Graph Cuts Segmentation by Using Local Texture Features of Multiresolution Analysis

Keita FUKUDA†, Nonmember, Tetsuya TAKIGUCHI††, and Yasuo ARIKI††, Members

SUMMARY This paper proposes an approach to image segmentation using Iterated Graph Cuts based on local texture features of wavelet coefficients. Using Haar Wavelet based Multiresolution Analysis, the low-frequency range (smoothed image) is used for the n-link and the high-frequency range (local texture features) is used for the t-link along with the color histogram. The proposed method can segment an object region having not only noisy edges and colors similar to the background, but also heavy texture change. Experimental results illustrate the validity of our method.

key words: image segmentation, graph cuts, multiresolution analysis, local texture feature

1. Introduction

Extracting the foreground objects in static images is one of the most fundamental tasks in image content analysis, object detection and image editing. The task can be formulated as an image segmentation problem.

In recent years, the image segmentation problem has been formalized as an optimal solution problem. The graph cuts technique proposed by Boykov [1], [2] provides a globally optimal solution for segmentation. In Snakes [3] and Level Set Method [4], the local minimum solution for boundary property cost function is computed. On the other hand, in graph cuts, it is possible to compute the global minimum solution, and the cost function is general enough to include both region and boundary properties of the segments.

However, it is difficult to segment an image with complex noisy edges because local noisy edges influence the n-link cost calculated between neighboring pixels. To solve this problem, Nagahashi [6] proposed a coarse-to-fine approach to detect the true boundaries using graph cuts.

In addition, it is difficult to segment an image with an object whose colors or texture are similar to the background. To solve this problem, in this paper, we propose employing image segmentation using Iterated Graph Cuts based on the smoothed image obtained from the low-frequency range and also the local texture features obtained from the high-frequency range for t-link as well as color likelihood. For the integrated formalization, we employ multiresolution wavelet transformation to obtain the smoothed image and the local texture features.

2. Graph Cuts

2.1 Theory of Graph Cuts

This section describes the Graph Cuts based on segmentation proposed by Boykov and Jolly [1], [2]. An image segmentation problem can be viewed as a binary labeling problem. From a given image, we can construct a weighted graph \( G = (V,E,W) \) that consists of nodes \( V \), edges \( E \) and nonnegative weights (costs) \( W \). The nodes are pixels \( p \) on the image \( P \) and the edges have adjacency relationships with four or eight connections between neighboring pixels \( q \in N \). \( N \) is a set of neighboring pixels. The labeling problem is to assign a unique label \( A \) to each node. \( A = (A_1,A_2,\ldots,A_p,\ldots,A_P) \) can be obtained by minimizing the energy \( E(A) \) in Eq. (1). \( A \) is a binary vector i.e. \( A_p \in \{\text{"ob"}, \text{"bkg"}\} \). \(|P|\) is the number of pixels on the image.

\[
E(A) = \lambda \cdot R(A) + B(A) \tag{1}
\]

The coefficient \( \lambda \geq 0 \) in Eq. (1) specifies the relative importance of the region properties term \( R(A) \) expressed as Eq. (2), to the boundary properties term \( B(A) \), expressed as Eq. (3).

\[
R(A) = \sum_{p \in P} R_p(A_p) \tag{2}
\]

\[
B(A) = \sum_{(p,q) \in N} B_{[p,q]} \cdot \delta(A_p,A_q) \tag{3}
\]

The term \( R(\cdot) \) may reflect how the intensity of pixel \( p \) fits into a known intensity model of object and background. The term \( B(\cdot) \) comprises the “boundary” properties of segmentation. \( B(\cdot) \) should be interpreted as a penalty for discontinuity between pixels \( p \) and \( q \). \( B_{[p,q]} \) is normally large when pixels \( p \) and \( q \) are similar. \( \delta(A_p,A_q) = 1 \) if \( A_p \neq A_q \); otherwise \( \delta(A_p,A_q) = 0 \). The process to obtain \( R_p \) and \( B_{[p,q]} \) is described in more detail in the next subsection.

