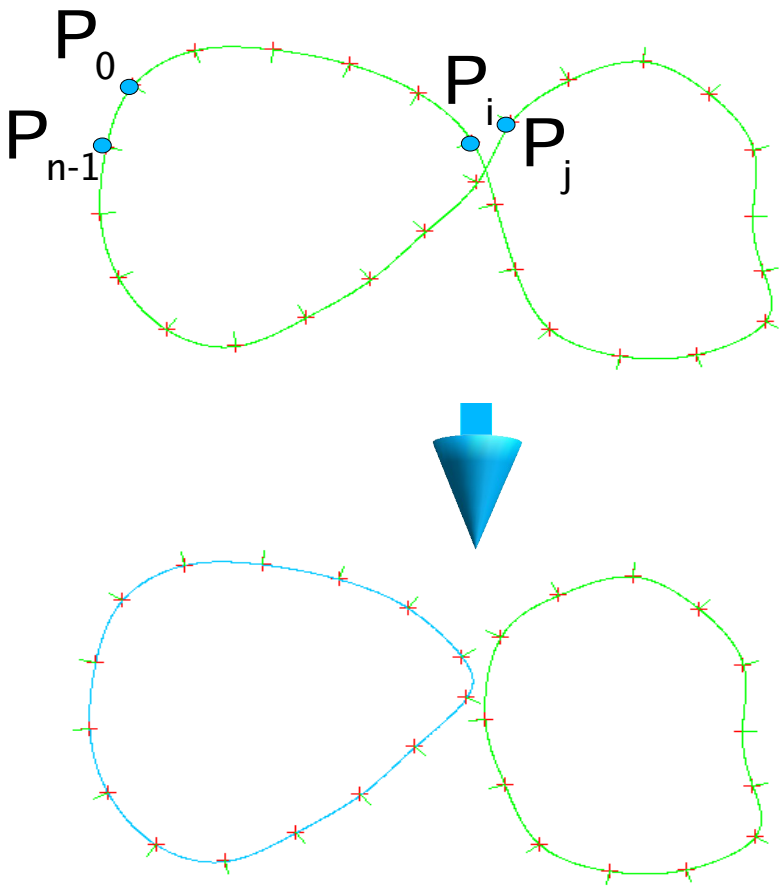
- Advantages:
  - Geometrically constrained 
  - Possibility of overlapping regions
  - Easy to compute normal vectors and curvature
  - Light in memory space and computation time

Bspline : how to detect self intersections ?



- We draw the curve in an array. We use different colors for different segments (segment i: color i+1)
- While drawing a segment, we verify that each point was not previously colored using another color
  - except for the 1st point of a segment
- Be careful to draw curves without holes !
  - when both x and y are modified, we also examine (xOld;y) and (x;yOld)

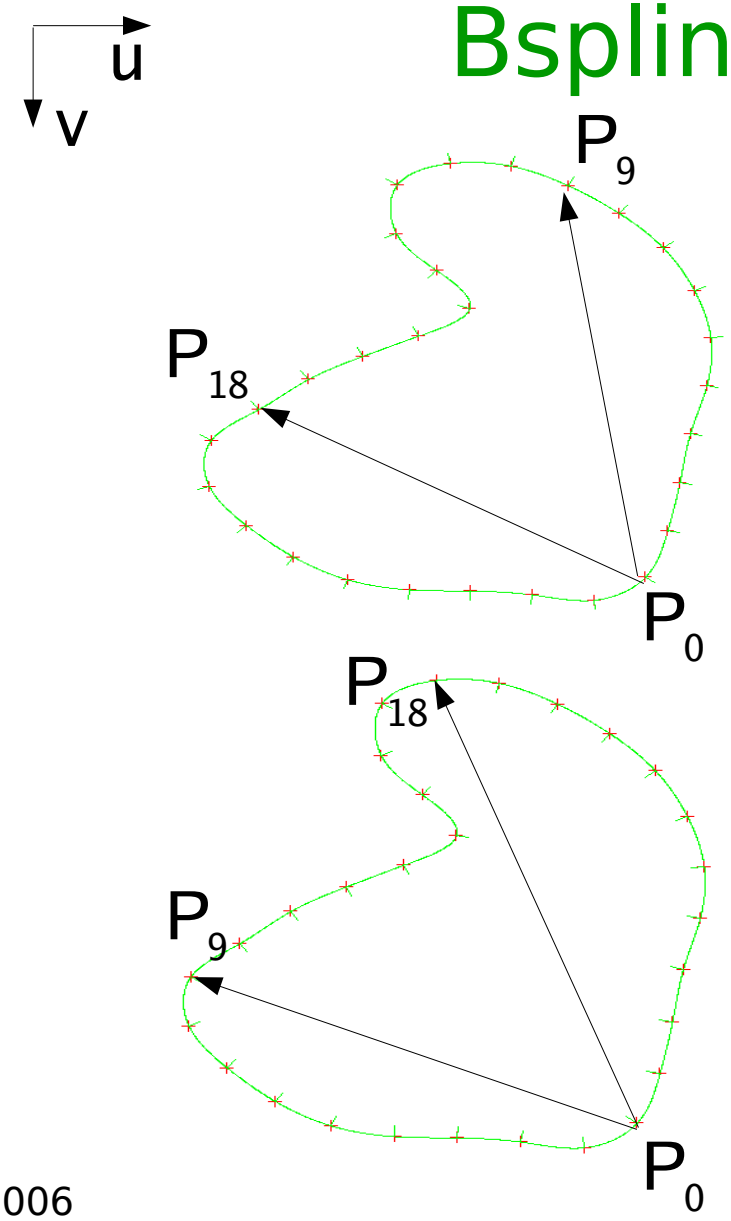
Bspline : how to deal with self intersections ?



- 1. Determination of the 2 segments that intersect : P_i and P_j

  $(P_0 \leq P_i < P_j \leq P_{n-1})$
- 2. Construction of 2 new Bsplines :
  - From P_0 to P_i and from P_{j+1} to P_{n-1}
  - From P_{i+1} to P_j

Bspline : orientation



- The normal vectors must go inward
- Then : P_0, $P_{n/3}$, $P_{2n/3}$ must be direct : we can verify it using the cross product:

$$\mathbf{P_0P_9} \wedge \mathbf{P_0P_{18}} = k\,\mathbf{w} \quad \text{with } k > 0$$

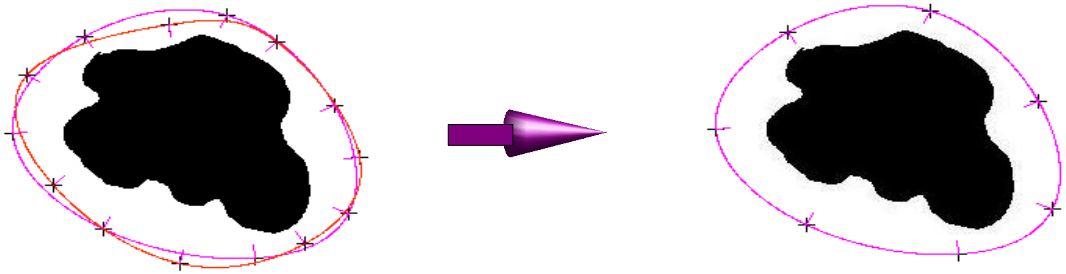
- The probability of P_0, $P_{n/3}$, $P_{2n/3}$ not direct and orientation correct is very low: this is a rapid test.

Bspline: inside / outside

- We use the region labeling method with some modifications:
  - each Bspline b is draw using color (-b)
  - regions are colored using positive values
- Assuming that point (0,0) is outside all curves, we determine the background
- Each region neighboor of a point of color (-b) become of color b.
- This is true only if there is no intersection between bsplines.
- This is true only if the drawn curves are closed (connectivity 4 or 8)

Bsplines : similar curves

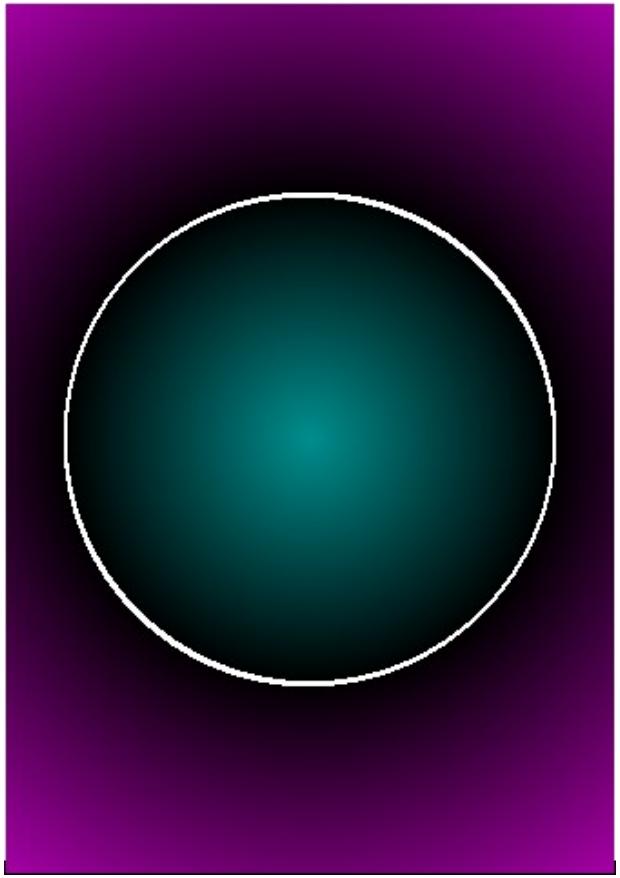
- During segmentation, we do not want to segment several times the same object. We want to eliminate similar curves.
- A similarity criterion : if 80% of points inside curve 1 are also inside curve 2 and constitute at least 80% of the points inside curve 2, the curves are similar.



Bsline : uniform !

