Reproduce GrabCut

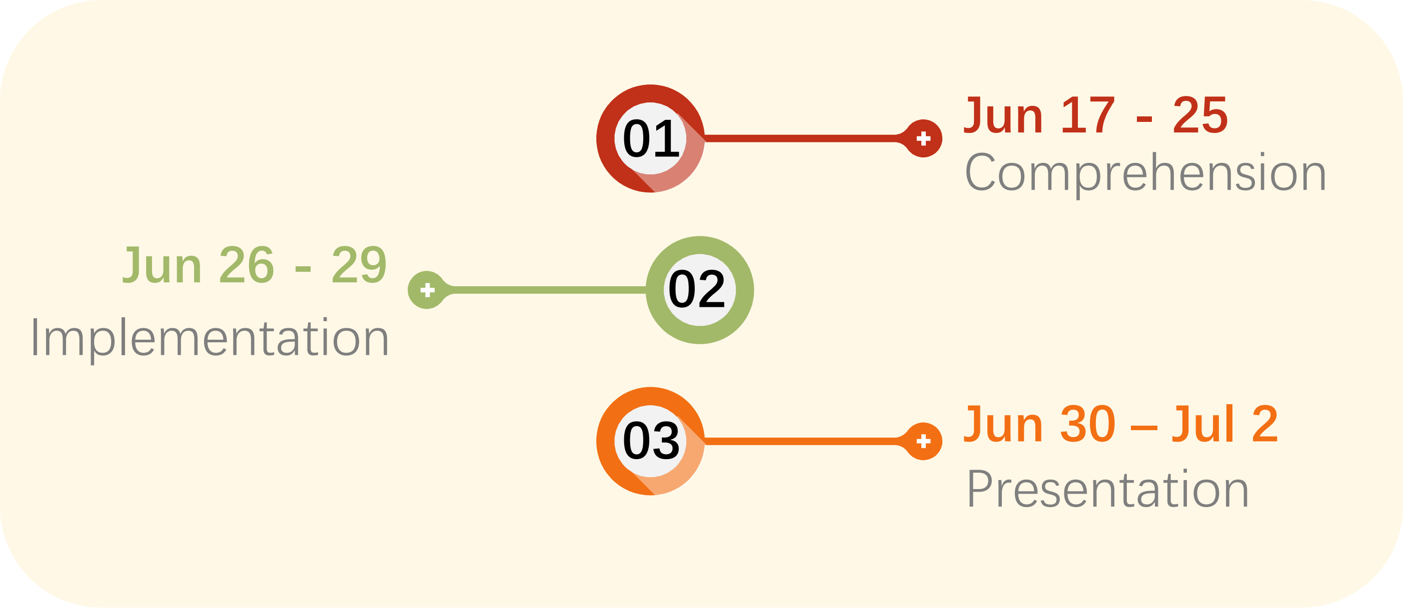
Zhong Tao
Nankai University

Outline

1. Time Overview
2. GrabCut Review
3. Implementation Detail
4. Results Analysis
5. Thoughts

1. Time Overview

- Comprehension
- Implementation
- Presentation



Outline

1. Time Overview
2. GrabCut Review
3. Implementation Detail
4. Results Analysis
5. Thoughts

2. GrabCut Review

"GrabCut" — Interactive Foreground Extraction using Iterated Graph Cuts [1]

Carsten Rother* Vladimir Kolmogorov† Andrew Blake‡
Microsoft Research Cambridge, UK

Figure 1: **Three examples of GrabCut.** The user drags a rectangle loosely around an object. The object is then extracted automatically.

[1] Rother, Carsten, Vladimir Kolmogorov, and Andrew Blake. "Grabcut: Interactive foreground extraction using iterated graph cuts." In *ACM transactions on graphics (TOG)*, vol. 23, no. 3, pp. 309-314. ACM, 2004.

2. GrabCut Review

"GrabCut" — Interactive Foreground Extraction using Iterated Graph Cuts [1]

Carsten Rother* Vladimir Kolmogorov† Andrew Blake‡
Microsoft Research Cambridge, UK

Figure 1: **Three examples of GrabCut.** The user drags a rectangle loosely around an object. The object is then extracted automatically.

[1] Rother, Carsten, Vladimir Kolmogorov, and Andrew Blake. "Grabcut: Interactive foreground extraction using iterated graph cuts." In *ACM transactions on graphics (TOG)*, vol. 23, no. 3, pp. 309-314. ACM, 2004.

2. GrabCut Review

- What GrabCut does
  - Interactive foreground/background **segmentation**
  - i.e. extract a foreground element from a background image [1]

Segmentation vs **Matting**

$\alpha \in \{0,1\}$ $\alpha \in (0,1)$

opacity

[2]

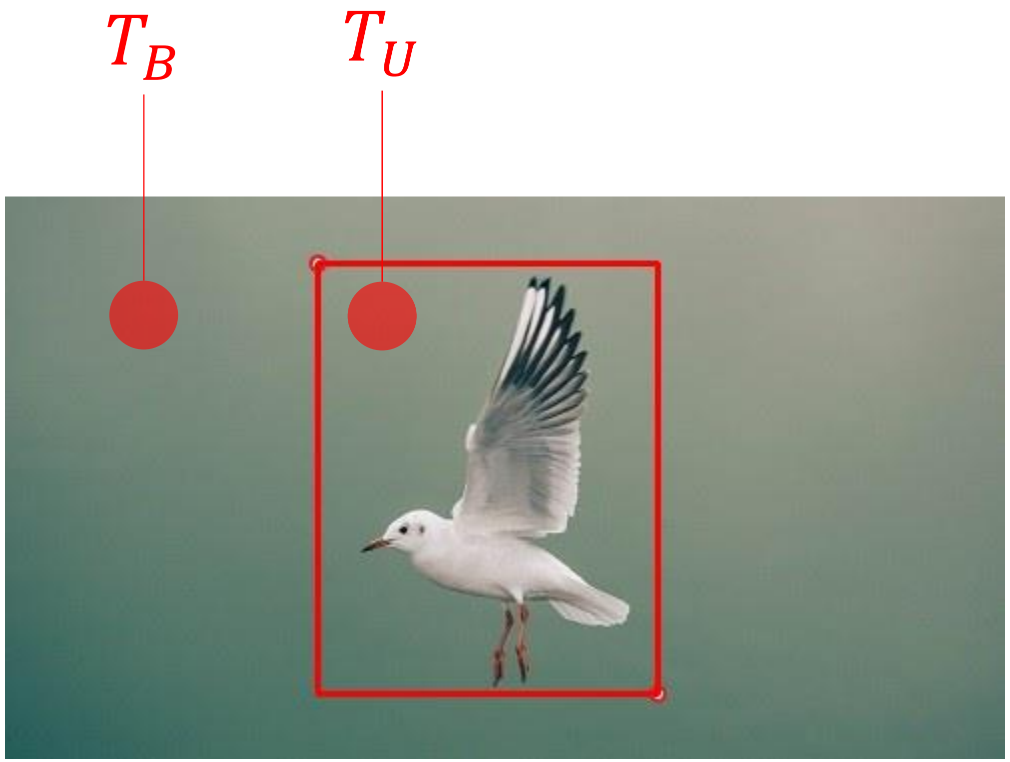
$$C = \alpha F + (1 - \alpha) B$$

composition foreground background

[1] Chuang, Yung-Yu, Brian Curless, David H. Salesin, and Richard Szeliski. "A bayesian approach to digital matting." In *CVPR (2)*, pp. 264-271. 2001.
[2] Zhu, Qingsong, Pheng Ann Heng, Ling Shao, and Xuelong Li. "What's the Role of Image Matting in Image Segmentation?." In *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1695-1698. IEEE, 2013.

2. GrabCut Review

- What GrabCut bases
  - Bayes matting, Graph Cut, ……
- Keys:
  - trimap / seeds / hard constraints
    - T_F: $definitely\ foreground$
    - T_B: $definitely\ background$
    - T_U: $unknown\ region$
    - (note: in GrabCut, no T_F inputs.)



[1]

[1] Photo by janer zhang on Unsplash.

2. GrabCut Review

- What GrabCut bases
  - Bayes matting, Graph Cut, ……
- Keys:
  - energy function / cost function / soft constraints
    - find it expensive to assign a pixel to low-probability region.
    - find it expensive to separate similar parts.

