Level Set Method

Yuanfeng Shi, 1800010697

November 4, 2019

1 Level Set Method

Let us first take a look at the curve evolution equations:

$$\begin{cases} \frac{\partial c}{\partial t} = FN \\ c(0,q) = c_0(q) \end{cases}$$

A key observation is that a curve c can be seen as the zero level set of some function u in higher dimension:

$$u(t, c(t,q)) = 0,$$

$$\frac{\partial u}{\partial t} + \left\langle \nabla u, \frac{\partial c}{\partial t} \right\rangle = \frac{\partial u}{\partial t} + \langle \nabla u, FN \rangle = 0.$$

$$\frac{\partial u}{\partial t}(t, c(t,q)) = F|\nabla u(t, c(t,q))|.$$

2 Active Contour

The active contour method aims to minimize the energy of the curve:

$$J(c) = \int g(|\nabla I(c(q))|)|c'(q)|dq.$$

First we take a look at the flow of the Geodestic active cintours model:

$$\frac{\partial c}{\partial t} = (\kappa g - \langle \nabla g, N \rangle)N.$$

Here κ is the curvature of the curve. The drawback of the original model is that it is hard to detect nonconvex objects. In order to let the curve cling to the edge, we propose an improved equation to plug the contour into the nonconvex region:

$$\frac{\partial c}{\partial t} = (\kappa g - \langle \nabla g, N \rangle + \alpha g)N.$$

We solve this method using level set method, suppose u is the level set function, then we induce the evolution equation of the u:

$$\frac{\partial u}{\partial t} = g(|\nabla I|) \left(div \left(\frac{\nabla u}{|\nabla u|} \right) + \alpha \right) |\nabla u| + \langle \nabla g, \nabla u \rangle. \tag{1}$$

3 Convexified CV

3.1 Model

We list our energy model, which is slight different from the formula in the SLIDE (the limit of u):

$$\min_{|u|\leq 1} \int_D |\nabla u| + \lambda \int_D \{(c_1 - f(x))^2 - (c_2 - f(x)^2\} u(x).$$

3.2 Algorithm

We set $\boldsymbol{d} = \nabla u$ and use the augmented Lagrangian method to solve the constrained problem:

$$\min_{\boldsymbol{d}, u \in [-1,1]} \|\boldsymbol{d}\|_1 + \mu \langle u, r \rangle \quad s.t. \;\; \boldsymbol{d} = \nabla u.$$

We obtain the following algorithm:

$$\begin{cases} (u^{k+1}, \boldsymbol{d}^{k+1}) & = argmin_{u\in[-1,1],\boldsymbol{d}} \|\boldsymbol{d}\|_1 + \mu\langle u, r\rangle + \frac{\lambda}{2}\|\boldsymbol{d} - \nabla u - \boldsymbol{b}^k\|_2^2 \\ \boldsymbol{b}^{k+1} & = \boldsymbol{b}^k + \delta(\nabla u - \boldsymbol{d}^k) \end{cases}$$

We alternatively optimize the u and $\boldsymbol{d}$. In order to avoid the near singularity of $\nabla^T\nabla$ when solving it using FFT, we consider replace $\nabla^T\nabla$ by $\nabla^T\nabla + \epsilon I$, where I is the identity operator.

The whole algorithm is shown in Algorithm

Algorithm 1 $u = ms(f, lambda, outeriter, inneriter)$

```
Initialize the level set function $u$.
for $i$=1:outer iter/inner iter do
    Update two area $u > 0$ and $c < 0$.
    Calculate the mean value of $f$ in two area: $c_1$, $c_2$.
    for $j$ = 1:inner iter do
        $u = (\Delta^T\Delta + \epsilon I)^{-1}(\Delta^T(\boldsymbol{b} - \boldsymbol{d}) - \mu r)$
        $\boldsymbol{d} = \mathcal{T}_{1/\lambda}(\boldsymbol{d})$
        $\boldsymbol{b} = \boldsymbol{b} + \delta(\nabla u - \boldsymbol{d})$
    end for
end for
```

