Monte Carlo Method in Image Segmentation

Zhongkai Shangguan

December 18, 2019

Abstract

Image segmentation is an attractive and challenging domain in digital signal and image processing. Segmentation techniques are valuable tool in a lot of fields: it can help people extract information they would like to focus on. For example, segmentation techniques are often used in medical images to help diagnostic for cancer tumors and cysts; in industry field, it can improve the efficiency in product quality inspection and defects detection. In this report, we detailed two papers I selected in image segmentation domain, discussed their similarities and difference of their algorithm, and finally proposed some possible future work.

1. Introduction

Image segmentation is to partition the image into different meaningful regions, within which the measurement values are relatively homogeneous. However, segmenting low quality images, especially for medical images, or with missing data, is still a challenging problem. This is what [1, 2] aiming to solve and the reason why I choose these two papers.

Historically, many algorithm models for image segmentation and edge detection have been proposed. [4] raised an algorithm called active contours model, also called snake, which can extract contour lines of objects from 2D images that contain noise. Both [1, 2]using active contours algorithm in their models. [5] proposed an algorithm that uses the linear system state equation to optimally estimate the system state through the system input and output observation data, which called Kalman filter. Based on Monte Carlo method, [7] developed particle filter. Particle filter is a generalized method of Kalman filter. Kalman filter is based on linear state space and Gaussian noise; while the state space model of particle filter can be nonlinear and noise. Therefore, particle filter is widely used in medical images as most medical images have a multimodal distribution. Then, a robust particle filtering algorithm for contour following is developed in [8], which is called JetStream. This algorithm is demonstrated in the context of the interactive cut-out in photo editing applications. JetStream is a universal tool for designing contour tracking algorithms in different application areas. Designers are free to choose the appropriate task-oriented elements: dynamics and measurement models, likelihood ratios or likelihood ratios and constraints. [9] raised an interactive model for image segmentation, it uses a parametric model for an implicit representation of the segmenting curve by applying principal component analysis to a training data set. Based on the prior information of the image, [10] proposed nonparametric methods by extending density estimation to the space of shapes to segment images with multimodal shape densities.

Previous discussion introduced main algorithms referenced or used in [1,2], and these two papers extending and combining the above methods, and proposed new algorithms based on their own ideas to obtain better accuracy in image segmentation and apply to different scenarios. This report will discuss their common ideas using the above method in section 2.

Then the report will describe the two papers in section 3 and section 4 respectively. In section 5, we will talk about the similarities and differences between [1] and [2], then evaluate their respective strengths and weaknesses. Finally, the report will give a summary of what was discussed in these two papers and forward some further work that may be carried out in the future.

2. Common Ideas

Monte Carlo Method

Monte Carlo method is not the name of a specific algorithm, but a summary of the characteristics of a class of random algorithms. It was proposed by John von Neumann and others in the 1940s[11]. This method is a numerical calculation method guided by the theory of probability and statistics, it is a method that uses random numbers to solve many computational problems.

The characteristic of Monte Carlo method is that the approximate result can be calculated on random sampling. As the number of samples increases, the probability of getting the correct result gradually increases, but before the real result is obtained, it was impossible to know whether the results obtained so far were real.

In general, the Monte Carlo method can be roughly divided into two categories: one is that the problem itself is inherently random, and this random process can be directly simulated by the computing power of a computer. Another type is that the problem to be solved can be transformed into some randomly distributed feature number, such as the probability of a random event, or the expected value of a random variable. By random sampling, the probability of a random event is estimated, or the digital characteristics of a random variable are estimated using the sampled digital characteristics as the solution to the problem. In this report, the Monte Carlo method we are talking about is the second type.

Active Contours

Active contours[5] can extract contour lines of objects from 2D images that contain noise.

The specific method includes three steps. Firstly, create an initial curve in the image, the shape is free, but the outline of the target object needs to be wrapped inside. Next, the "energy equation" is established, including the "internal energy" for the purpose of the standard curve shape, and the "external energy" for how close the standard curve is to the contour of the target object. In the calculation process, minimizing the internal energy can make the curve continue to tighten and keep smooth; while minimizing the external energy can make the curve continue to be close to the contour of the target object until it is consistent.