In GrabCut:

$$E\left(\underline{\alpha},\vec{k},\ \underline{\theta},\ \vec{z}\right)=U\left(\underline{\alpha},\vec{k},\ \underline{\theta},\ \vec{z}\right)+V(\underline{\alpha},\vec{z})$$ [2]

regional penalty

$$E(A)=\lambda\cdot R(A)+B(A)$$ [1]

boundary penalty

[1] Boykov, Yuri Y., and M-P. Jolly. "Interactive graph cuts for optimal boundary & region segmentation of objects in ND images." In *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, vol. 1, pp. 105-112. IEEE, 2001.
[2] Rother, Carsten, Vladimir Kolmogorov, and Andrew Blake. "Grabcut: Interactive foreground extraction using iterated graph cuts." In *ACM transactions on graphics (TOG)*, vol. 23, no. 3, pp. 309-314. ACM, 2004.

2. GrabCut Review

- What GrabCut bases
  - Bayes matting, Graph Cut, ……
- Keys:
  - energy function / cost function / soft constraints
    - find it expensive to assign a pixel to low-probability region.
    - find it expensive to separate similar parts.

In GrabCut:

$$E\left(\underline{\alpha},\vec{k},\ \underline{\theta},\ \vec{z}\right)=U\left(\underline{\alpha},\vec{k},\ \underline{\theta},\ \vec{z}\right)+V\left(\underline{\alpha},\vec{z}\right)$$ [2]

regional penalty

$$E(A)=\lambda\cdot R(A)+B(A)$$ [1]

boundary penalty

[1] Boykov, Yuri Y., and M-P. Jolly. "Interactive graph cuts for optimal boundary & region segmentation of objects in ND images." In *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, vol. 1, pp. 105-112. IEEE, 2001.
[2] Rother, Carsten, Vladimir Kolmogorov, and Andrew Blake. "Grabcut: Interactive foreground extraction using iterated graph cuts." In *ACM transactions on graphics (TOG)*, vol. 23, no. 3, pp. 309-314. ACM, 2004.

2. GrabCut Review

- What GrabCut bases
  - Bayes matting, Graph Cut, ……
- Keys:
  - energy function / cost function / soft constraints
    - find it expensive to assign a pixel to low-probability region.
    - find it expensive to separate similar parts.

In GrabCut:

$$E\left(\underline{\alpha}, \vec{k}, \underline{\theta}, \vec{z}\right) = U\left(\underline{\alpha}, \vec{k}, \underline{\theta}, \vec{z}\right) + V(\underline{\alpha}, \vec{z})$$ [2]

Where:

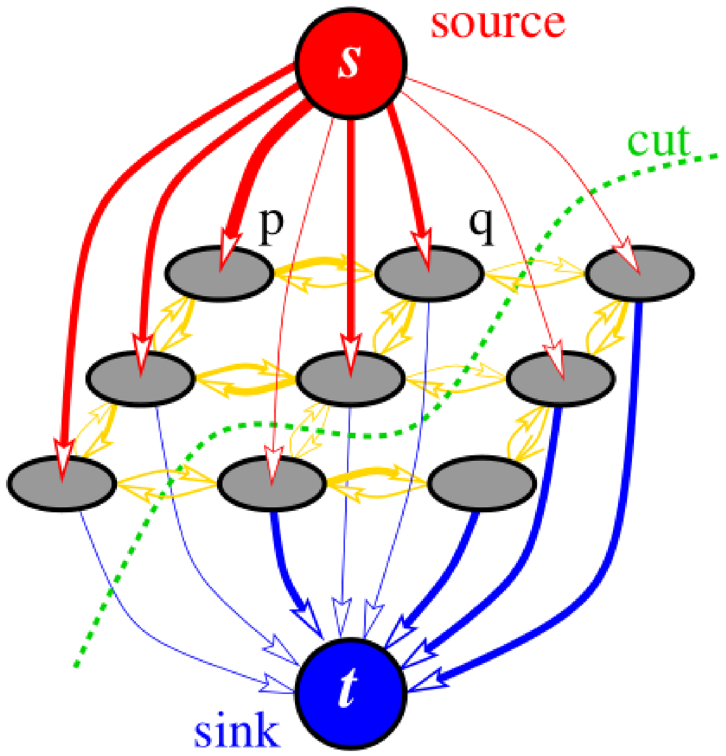
U: negative log-likelihoods
(entropy, the lower probability, the higher gain.)

V: gaussian form $\frac{1}{\mathrm{dis}} \cdot e\hat{}(-\frac{\mathrm{diff}^2}{2\sigma^2})$
(3σ rule, when difference between neighboring pixels exceeds σ, the penalty get small.)

[1] Boykov, Yuri Y., and M-P. Jolly. "Interactive graph cuts for optimal boundary & region segmentation of objects in ND images." In *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, vol. 1, pp. 105-112. IEEE, 2001.
[2] Rother, Carsten, Vladimir Kolmogorov, and Andrew Blake. "Grabcut: Interactive foreground extraction using iterated graph cuts." In *ACM transactions on graphics (TOG)*, vol. 23, no. 3, pp. 309-314. ACM, 2004.

2. GrabCut Review

- What GrabCut bases
  - Bayes matting, Graph Cut, ……
- Keys:
  - energy minimization / min-cut / max-flow



[1]

[1] Boykov, Yuri, and Vladimir Kolmogorov. "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision." *IEEE Transactions on Pattern Analysis & Machine Intelligence* 9 (2004): 1124-1137.

2. GrabCut Review

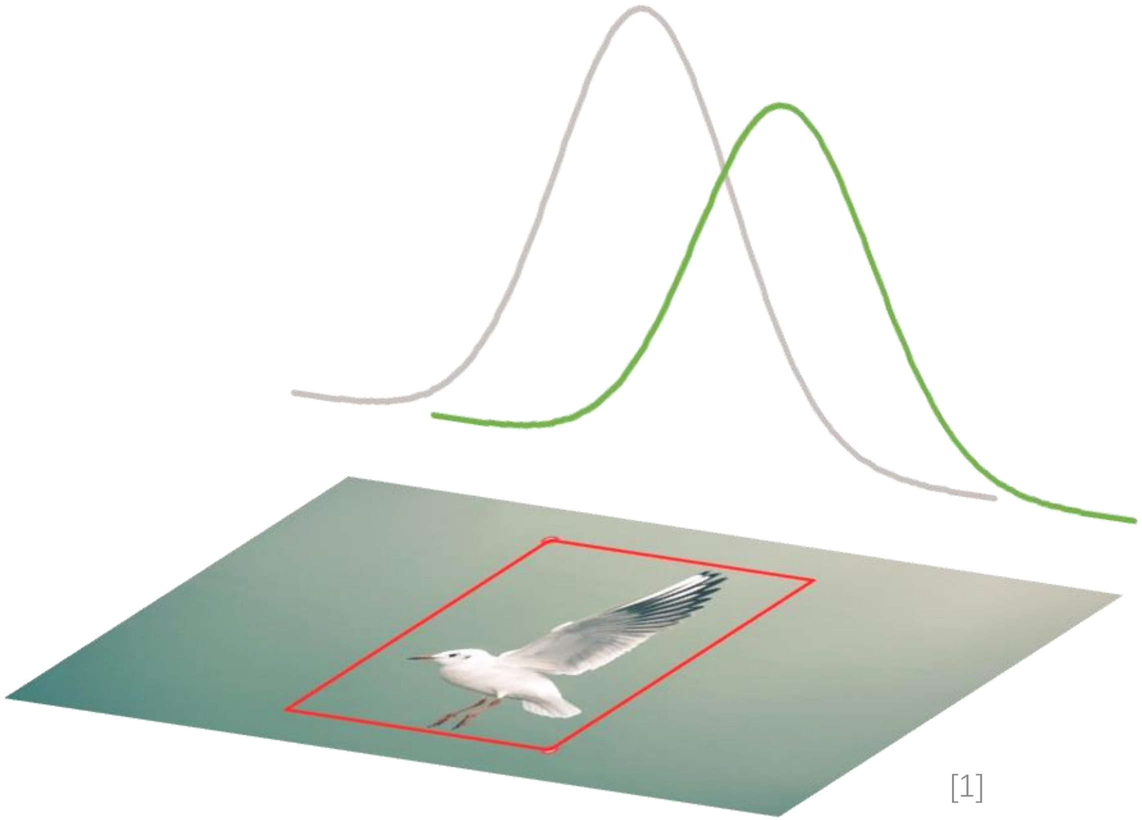
- What GrabCut makes
  - Color models
    - GMMs (Gaussian Mixture Models)
  - Iterative
    - Gaussian parameter estimation
  - Border matting
    - I didn't implement

3-D Gaussian probability distribution

$$\frac{1}{(2\pi)^{\frac{3}{2}}(\det \boldsymbol{C})^{\frac{1}{2}}}\exp\{-\frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu})^T\boldsymbol{C}^{-1}(\boldsymbol{X}-\boldsymbol{\mu})\}$$

2. GrabCut Review

- What GrabCut makes
  - Color models
    - GMMs (Gaussian Mixture Models)
  - Iterative
    - Gaussian parameter estimation
  - Border matting
    - I didn't implement