4 Numerical Aspect

4.1 Result: Active contour

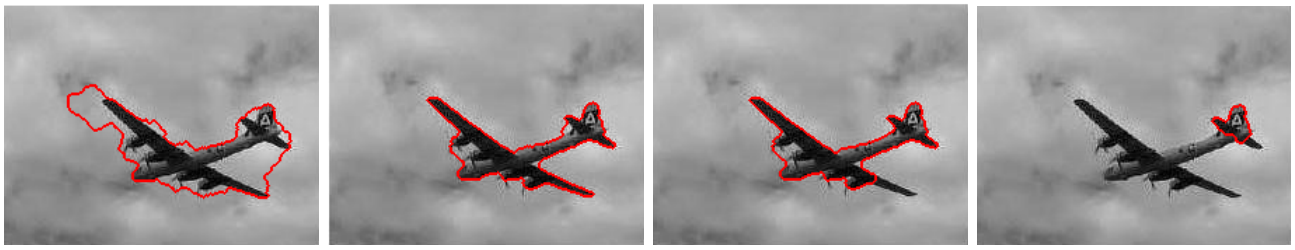
We first show the result of the active contour method. All the experiment uses the same initialization method and share the same reinforce iteration time 20.

First we discuss the different choice of

$$g(s) = \frac{1}{1 + k|s|^2},$$

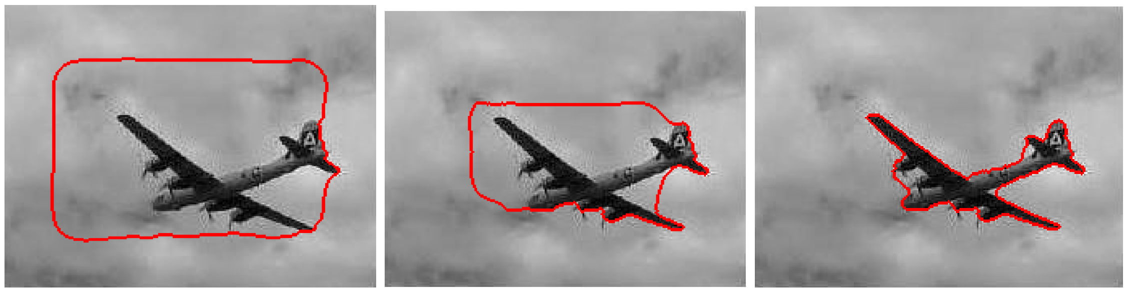
where k is the only parameter to control the stiffness of g. Qualitatively, when k is larger, the gradient is small hence negligible in the whole image. Thus the active contour method will fill in the sense all the level set function will be positive eventually, and in consequence the zero level set vanishes. When k is small, the evolution will be stuck at some smooth area and thus fail.

Figure 1: The affects of the choice of k: k=0.1,0.01(optimal),0.005,0.001



Then we show the results of different iteration time.

Figure 2: The result of the active contour: iter _ time = 1000,2000,4000, $g(s) = \frac{1}{1+0.01s^2}$



The effects of noise and blurring: When the noise size become larger, the g is supposed to chose more stiff to make the evolution faster.

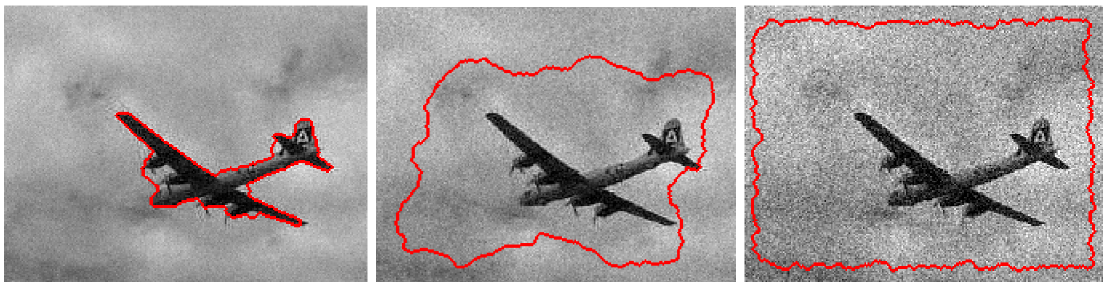
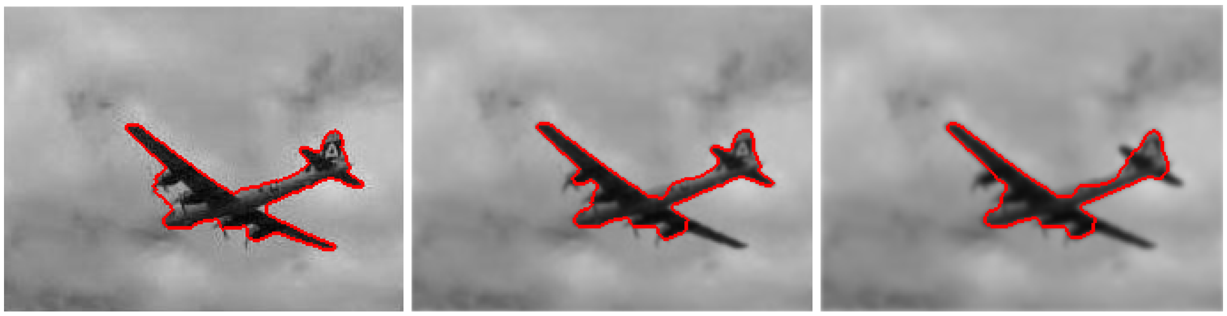