The energy function of the active contour is the sum of its external energy and internal energy, defined in equation (1).

$$\mathrm{E}^{*}_{\mathrm{snake}} = \int_0^1 E_{snake}\big(v(s)\big)ds = \int_0^1 \Big(E_{internal}\big(v(s)\big) + E_{image}\big(v(s)\big) + E_{con}\big(v(s)\big)\Big)\,ds, \quad (1)$$

This method has some limitations. It relies on the calculation of image energy, so when the background color of the image is very close to the object to be segmented, it is difficult to segment the foreground and background well. *Figure 1* shows an example using active contour. Left image is segment images with a high level of noise, and the image on the right indicate that this method can not do perfectly segmentation when some part of the object is similar to the background.

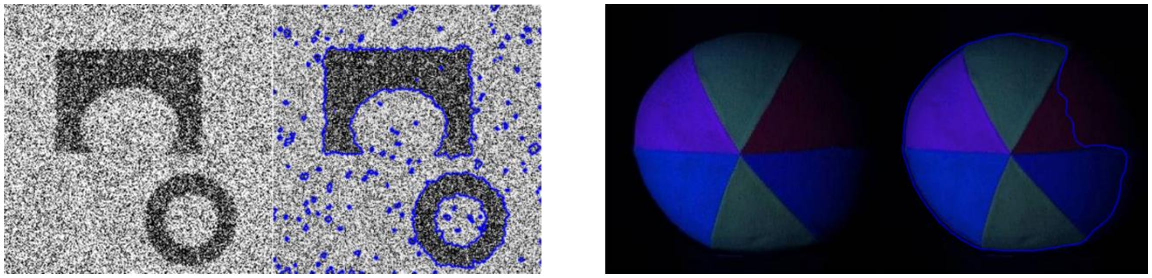

Figure 1. Left: segment images with a high level of noise; Right: images with similar object and background.

3. "Contour segmentation in 2D ultrasound medical images with particle filtering" [1]

In section 1, we introduced some algorithms for image segmentation, including particle filter and JetStream algorithm. Paper 1 extends some of the capabilities of JetStream and developed a new segmentation algorithm for ultrasonic images.

Overall Process

The whole process can be divided into three steps.

A. Pre-processing

Because medical images are always noisy and not clear, so we must use some filters to increase the quality of images. In image segmentation, a fundamental requirement of the noise filtering method is to preserve the important feature for object boundaries. The author used a non-linear Gaussian filter and edge preserving diffusion to adopt the original image.

B. Multiple Model Particle Filter

In this paper, author realized a multiple model particle filter (MMPF) to do segmentation, and each particle represents a point around the contour. Within one recursive of particle filter, each particle updates its weight, and based on the current state to predict the moving space for the next sampled particle. In this algorithm, each particle can represent a particle filter, so it is called multiple model particle filter.

C. Smoothing

The particles we sampled as the candidate of the contour are discrete, and if the amount of the particles is small, the contour we segmented is not smooth. So the last step is using a standard moving average to smooth the contour.

Stage of Algorithm

A. Prior Information.

At the beginning of the algorithm, people should manually select 4 points, $\mathrm{x_{seed}}$, $\mathrm{x_{start}}$, $\mathrm{x_{min}}$, and $\mathrm{x_{max}}$. $\mathrm{x_{seed}}$ should be inside the object we are aiming to segment; $\mathrm{x_{start}}$ should be the first sample point which on the contour, $\mathrm{x_{min}}$ and $\mathrm{x_{max}}$ defined the gate of the contour: we will use the seed point and min point to generate a circle, with radius r, and the distance between $\mathrm{x_{seed}}$ and $\mathrm{x_{max}}$ as the semi-major axis of ellipse, called d, then define the length of semi-minor axis is $\frac{2}{3}d + r$. The contour should between the area of the circle and ellipse. This four points manually give the prior information to guide the area we are interested in.

B. Particle Filter

A particle filter is a recursive filter using the Monte Carlo method. It uses a set of weighted random samples (called particles) to represent the posterior probability of a random event. Noise or incomplete observation sequences can estimate the state of a dynamic system.

The flowchart of particle filter is shown in *Figure 2*. Usually, at the beginning, we will initial equal weights for each pixel in the image, and then sample the first particle. In our case, the first particle is manually given by the start point, and weights for area outside the gate area should be zero. After that, we will update the weights of particles based on the previous measurement space. It is also important to normalize the weight in case gradient exploding problem. Then we will measure current state in measurement space, and calculate the expectation for likelihood, and define whether to accept the new weights. If yes, we will resample the particles based on the new weights; or, we will keep the current state as the next state.