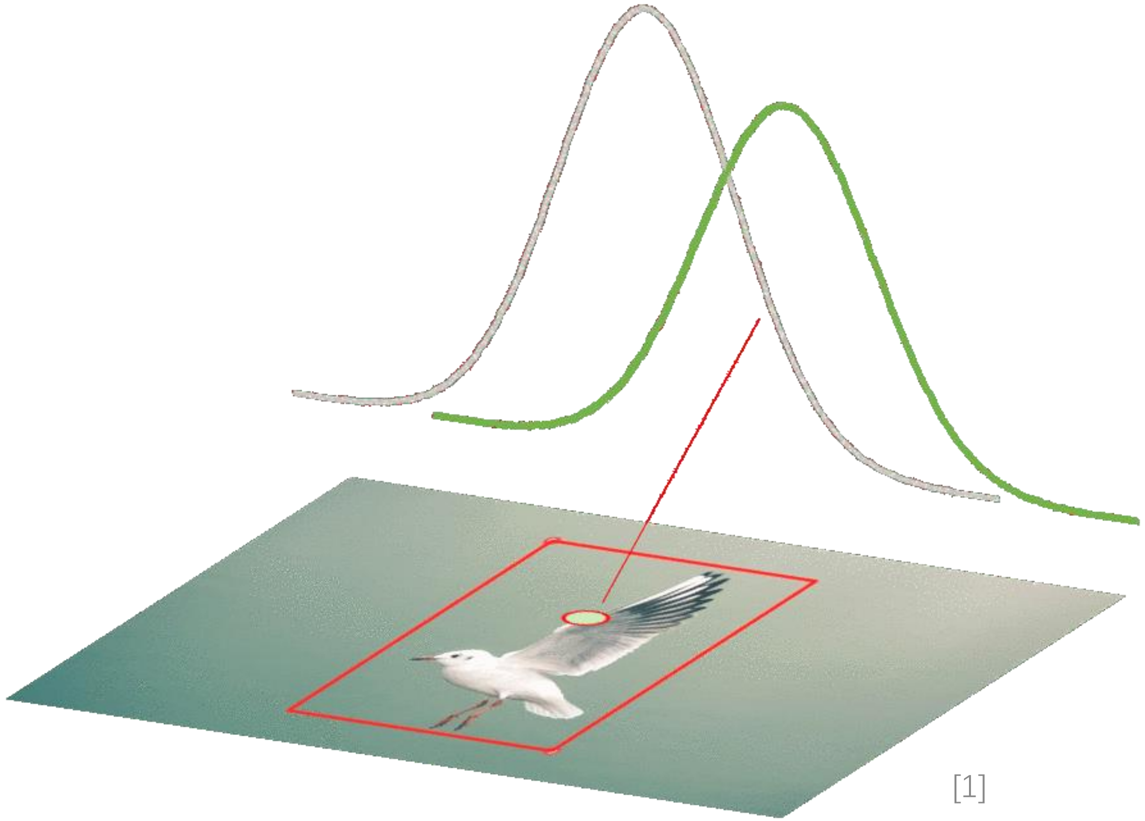


[1]

[1] Photo by janer zhang on Unsplash.

2. GrabCut Review

- What GrabCut makes
  - Color models
    - GMMs (Gaussian Mixture Models)
  - Iterative
    - Gaussian parameter estimation
  - Border matting
    - I didn't implement

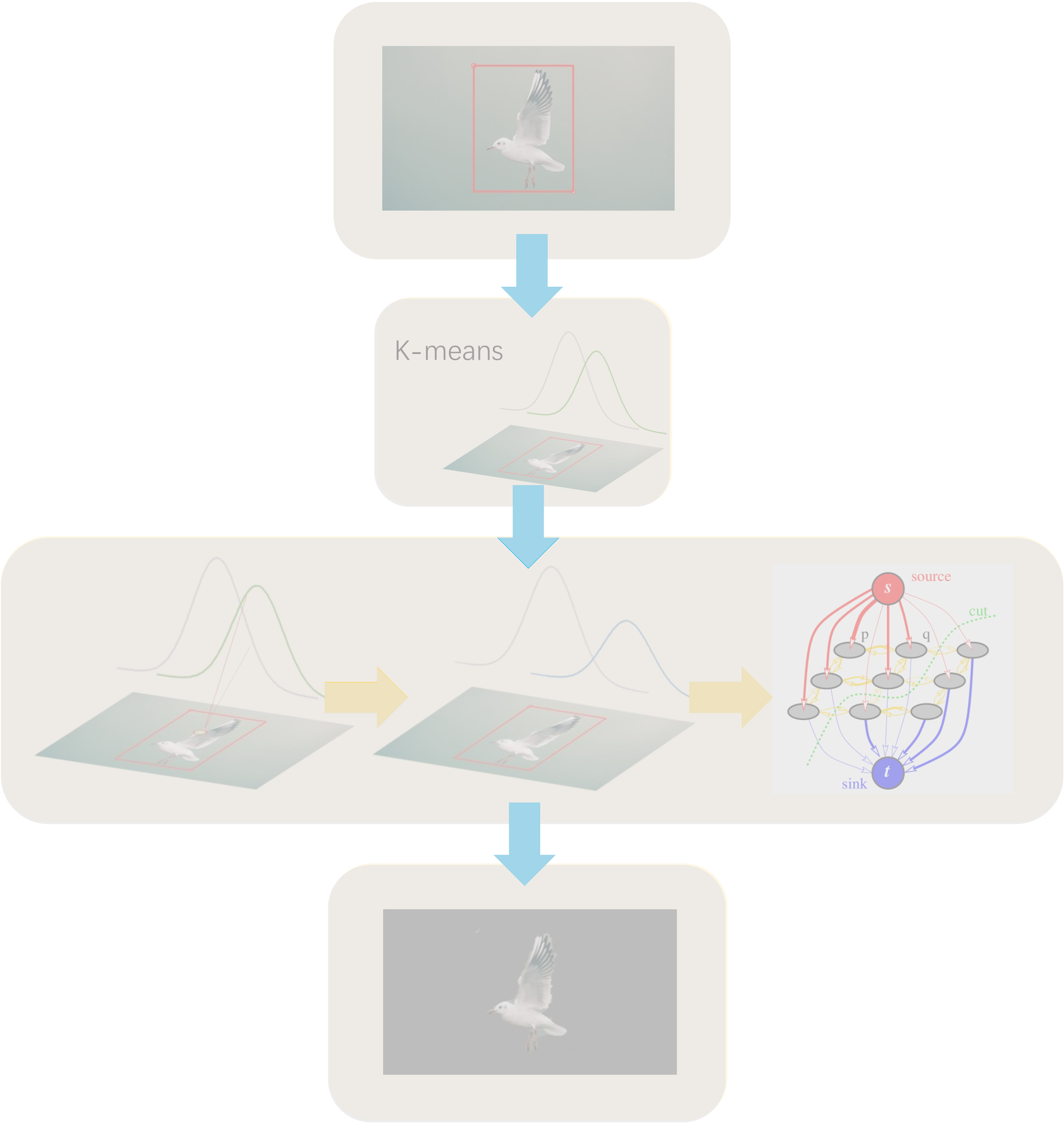


[1]

[1] Photo by janer zhang on Unsplash.