- This property must be saved during the evolution of the curve.
- This needs :
  - computation of curve length
    - computation of segment length
  - computation of the point P(x(t),y(t)) between P_i and P_{i+1} such as dist(P_i,P) = d
  - closed-form solution does not existe : we need to approximate the integration

Levelset representation



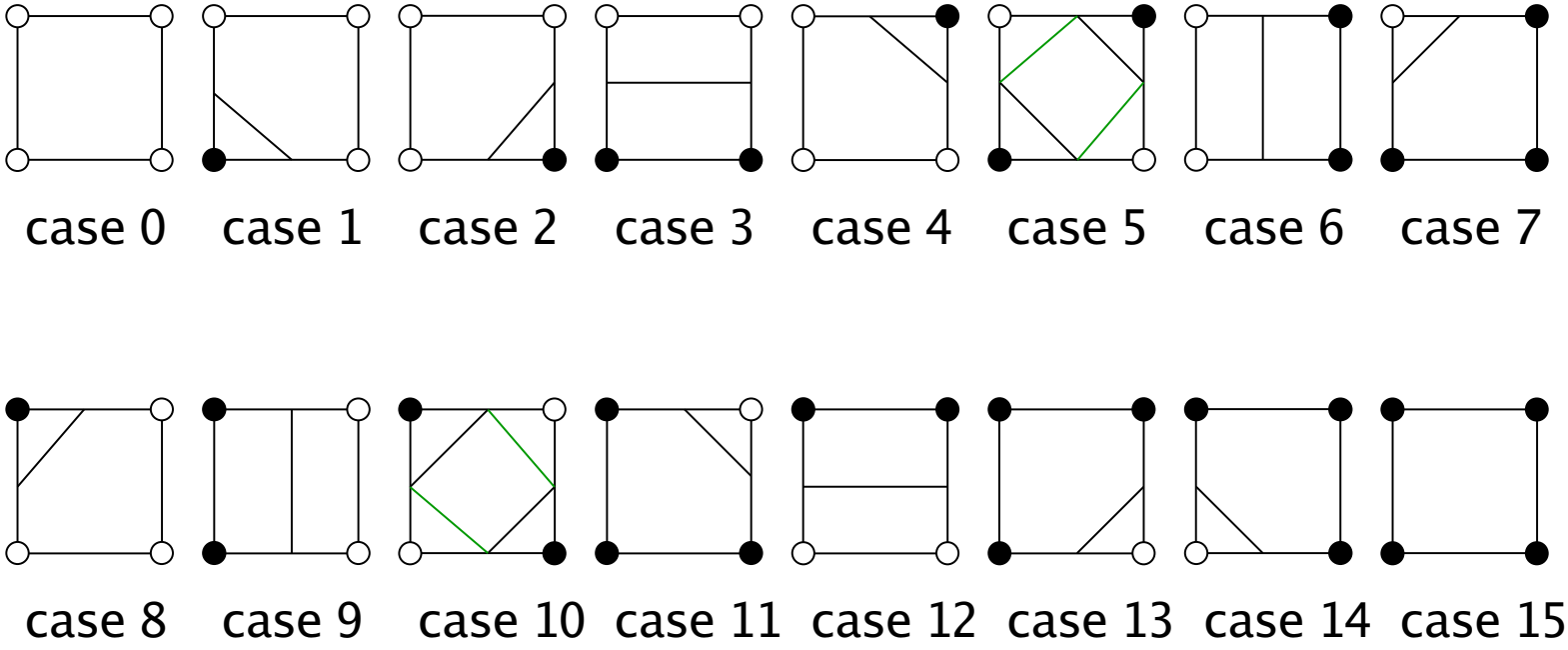
distance map

$$\frac{\partial u}{\partial \tau} = F\,\mathbf{n}$$

levelsets

- Let C be a curve. We define an image where each pixel has for value its distance to this curve C. Thus, C is the level 0 of this image.O
- Drawing this curve
  - Marching Squares in 2D

Marching Squares (2D)

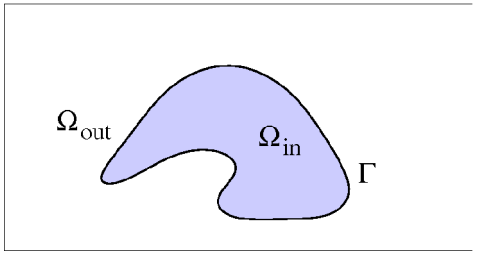


in 3D : http://www.marchingcubes.fr.st
by Raphaël Gervaise and Karen Richard, ESSI2 2001/02

Usual drawbacks and advantages

- Drawbacks
  - Slower
  - More memory
- Advantages
  - Easy to implement
  - Naturally handle topology changes
  - Easy to determine inside and outside a curve
  - Easy to adapt to higher dimension

Uniform object / uniform background



$$E = \lambda_1 \int \int_{\Omega_{\text{in}}} (I(x,y) - \mu_{\text{in}})^2 dxdy + \lambda_2 \int \int_{\Omega_{\text{out}}} (I(x,y) - \mu_{\text{out}})^2 dxdy$$

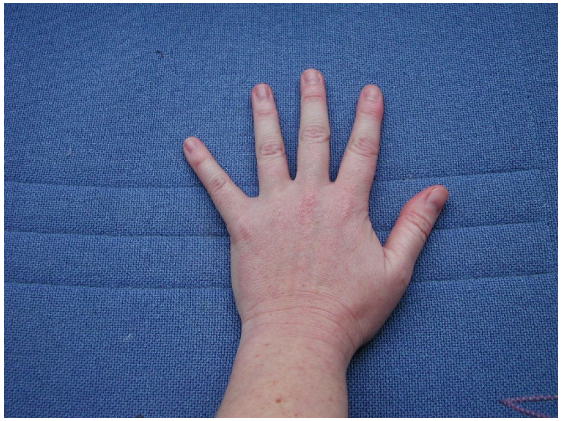
$$F = \lambda_1 (I - \mu_{in})^2 - \lambda_2 (I - \mu_{out})^2$$

$$E = \lambda_1 \int \int_{\Omega_{in}} (I(x,y) - \mu_{in})^2 dxdy + \lambda_2 \int \int_{\Omega_{out}} (I(x,y) - \mu_{out})^2 dxdy + \lambda_3 \int_{\Gamma} ds$$

$$E = \lambda_1 \sum_{\Omega_{in}} \sum (I(i,j) - \mu_{in})^2 + \lambda_2 \sum_{\Omega_{out}} \sum (I(i,j) - \mu_{out})^2 + \lambda_3 L_{\Gamma}$$

$$F = \lambda_1 (I - \mu_{in})^2 - \lambda_2 (I - \mu_{out})^2 + \lambda_3 \kappa$$

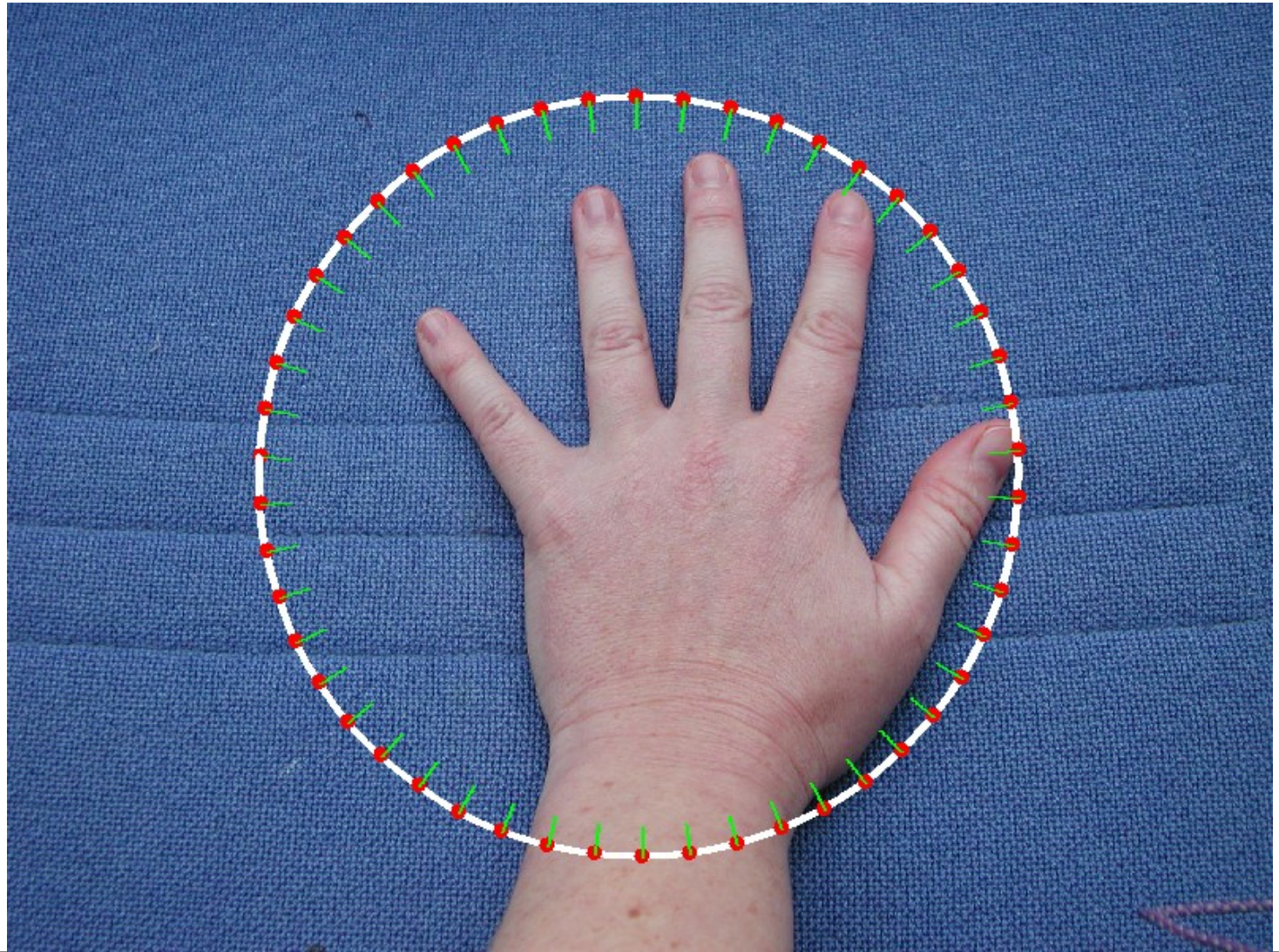
First experiment: Segmenting a hand from a uniform background