C. Measurement Space

Measurement space is very important as it is used to determine whether accept candidate state or not, and it is also used to judge if the model converge. In [1], the measurement space considered the density of the pixels and the gradient of the image, but it is also possible to define different measurement criteria. In measurement space, the likelihood is computed for each particle point, situated inside and on the boundaries.

Denote by $\mathrm{p_{on}} \coloneqq p_{on}(y(x_k)|x_{0:n})$ the likelihood of the pixel $\mathrm{x_k}$, if it belongs to the contour. Denote by $\mathrm{p_{off}} \coloneqq ff(y(x_k)|x_{0:n})$ the likelihood of the same pixel $\mathrm{x_k}$, if it does not belong to the contour. Then, the likelihood ratio can be written as (2), extracting useful information from the image data.

$$\ell = \frac{\mathrm{p_{on}}}{p_{off}}, \;\; \ell \propto p\big(y(x_k)\big), \tag{2}$$

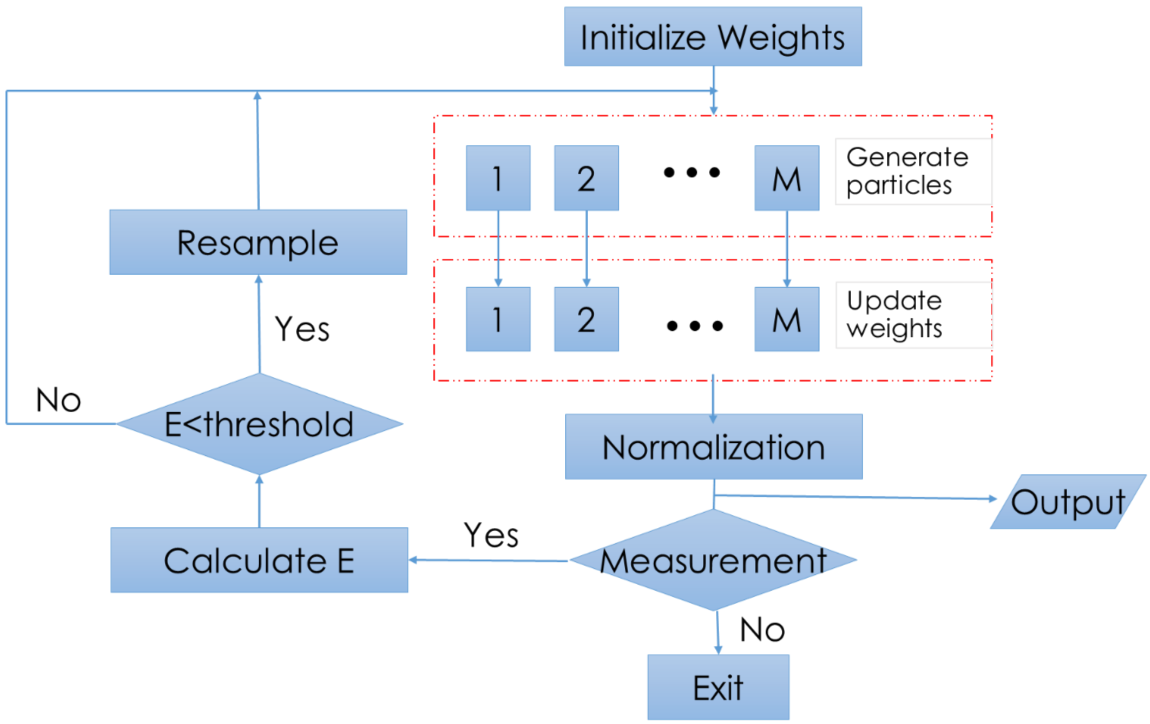

Figure 2. The flowchart of particle filter.

Example and result

Figure 3 is an example from [1] in order to explain the process more intuitive.