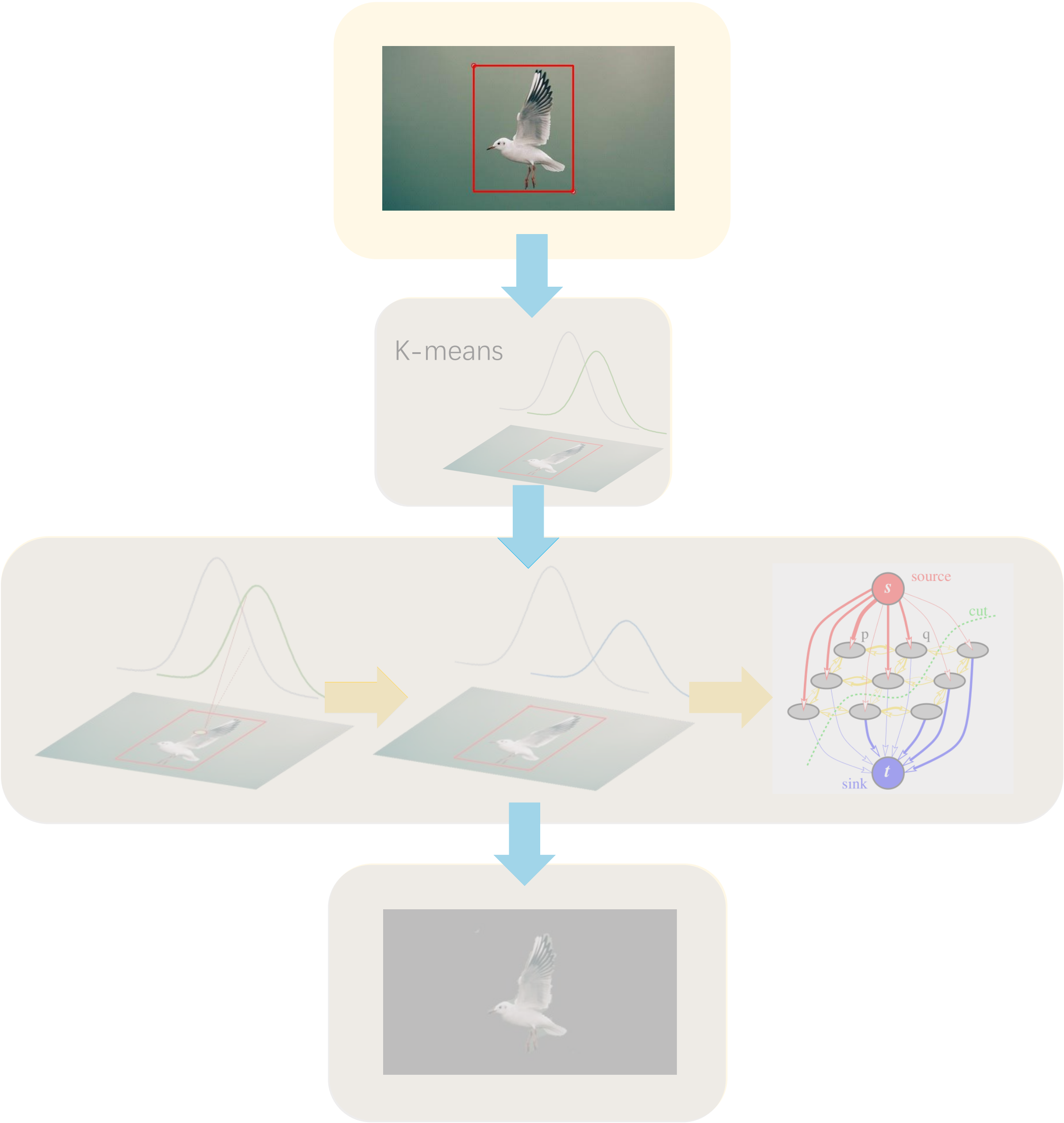
2. GrabCut Review

- Pipeline
- Initialization
  1. User initializes trimap
  2. Initialize alpha label
  3. Initialize GMM
- Iterative minimization
  4. Reassign GMM to pixels
  5. GMM learn from new clusters
  6. Build graph and cut
  7. Repeat 4-6 until convergence



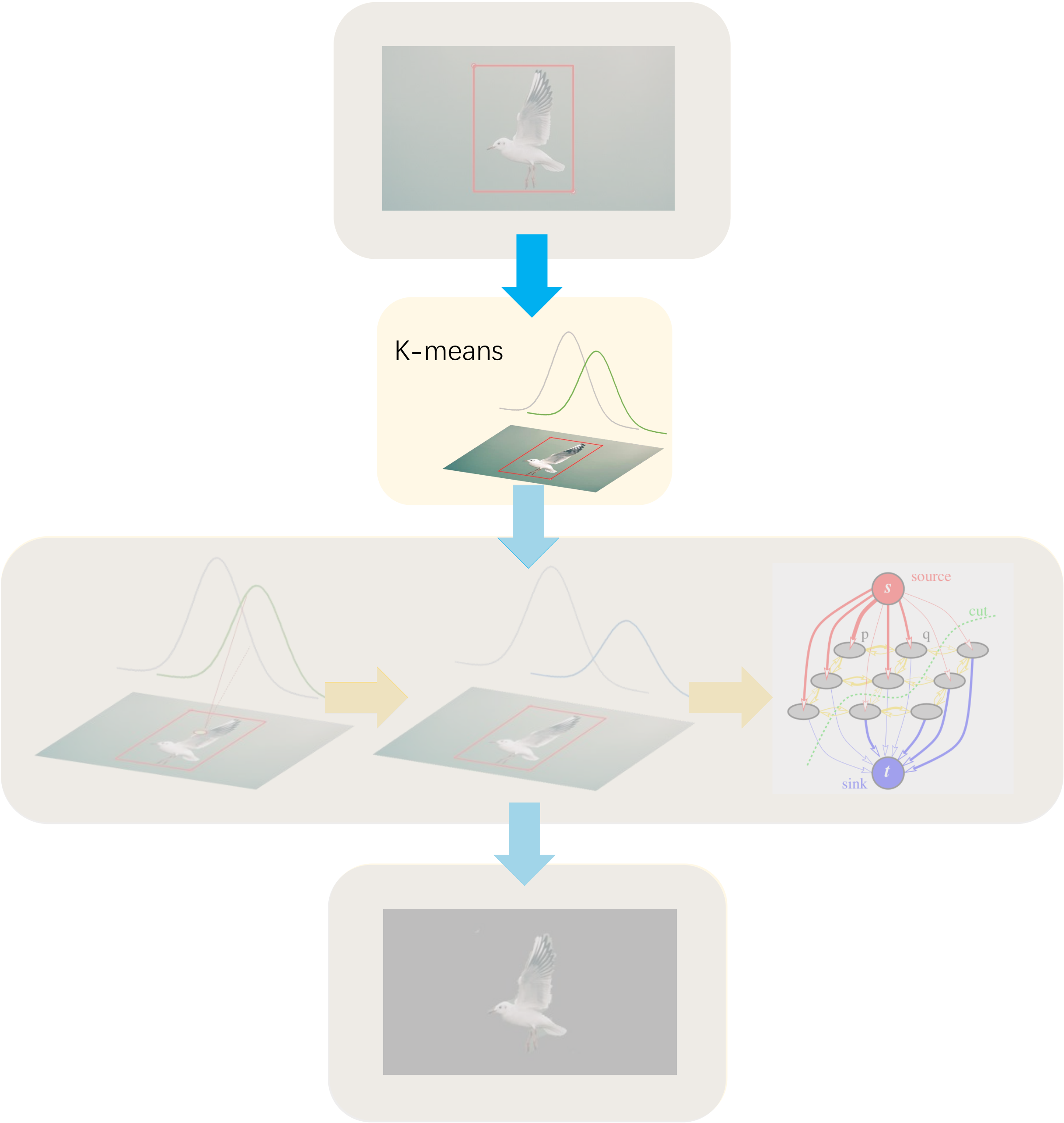
2. GrabCut Review

- Pipeline
- Initialization
  1. User initializes trimap
  2. Initialize alpha label
  3. Initialize GMM
- Iterative minimization
  4. Reassign GMM to pixels
  5. GMM learn from new clusters
  6. Build graph and cut
  7. Repeat 4-6 until convergence



