https://doi.org/10.20965/jaciii.2010.p0645

Paper:

Spatial Object Segmentation Using Stereo Images

Yong Hao*, Lifeng He, Tsuyoshi Nakamura*, Yuyan Chao***, and Hidenori Itoh***

*Department of Computer Science and Engineering, Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya, Aichi 466-8555, Japan
E-mail: {haoyong@juno.ics., tnaka@, itoh@ics.}nitech.ac.jp
**Graduate School of Information Science and Technology, Aichi Prefectural University
Nagakute-cho, Aichi 480-1198, Japan
E-mail: helifeng@ist.aichi-pu.ac.jp
***Graduate School of Environment Management, Nagoya Sangyo University
Owariasahi-city, Aichi 488-8711, Japan
E-mail: chao@nagoya-su.ac.jp
[Received January 23, 2010; accepted July 15, 2010]

The framework of our proposed for segmenting objects using spatial location information from stereo images. An efficient graph-based image segmentation algorithm within this framework for combining changes in optical features and physical location to segment reality scenes into perceptually and semantically uniform regions. Optical and physical location are extracted using *k*-means clustering, and we propose a rules table for combining optical and spatial features together. The performance of our proposed framework is demonstrated in a series of reality-scene images using experimental data from the Middlebury stereo image data (http://vision.middlebury.edu/stereo/data/).

Keywords: object segmentation, image segmentation, stereo image, minimum spanning tree clustering

1. Introduction

Problems of image segmentation and grouping on computer vision even though increasing numbers of approaches based on perceptual and semantic segmentation have been applied to artificial intelligence in image recognition, environment learning, etc. Studies on image segmentation for low-level feature extraction include threshold methods [1], stochastic model-based approaches [2–4], watershed techniques [5], edge flow techniques [6], and normalized cuts [7]. These approaches are based on changes in optical features such as luminance, color, textures and shapes, however, it is difficult to distinguish a region in which a new object or background texture is based on optical feature changes alone. Our framework, which segments objects using spatial location information, originates in binocular vision combining obvious changes in color, textures, and physical location to segment reality scenes into uniform regions. Such an approach similar to ours has not existed until now, and the results of this framework can be used to assist robust systems in, for example, environment learning and object/obstacle recognition.

Our proposed framework is based on two types of input images. One a reality-scene image describing the optical physical world, and the other one a stereo image of location and distance information on individual objects from the reality-scene image. With many approaches already proposed to deal with stereo matching [8–10], we do not deal with this problem here.

2. Algorithm Overview

Our proposed framework segments a reality image using physical space. The goal is to segment objects in physical space similar to what a human being would see. In segmentation, the central question is how to estimate a pair of neighboring pixels by either splitting or combining them. The segmentation of our framework uses an efficient graph-based image segmentation algorithm which defining a predicate for measuring evidence for a boundary between two regions using graph-based image representation as shown in **Fig. 1**.

Once, both types of images, reality and stereo, are input, we use a Gaussian filter to smooth both images to compensate for digitization artifacts before segmentation. Thanks to our efficient graph-based image segmentation algorithm, which is a kind of minimum spanning tree clustering, in the next block, we attempt to construct respective graphs from the two types of images. In **Fig. 1**, G_r is a graph describing optical features from the reality image and G_s is the graph describing spatial information from the stereo image. When graph-construction is completed, G_r and G_s are fused (G_f) with an array of our proposed rules, and we will discuss the rules in detail in the next section. The graph-based segmentation algorithm that runs once in $O(m \log m)$ for m graph edges [11]. In the last block, the segmentation algorithm runs with merging intensity that continues increasing.

3. Efficient Graph-Based Image Segmentation

The section that follows reviews efficient graph-based image segmentation.

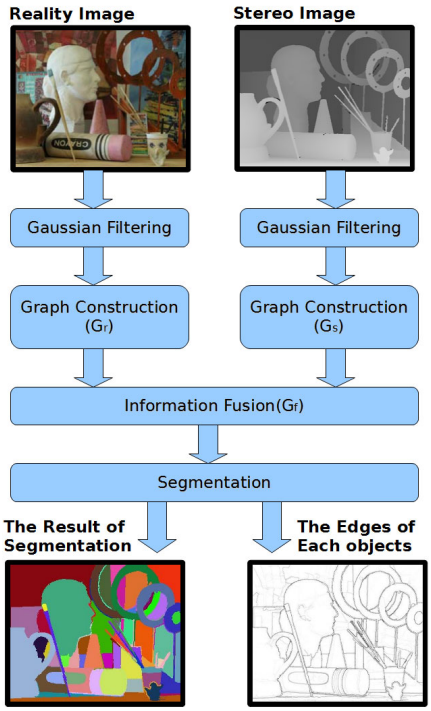


Fig. 1. Block diagram of proposed framework.