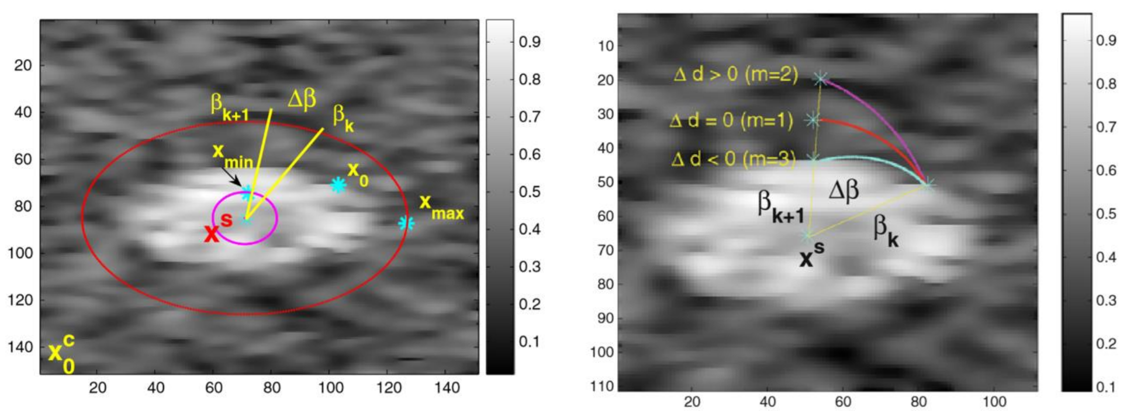

Figure 3. Left: An ultrasound image with a center of the contour Xs. Right: The distance increments for different modes. [1]

For the left image, $X^s, x_o, x_{min}, x_{max}$ are seed point, start point, and gate points respectively, Δβ defined how many particles we want to sample. Current state is described as state k, and the next state should be k+1. We will predict the next particle state based on our measurement space, then the particle in current state may have three possible actions: longer, shorter, keep the same, as shown in the right image. If Δd for all particles are near zero, then that means the overall model converge.

Figure 4 is two results shown in [1], remember that smoothing in the last step, which create the blue line and yellow line respectively.

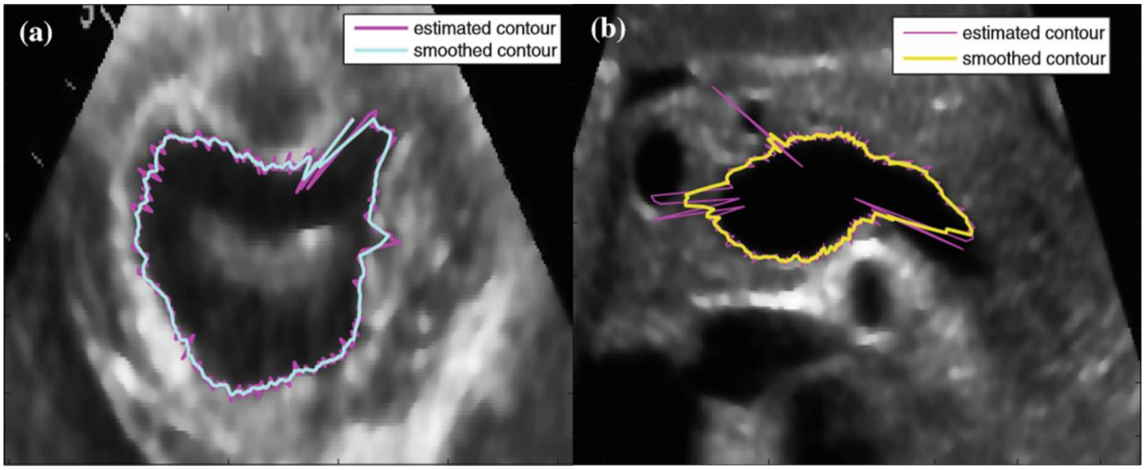

Figure 4. Two results generated using MMPF. [1]

Contributions

1. Proposed a multi-model structure that captures the previous state space and controls the growth process of the predicted contour;
2. A combined likelihood is proposed, which involves the projection along the x, y axis and the contour from the seed point to the contour intensity gradient of the radius;
3. Considering the contour convexity in conjunction with the constraints.

4. "MCMC Shape Sampling for Image Segmentation with Nonparametric Shape Priors"

In section 3, we discussed the particle filter, but there exists a main problem that a large number of samples are needed to approximate the posterior probability density of the system. The more complex the environment the particle faces, the more samples are required to describe the posterior probability distribution, and the higher the complexity of the algorithm.

Also based on Monte Carlo method, [2] proposed an improvement method using Markov Chain to solve low-quality images with high level noise or occluded objects. This model can also do multi-classes segmentation. The code is opensource, and the dataset using by this paper is Aircraft Benchmark [13] and the MNIST database of handwritten digits [12].