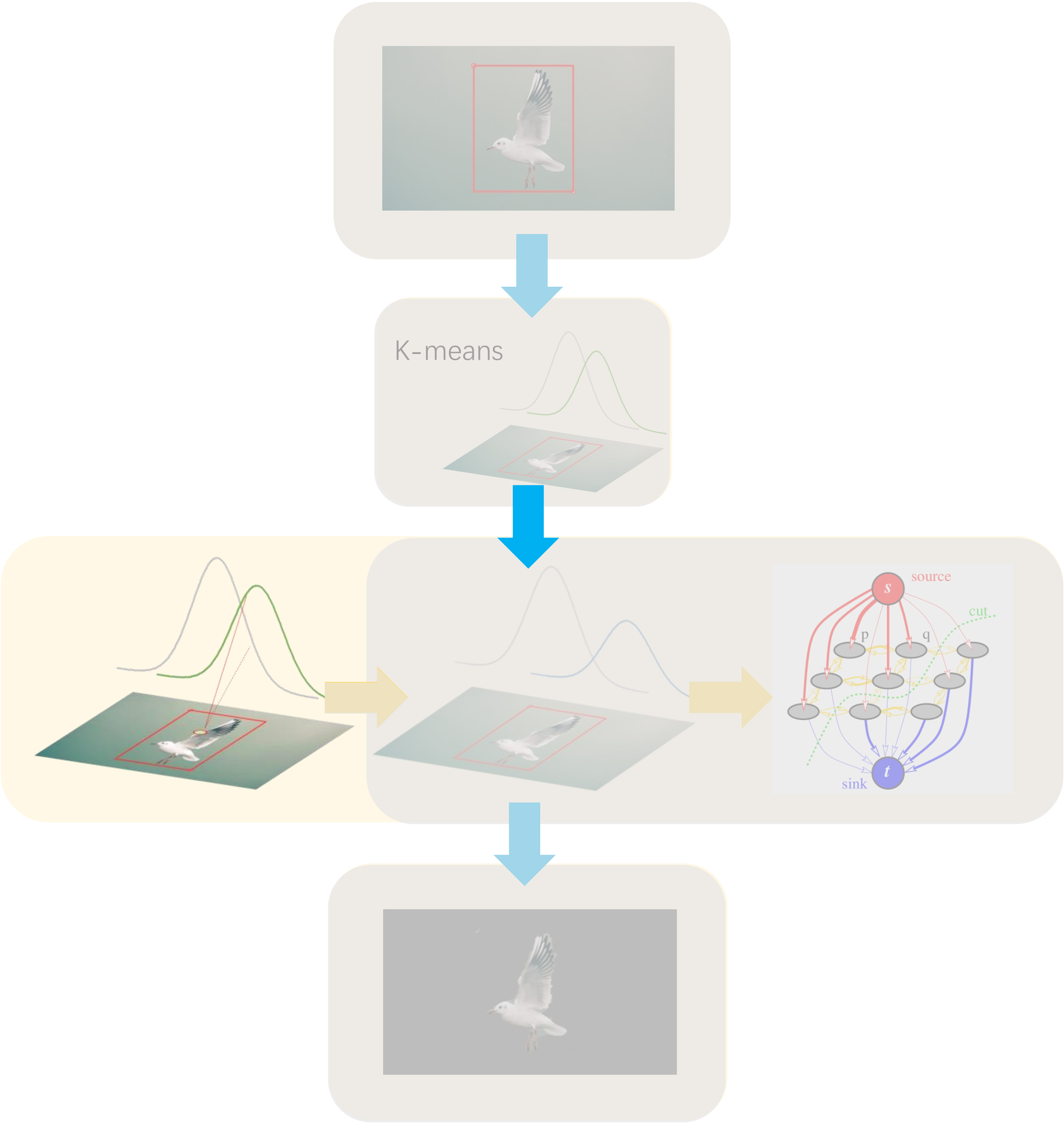
2. GrabCut Review

- Pipeline
- Initialization
  1. User initializes trimap
  2. Initialize alpha label
  3. Initialize GMM
- Iterative minimization
  4. Reassign GMM to pixels
  5. GMM learn from new clusters
  6. Build graph and cut
  7. Repeat 4-6 until convergence



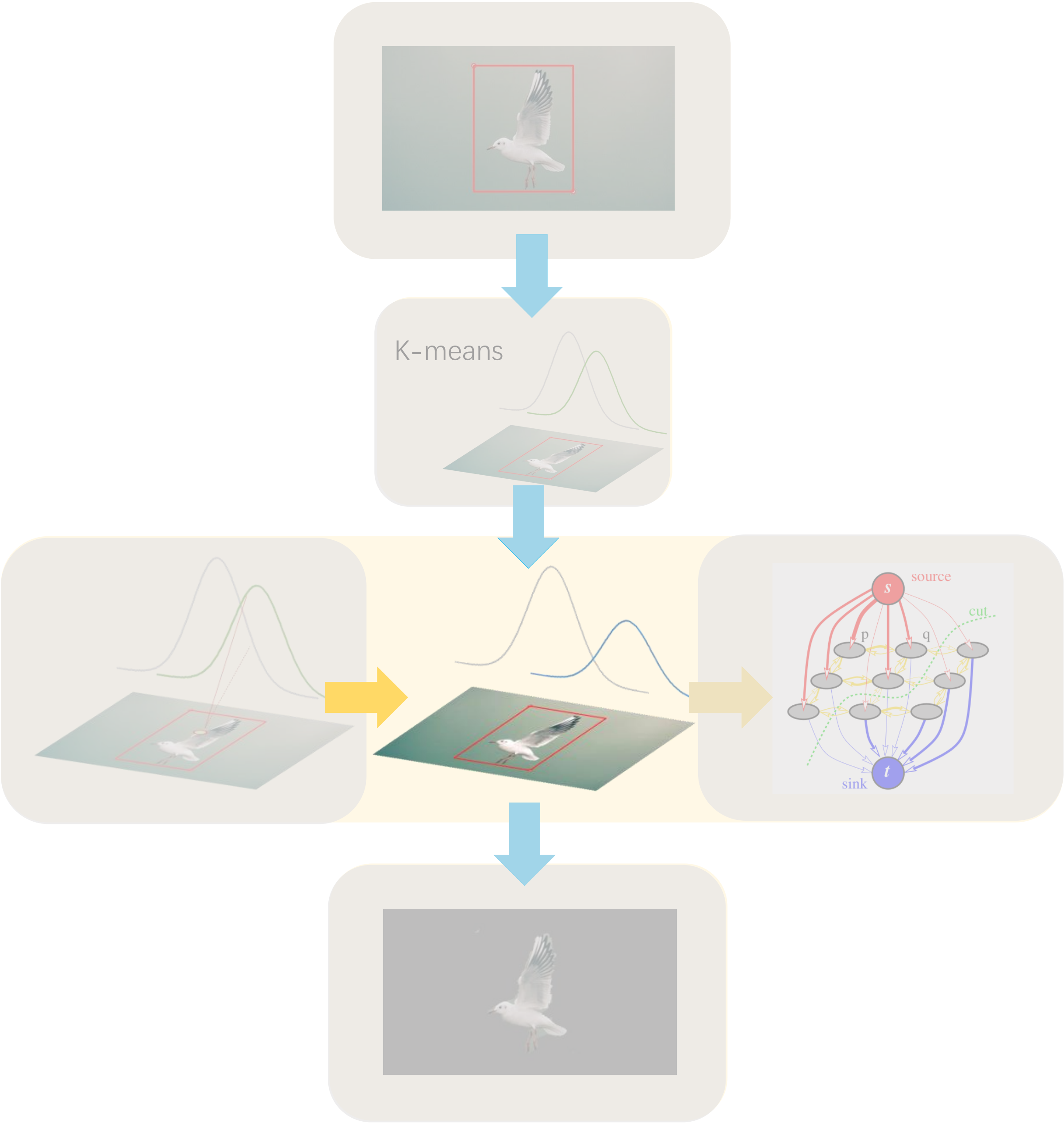
2. GrabCut Review

- Pipeline
- Initialization
  1. User initializes trimap
  2. Initialize alpha label
  3. Initialize GMM
- Iterative minimization
  4. Reassign GMM to pixels
  5. GMM learn from new clusters
  6. Build graph and cut
  7. Repeat 4-6 until convergence



