Graphical models
Globally Optimal Image Segmentation with an Elastic Shape Prior
Final Project Report

Aymane Oubella / Aurélien Houdbert / Mohammed El Aichi

March 2021

1 Abstract

Computer vision has always generated huge interest among researchers and in the entrepreneurial world. Today most of state of the art tasks performed on images, such as image classification or image segmentation, are carried with deep learning algorithms. But deep learning draws attention only since 2012. Image segmentation and shape matching are two tasks very similar and in this work we implemented an algorithm able to perform these two tasks simultaneously based on the work of T. Schoenemann and D. Cremers. We tested our solution on different small shapes and images, testing scaling and translation invariance properties.

2 Introduction

Most of the optimal approaches to image segmentation are based on low level cues such as edges or region statistics. There are very few case in which the corresponding optimization problem can be solved globally such as the approach of Jermyn and Ishikawa [1]. Recently, researchers have suggested to enhance purely low-level segmentation schemes by imposing prior knowledge, favoring segmentation that are in some sense similar to a given shape. Sophisticated shape priors and shape similarity measures typically involve the computational challenge of determining appropriate matches between the points of the segmenting contour and points on a shape template. Such shape alignments are usually computed via dynamic time warping. Yet, determining efficiently a segmentation which globally optimizes both the low-level edge consistency and the similarity to one or more shape templates has so far remained an open challenge. In this project we study an approach in which segmentations and shape alignments can be determined simultaneously, resulting in a globally optimal image segmentation with a translation invariant elastic shape prior. To this end, we will compute cycles of minimal ratio in a large graph representing the product space spanned by the input image and all points of the shape template.

3 Related Work

Most of the work applied in this project was described in the article "Globally Optimal Image Segmentation with an Elastic ShapePrior." [3]. The authors of this article Thomas Schoenemannet and Daniel Cremers have extended their approach to take into account local rotations in their article titled " "Matching Non-rigidly Deformable Shapes Across Images : A Globally Optimal Solution." [4]. The application of the segmentation method described in [3] relies on the understanding of the Minimum Ratio Cycle algorithm described in [1] and [2].

4 Methodology

The problem we are considering is an elastic matching in which we are looking for an optimal way to deform a given shape C into a shape S where both shapes are parameterized by arc length . Formally we are given C : [0, l(C)] -> $\mathbf{R}^2$ and S : [0,l(S)] -> $\mathbf{R}^2$ where l denotes the length of the curve and we are looking for a increasing diffeomorphism m : [0 : l(C)] -> [0 : l(S)] which maps C points into S points. The choice of m is determined given an appropriate measure of the total deformation.

The problem of image segmentation might be tackled through different approaches as stated in [3]. In this project, we will aim to study the image segmenation method proposed by Jemyn and Ishikawa[1]. This method considers the ratio of two line integrals. a problem that might be discretised and reduced to the problem of finding a cycle C in a graph where C^* minimizes :

$$C^* = argmin_C \frac{\sum_{e \in C} n(e)}{\sum_{e \in C} d(e)}$$

where **e** is an edge of the cycle. Each edge **e** is assigned a numerator weight n(e) and a denominator weight d(e) representing a small piece of the respective integral. This optimisation problem might be solved using the Minimum Ratio Cycle algorithm [1]. The key idea is that for a ratio $\tau > \tau^*$ where τ^* is the optimal ratio computed with C^*, then a graph with the same topology and edge weights $w(e) = n(e) - \tau d(e)$ must posses a negative cycle.

the graph construction will be presented later, for now we suppose we dispose of a graph (V, E) where V is the set of vertices (n = #V) and E is the set of edges. we will assign a distance label d(v) and a parent entry p(v) for each node v in the graph, Negative cycles can be found as follow :

```
- initialize the distance of a source node r to 0 and set d(v) = ∞ for all other nodes v
- add the root node r to a queue
- iteration = 0
while queue not empty and iteration < max_iter do
    iteration = iteration + 1;
    v = queue.dequeue();
    for (v,v') in E do
        if d(v) + w(v,v') < d(v') then
            d(v') = d(v) + w(v,v') ;
            p(v') = v ;
        end
        if v' not in queue then
            queue.enqueue(v')
        else
    end
    if iteration mod (α * n) = 0 then
        if negative_cycle_exists then
            break
        end
    end
end
```

the optimisation of the ratio defined above is then as follow :

```
τ = initial_value; (upper bound)
C* = None;
while exist_negative_cycle(graph, τ) do
    τ = negative_cycle_tau ;
    C* = negative_cycle ;
end
return τ, C*
```