Overall Process

The main process contains 3 parts.

A. Initial Curve

In this paper, the prior information is given by a seed point inside the object people are interested in of the image. Then, based on the given point, evolve a curve which is used as the first sample curve.

B. Data-loader

The second part is construct level set representation of shapes in training set. This algorithm demands training set and test set, and one thing need to pay attention is that the data in the training set are well designed, without missing data. However, in the test set, the object we are aiming to segment is occluded.

C. Training and Prediction

The last step is to do training and prediction. The training algorithm is also based on Markov Chain Monte Carlo method but using Metropolis–Hastings sample algorithm. Based on the initial curve evolved by prior information, generate a candidate curve as the next state. Then based on measurement, determine the probability of whether the state will move from current state to our candidate state. After that, randomly generate a number between 0 and 1 based on uniform distribution to determine accept the prediction or not.

Stage of Algorithm

A. Markov chain Monte Carlo Method

The Markov chain Monte Carlo (MCMC) is a set of algorithms that use Markov chains to sample from a random distribution, with the previous steps as the basis. The more steps, the better the result.

In section 3, we talked about particle filters. The main problem of particle filter is that a large number of samples are needed to approximate the posterior probability density of the system. Therefore, an adaptive sampling strategy that can effectively reduce the number of samples is the focus of this algorithm. In addition, the resampling phase will result in a loss of sample validity and diversity, leading to sample depletion. How to maintain the validity and diversity of particles and overcome sample depletion is also the focus of this algorithm.

The Markov chain Monte Carlo (MCMC) method generates a sample from the target distribution by constructing a Markov chain, and has good convergence. In each iteration, combining MCMC enables particles to move to different places, thereby avoiding degradation, and the Markov chain can push particles closer to the state probability density function (PDF), making the sample The distribution is more reasonable.

B. Metropolis–Hastings Algorithm

With a Bayesian perspective, segmentation can be viewed as the problem of estimating the boundary *C* based on image data [2]:

$$\mathrm{p}(\mathrm{C}|\mathrm{data}) \propto \exp\big(-\mathrm{E}(\mathrm{C})\big), \tag{3}$$

$$\mathrm{E}(\mathrm{C}) = E_{data}(C) + E_{shape}(C) = -\log p(data|C) - \log p_C(C), \tag{4}$$

In [2], author presents an algorithm based on Metropolis–Hastings Algorithm to draw samples from p(C|data).

Metropolis-Hastings algorithm is given in *Algorithm 1*, shown below.

Algorithm 1 Metropolis-Hastings algorithm

Initialize $x^{(0)} \sim q(x)$
for iteration $i = 1, 2, \ldots$ **do**
  Propose: $x^{cand} \sim q(x^{(i)}|x^{(i-1)})$
  Acceptance Probability:
    $\alpha(x^{cand}|x^{(i-1)}) = \min\left\{1, \frac{q(x^{(i-1)}|x^{cand})\pi(x^{cand})}{q(x^{cand}|x^{(i-1)})\pi(x^{(i-1)})}\right\}$
  $u \sim \text{Uniform}\ (u; 0, 1)$
  if $u < \alpha$ **then**
    Accept the proposal: $x^{(i)} \leftarrow x^{cand}$
  else
    Reject the proposal: $x^{(i)} \leftarrow x^{(i-1)}$
  end if
end for

To begin with, it will choose an arbitrary point to be the first sample and choose an arbitrary probability density. In our case, our first sample is a curve generated from the seed point given by prior information. Then, for each iteration t, it generates a candidate curve $\mathrm{x}^{\mathrm{cand}}$ for the next sample by picking from the distribution. After that, calculate the acceptance ratio α based on equation (5). Finally, generate a uniform random number $u \in [0, 1]$, if $u \leq \alpha$, then accept the candidate; else, reject the candidate and set $\mathrm{x}^{\mathrm{t+1}} = x^t$ instead.

$$\alpha(\mathrm{x}^{\mathrm{t}}|\mathrm{x}^{\mathrm{t-1}}) = \min\left\{1, \frac{q(x^{t-1}|x^t)\pi(x^t)}{q(x^t|x^{t-1})\pi(x^{t-1})}\right\}, \tag{5}$$

Implementation and Result

The code is adjust from [2], and running on the MNIST database of handwritten digits [12] and Aircraft Benchmark [13] respectively.