2.2 The Image Viewed as a Graph

The general approach to constructing a graph from an image is shown in Fig. 1. Each pixel in the image is viewed as a node in a graph. Edges are formed between the nodes with the edge costs corresponding to how similar two pixels are (i.e., neighbor-link). The two terminal nodes, the...
source (S) and the sink (T), do not correspond to any pixels in the image but instead are viewed as representing the object and background, respectively. The terminal edge costs are computed using models for the object and background, respectively. The terminal edge costs in the image but instead are viewed as representing the object (S) and the sink (T), do not correspond to any pixels

In Eq. (5), the likelihood is computed based on Gaussian Mixture Model. In Eq. (5), \( \sigma \) is the distribution given experimentally and \( \text{dist}(p, q) \) is the distance between pixel \( p \) and \( q \). The boundary between the object and the background is found by searching for the minimum cost cut [5] on graph \( G \).

2.3 Problem with Graph Cuts

It has been difficult to segment images that include complex noisy edges in interactive graph cuts (Problem 1 in Fig. 2). This is because the cost of the n-link due to noisy edges becomes larger than that of t-link. The edge has a strong influence when there is a large n-link. This problem was solved by using iterated graph cuts based on multi-scale smoothing to avoid the noisy edges [6]. But a new problem arose because it was difficult to segment images with an object whose color is similar to the background (Problem 2 in Fig. 2). This is because the t-link is calculated based only on the color likelihood of the object and background. This problem can be solved by employing texture likelihood in conjunction with color likelihood.

In this paper, we present an approach to image segmentation using iterated Graph Cuts based on local texture features as well as low-frequency features of wavelet coefficients.

3. Proposed Method

3.1 Overview of Proposed Method

The proposed method is shown in Fig. 3. “Object” and “background” seeds are given by the user on an input image. After initializing the level \( k \), the input image is decomposed into subbands (LL, LH, HL, HH) by using Multiresolution Wavelet Analysis at the level \( k \). The smoothed image defined in the low-frequency range (LL) is used for the n-link, and the local texture features defined in the high-frequency range (LH, HL, HH) are used for the t-link. The likelihood is derived from local texture features as well as color features. The prior probabilities are defined by a distance transform from the object edge of the previous segmentation result. The posterior probability is obtained by multiplying the prior probability with the feature likelihood and is set to t-link edge as the edge cost. Graph Cut segmentation is carried out, and these processes are repeated until \( k = 0 \).

3.2 Multiresolution Analysis Based on Wavelet Transform

Multiresolution Analysis using Wavelet Transform starts from the resolution level +1. Scale function \( c_k^{(n)} \) at the position \( k \) on the image and wavelet function \( d_k^{(n)} \) at the position \( k \) on the image are shown as follows:

\[
c_k^{(n)} = \frac{1}{2} (c_{2n}^{(k-1)} + c_{2n+1}^{(k-1)})
\]

\[
d_k^{(n)} = \frac{1}{2} (c_{2n}^{(k-1)} - c_{2n+1}^{(k-1)})
\]

The signal is down-sampled after using Haar Wavelet transform, so that each subband contains one quarter of the pixels of the input image. Three subbands contain high-frequency information in different orientations: vertical (LH), horizontal (HL) and diagonal (HH). The remaining subband (LL) contains low-pass information. In Fig. 4, for example, the signal is first decomposed into subbands HL1, LH1, HH1, and LL1. Then, image LL1 is further decomposed into HL2, LH2, HH2 and LL2.

3.3 Multi-Scale Smoothing (n-Link)

The low-frequency image (LL \( k \)) is obtained by multireso-
Proposed method. To solve problem (1) with Graph Cuts, low pass subband LL is used for n-link (smoothing process). To solve problem (2), high pass subbands LH, HL, HH are used for t-link (local texture features).