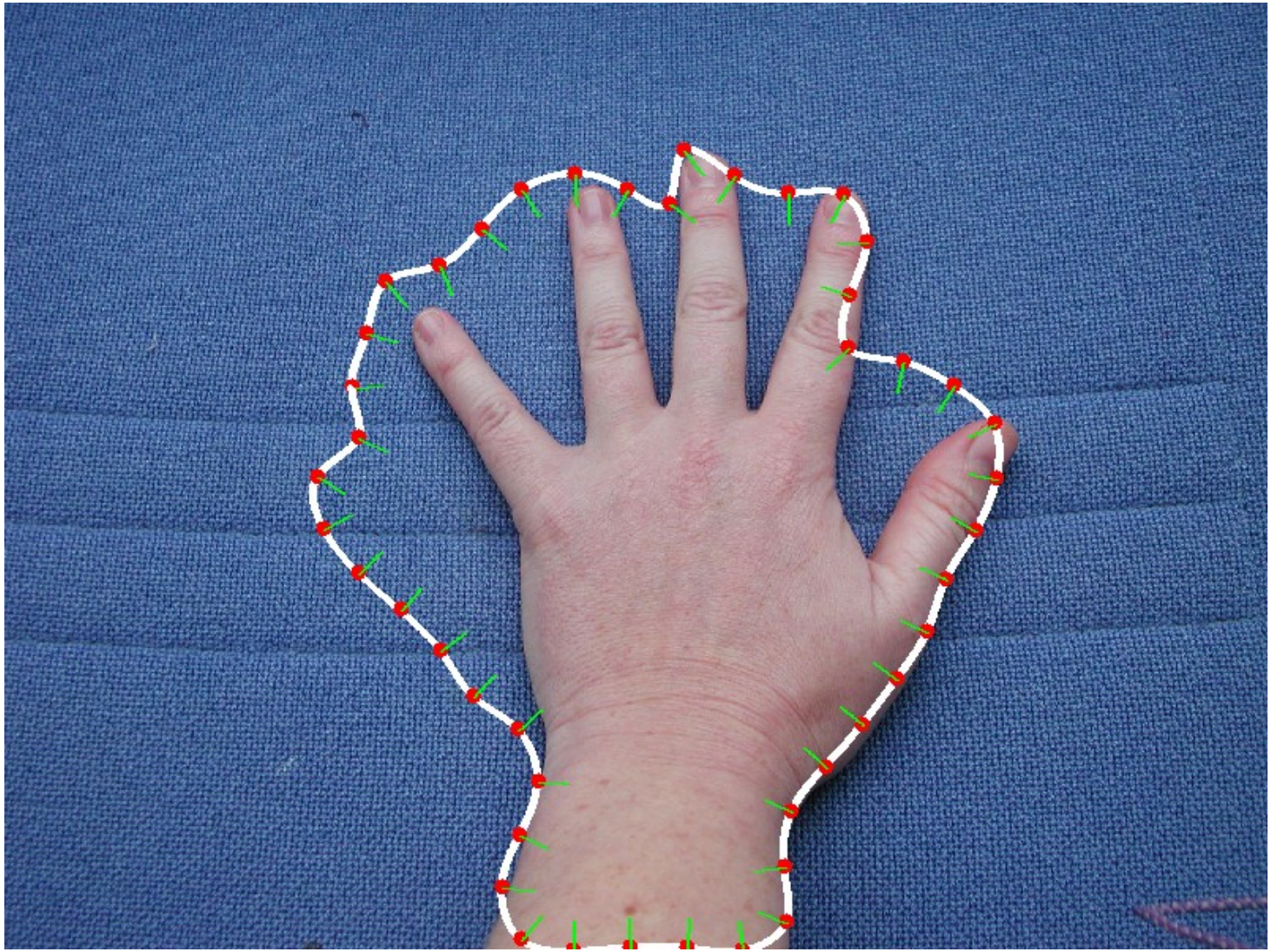


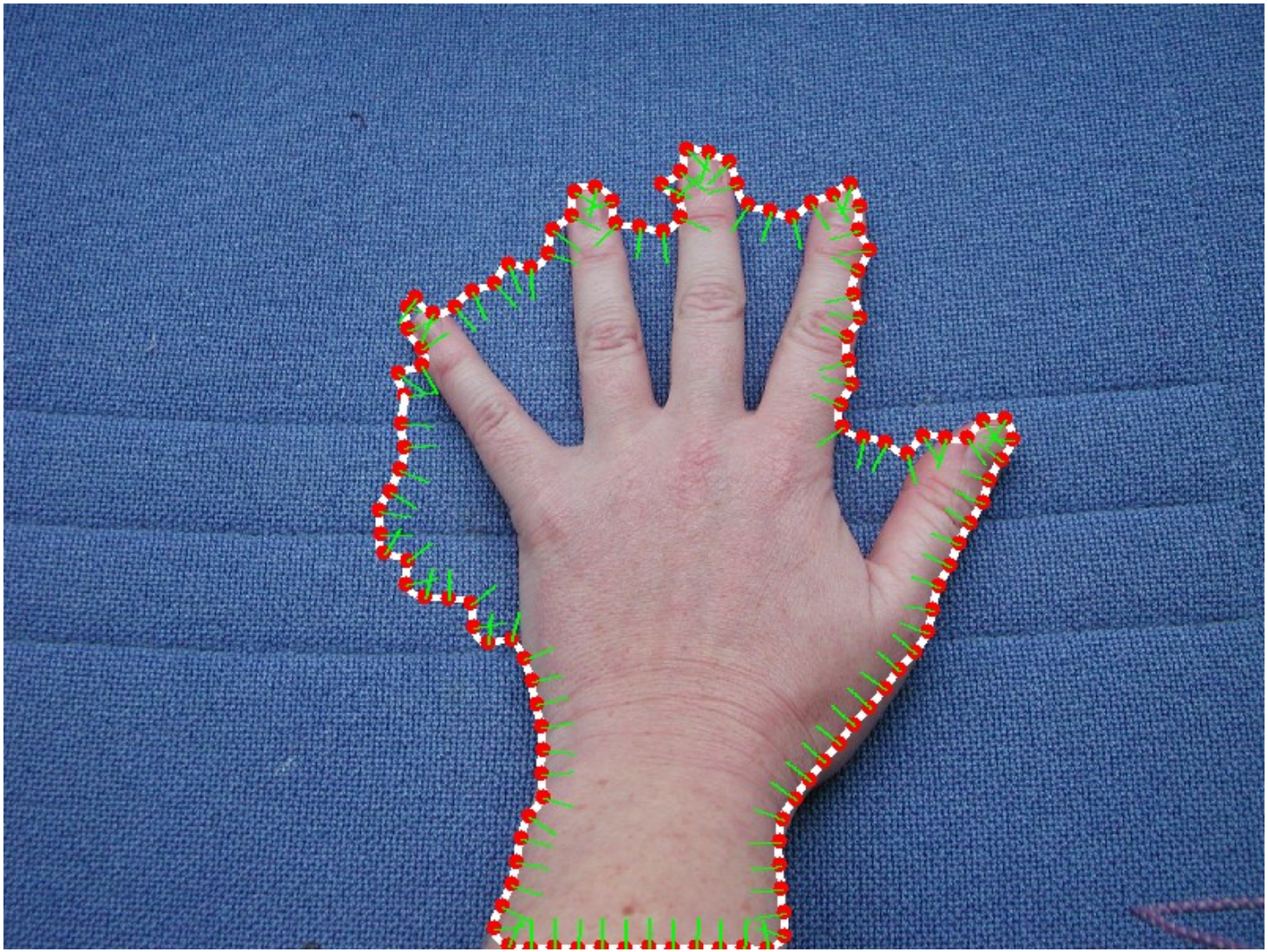
$$F = \lambda_1(I - \mu_{in})^2 - \lambda_2(I - \mu_{out})^2 + \lambda_3\kappa$$

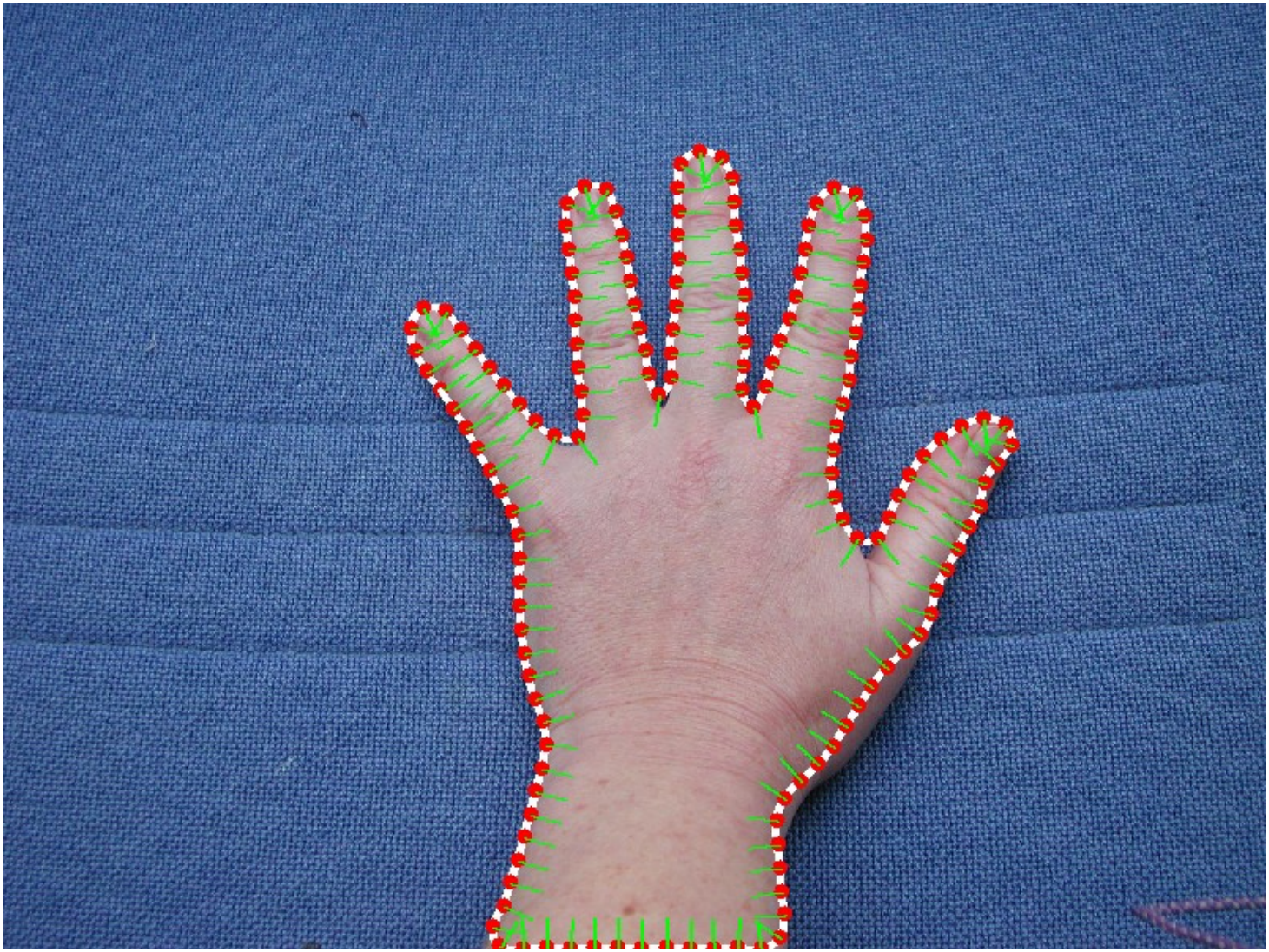
- Assuming that the background is uniform
- Assuming that the object is uniform