3.1. Graph-Based Image Segmentation

In graph theory, a graph is a pair $G = (V,E)$ of sets in which $E \subseteq [V]^2$ ($[A]^k$ is the set of all k-element subsets of A). In image segmentation, each node $v_i \in V$ corresponds to the pixels of an image, and edges $e_{ab} \in E$ connect current pixel v_a to its neighboring pixel v_b. Weight is associated with edge length based on pixels property, such as image intensity or *RGB* color. Earliest graph-based methods use fixed thresholds and local measures in computing segmentation. The segmentation algorithm of our framework is based on minimum spanning tree (*MST*) clustering, detailed in the section that follows.

3.2. Segmentation Algorithm

In Felzenszwalb's work [11] defining a predicate for measuring evidence for the boundary between two regions, a region consists of a minimum spanning tree, and the predicate is based on measuring the dissimilarity between elements along the boundary of the two components relative to a measure of dissimilarity among neighboring elements within each of the two components.

Felzenszwalb proposed the comparison predicate is defined as follows:

$$D(C_1,C_2) = \begin{cases} \text{true} & Dif(C_1,C_2) > MInt(C_1,C_2) \\ \text{false} & \text{otherwise} \end{cases} \quad (1)$$

$MInt(C_1,C_2)$ returns the minimum internal difference for components C_1 and C_2, and $Dif(C_1,C_2)$ returns the minimum weight edge connecting the two components C_1 and C_2. The two functions are defined as follows:

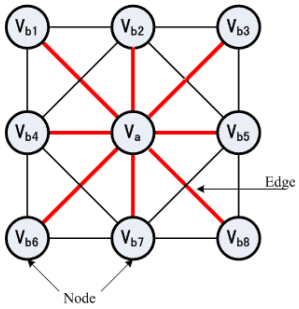


Fig. 2. 8-connected neighbor graph. v_a is the center node, v_{bi} are 8-connect neighbor nodes of v_a, and thick lines edges connecting neighbor nodes.

$$MInt(C_1,C_2) = \min(Int(C_1)+r(C_1), Int(C_2)+r(C_2)) \quad (2)$$

$$Dif(C_1,C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i,v_j) \in E}(weight(v_i,v_j)) \quad (3)$$

$Int(C_i)$ is the internal difference of component C_i ($C_i \subseteq V$) determined to be the largest weight in minimum spanning tree ($G_{MST}(V,E)$) of the component. $Int(C_i)$ is defined as follows:

$$Int(C_i) = \max_{e \in G_{mst}(V,E)}(weight(e)) \quad (4)$$

In Eq. (2), $r(C_i)$ is a threshold function controlling the difference in degree between two components, which must exceed their internal differences to yield evidence of a boundary between them ($D(C_1,C_2)$ is true). Function $r(C_i)$ is defined as follows:

$$r(C_i) = \frac{k}{|C_i|} \quad (5)$$

$|C_i|$ is the size of component C_i and k is the initial parameter. According to Eqs. (1)-(5), parameter k is the merging intensity of this algorithm, meaning that larger k causes a preference for larger components.

4. Spatial Object Segmentation

This section gives more detailed introduce about our approach as graph construction and information fusion.

4.1. Graph Construction

Our approach involves two graphs, G_r of the color intensities from a reality image and G_s of physical space information from a stereo image. Reality and stereo images are shown in **Fig. 1**. For the stereo image, the higher the luminance of object pixels, the nearer they are to the camera in physical space.

Graphs G_r and G_s are constructed as the 8-connection neighbor shown in **Fig. 2**.

Section 3.1 introduced the image segmentation problem in which each node $v_i \in V$ corresponds to pixels of an image, and edges $e_{ab} \in E$ connect current pixel v_a to neighboring pixel v_b. Weight is associated with edge

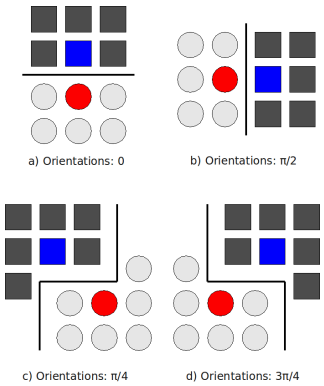


Fig. 3. Four different neighborhood partitions.

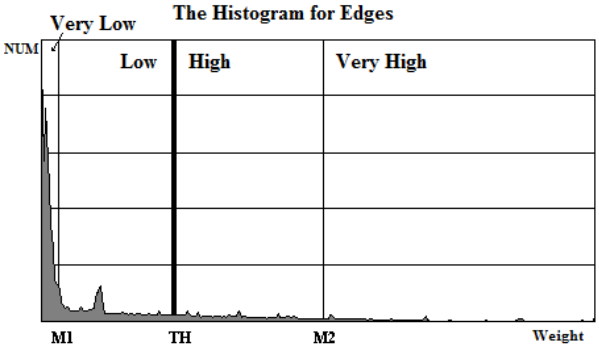


Fig. 4. Edges histogram, where M1 and M2 are mean point, and TH is the threshold plot between Groups 1 and 2.

length based on a pixels property and calculated as follows:

$$e_{ab} = \|V_a - V_b\| \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (6)$$

For reality images, nodes are 3-elements RGB vectors, so weight between two neighboring nodes is calculate as follows:

$$e_{ab} = \|V_a - V_b\| = \sqrt{(R_{va} - R_{vb})^2 + (B_{va} - B_{vb})^2 + (G_{va} - G_{vb})^2} \quad (7)$$

R, G and B are vector v elements for obtaining the color values. For stereo images, nodes V_s have only one luminance element, making introducing its formula unnecessary.