4.1 Graph structure

The graph we are building is a product graph. It is the product of the pixel set of the image I we are considering and the set of numbers between 0 and $|S| \times K$ where $|S|$ is the shape length and K the number of maximal representation of each shape point. K's role is to enable the possibility for successive pixels in the image I to be aligned with the same shape point. Therefor each node can be express as $(p, s \times K + l)$ where $l < K$. The graph has a cyclic structure and therefor $(p, |S| \times K) = (p, 0)\ \forall p \in I$

Before defining the edges of the graph we first we need to define specific functions as they did in [3] in order to build the edges' weights :

- $g(.)$ **the edge indicator function** : This edge indicator function is image intensity is denoted I and $\Delta I(.)$ denotes the gradient of the image which is calculated through convolutions of the image with Sobel filters.

$$g(.) = \frac{1}{1 + |\Delta I(.)|}$$

- $alpha_{.}(.)$ **the tangent angle function** : $\alpha_S(s)$ returns the tangent angle at a given point s in the Shape S.
- $\phi(.)$ **returns the angle of a vector** : $\phi(p-q)$ with p and q two pixels will return the angle of the vector $\vec{p}-\vec{q}$

The edges of the graph split into two different categories represented in figure 1. We denote p and q the pixels of the source and the destination node of an edge respectively and s a shape point. The first category corresponds to edges between two node aligned with the same shape point :

$$(p, s \times K + l) \rightarrow (q, s \times K + l + 1)\,, \text{ for } l < K - 1 \text{ and q} \in \mathcal{N}(p)$$

In this case and according to [3], the numerator part of the weight can be expressed as follow :

$$\begin{aligned} n(e) = &\frac{1}{2}|q-p|\,(g(p) + g(q))) \\ &+ \nu|q-p||\alpha_S(s) - \phi(q-p)|^2 \\ &+ \lambda \end{aligned} \tag{1}$$

The second category corresponds to edges that links two different shape points. In this case, we bound the "skip" size to a maximum of K in order to allow variations in the search shape while maintaining the global structure of the prior shape :

$$(p, s \times K + l) \rightarrow (q, (s+i) \times K)\,, \text{ for } l < K,\ 1 \leqslant i \leqslant K \text{ and q} \in \mathcal{N}(p)$$

In this second category and according to [3], the numerator part of the weight can be expressed as follow :

$$\begin{aligned} n(e) = &\frac{1}{2}|q-p|\,(g(p) + g(q))) \\ &+ \nu|q-p| \min_{s \leqslant s' < s+i} |\alpha_S(s') - \phi(q-p)|^2 \\ &+ (i-1)\lambda \end{aligned} \tag{2}$$

In both cases, the numerator weight is composed of three main components : the first one $\frac{1}{2}|q-p|\,(g(p)+g(q)))$ is useful to indicate whether the pixels p and q that we are considering are on an image edge or not. The second one $\nu|q-p||\alpha_S(s) - \phi(q-p)|^2$ quantifies the angle difference between $\vec{p}-\vec{q}$ and the tangent angle at point s in shape S. Indeed ideally we want this different as close to 0 as possible because it indicates which direction to favor to predict next pixel in the detected shape. The third term λ or $(i-1)\lambda$ indicates whether the algorithm should predict q as aligned with the same point shape as p or if it should skip 1 or several point in the same shape. The denominator weight remains the same $d(e) = |p-q|$ in both cases and the total weight of an edge is given by $w(e) = n(e) - \tau d(e)$.