Multiresolution analysis at level $k$ in the smoothed image. A stepwise process from global to local segmentation is performed by the Iterated Graph Cuts process with the multiresolution analysis of the coarse-to-fine level (in Fig. 5) in the similar to the way proposed by Nagahashi [6].

The difference is that Nagahashi employed Gaussian Smoothing and we employed the LL image obtained by multiresolution analysis. The obtained neighbor similarity is set to the n-link as edge cost.

3.4 Local Texture Features (t-Link)

High-frequency (LH $k$, HL $k$ and HH $k$) wavelet images are obtained by using multiresolution analysis at level $k$. Local texture features defined by wavelet coefficients of LH $k$, HL $k$, and HH $k$ are used for the t-link along with color features, as described in Sect. 3.5. Local texture features are defined by averaging the absolute wavelet coefficient $d^{(k)}$ in the window $(3 \times 3)$ surrounding pixel $p$ as follows:

$$T_p = \frac{1}{9} \sum_{p,q \in N} |d^{(k)}| \text{ for LH}_k, HL_k, HH_k$$

Local texture features $T_p$ are larger in a complex region and smaller in a flat region. In Fig. 6, local texture features at level 1 are shown.

There have been many studies on texture analysis [7], [8]. Figure 7 shows texture features derived from Haar Wavelet, Texton [7] and LBP [8]. Texton [7] is computed by applying the orientation and spatial-frequency selective filter to images and clustering the responses into a small set of prototype response vectors. Texton [7] can be used for texture analysis, but it requires much more computation time than Haar Wavelet because it has larger dimensions and the clustering has to be done at each level in this study.

Local Binary Pattern [8] is computed at each pixel by thresholding eight surrounding pixel values, multiplying them with binominal weights and summing their values as LBP number. LBP [8] is finer textures than Haar Wavelet.
But in this study, seeds given by a user for learning the likelihood described in Sect. 3.5 are very small. For this reason, Haar Wavelet is suitable as coarse texture feature in this study.

3.5 The Posterior Probability (t-Link)

6 dimensional features \( Y_p = \{ C_p, T_p \} \) are derived from RGB color features \( C_p \) and local texture features \( T_p \). In Eq. (4), the t-link edge costs are transformed to the posterior probability to achieve greater further accuracy as follows:

\[
\begin{align*}
R_p(\text{"obj"]) &= -\ln Pr(O \mid Y_p) \\
R_p(\text{"bkg"]) &= -\ln Pr(B \mid Y_p)
\end{align*}
\]

The posterior probability is proportional to the product of the prior probability and the features likelihood according to Bayes’ theorem as follows:

\[
\begin{align*}
Pr(O \mid Y_p) &= \frac{Pr(Y_p \mid O)Pr(O)}{Pr(Y_p)} \\
Pr(B \mid Y_p) &= \frac{Pr(Y_p \mid B)Pr(B)}{Pr(Y_p)}
\end{align*}
\]

The feature likelihoods \( Pr(Y_p \mid O) \), \( Pr(Y_p \mid B) \) are derived by using Gaussian Mixture Model, and the prior probabilities \( Pr(O) \) and \( Pr(B) \) are derived using the distance transform of the segmentation result image at one previous multiresolution level. The brief shape information can be used as the prior probability. The distance from the boundary is normalized from 0.5 to 1.0. \( d_{(obj)} \) is defined as the normalized distance to the object, and \( d_{(bkg)} \) is defined as the normalized distance to the background. The prior probability is defined by using \( d_{(obj)} \) and \( d_{(bkg)} \) as follows:

\[
\begin{align*}
Pr(O) &= \begin{cases} 
  d_{(obj)} & \text{if } d_{(obj)} \leq d_{(bkg)} \\
  1 - d_{(obj)} & \text{otherwise} 
\end{cases} \\
Pr(B) &= 1 - Pr(O)
\end{align*}
\]

The flow of estimating the posterior probability of \( \{ p, T \} \) and \( \{ p, S \} \) is shown in Fig. 9.