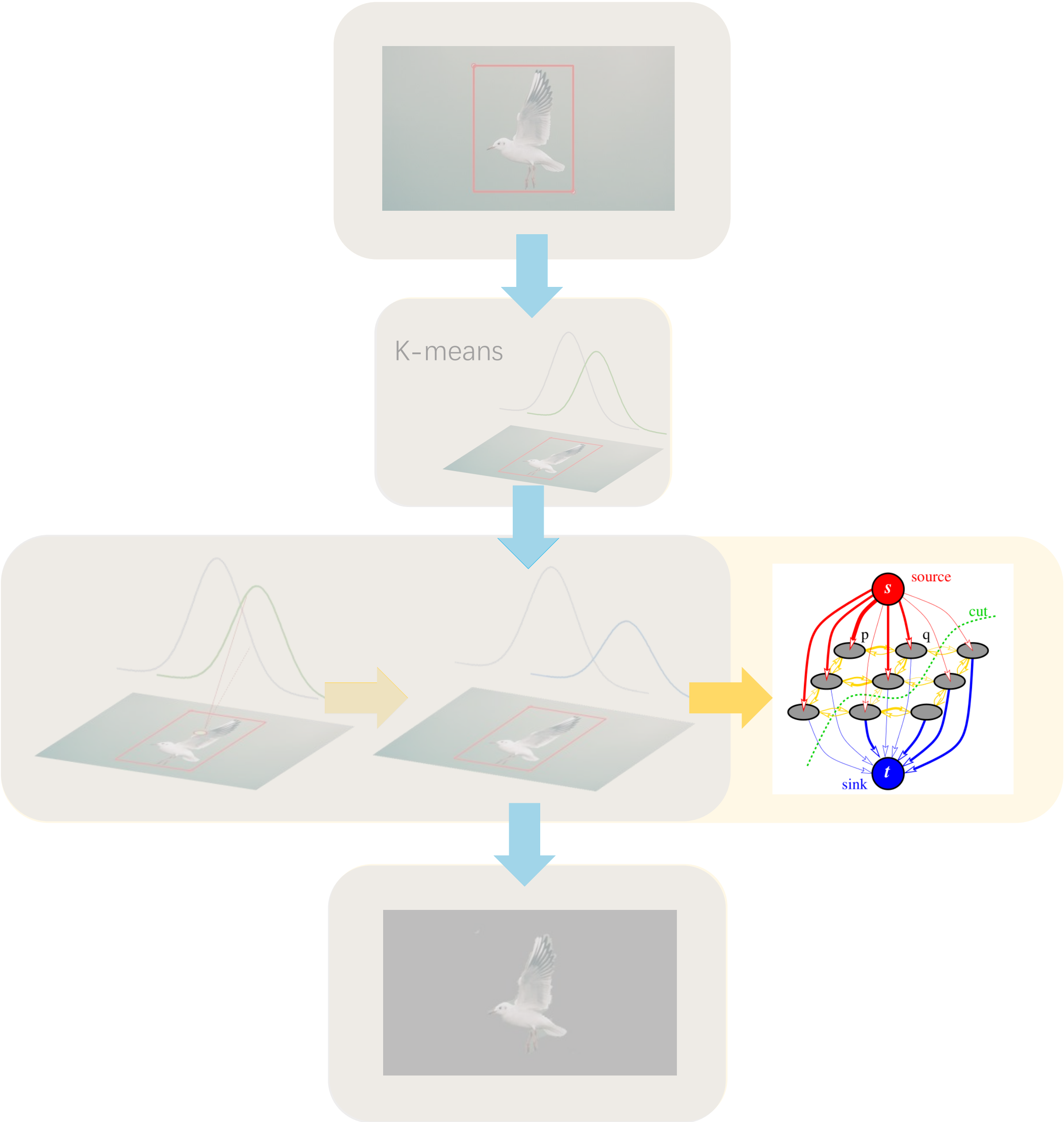
2. GrabCut Review

- Pipeline
- Initialization
  1. User initializes trimap
  2. Initialize alpha label
  3. Initialize GMM
- Iterative minimization
  4. Reassign GMM to pixels
  5. GMM learn from new clusters
  6. Build graph and cut
  7. Repeat 4-6 until convergence



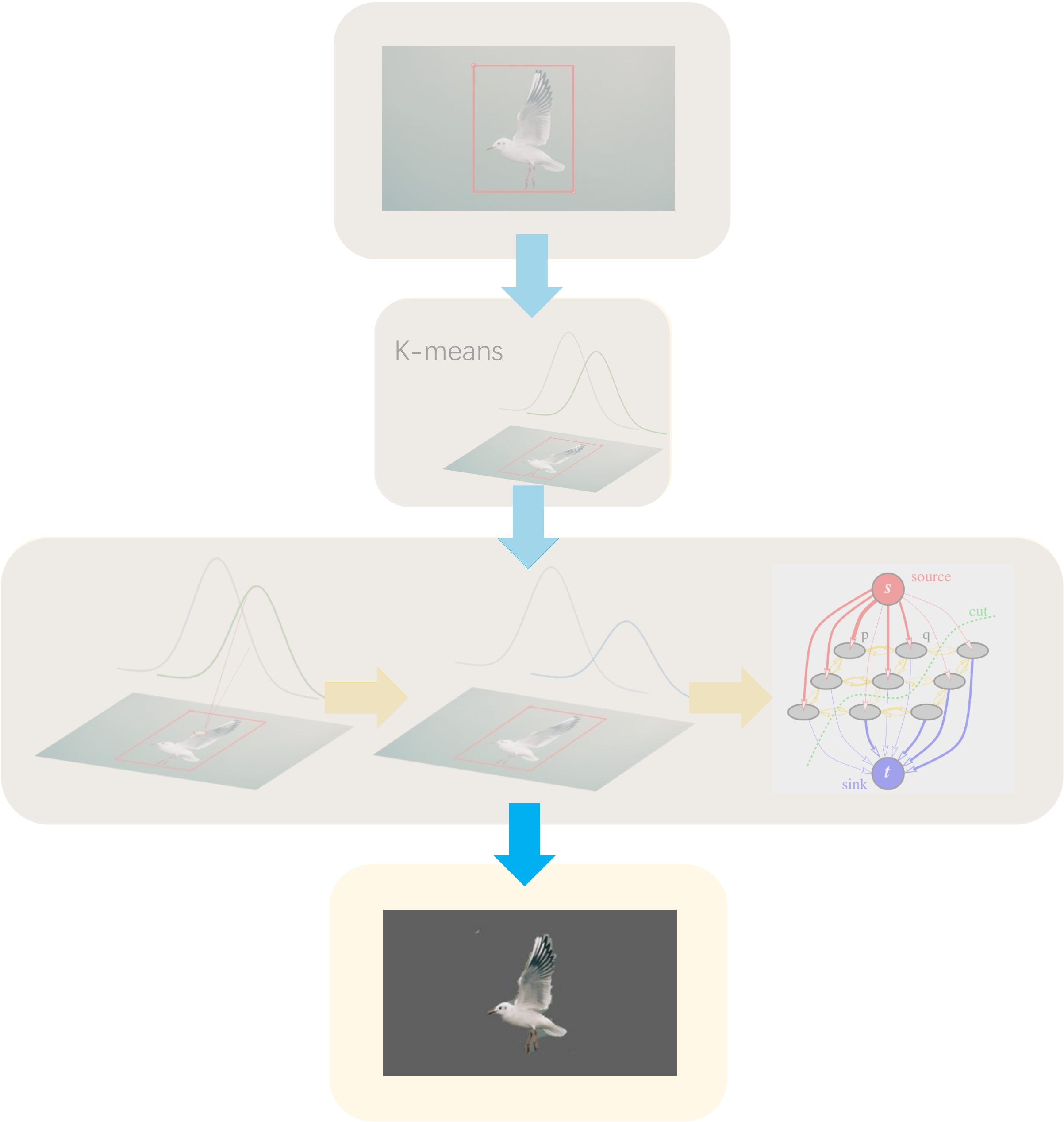
2. GrabCut Review

- Pipeline
- Initialization
    1. User initializes trimap
    2. Initialize alpha label
    3. Initialize GMM
- Iterative minimization
    4. Reassign GMM to pixels
    5. GMM learn from new clusters
    6. Build graph and cut
    7. Repeat 4-6 until convergence



2. GrabCut Review

- Pipeline
- Initialization
  1. User initializes trimap
  2. Initialize alpha label
  3. Initialize GMM
- Iterative minimization
  4. Reassign GMM to pixels
  5. GMM learn from new clusters
  6. Build graph and cut
  7. Repeat 4-6 until convergence

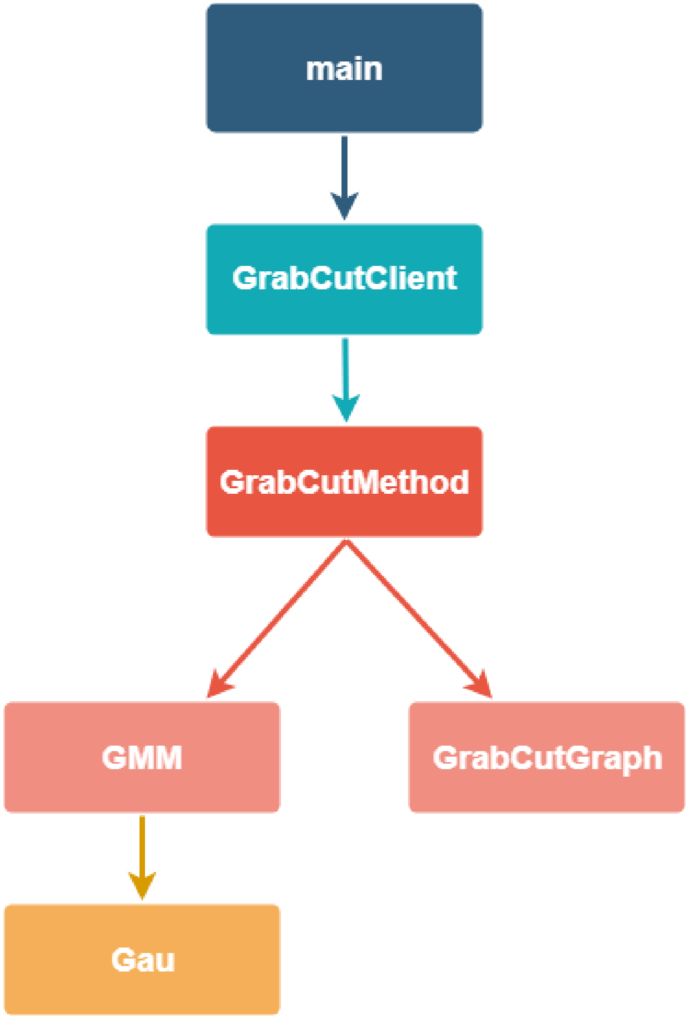


Outline

1. Time Overview
2. GrabCut Review
3. Implementation Detail
4. Results Analysis
5. Thoughts

3. Implementation Detail

- GrabCutClient
- GrabCutMethod
- GrabCutGraph
- GMM, Gau
- cv::kmeans, cv::calcCovarMatrix,[1] maxflow[2]
- ……



[1] Opencv. "Opencv/opencv." GitHub. Accessed July 02, 2019. https://github.com/opencv/opencv/blob/master/modules/imgproc/src/grabcut.cpp.
[2] Boykov, Yuri, and Vladimir Kolmogorov. "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision." *IEEE Transactions on Pattern Analysis & Machine Intelligence* 9 (2004): 1124-1137.