We defined the four different neighborhood partitions, as shown in **Fig. 3**.

Blue and red nodes are pairs of neighbor nodes, whose weight between them pairs is calculated as follows:

$$e_{ab} = \frac{1}{N} \sum_{\substack{v_{ai} \in \{\text{Neigborhood of } v_a\} \\ v_{bi} \in \{\text{Neigborhood of } v_b\}}} \|V_{ai} - V_{bj}\| \quad \ldots\ldots\ldots \quad (8)$$

N is the amount of current neighborhood. The graph is constructed with four neighborhood partitions selected with orientations between a pair of neighbor nodes.

4.2. Information Fusion

When graph construction is completed, it must be determined how to fuse the two types of information (G_r, G_s). In human vision, boundary is important in object recognition. Two types of boundaries from binocular vision are optical and spatial, but changes in physical location have stronger probability than optical one for spatial object segmentation, so we propose a rules table for modifying the weight between a pair of neighbor nodes in color Graph G_r, based on this pair of neighbor node weights in physical location in Graph G_s.

Due to the segmentation problem having two types of operators that merge and split, we define k-means clustering as $k = 2$ by grouping histograms into two groups each $\{Group\ 1,\ Group\ 2\}$. *Group 1* has a strong probability of merging operators and *Group 2* has a strong probability of splitting operators. Mean points for each group are also divided into two groups, meaning, the histograms are divided into four groups $\{Very\ Low,\ Low,\ High,\ and\ Very\ High\}$. k-means clustering results are shown in **Fig. 4**.

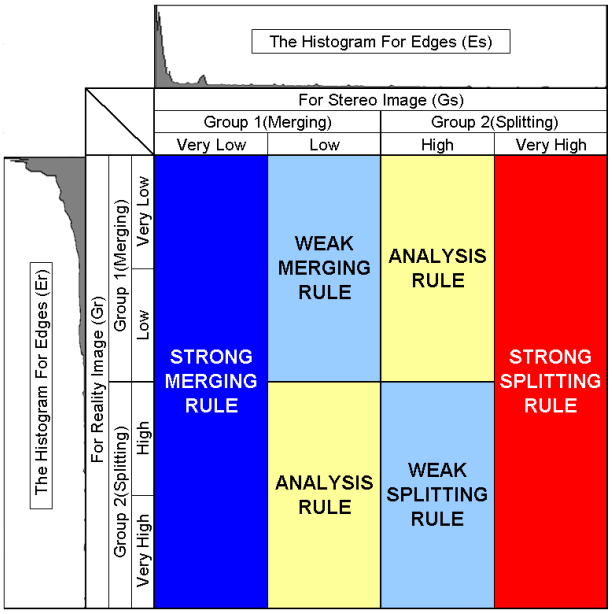


Fig. 5. Weight update rules table.

4.3. Update Rules

We defined five update rules $\{Strong\ Merging,\ Weak\ Merging,\ Analysis,\ Weak\ Splitting,\ Strong\ Splitting\}$ in **Fig. 5**, and below is detailed introduce about the five update rules.

A) STRONG MERGING RULE

IF:	$es_{ab} \in [0, Ms_1)$	{Very Low}
THEN:	STRONG MERGING	{SM}

This update rule assumes that current pairs of neighbor nodes vr_a and vr_b ($vr_a, vr_b \in V_r$) have a strong probability of being combined, thus, the weight of er_{ab} (from color

graph G_r) is modified into *Group 1*, which has a strong probability of being combined in object segmentation. The specific weight is calculated as follows:

$$ef_{ab} = \frac{TH_r}{\underset{er_{ab}\in\{VeryLow\}}{MAX(\cup er_{ab})}} \cdot er_{ab} \quad \ldots\ldots\ldots (9)$$

er_{ab} is an edge of graph G_r, and TH_r threshold points from the graph G_r histogram. $MAX()$ is a function for finding the maximum weight in graph G_r under its corresponding weight es_{ab} in the $\{Very\ Low\}$ group. ef_{ab} is an edge of graph $G_f(G_f = \{V_f, E_f\})$, and graph(G_f) will be segmented in the last block of our framework.