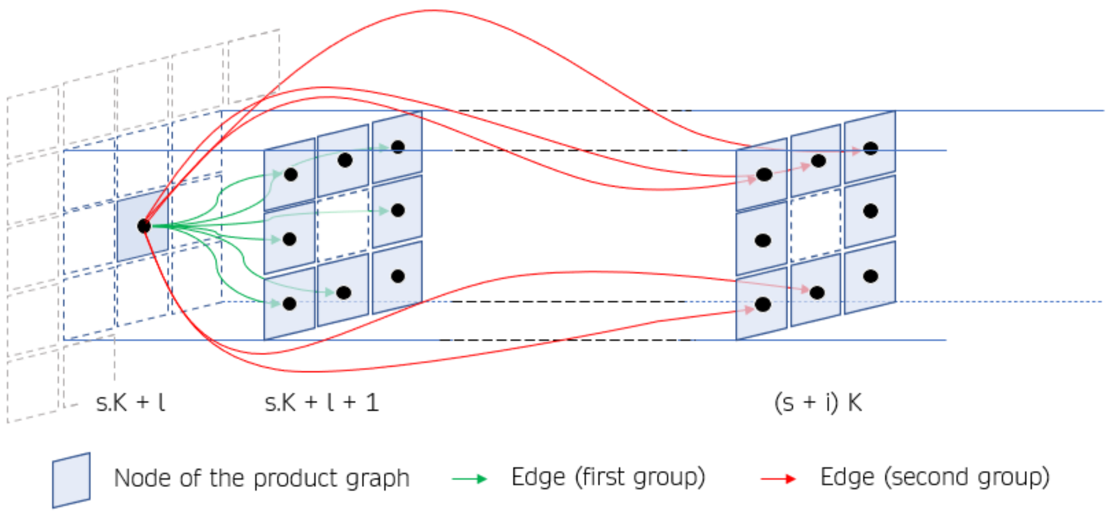

FIGURE 1 – Graph structure representation. Each instance is the size of the image I. **Green** edges are of the first category and **red** edges are of the second category

5 Results

We faced two main issues during this project. The first one is that we had huge troubles finding shape data appropriate to our case and the second one is related to our python implementation of the algorithm that is not optimized enough which didn't allow us to perform tests on large images or very long shapes.

Finally we decided to make experiments with small images (between 10×10 and 20×20 pixels) with different shape templates.

We fixed $K = 3$ $\nu = 0.25$ and $\lambda = 0.25$ in all our experiments. In the figure below, we experimented the performance of the approach through many examples :

- In the first line of figure images, we provided our algorithm with a square shape colored green and an image with full square with same size blue colored. the matching algorithm managed to extract the desired cycle red colored which is the same as the green colored shape.
- In the second line, we used a random shape, a full image with the same boundaries and size and provided them to the matching algorithm which managed this time also to detect accurately the expected cycle.
- In the third line, we used again a square shape, but this time with scaled up image. the expected detection is again accurate
- In the fourth line, we did the same as in the third line yet this time with down scaling. The matching algorithm have not managed to provide the expected cycle, instead it provides a cycle with the good shape (square) but bigger than expected. we expect that this might be impacted with the parameters λ which affects the scaling in the energy minimization.