![Fig. 7 Comparison of texture analysis. (Haar Wavelet, Texton [7] and LBP[8])](image)

![Fig. 8 Multiresolution when level is set to k – 1. When graph cuts segmentation is carried out at level k, local texture features (LHk, HLk, HHk) and smoothing image (LLk) are computed at level k – 1. The prior probability is computed from the segmentation result, and graph cuts segmentation is carried out at level k – 1. This examples is at level k = 2.](image)
In Eq. (10), (11) and (12), edge costs are calculated in \( p;S \) and \( p;T \) t-link. When one level of Graph Cuts segmentation finished, the multiresolution level is set to \( k = 1 \) in Fig. 8.

For example, at multiresolution level \( k = 2 \), 6 features \( Y_p = \{ C_p, T_p \} \) are derived from RGB color features \( C_p \) and 3 dimensional local texture features \( T_p \) defined as \( LH2, HL2 \) and \( HH2 \) subbands. Then, at multiresolution level \( k = 1 \), local texture features \( T_p \) defined as \( LH1, HL1 \) and \( HH1 \) subbands. The prior probability \( Pr(O) \) and \( Pr(B) \) are computed using the previous segmentation result. The likelihood \( Pr(Y_p | O) \) and \( Pr(Y_p | B) \) are recomputed on the image at level \( k = 1 \) using the previous segmentation result. If \( k = 0 \), texture features are not defined, but the previous segmentation result affects segmentation result at \( k = 0 \) by updating the parameters.

In the same way, all edge costs are computed. The prior probabilities are defined by the distance transform of the previous segmentation result and the obtained posterior probability is set to t-link as edge cost. This process is repeated until \( k = 0 \).

4. Experiment

4.1 Experimental Conditions

Segmentation experiments were carried out using 50 images provided by the Grab Cuts Database [9]. This image database has the original images and mask images of humans, animals, landscapes and so on. The user gives seeds to the original images and the differences between mask images given in the database and output images are computed as error rate as shown in Fig. 10.

Using the same seeds, we compared Interactive Graph Cuts (conventional method (1)), Iterated Graph Cuts using Smoothing (conventional method (2)) and the proposed method. The segmentation error rate is defined as:

\[
Err[\%] = \left( \frac{\text{undetected pixels in background}}{\text{image size}} + \frac{\text{undetected pixels in object}}{\text{image size}} \right) \times 100 \quad (15)
\]

4.2 Experimental Result

The error rate in segmentation results at each multiresolution level is shown in Table 2. Moreover, the error rate in segmentation results when changing each \( \lambda \) in Eq. (1) is shown in Table 3. The coefficient \( \lambda \) in Eq. (1) specifies the relative importance of t-link edge cost versus n-link edge cost. In particular, at multiresolution level \( k = 3 \) and at \( \lambda = 0.050 \), the proposed method has shown the lowest error rate.

The proposed method improved the error rate from 4.20% to 1.86% and 2.12% to 1.86% compared to the conventional methods (1) and (2), respectively. The reduction rate is 55.7% from conventional method (1) and 12.3% from conventional method (2). The reason why the absolute difference of the error rate between the conventional and the proposed methods is small is that the database includes many images that are conducive to reduction using conventional methods (1) and (2), so they could achieve low error rate. Figure 11 shows the examples of segmentation results with high error rates. In Fig. 11, “No mask” indicates that the images are not included in Grab Cuts database [9], so that the error rate can’t be computed.

For example, at the experiment result of the first line in Fig. 11, the extraction result is only one left person. If the user gives seeds to the right person region or two person regions, the regions corresponding to the location of the seeds can be extracted as follows.

The conventional method (2) and the proposed method can achieve better image segmentation for images with complex edges than conventional method (1) (see examples on rows 1-2 of Fig. 11). They can remove noise because the prior probability has object shape information and the smoothing process can reduce the influence of strong edges.