B) STRONG SPLITTING RULE

IF:	$es_{ab} \in [Ms_2, Es_{max}]$	{Very High}
THEN:	STRONG SPLITTING	{SS}

This update rule assumes that current pairs of neighbor nodes vr_a and vr_b ($vr_a, vr_b \in V_r$) have a strong probability of being divided into two different components, so the weight of er_{ab} from color graph G_r is modified into *Group 2*, which has a strong probability of being divided into two different objects in the next step. We calculate weight as follows:

$$ef_{ab} = \frac{\|MAX(\cup er_{ab}) - TH_r\|}{\underset{er_{ab}\in\{Very\ High\}}{MAX(\cup er_{ab})}} \cdot er_{ab} + TH_r \quad \ldots (10)$$

$MAX()$ is a function for finding the maximum weight in graph G_r under its corresponding weight es_{ab} in the $\{Very\ High\}$ group.

C) WEAK MERGING RULE

IF:	$es_{ab} \in [Ms_1, TH_s)$ AND $er_{ab} \in [0, TH_r)$	{Low} {Group1}
THEN:	WEAK MERGING	{WM}

This update rule, because es_{ab} belongs to $\{Low\}$ and er_{ab} belongs to $\{Group\ 1\}$, assumes that current pairs of neighbor nodes vr_a and vr_b ($vr_a, vr_b \in V_r$), tend to be combined, so we reduce the weight of graph G_r. This update rule is calculated as follows:

$$ef_{ab} = er_{ab} - \frac{\|TH_r - er_{ab}\|}{\underset{er_{ab}\in\{Group1\}\cap es_{ab}\in\{Low\}}{\|MAX(\cup d_{ab}) - MIN(\cup d_{ab})\|}} \cdot d_{ab} \quad (11)$$

$$d_{ab} = \underset{er_{ab}\in\{Group1\}\cap es_{ab}\in\{Low\}}{\|er_{ab} - TH_r\| + \|es_{ab} - TH_s\|} \quad \ldots (12)$$

D) WEAK SPLITTING RULE

IF:	$es_{ab} \in [TH_s, Ms_2)$ AND $er_{ab} \in [TH_r, Er_{max}]$	{High} {Group2}
THEN:	WEAK SPLITTING	{WS}

This update rule, Because es_{ab} belongs to $\{High\}$ and er_{ab} belongs to *Group 2*, assumes that current pairs of neighbor nodes vr_a and vr_b ($vr_a, vr_b \in V_r$), tend to be divided into two different objects, so we increase the weight of graph G_r. This update rule is calculated as follows:

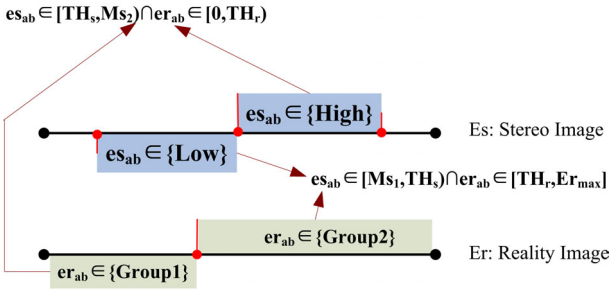


Fig. 6. *es* and *er* during the analysis rule.

$$ef_{ab} = er_{ab} + \frac{\|TH_r - er_{ab}\|}{\underset{er_{ab}\in\{Group2\}\cap es_{ab}\in\{High\}}{\|MAX(\cup d_{ab}) - MIN(\cup d_{ab})\|}} \cdot d_{ab} \quad (13)$$

$$d_{ab} = \underset{er_{ab}\in\{Group2\}\cap es_{ab}\in\{High\}}{\|er_{ab} - TH_r\| + \|es_{ab} - TH_s\|} \quad \ldots (14)$$

E) ANALYSIS RULE

Condition 1:	$es_{ab} \in [Ms_1, TH_s) \cap er_{ab} \in [TH_r, Er_{max}]$ {Low}∩{Group2}
IF:	$\|es_{ab} - TH_s\| > \|er_{ab} - TH_r\|$
THEN:	Tend to combine
ELSE:	Tend to split
Condition 2:	$es_{ab} \in [TH_s, Ms_2) \cap er_{ab} \in [0, TH_r)$ {High}∩{Group1}
IF:	$\|es_{ab} - TH_s\| > \|er_{ab} - TH_r\|$
THEN:	Tend to split
ELSE:	Tend to combine

This rule, because changes is physical location are not very low/high, optical changes would be major factor for object segmentation. Because es_{ab} and er_{ab} are located on both sides of the threshold, as shown in **Fig. 6**. We will analyze which is farther from the corresponding split on the histogram.

The update rule is calculated as follows:

$$ef_{ab} = \begin{cases} er_{ab} - (\|es_{ab} - TH_s\| - \|er_{ab} - TH_r\|) & Condition1: \\ er_{ab} + (\|es_{ab} - TH_s\| - \|er_{ab} - TH_r\|) & Condition2: \end{cases} \quad (15)$$

5. Experiments

This framework is based on two types of input images reality scene and stereo image. Experimental data came from Middlebury stereo image data (http://vision.middlebury.edu/stereo/data/). We selected three pairs of images to verify our algorithm in experiments. The first two pairs (**Figs. 7** and **8**) consisted of complex textures and many objects. For the third pair, in **Fig. 9**, we selected an image of objects with a similar texture.