... using B-Splines

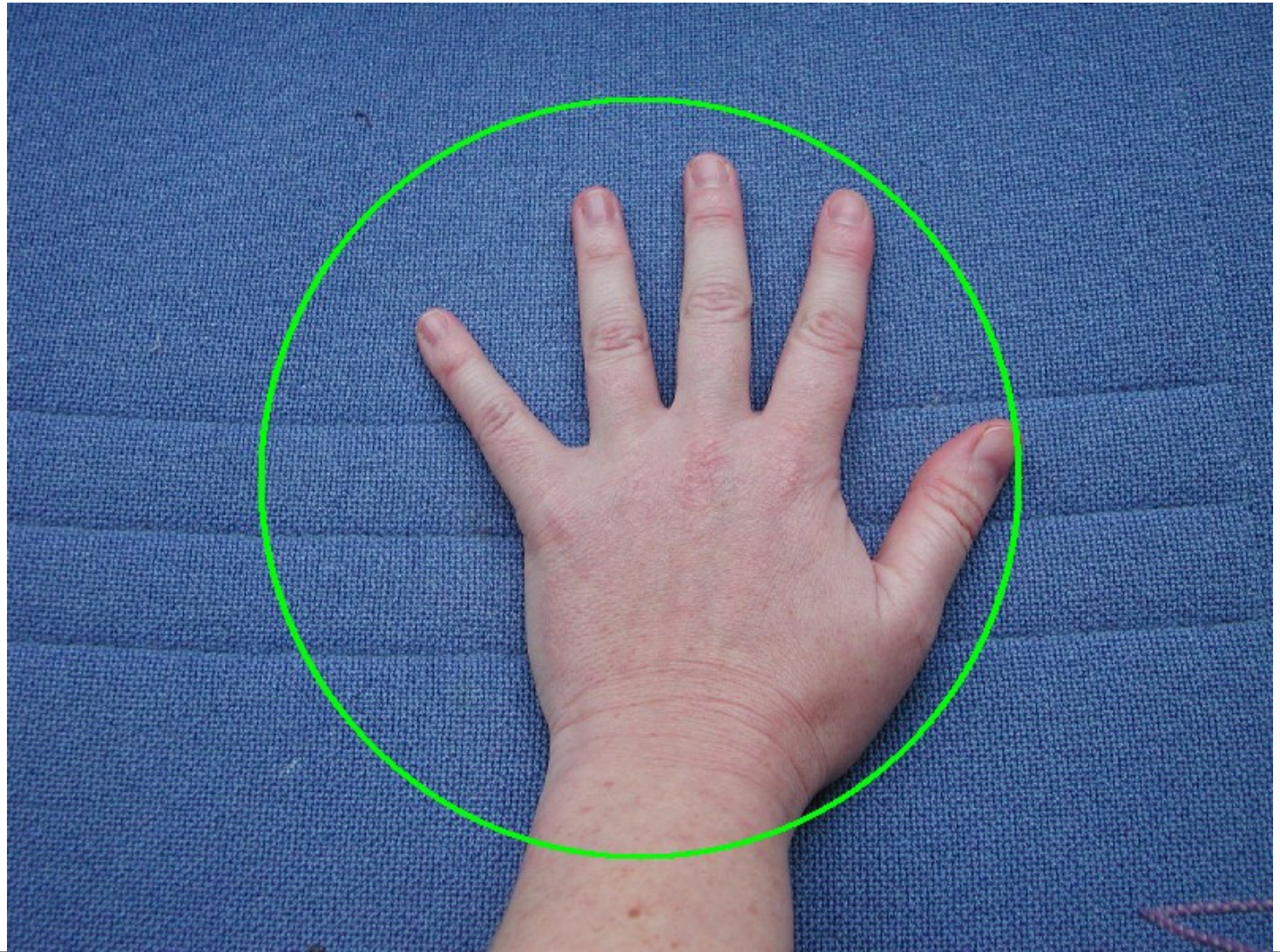


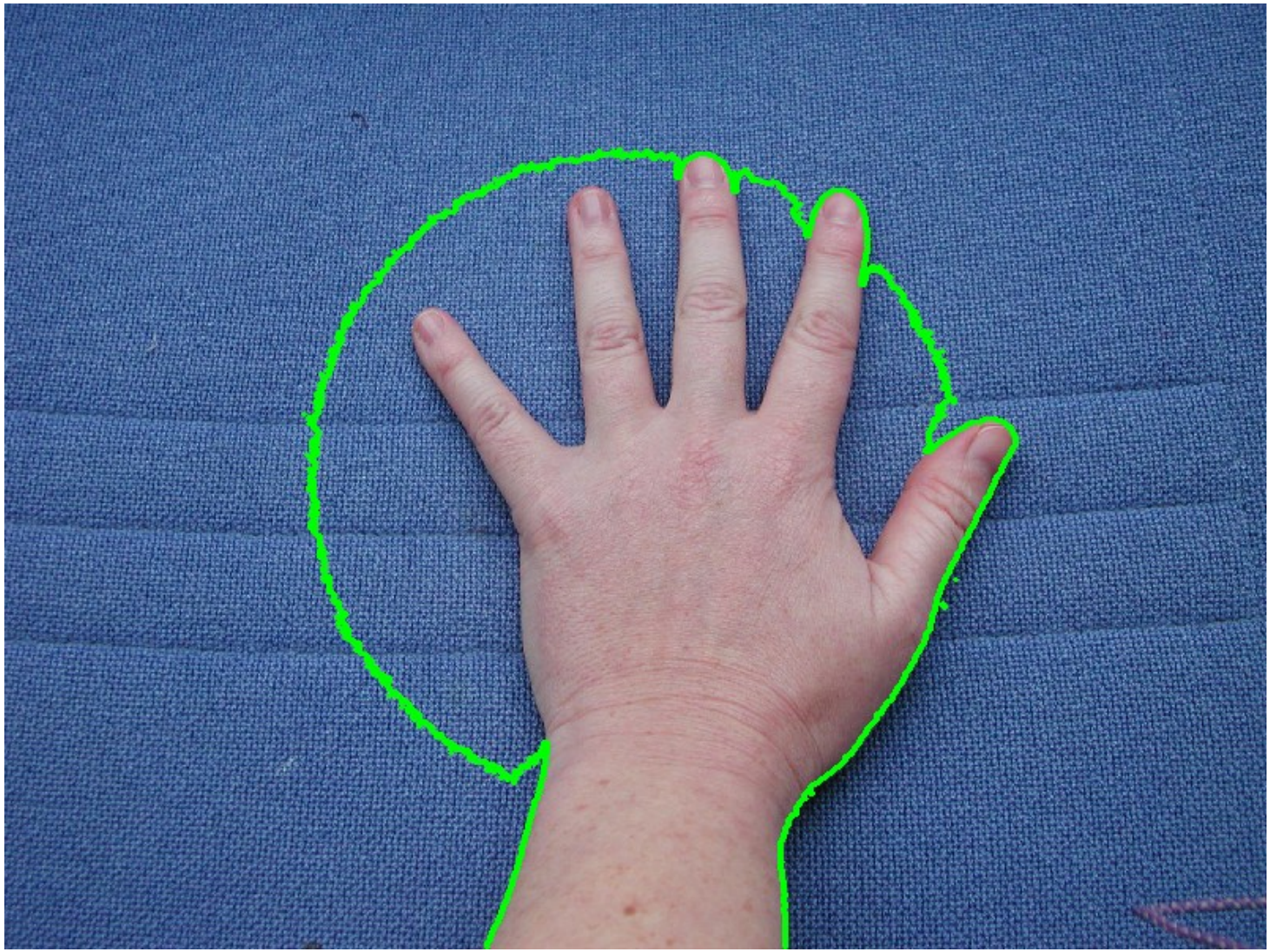


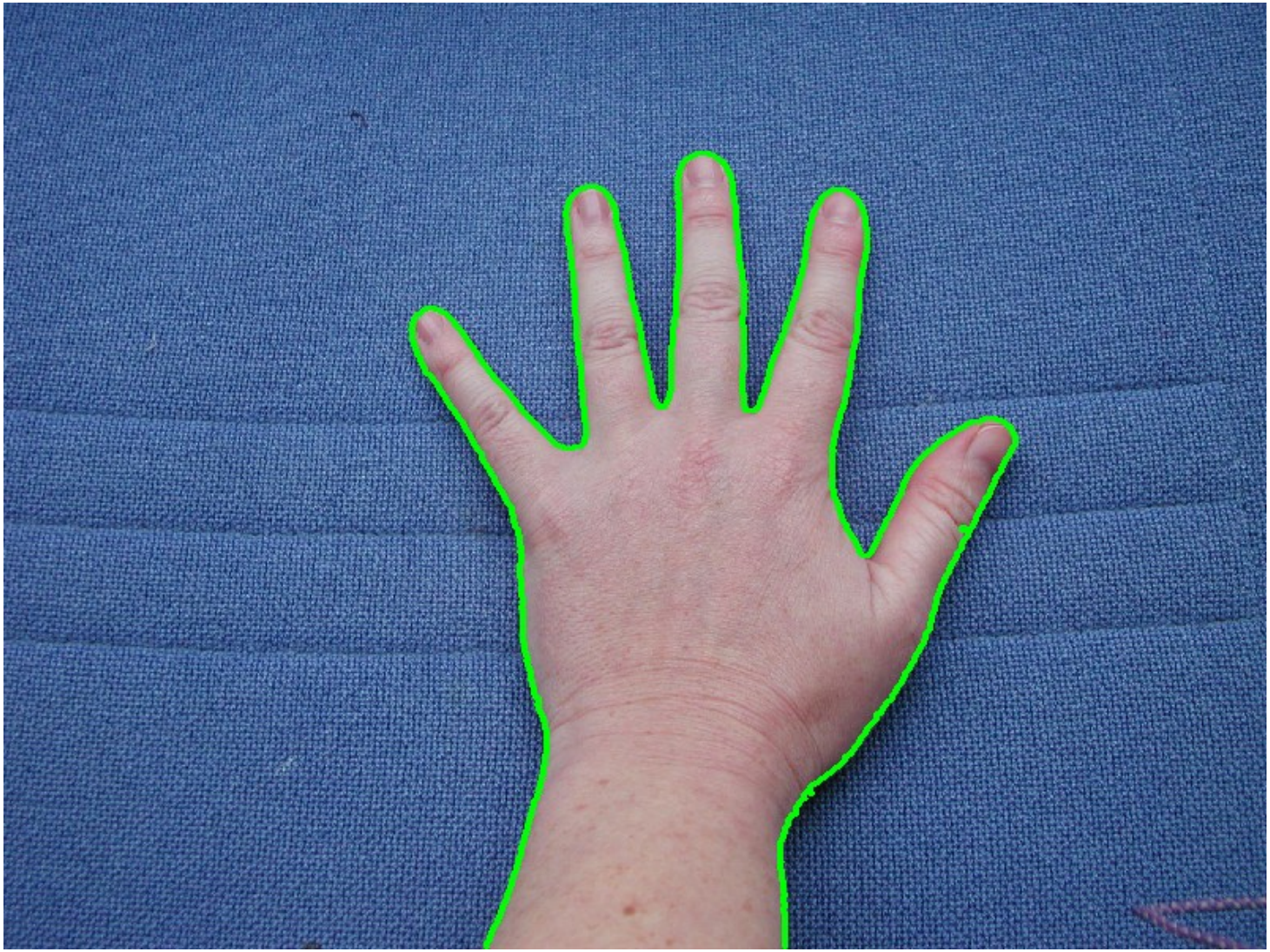


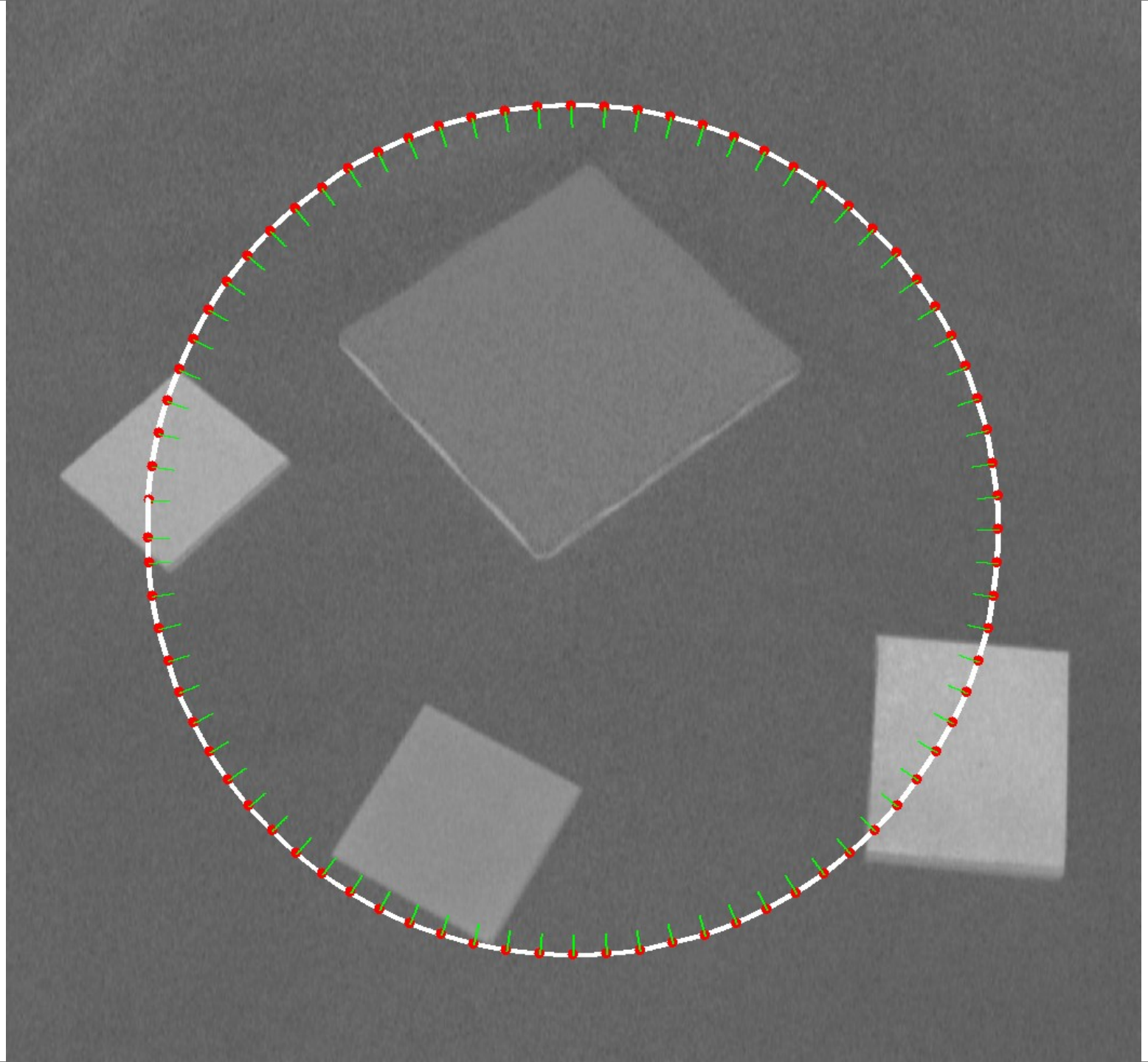


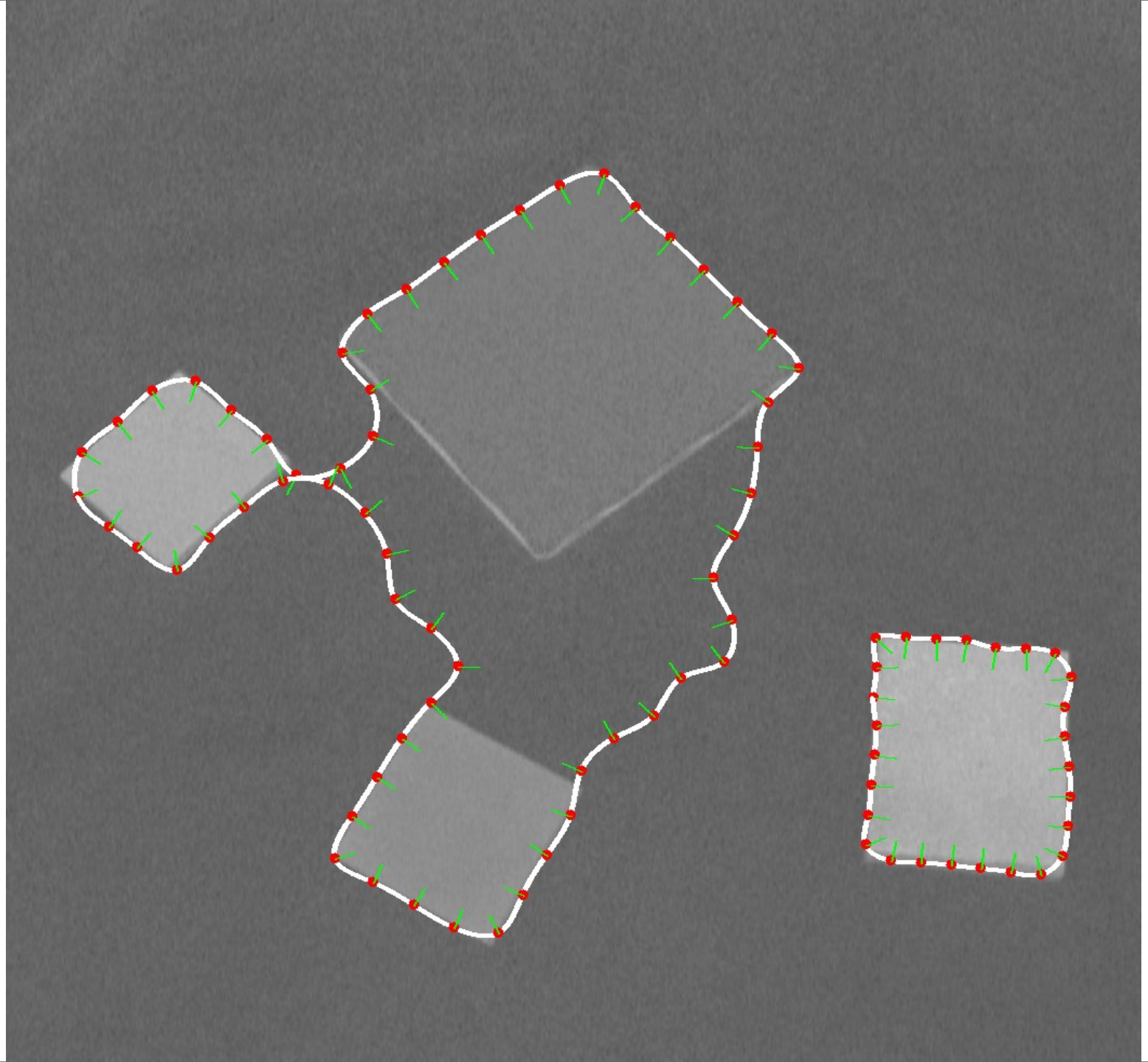