A. Experiment on the MNIST database of handwritten digits.

Figure 5 demonstrate one test result run on the MNIST database of handwritten digits. *Figure 5 A* is the original image, which was blocked in some part. *Figure 5 B* is the result for the first sample period; *Figure 5 C* is the result when we use the algorithm proposed by this paper and resampled for 30 times. This shows that with the number of sample iteration increases, the probability of getting the correct result gradually increases.

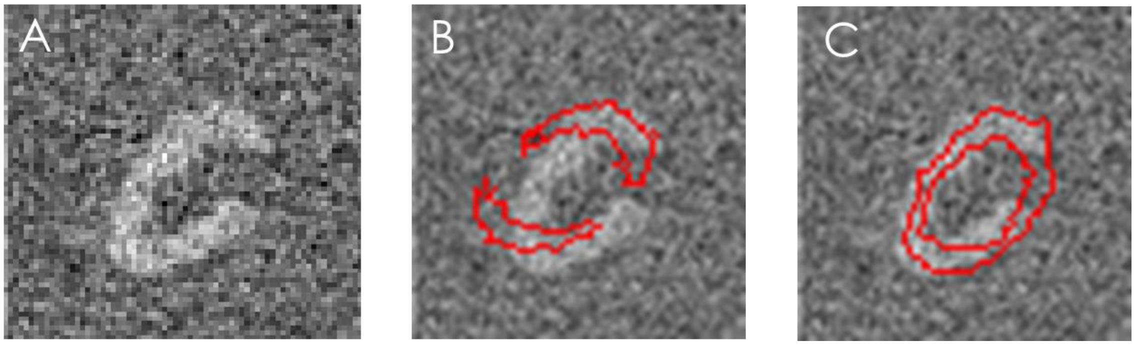

Figure 5. A: Original image with missing data. B: Result for the first iteration. C: Result for the 30-th iteration.

B. Experient on the Aircraft Benchmark dataset

Figure 6 demonstrates test result on the Aircraft Benchmark dataset. *Figure 6 A* is the original image with part of the missing data. I want to compare what will happen if give the incorrect prior information as the seed point, so I run a comparison experiment which shown in *Figure 6 B* and *Figure 6 C.* In *Figure 6 B*, I initial the seed point inside the plane and running for 30 sample periods, which gave a good result. However, I initial the seed point at the top-left corner in *Figure 6 C*, and run for 100 iterations, which still cannot converge to what we want.

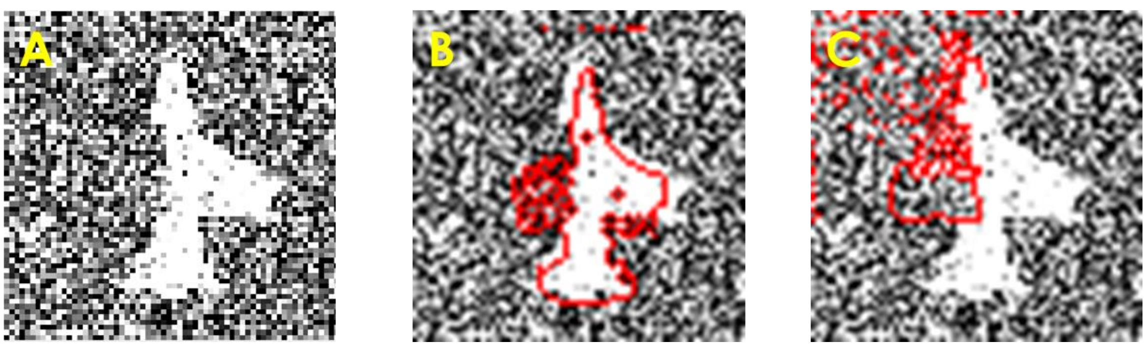

Figure 6. A: Original image with missing data. B: Result given correct prior information and run for 30 iterations. C: Result give incorrect prior information and run for 100 iterations

Contributions

1. Present a Markov chain Monte Carlo (MCMC) sampling approach that uses nonparametric shape priors for image segmentation.
2. Provide an extension within our MCMC framework, that involves a local shape prior approach for scenarios in which objects consist of parts that can exhibit independent shape variations.
3. Have the potential to address issues with getting stuck at local optima, suffered by existing shape-based segmentation methods.