Exp. 1

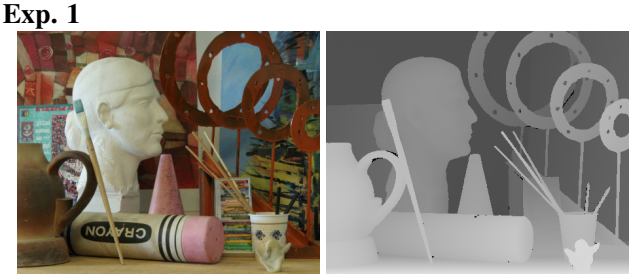


Fig. 7. Exp. 1 input images.

Exp. 2

Fig. 8. Exp. 2 input images.

Exp. 3

Fig. 9. Exp. 3 input images.

5.1. Parameter Settings

Our algorithm has three parameters, σ is used for the Gaussian filter, k is the merging intensity, and *size* is the minimum object size for object segmentation. We selected $k = 500$ for the maximum merging intensity to compare the result with different *size* from 100 to 500, and the step is 100. Parameter setting results are shown in **Figs. 10**-**12(a)**.

The Gaussian filter in our framework smooths digitization artifacts, and the Gaussian filter as $\sigma = 0.15$ in experiments, which produces no visible changes in the image. The object number result decreases with increasing merging intensity k, and converges at a stable order of magnitude. **Figs. 10**-**12(b)**-**(e)**, compares image results from a relatively stable convergence curve with the same parameter *size* minimum object size.

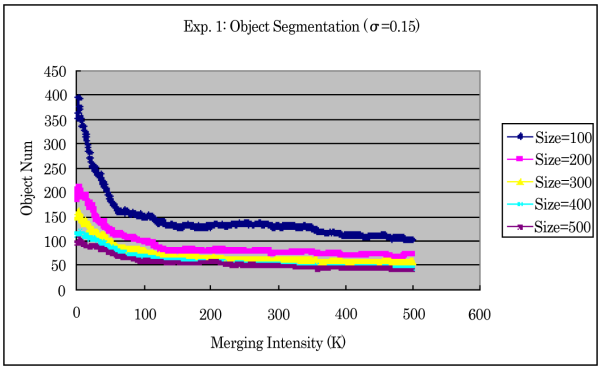
5.2. Results of Experiments

This section shows reasonable results for Experiments 1-3. The evaluation function is calculated as follows:

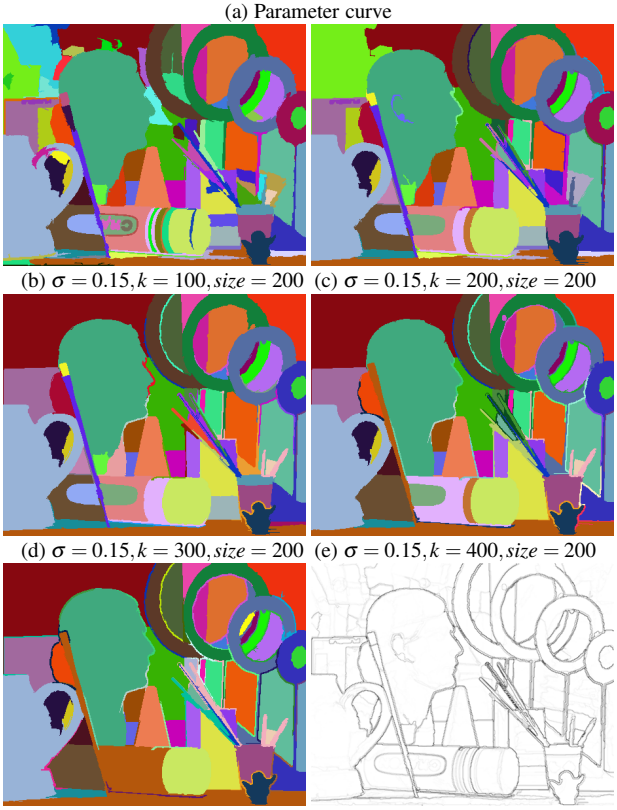
$$var[NUM(T-50):NUM(T)] \leq Threshold \quad . \quad (16)$$

$NUM()$ is the number of segmentation objects, T is the iteration number of merging intensity increasing, and $var()$ is the variance function. In Exps. 1 and 2, due to the two images consisting of complex textures and many objects, segmentation results are converge slowly. Curves are shown in **Figs. 10(a)** and **11(a)**. We defined the threshold as 0.5 and the *size* minimum object size as 300. For Exp. 3, which has a relatively simple image, the segmentation result converge is rapidly as shown **Fig. 12(a)**. We defined the threshold for Exp. 3 as 0.1 and the *size* minimum object size as 400. The three segmentation results are shown in **Figs. 13**-**15**.