Figure 3: Different Gaussian white noise size affects the segmentation result. Noise size = 0.03, 0.05, 0.1.

Figure 4: Different blur kernel size affects the segmentation result. Sigma = 0.5, 1, 2. Kernel_size = 15



At last we show some other excellent results.

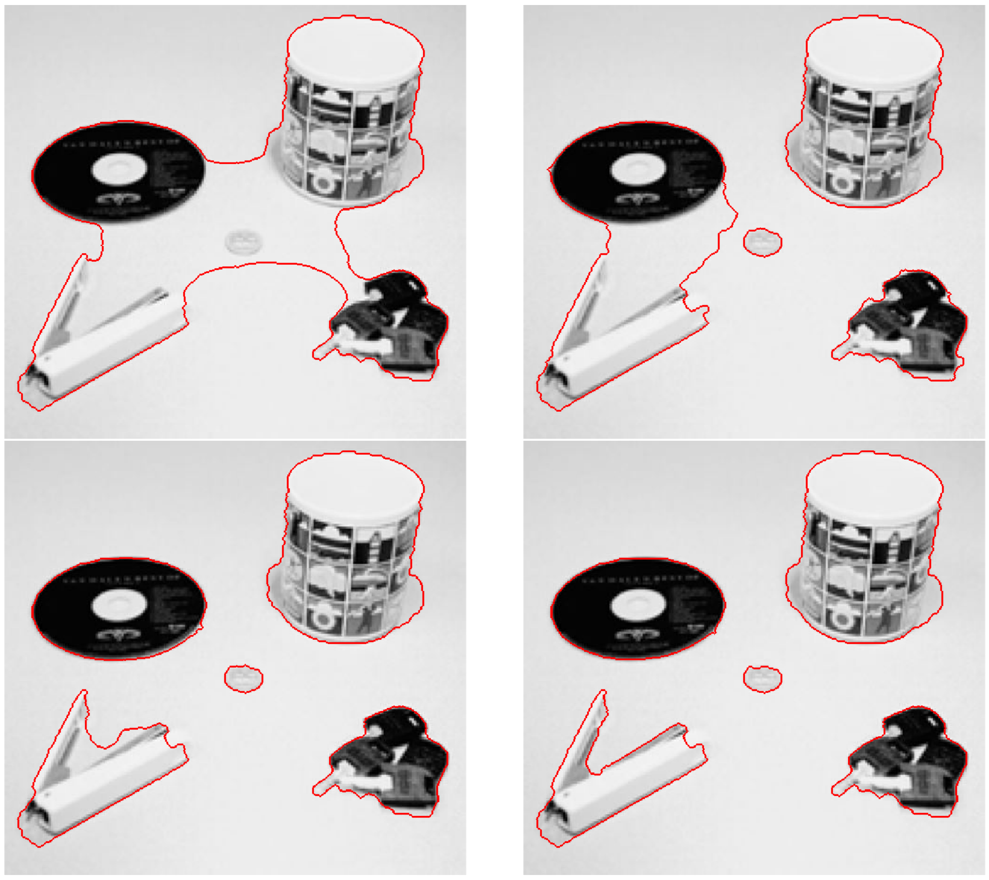
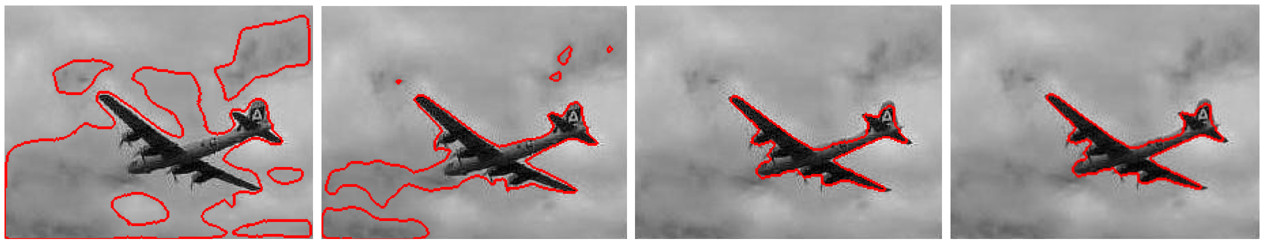