... using LevelSets

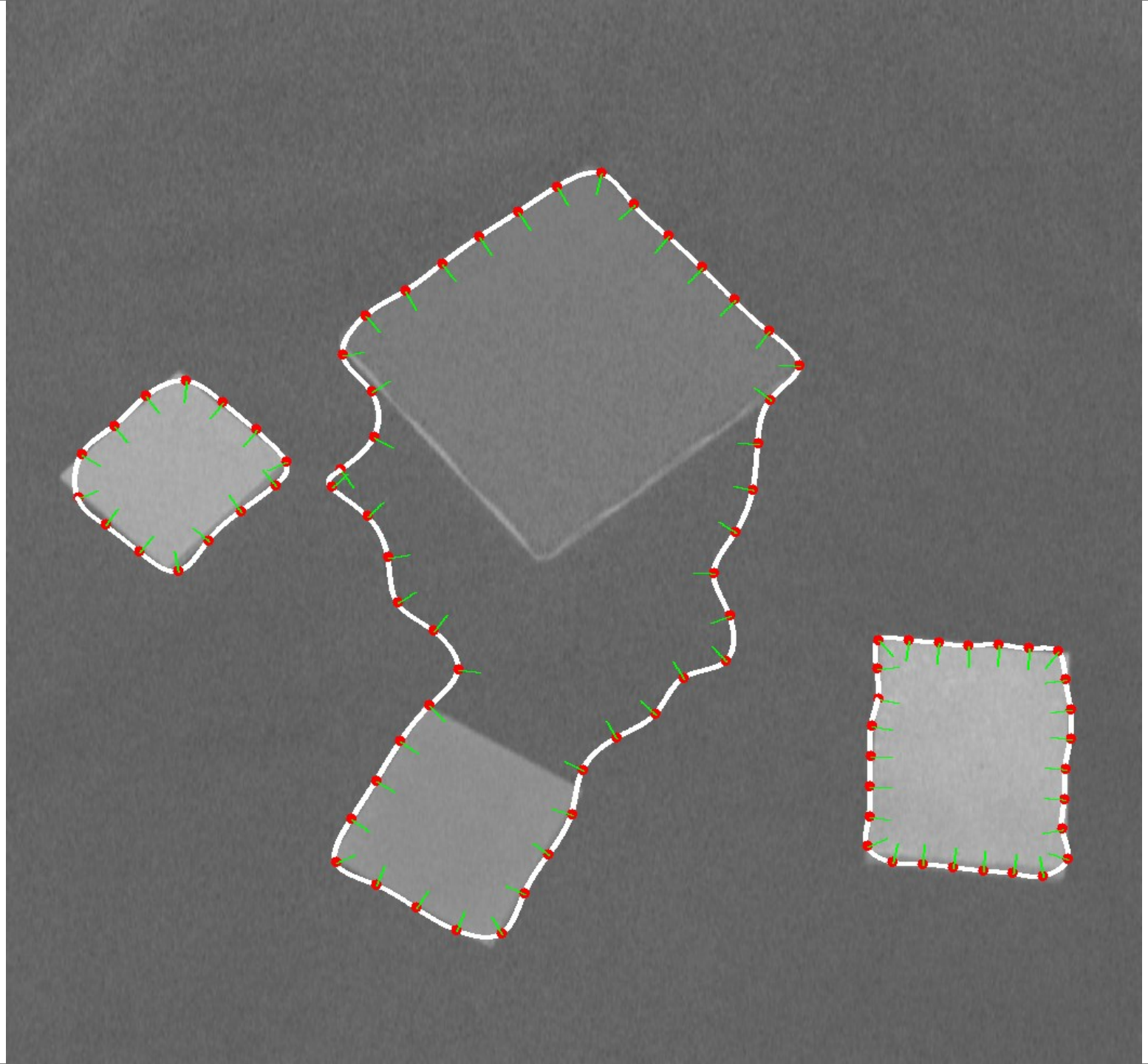


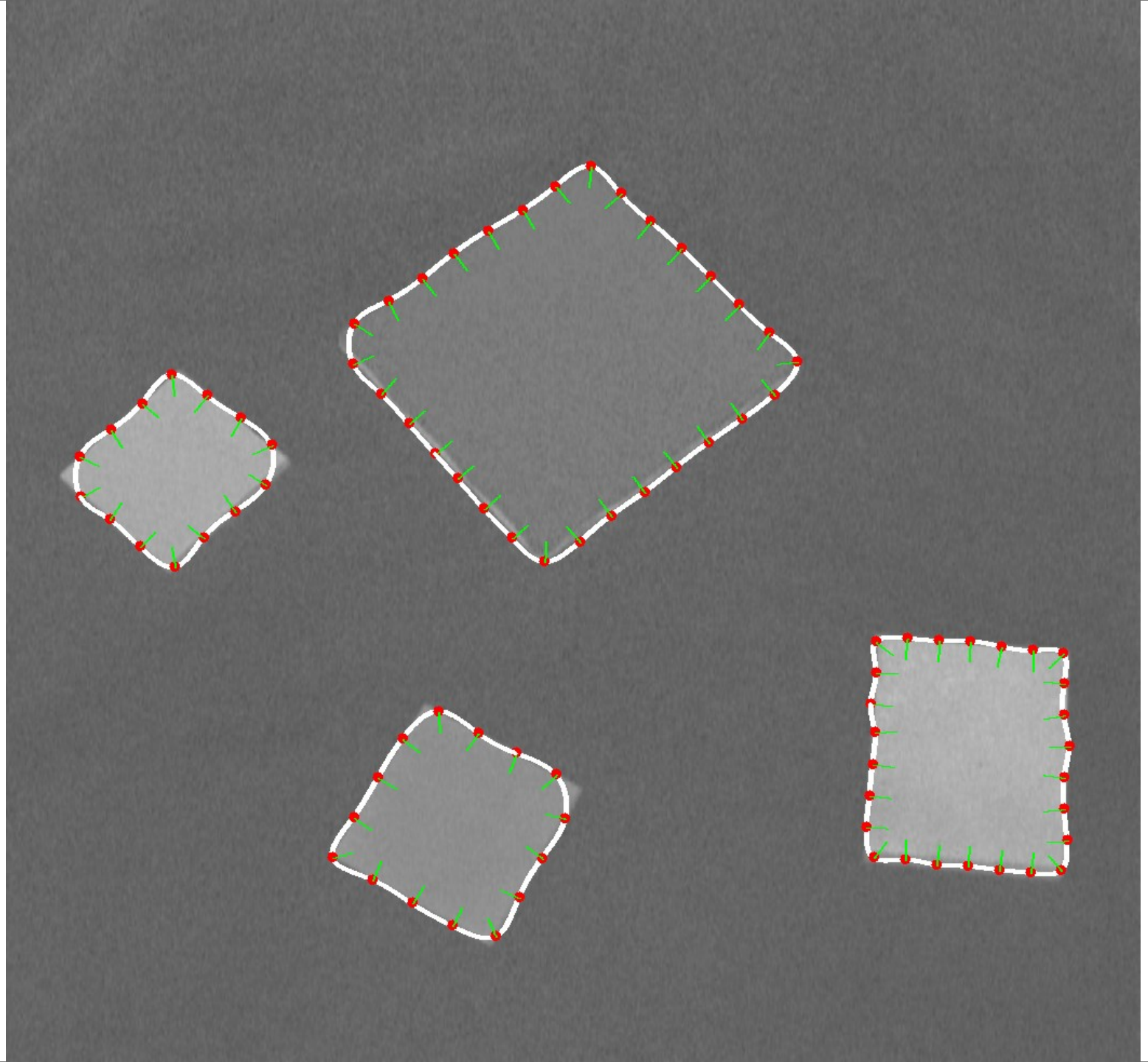




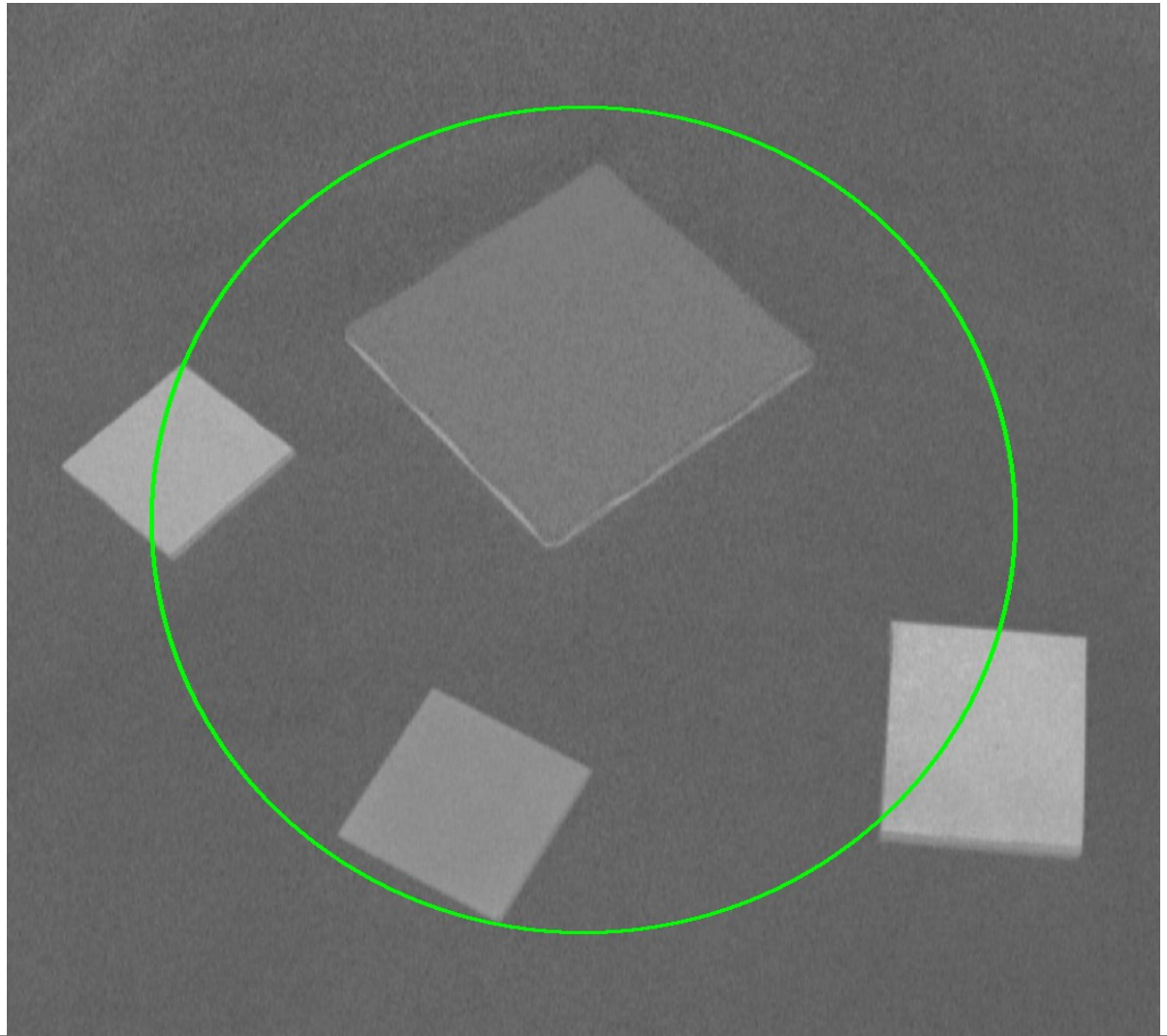


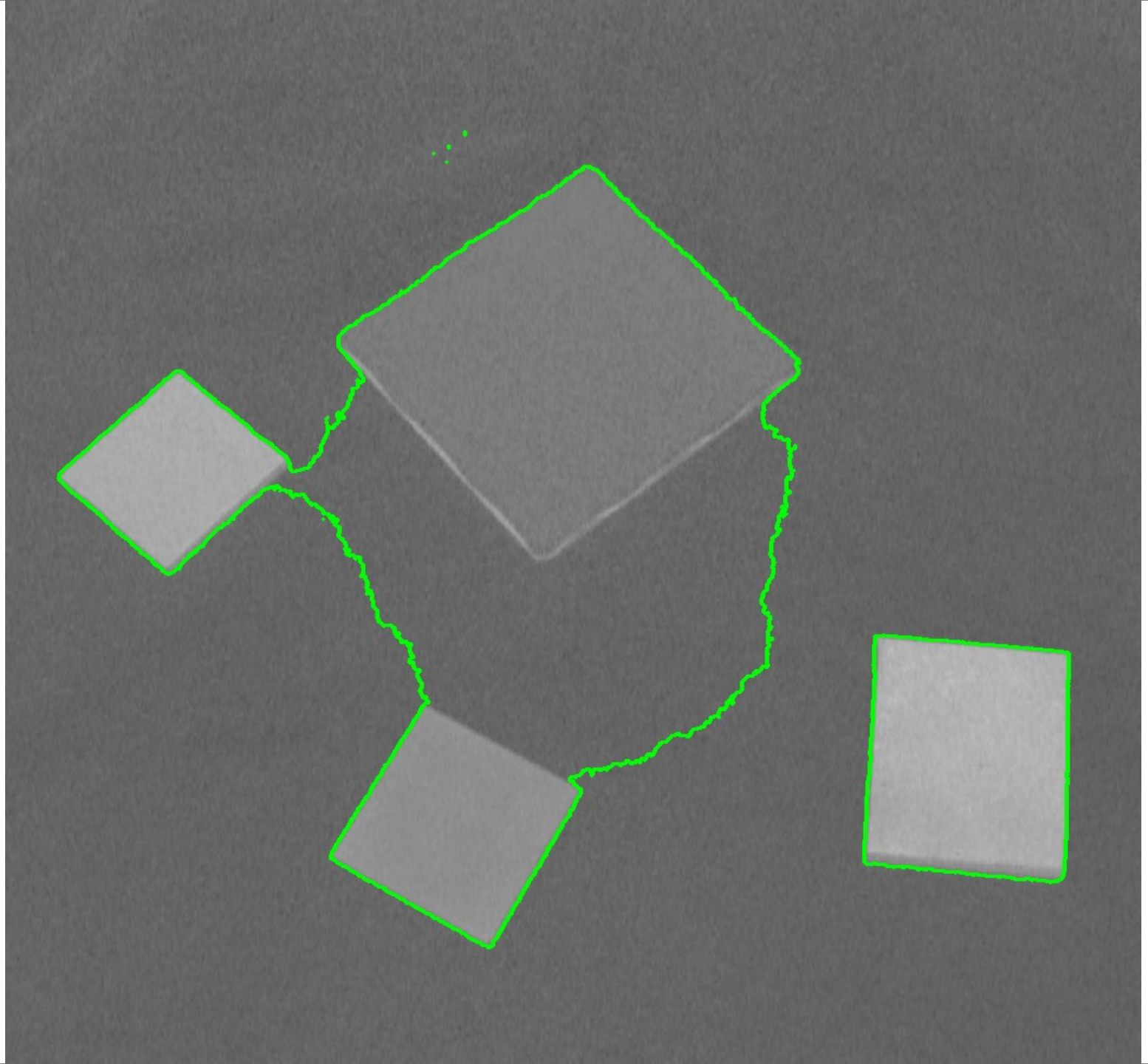


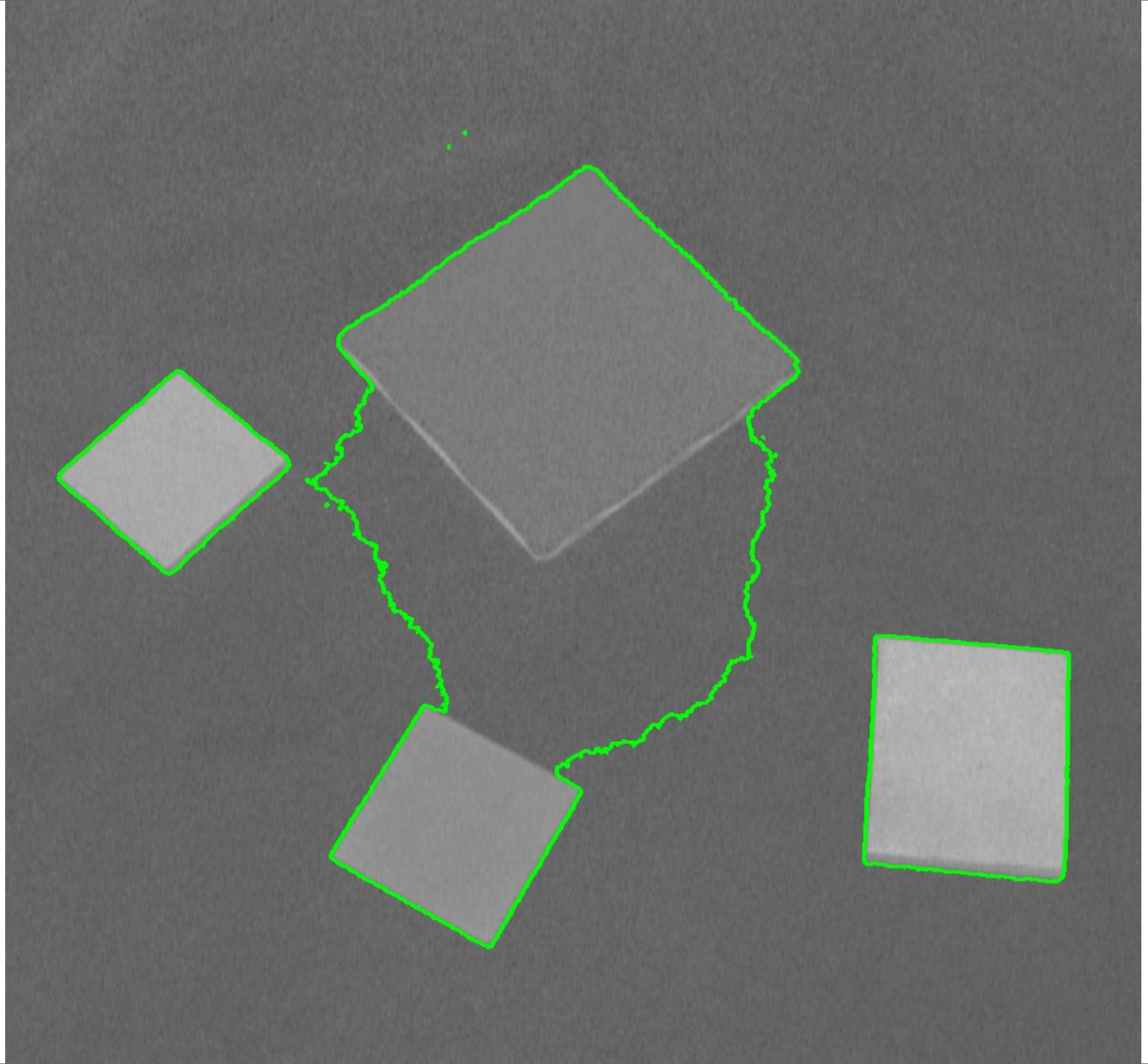