The proposed method can achieve better image segmentation for images with object colors similar to background than conventional methods (1) and (2) (see examples on rows 3-6 in Fig. 11). Also, Fig. 12 shows that the regions corresponding to the location of the seeds can be extracted.

It is effective to add texture features \( T_p \) to t-link cost for images with complex edges such as natural objects. These texture features \( T_p \) tend to represent the texture difference. Figure 13 shows the segmentation result using only texture features.

| Table 2 Error rates at each multiresolution level [%]. |
|---|---|---|---|
| Level | Conventional (1) | Conventional (2) | Proposed |
| 1 | 4.40% | 2.29% | 1.95% |
| 2 | 4.40% | 2.14% | 1.90% |
| 3 | 4.40% | 2.12% | 1.86% |

| Table 3 Error rates when changing each \( \lambda \) in Eq. (1) [%]. |
|---|---|---|
| \( \lambda \) | Conventional (1) | Conventional (2) |
| 0.100 | 5.81% | 3.21% |
| 0.050 | 4.40% | 2.12% |
| 0.010 | 2.16% | 2.80% |
| 0.005 | 2.19% | 3.52% |
features for t-link cost without color features at each level. The effectiveness can be seen in the examples at rows 1-3 in Fig. 13 because the difference of textures between object and background regions is clear. However, it is ineffective for the examples at row 4 in Fig. 13 because textures are almost same. From this view point, it is effective to use color features as well as texture features.

We carried out segmentation experiments by non-iterated Graph Cuts using all textures defined at level $k$. For example, at multiresolution level $k = 3$, 12 dimensional features $Y_p = \{C_p, T_p\}$ are derived from 3 dimensional RGB color features $C_p$ and 9 dimensional local texture features $T_p$ defined as $LH_1, HL_1, HH_1, LH_2, HL_2, HH_2, LH_3, HL_3$ and $HH_3$ subbands as shown in Fig. 14.

We compared “Graph Cuts using all textures defined at level $k$ (Method (1))”, “Graph Cuts using all textures de-
Fig. 2  Problem with Graph Cuts. Problem 1: Segment images that include complex noisy edges. Problem 2: Segment images with an object whose color is similar to the background.

Fig. 10  Error region computed from the difference between output image and mask image given in image database.

Fig. 12  The regions corresponding to the location of the seeds can be extracted.

Fig. 13  Segmentation results using t-link cost derived from local texture features without color features at each level.

Fig. 14  Texture features $T_p$ at level 3 are extracted from LH1, HL1, HH1, LH2, HL2, HH2, LH3, HL3, and HH3 at level k = 3.

Fig. 15  Problem of Method (1). Many noise regions are detected because n-link edge cost is computed without smoothed image.

---

Table 4  Error rates when changing features $Y_p$ at each level $k$.

| Level | Method (1) | Method (2) | Proposed |
|-------|------------|------------|----------|
| 1     | 3.54       | 2.84       | 1.95     |
| 2     | 4.49       | 3.31       | 1.90     |
| 3     | 5.05       | 3.88       | 1.86     |

---

In Method (1), many noise regions are detected because n-link edge cost is computed without smoothed image. These samples are shown in Fig. 15.

Segmentation results are coarse in Method (2) because n-link edge cost is computed with smoothed image at level $k$, while the proposed method can perform segmentation from

fined at level $k$ and smoothed image $LL_k$ at level $k$ for n-link (Method (2)) and proposed method. Table 4 shows the error rate when changing features $Y_p$ at each multiresolution level $k$ and $\lambda = 0.05$. 
coarse to fine level by iterated Graph Cuts. These examples are shown in Fig. 16.

Moreover, Table 4 shows the higher error rate at level \( k = 3 \) in Method (1) and Method (2). In this case, it is ineffective for images without textures because the ratio of texture features to color features is higher than that of color features. For this reason, iterated Graph Cuts using each texture feature to color features is less evaluated. For this reason, iterated Graph Cuts using all textures defined at level \( k \) is more effective.