3. Implementation Detail

```
GrabCutClient gc_client("images/white-bird-2-test.jpg", false ,
                        2 /*i_comp*/, 1 /*i_iterate*/);
```

Note: Algorithms were tested using a 4 cores 1.8 GHz machine with 8GB RAM.

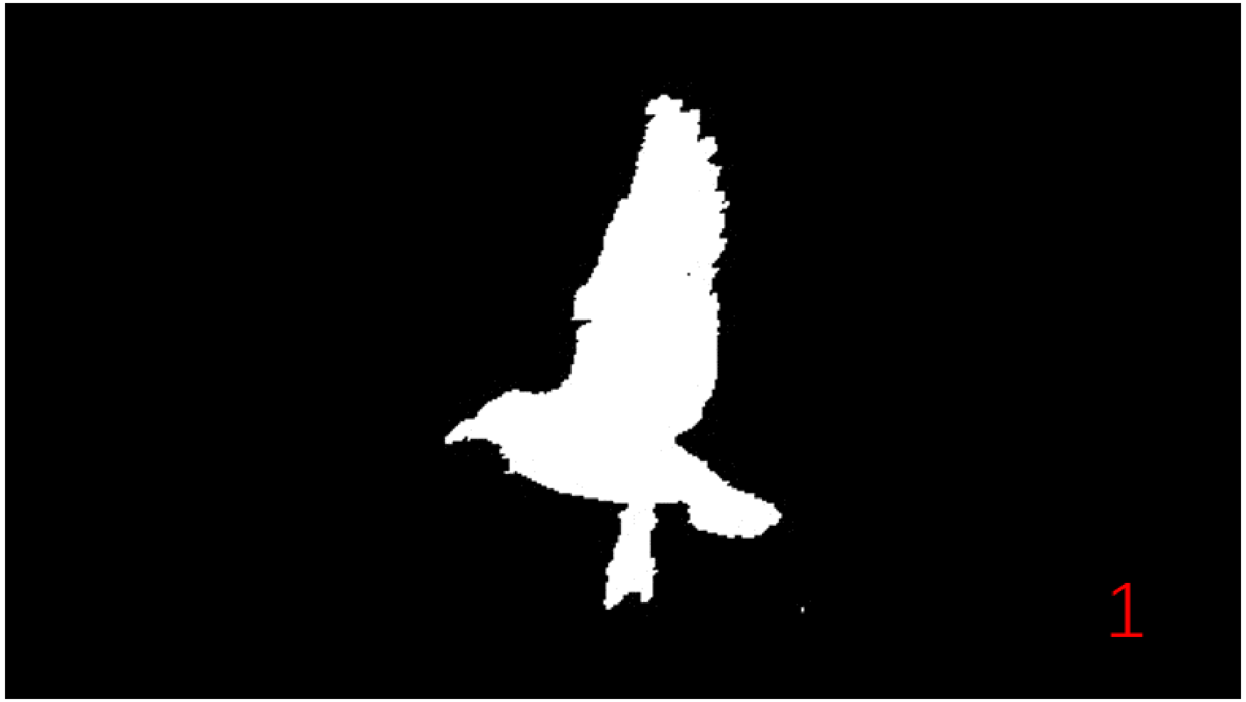
Outline

1. Time Overview
2. GrabCut Review
3. Implementation Detail
4. Results Analysis
5. Thoughts

4. Results Analysis

Ground truth

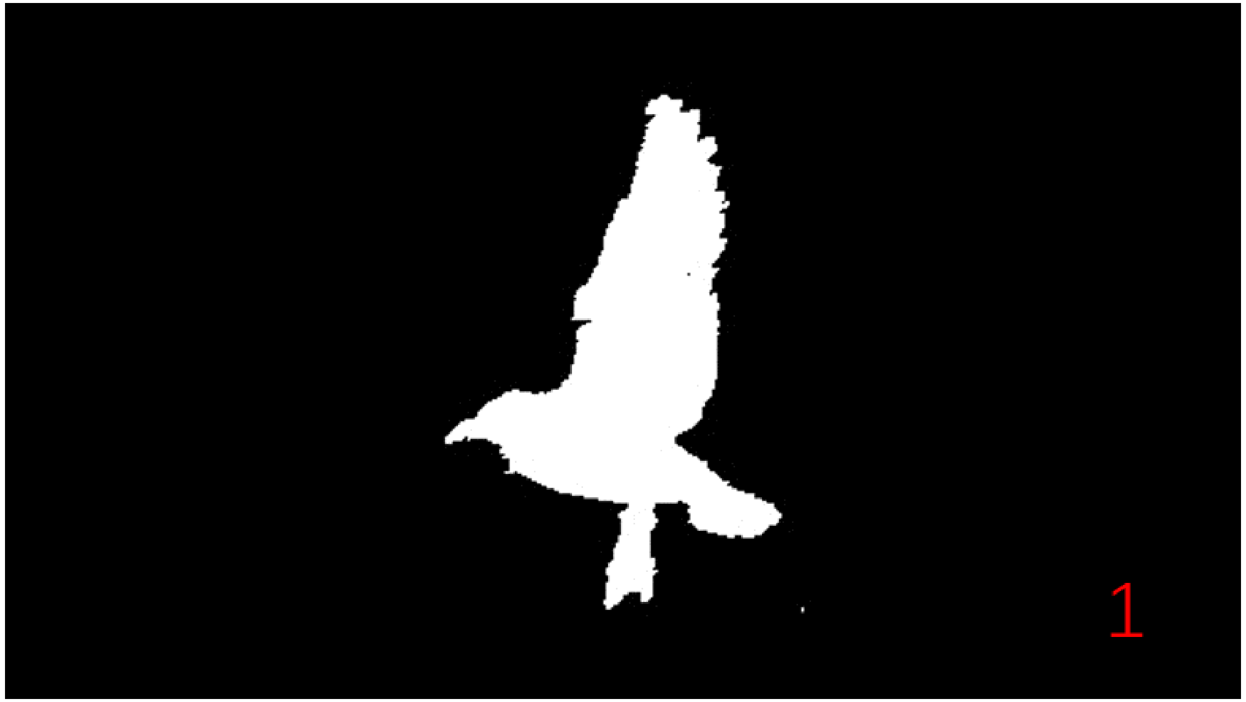
Components increases



4. Results Analysis

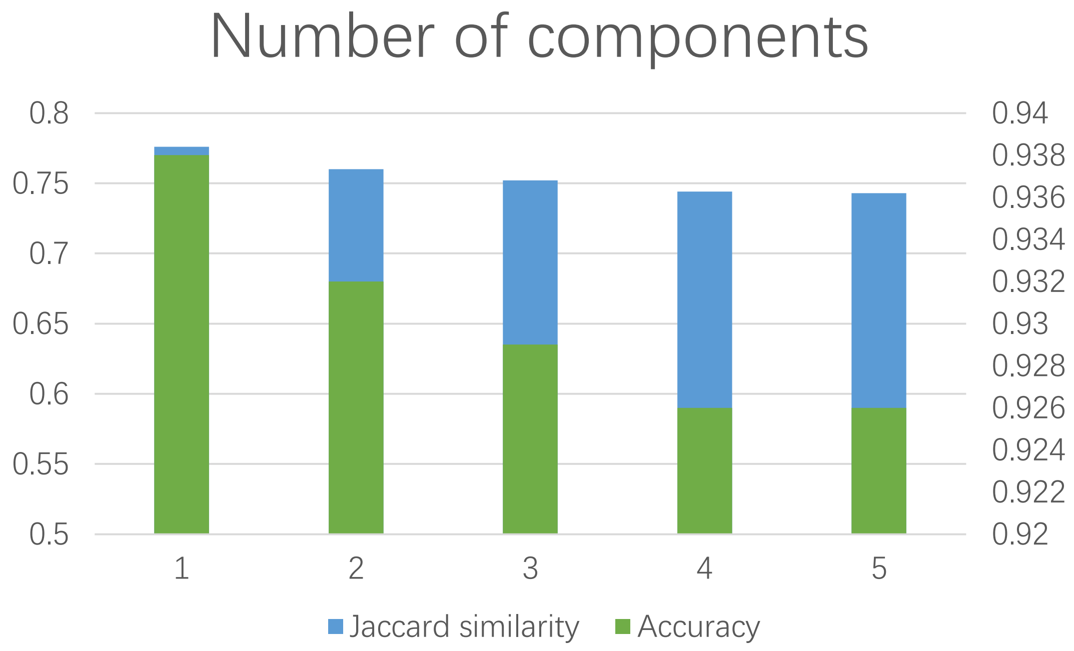
Ground truth

Iteration increases



4. Results Analysis

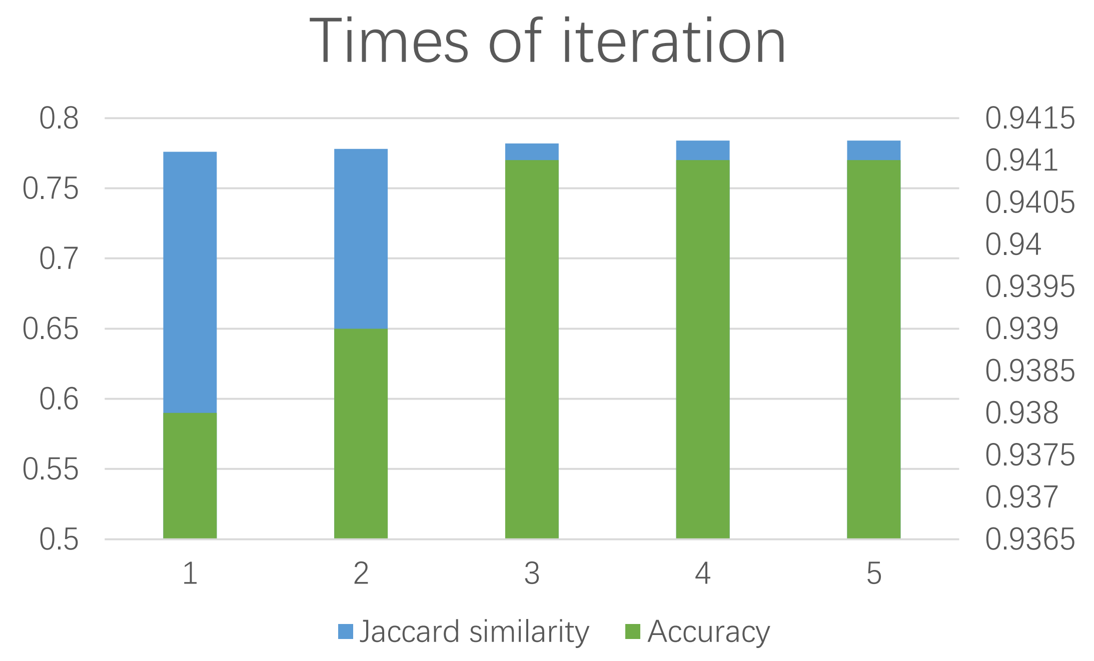
- Dependent variables:
    - Execution time
    - Error metrics
- Independent variables:
    - Number of components
    - Times of iteration
- Energy convergence……



Note: The chart shows the data tested under 1 iteration, and the number of components is incremented from 1 to 5.

4. Results Analysis

- Dependent variables:
  - Execution time
  - Error metrics
- Independent variables:
  - Number of components
  - Times of iteration
- Energy convergence……

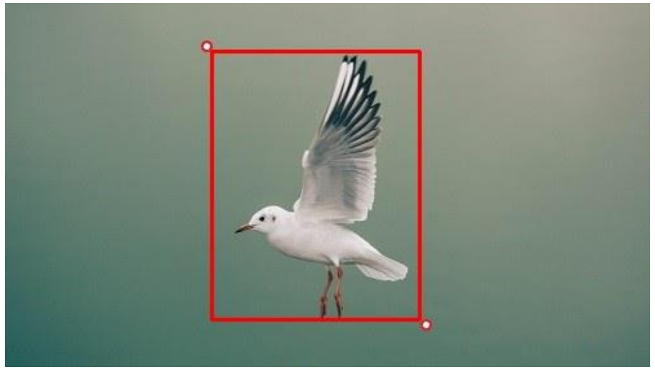


Note: The chart shows the data tested under 1 component, and the times of iteration is incremented from 1 to 5.