... using LevelSets







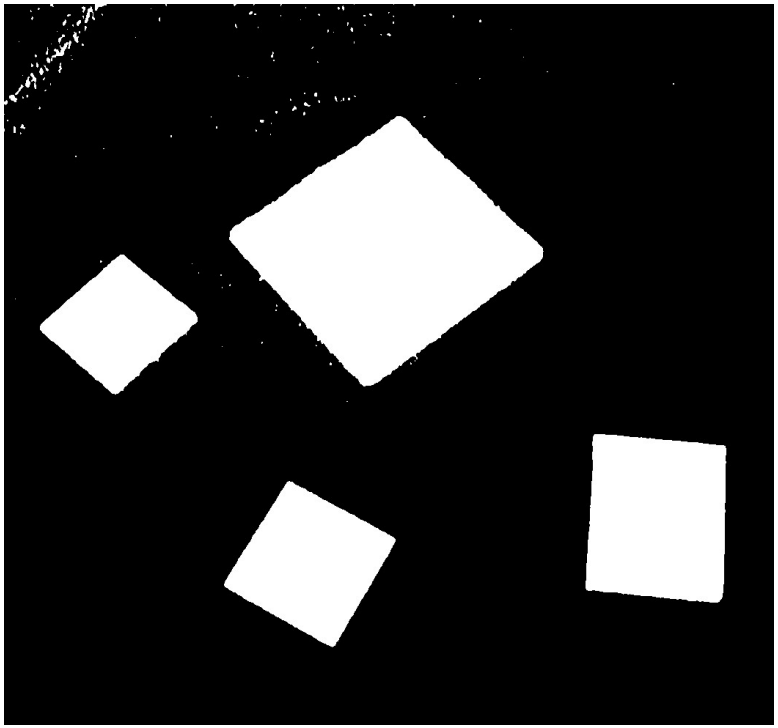
Several objects ....

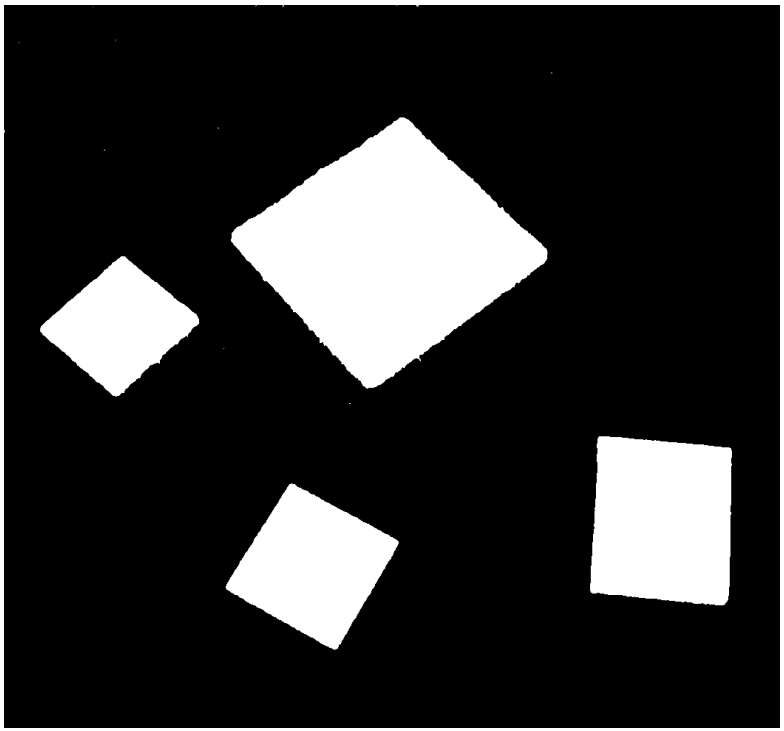
$$F = \lambda_1 (I - \mu_{in})^2 - \lambda_2 (I - \mu_{out})^2 + \lambda_3 \kappa$$

- ... means several curves for the B-Spline representation
- ... but means only one curve for the levelset representation
- Thus, depending on the mean intensity values of the different objects and the background, levelset based methods may fail.