Exp. 1



(a) Parameter curve



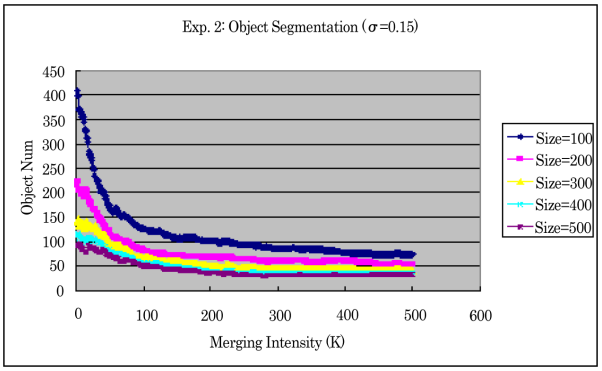
(b) $\sigma = 0.15, k = 100, size = 200$ (c) $\sigma = 0.15, k = 200, size = 200$

(d) $\sigma = 0.15, k = 300, size = 200$ (e) $\sigma = 0.15, k = 400, size = 200$

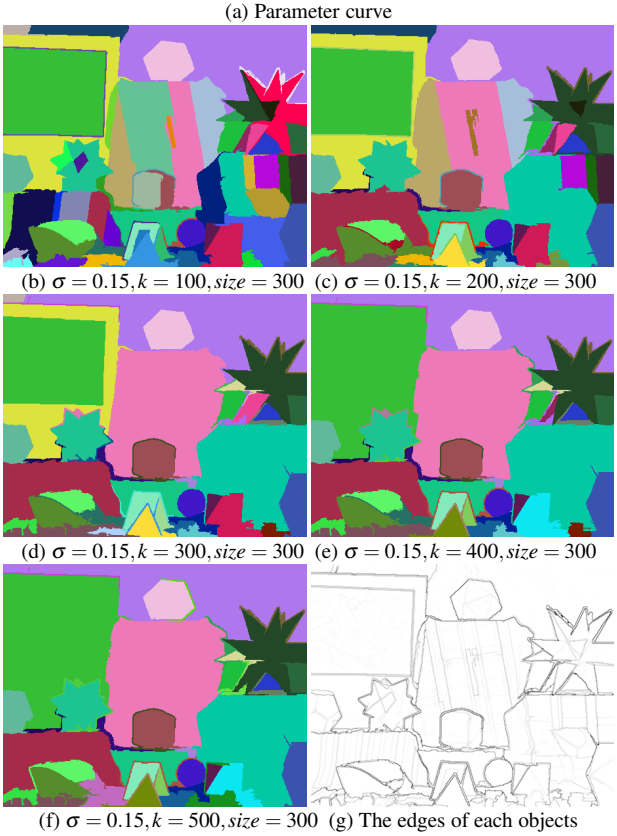
(f) $\sigma = 0.15, k = 500, size = 200$ (g) The edges of each objects

Fig. 10. Exp. 1 parameter curve and segmentation results.

Exp. 2



(a) Parameter curve



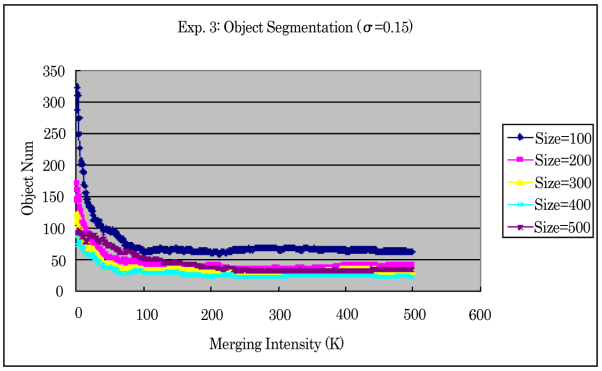
(b) $\sigma = 0.15, k = 100, size = 300$ (c) $\sigma = 0.15, k = 200, size = 300$

(d) $\sigma = 0.15, k = 300, size = 300$ (e) $\sigma = 0.15, k = 400, size = 300$