Finally, we carried out segmentation experiments by “iterated” Graph Cuts using all textures defined at level \( k \). In other words, this method is iterated Graph Cuts based on local texture features of wavelet coefficients. Using Haar Wavelet-based Multiresolution Analysis, the low-frequency range (smoothed image) is used for the \( n \)-link and the high-frequency range (local texture features) is used for the \( t \)-link along with the color histogram. The proposed method improved the segmentation error rate compared to the conventional methods. Future work is required including optimization of the weight to texture and color for segmentation.

5. Conclusions

This paper has proposed an approach to image segmentation using Iterated Graph Cuts based on local texture features of wavelet coefficients. Using Haar Wavelet-based Multiresolution Analysis, the low-frequency range (smoothed image) is used for the \( n \)-link and the high-frequency range (local texture features) is used for the \( t \)-link along with the color histogram. The proposed method improved the segmentation error rate compared to the conventional methods. Future work is required including optimization of the weight to texture and color for segmentation.

References

[1] Y. Boykov and M.P. Jolly, “Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images,” IEEE International Conference on Computer Vision and Pattern Recognition, vol.2, pp.731–738, 2004.
[2] Y. Boykov and G. Funka-Lea, “Graph cuts and efficient N-D image segmentation,” Int. J. Comput. Vis., vol.70, no.2, pp.109–131, 2006.
[3] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” Int. J. Comput. Vis., vol.1, no.4, pp.321–331, 1988.
[4] J.A. Sethian, Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry Fluid Mechanics, Computer Vision, and Materials Science, Cambridge University Press, 1999.
[5] Y. Boykov and V. Kolmogorov, “An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision,” IEEE Trans. Pattern Anal. Mach. Intell., vol.26, no.9, pp.1124–1137, Sept. 2004.
[6] T. Nagahashi, H. Fujiyoshi, and T. Kanade, “Image segmentation using iterated graph cuts based on multi-scale smoothing,” ACCV 2007, Part II, LNCS 4844, pp.806–816, 2007.
[7] T. Leung and J. Malik, “Representing and recognizing the visual appearance of materials using three-dimensional textons,” Int. J. Comput. Vis., vol.43, no.1, pp.29–44, 2001.
[8] T. Maenpaa, M. Pietikainen, and T. Ojala, “Texture classification by multipredicate local binary pattern operators,” Proc. ICPR, pp.3951–3954, Barcelona, 2000.
[9] http://research.microsoft.com/en-us/um/cambridge/projects/visionimagevideoediting/segmentation/grabcut.htm
Keita Fukuda received a B.E. degree in computer science from Kobe University in 2007. His current research interests include object recognition and image segmentation.

Tetsuya Takiguchi received the B.S. degree in applied mathematics from Okayama University of Science, Okayama, Japan, in 1994, and the M.E. and Dr. Eng. degrees in information science from Nara Institute of Science and Technology, Nara, Japan, in 1996 and 1999, respectively. From 1999 to 2004, he was a researcher at IBM Research, Tokyo Research Laboratory, Kanagawa, Japan. He is currently a Lecturer with Kobe University. His research interests include statistic signal processing and pattern recognition. He received the Awaya Award from the Acoustical Society of Japan in 2002. He is a member of the IEEE, the IPSJ, and the ASJ.

Yasuo Ariki received his B.E., M.E. and Ph.D. in information science from Kyoto University in 1974, 1976 and 1979, respectively. He was an assistant professor at Kyoto University from 1980 to 1990, and stayed at Edinburgh University as visiting academic from 1987 to 1990. From 1990 to 1992 he was an associate professor and from 1992 to 2003 a professor at Ryukoku University. Since 2003 he has been a professor at Kobe University. He is mainly engaged in speech and image recognition and interested in information retrieval and database. He is a member of IEEE, IPSJ, JSAI, ITE and IIEJ.