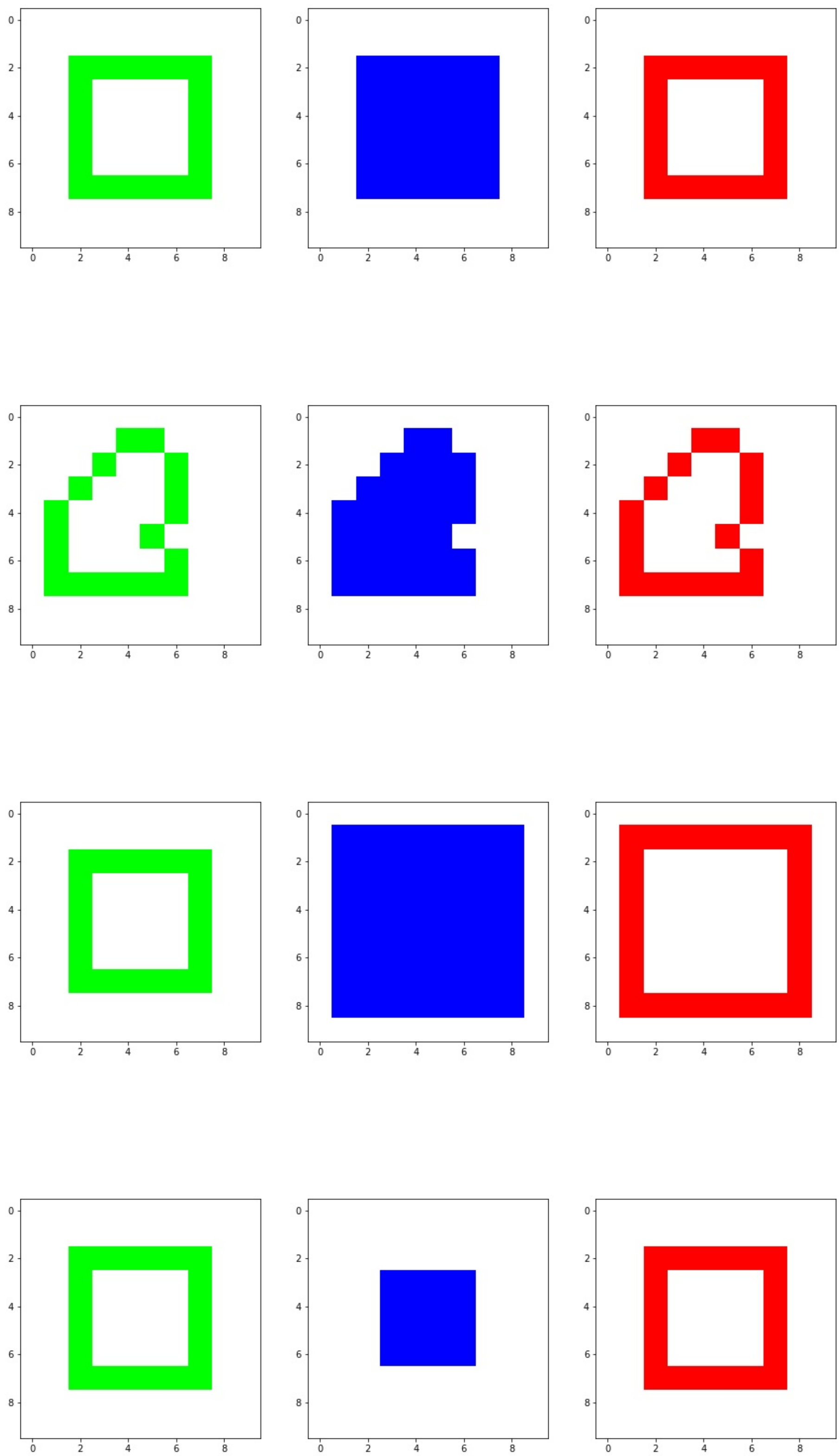

FIGURE 2 – Segmentation of simple shapes in small images of size 10×10. **Left** prior shape we try to reconstruct. **middle** image we are considering. **Right** matching shape retrieved from the image

6 Conclusion

In this project, we've implemented a Graph-theoretic approach able to perform Image segmentation and shape matching based on the work of TSchoenemann and D. Cremers. Our implementation gave satisfying results on the basic small shapes we tested but we were enable to scale our solution to perform segmentation bigger size images and longer shapes for computing time reasons. The graph structure makes the shape matching invariant to translation or up scaling but show defaults on the down scaling task. It would be interesting to test the implementation on long and complex shapes to really measure its performances and observe whether the down scaling error comes from the implementation or from the small size of our tested images.

Références

[1] I. H. Jermyn et H. Ishikawa. "GLOBALLY OPTIMAL REGIONS AND BOUNDARIES AS MINIMUM RATIO WEIGHT CYCLES." In : (2001). URL : http://www.f.waseda.jp/hfs/GORBAMRC.pdf.

[2] Nimrod Megiddo. "Combinatorial optimization with rational objective functions." In : (1979). URL : http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.79.5610&rep=rep1&type=pdf.

[3] Thomas Schoenemann et Daniel Cremers. "Globally Optimal Image Segmentation with an Elastic Shape Prior." In : (2007). URL : https://vision.cs.tum.edu/_media/spezial/bib/shape_iccv07.pdf.

[4] Thomas Schoenemann et Daniel Cremers. "Matching Non-rigidly Deformable Shapes Across Images : A Globally Optimal Solution." In : (2008). URL : https://vision.in.tum.de/_media/spezial/bib/deform_cvpr08.pdf.