4. Results Analysis

Mine

OpenCV



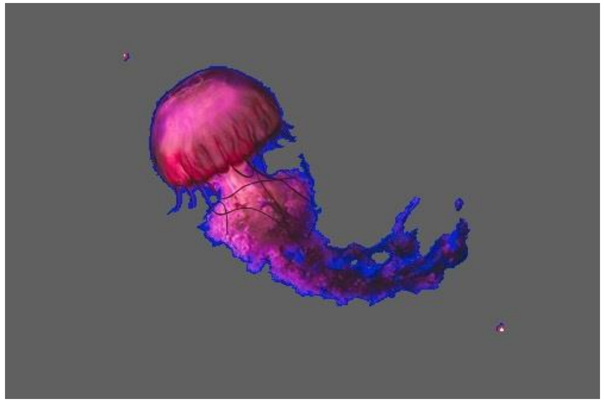
[1]

Note: Both are tested under 5 iterations, 5 components.
[1] Photo by janer zhang on Unsplash.

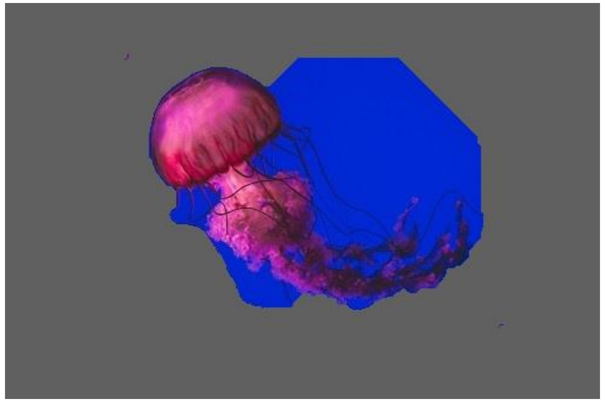
4. Results Analysis

Mine

Failed OpenCV



[1]

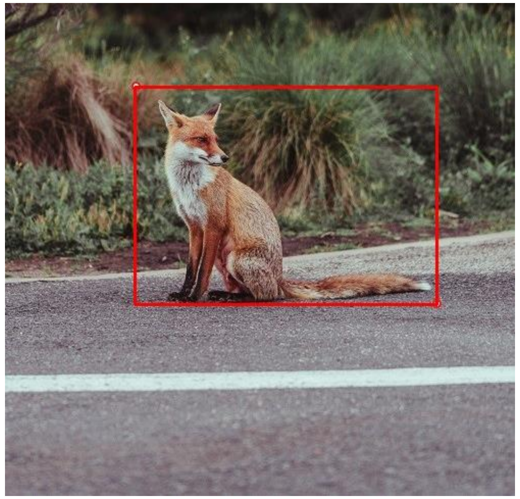


Note: Both are tested under 1 iterations. And my result is tested under 1 Gaussian component while OpenCV set 5 components as default.
[1] Photo by Chitbhanu Singh on Unsplash.

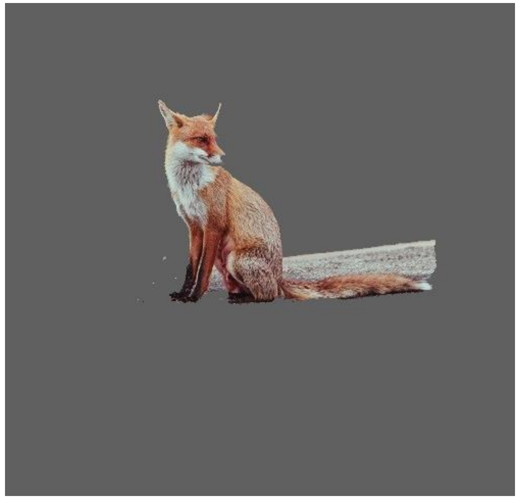
4. Results Analysis

Failed
Mine

OpenCV



[1]



Note: Both are tested under 5 iterations. And my result is tested under 1 Gaussian component while OpenCV set 5 components as default.
[1] Photo by guille pozzi on Unsplash.

Outline

1. Time Overview
2. GrabCut Review
3. Implementation Detail
4. Results Analysis
5. Thoughts

5. Thoughts

- GrabCut is not that perfect, the point is what should be foreground elements?
- Reproducing is not simply copying, it helps to understand some background and will inspire you.
- Code writing needs more practice, knowledge about data structure and algorithm.
- Try to think problems in a different way.
- ……

Thank you!