5. Comments and Future Work

As images are 2-D discrete signals, these two amazing papers have a strong correlation with our course.

These two papers have a lot of similarities. Firstly, they both combined sampling method in digital signal processing with Monte Carlo method in statistics. Secondly, prior information is both needed to initial the first sample. Additionally, they both aiming to deal with images of low-quality.

These two papers also have some differences. The first difference is how they initial the first sample point: paper 1 needs 4 points while paper 2 just need a seed point. They are also different in their sample method: paper 2 use Metropolis-Hastings sample method which is an improvement compared with paper 1. Moreover, paper 1 can only segment images with only one class; paper 2 can segment for multi-classes and can even multi-predictions, which is also an improvement.

Overall, from my point of view, paper 2 is much better than paper 1 as it improves in many aspects. However, both two papers have the same limitations. They both heavily rely on the prior shape information, as discussed in the implementation and result in section 5; to define a good measurement space is another problem; and it is also hard to balance time and accuracy: definitely, we will achieve better result by adding more samples or iterations, but that is computational and time consuming.

To solve these limitations, I proposed some possible aspects which we may concentrate on in the future. Firstly, we can focus on how to improve the update strategy, in order to make the MC process converge faster or less computational. Nowadays comes many different optimizers in machine learning field, which we can reference on. Secondly, deep neural network is a strong tool in different fields, and I think we would get better result if we can replace the measurement space by a DNN model. Additionally, reinforcement learning is another booming method based on MCMC, this is also another domain we can focus on. Lastly, the size of images in reality are much more bigger than the data we run in these two paper, so it is also very meaningful to apply this method to solve problems happen in our daily life.

Acknowledgement

I am very grateful to the professor, Mujdat Cetin, for one semester of teaching digital signal processing. He is a great teacher and I do learn a lot from the course, and he is also one of the authors of the second paper I selected in the final project. I need also thanks to our five teaching assistants, they are all patient to help me solving my questions during the course. Lastly, I also discussed a lot with Jingwen Wang, Yue Zhao and Shiyu Sun for this course, they are all intellectual and amazing classmates who offer me a lot of help, I wish they all get their ideal result in this course.

References

[1] Angelova, Donka , and L. Mihaylova . "Contour segmentation in 2D ultrasound medical images with particle filtering." Machine Vision & Applications 22.3(2011):551-561.

[2] Erdil, Ertunc, et al. "MCMC Shape Sampling for Image Segmentation with Nonparametric Shape Priors." Computer Vision & Pattern Recognition 2016.

[3] Fan, Ayres C., et al. "MCMC Curve Sampling for Image Segmentation." 2007.

[4] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. "Snakes: Active Contour Models." International Journal of Computer Vision 1.4(1988):321-331.

[5] Kalman, and R. E. "A New Approach to Linear Filtering and Prediction Problems." *Journal of Basic Engineering Transactions* 82.1:35.

[6] Zarchan, P. & Muso, H.. (2005). Fundamentals of Kalman Filtering: A Practical Approach. 190.

[7] A. Doucet. "On sequential Monte Carlo methods for Bayesian filtering." *Statistics & Computing* 10.3(1998):197-208.

[8] Patrick Pérez, Andrew Blake, and Michel Gangnet. "JetStream: probabilistic contour extraction with particles." *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on* IEEE, 2001.

[9] Tsai, A. et al. "A shape-based approach to the segmentation of medical imagery using level sets." *Medical Imaging IEEE Transactions on* 22.2:137-154.

[10] Daniel Cremers, Stanley J. Osher, and Stefano Soatto. "Kernel Density Estimation and Intrinsic Alignment for Shape Priors in Level Set Segmentation." *International Journal of Computer Vision* 69.3:335-351.

[11] Giles, Michael B. "Multilevel Monte Carlo methods." Acta Numerica 24.3(2013):259-328.

[12] Yann LeCun, et al. *THE MNIST DATABASE of handwritten digits,* Retrieved December 18, 2019, from http://yann.lecun.com/exdb/mnist/

[13] Fine-Grained Visual Classification of Aircraft, S. Maji, J. Kannala, E. Rahtu, M. Blaschko, A. Vedaldi, arXiv.org, 2013

[14] Metropolis, N. . "Equation of state calculations by fast computing machines." Journal of chemical physics 21(1953).

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE, 86(11):2278-2324, November 1998.