Don't forget simple methods !

- with only 2 iterations



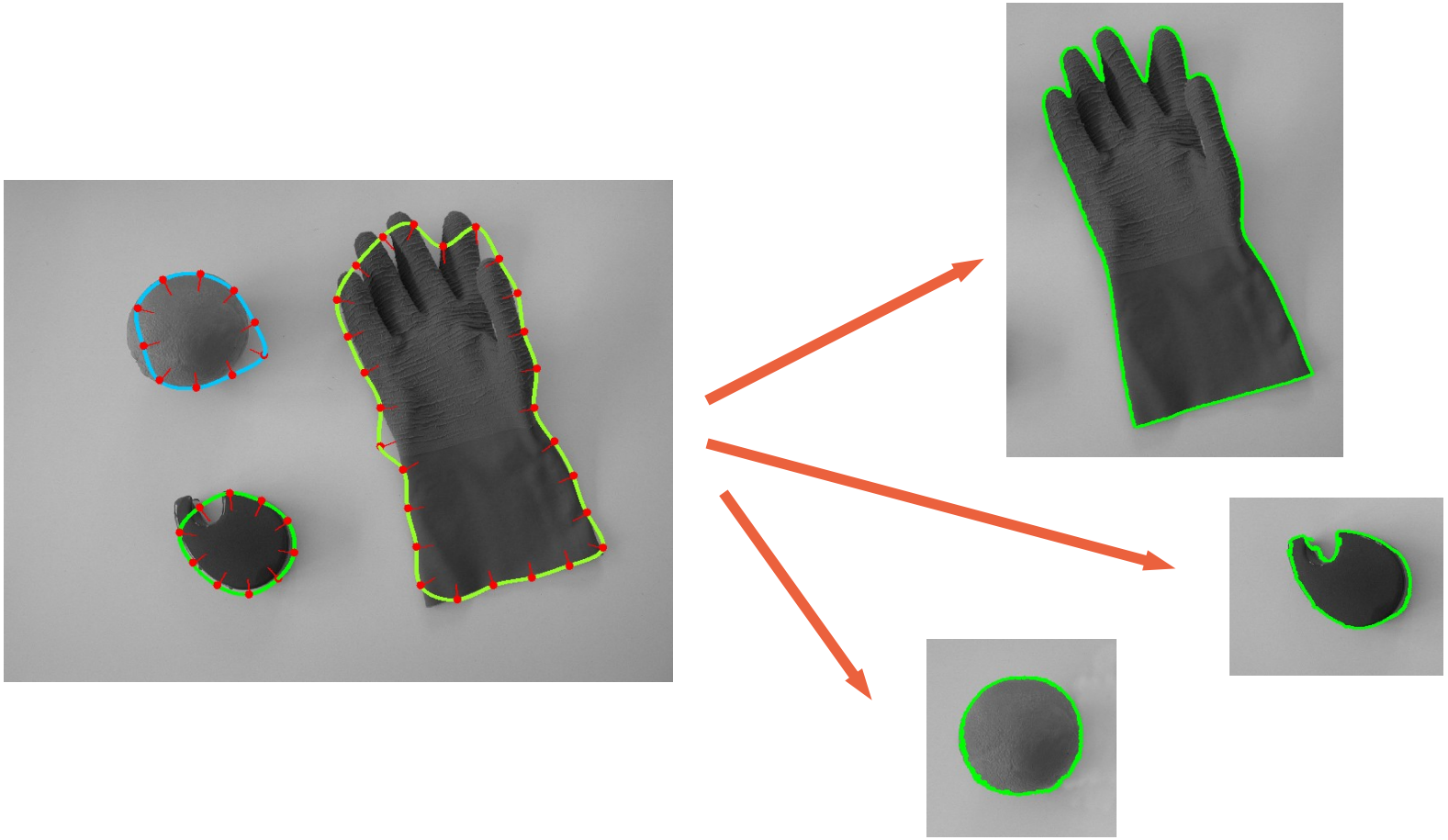


... but how to choose the threshold ?

... in two steps.

- B-splines have difficulties to segment objects with high curvature
- Levelsets are unable to distinguish one region to the others
- Segmentation in two steps:
  - First step: segmenting regions each containing only 1 object using B-splines
  - Second step: refining the result using levelset

2 steps example



Conclusion

- Both methods have drawbacks and advantages
- Combining these methods may improve segmentation
- It is always better to know what you are looking for ...

And with non uniform objects ?

- Gradient

$$\lambda_1(c\kappa - \vec{grad}(c).\vec{n}) \quad \text{with } c = \frac{1}{(1+|\vec{grad}(I)|)}$$
$$\lambda_1 c(\lambda_2 + \kappa) \quad \text{with } c = \frac{1}{(1+|\vec{grad}(I)|)}$$
$$\lambda_1(c\kappa - \vec{grad}(c).\vec{n}) \quad \text{with } c = \frac{1}{(1+|\vec{grad}(I)|^2)}$$
$$\lambda_1 c(\lambda_2 + \kappa) \quad \text{with } c = \frac{1}{(1+|\vec{grad}(I)|^2)}$$

- Variance

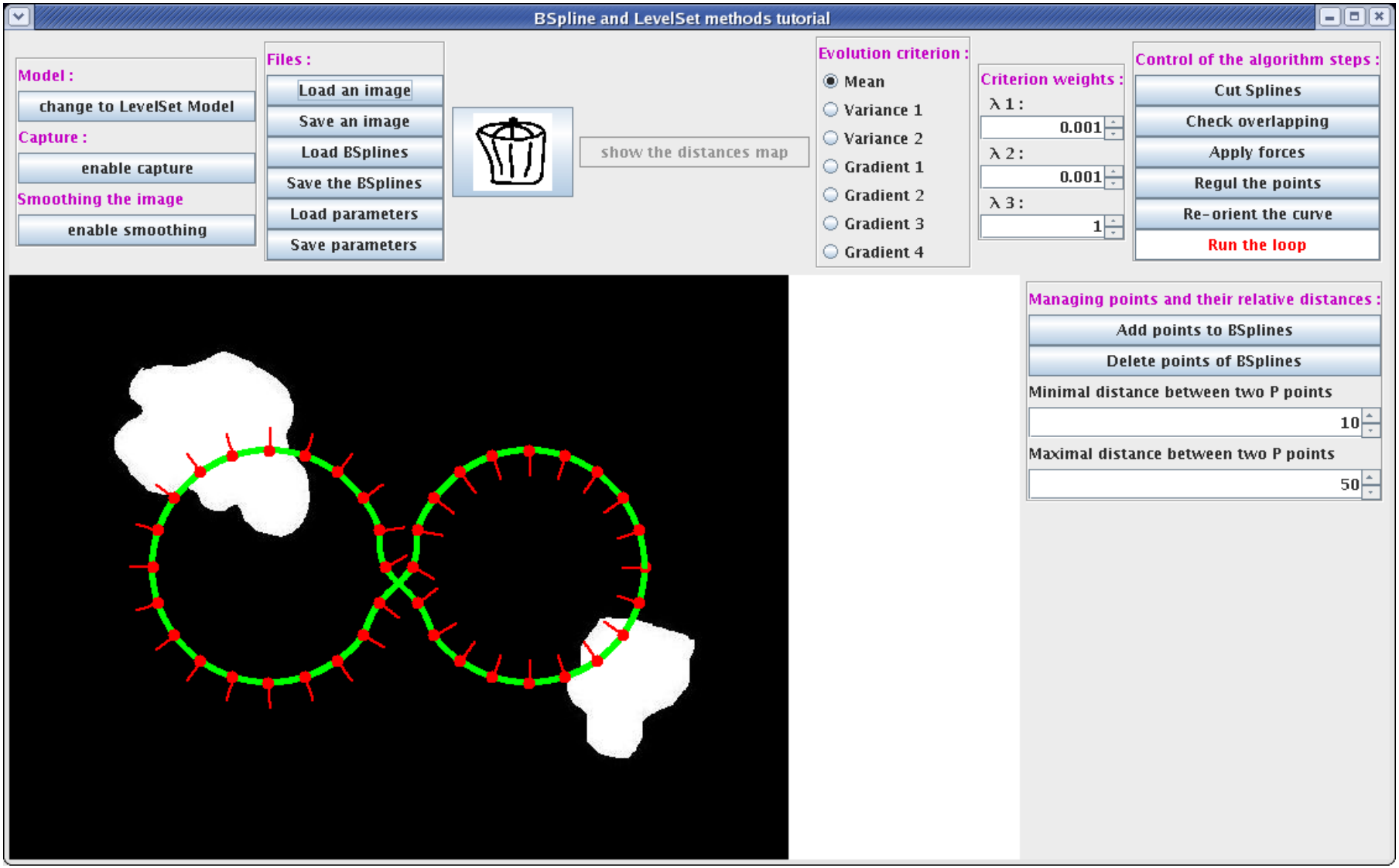
$$F = \lambda_1 \left(\frac{(I-\mu_{in})^2 - \sigma_{in}^2}{1+\sigma_{in}^2} \right) - \lambda_2 \left(\frac{(I-\mu_{out})^2 - \sigma_{out}^2}{1+\sigma_{out}^2} \right) + \lambda_3 \kappa$$

or

$$F = \lambda_1 \left(\log(1+\sigma_{in}^2) + \frac{(I-\mu_{in})^2 - \sigma_{in}^2}{1+\sigma_{in}^2} \right) - \lambda_2 \left(\log(1+\sigma_{out}^2) + \frac{(I-\mu_{out})^2 - \sigma_{out}^2}{1+\sigma_{out}^2} \right) + \lambda_3 \kappa$$

- Histogram
- ...

You want to play with it ?



http://www.i3s.unice.fr/~lingrand/ImageSegmentation.html (GPL)