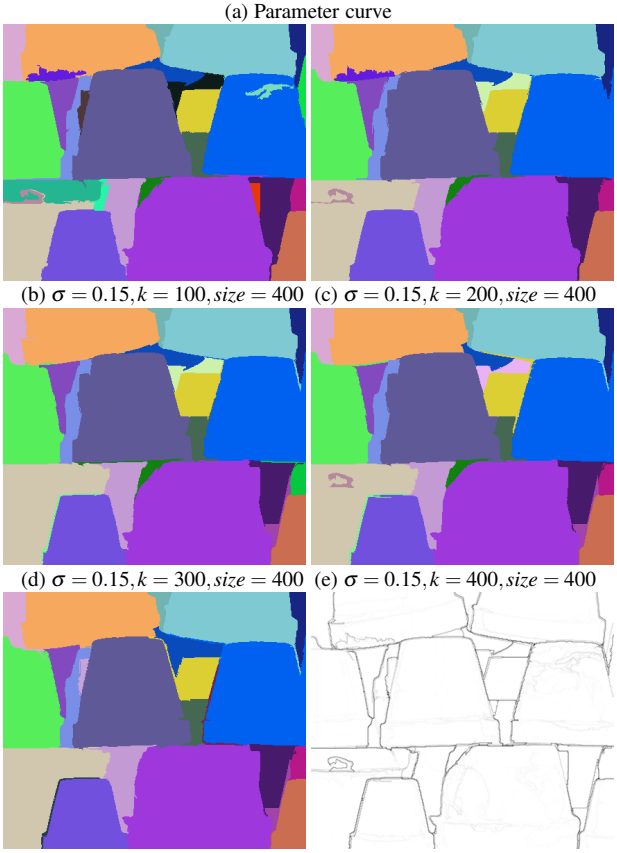
(f) $\sigma = 0.15, k = 500, size = 300$ (g) The edges of each objects

Fig. 11. Exp. 2 parameter curve and segmentation results.

Exp. 3



(a) Parameter curve



(b) $\sigma = 0.15, k = 100, size = 400$ (c) $\sigma = 0.15, k = 200, size = 400$

(d) $\sigma = 0.15, k = 300, size = 400$ (e) $\sigma = 0.15, k = 400, size = 400$

(f) $\sigma = 0.15, k = 500, size = 400$ (g) The edges of each objects

Fig. 12. Exp. 3 parameter curve and segmentation results.

5.3. Comparison of Results

We proposed a framework for segmenting objects using physical location, and this subsection we want to compare our framework to only reality image input framework. The segmentation algorithm of the two frameworks used the same segmentation algorithm, *efficient graph-based image segmentation*, which was proposed by Felzenszwalb [11]. This algorithm runs in $O(m \log m)$ time for m graph edges and is fast in practice, generally running within a fraction of a second [11]. Results of the comparison are shown as follows.

Figures 16-18(b) show the same parameters as our results, and **(c)** shows reasonable results with only reality image input.

6. Conclusions

We have proposed a framework for segmenting objects using spatial location information from stereo images. Our framework segmentation is based on an efficient graph-based image for combining changes in optical features and physical location to segment reality scenes into perceptually and semantically uniform regions. We proposed a rules tables for combining optical and spatial features together, which are extracted using k-means clustering. A series of reality-scene images demonstrate the performance of our framework and experimental data are from the Middlebury stereo image data (http://vision.middlebury.edu/stereo/data/).

Exp. 1

Fig. 13. Exp. 1 result ($\sigma = 0.15, k = 330, size = 300$).

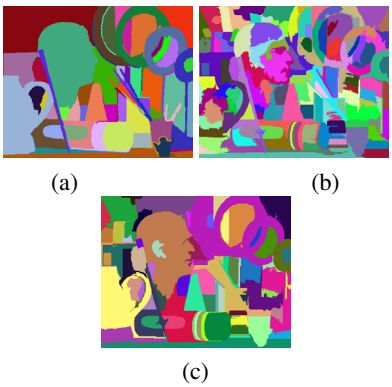
Exp. 2

Fig. 14. Exp. 2 result ($\sigma = 0.15, k = 340, size = 300$).

Exp. 3

Fig. 15. Exp. 3 result ($\sigma = 0.15, k = 350, size = 400$).

Exp. 1

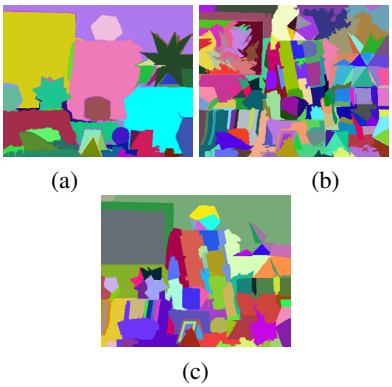


(a) (b)

(c)

Fig. 16. (a) Our result ($\sigma = 0.15, k = 330, size = 300$), (b) reality image only ($\sigma = 0.15, k = 330, size = 300$), (c) reality image only ($\sigma = 0.5, k = 500, size = 300$).

Exp. 2

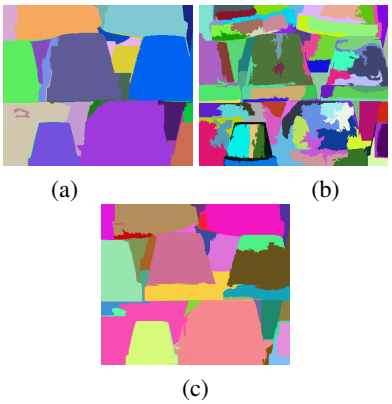


(a) (b)

(c)

Fig. 17. (a) Our result ($\sigma = 0.15, k = 340, size = 300$), (b) reality image only ($\sigma = 0.15, k = 340, size = 300$), (c) reality image only ($\sigma = 0.5, k = 400, size = 300$).