Figure 5: The result of the active contour: iter_ time = 8000,12000,16000,20000, $g(s) = \frac{1}{1+0.5s^2}$

4.2 Result: Convexified CV model

Now we focus on the result of the relaxed MS model. For simplicity, all the experiment uses the same initialization method and share the same inner iteration time 20. We first observe that Convexified CV model take shorter iteration time to stop.

Figure 6: The result of the active contour: iter _ time = 1,5,20,100



Besides, CV model show more robustness under different noise and blur settings. For Gaussian white noise, we implement it by "noise_size * randn(size(f))". For blurring, we use "fspecial('gaussian', 15, sigma)" as our blur kernel.

Figure 7: Different Gaussian white noise size affects the segmentation result. Noise size = 0.05, 0.1, 0.2.

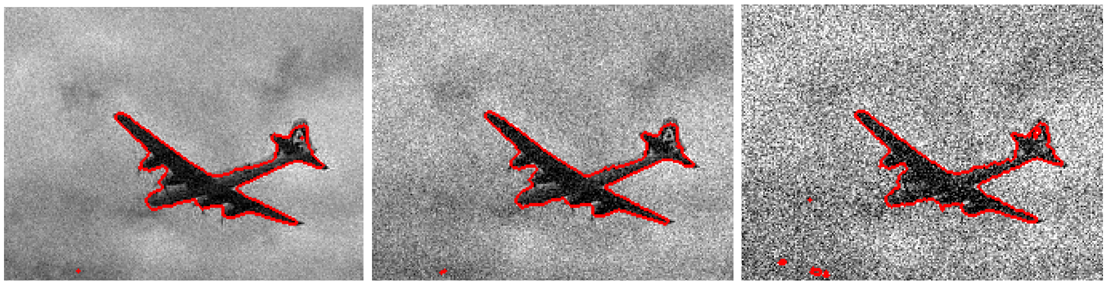
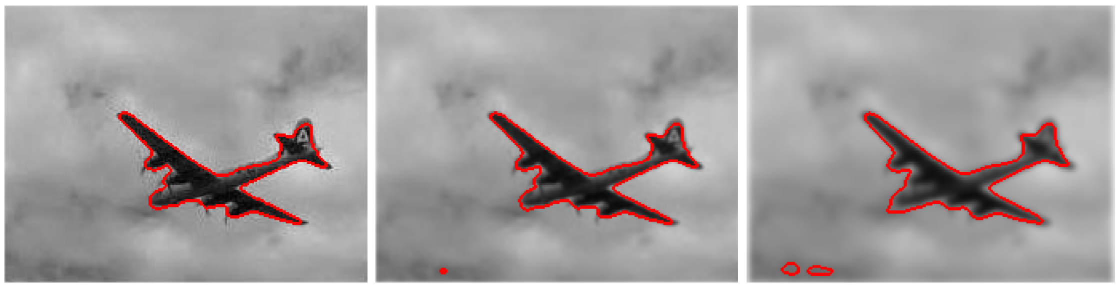
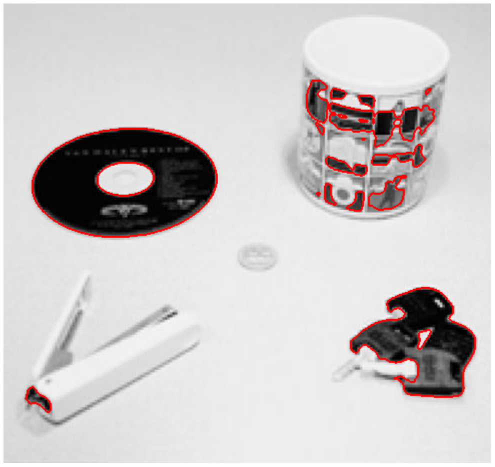


Figure 8: Different blur kernel size affects the segmentation result. Sigma = 0.5, 1, 2.





However, this method will fail to obtain the true result in some pictures.
We conclude that CV model cannot deal with object segmentation when the diversity of color in objects is high.