Exp. 3



(a) (b)

(c)

Fig. 18. (a) Our result ($\sigma = 0.15, k = 350, size = 400$), (b) reality image only ($\sigma = 0.15, k = 350, size = 400$), (c) reality image only ($\sigma = 0.5, k = 300, size = 400$).

Acknowledgements

This study is partly supported by Grant-in-Aid for Scientific Research #22700095 from Ministry of Education, Culture, Sports and Technology, Japan.

References:

[1] N. Otsu, "A threshold selection method from grey-level histograms," IEEE Trans. Syst., Man, Cybern., Vol.SMC-8, pp. 62-66, 1978.

[2] S. Belongie, C. Carson, H. Greenspan, and J. Malik, "Color and texturebased image segmentation using EM and its application to content based image retrieval," ICCV, pp. 675-682, 1998.

[3] D. K. Panjwani and G. Healey, "Markov random-field models for unsupervised segmentation of textured color images," PAMI, Vol.17, pp. 939-954, Oct. 1995.

[4] J. Wang, "Stochastic relaxation on partitions with connected components and its application to image segmentation," IEEE Trans. Pattern Anal. Machine Intell., Vol.20, No.6, pp. 619-636, 1998.

[5] L. Shafarenko, M. Petrou, and J. Kittler, "Automatic watershed segmentationof randomly textured color images," IEEE Trans. Pattern Anal. Machine Intell., Vol.6, No.11, pp. 1530-1544, 1997.

[6] W. Ma and B. S. Manjunath, "Edge flow: a technique for boundary detection and image segmentation," IEEE Trans. Image Processing, Vol.9, pp. 1375-1388, Aug. 2000.

[7] J. Shi and J. Malik, "Normalized cuts and image segmentation," IEEE Trans. Pattern Anal. Machine Intell., Vol.22, pp. 888-905, Aug. 2000.

[8] J. Sun, N.-N. Zheng, and H.-Y. Shum, "Stereo Matching Using Belief Propagation," IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.25, pp. 787-800, 2003.

[9] P.F. Felzenszwalb and D.P. Huttenlocher, "Efficient Belief Propagation for Early Vision," Proc. IEEE CS Conf. Computer Vision and Pattern Recognition, Vol.1, pp. 261-268, 2004.

[10] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," Int. J. Computer Vision, Vol.47, pp. 7-42, 2003.

[11] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," Int. J. of Computer Vision, Vol.59, No.2, pp. 167-181, 2004.

Appendix A.

Algorithm 1 The Algorithm of our Framework

Input:
  $I_r = \{ir_m | m = 0,1,2...N\}$ % Reality Image
  $I_r = \{is_m | m = 0,1,2...N\}$ % Stereo Image
Output:
  Num % The amount of Objects
  $I_o = \{io_m | m = 0,1,2...N\}$ % Result Image
  $I_e = \{ie_m | m = 0,1,2...N\}$ % The probability of current pixel $ie_m \in \{The\ boundary\ of\ any\ objects\}$
Parameters:
  σ, k_{max}, $size$
begin:
  $I_r \leftarrow gauss(I_r, \sigma)$;
  $I_s \leftarrow gauss(I_s, \sigma)$;
  $G_r = (V_r, E_r) \leftarrow graph(I_r)$;
  $G_s = (V_s, E_s) \leftarrow graph(I_s)$;
  $(Mr_1, Mr_2, TH_r) \leftarrow kmeans(E_r, k = 2)$;
  $(Ms_1, Ms_2, TH_s) \leftarrow kmeans(E_s, k = 2)$;
for each edges: $\begin{cases} er_{ab} \in E_r \\ es_{ab} \in E_s \end{cases}$ **do**
  if $es_{ab} \in \{Very\ Low\}$ **then**
    $G_f = (V_f, E_f) \leftarrow$ {Strong Merging}
  else if $es_{ab} \in \{Very\ High\}$ **then**
    $G_f = (V_f, E_f) \leftarrow$ {Strong Splitting}
  else if $es_{ab} \in \{Low\} \cap er_{ab} \in \{Group1\}$ **then**
    $G_f = (V_f, E_f) \leftarrow$ {Weak Merging}
  else if $es_{ab} \in \{High\} \cap er_{ab} \in \{Group2\}$ **then**
    $G_f = (V_f, E_f) \leftarrow$ {Weak Splitting}
  else if $(es_{ab} \in \{High\} \cap er_{ab} \in \{Group1\})$ OR $(es_{ab} \in \{Low\} \cap er_{ab} \in \{Group2\})$ **then**
    $G_f = (V_f, E_f) \leftarrow$ {Analysis}
  end if
end for
$T \leftarrow 0; k \leftarrow 1$
repeat
  $(NUM, I_o) \leftarrow Segment(k, size, G_f)$;
  for each pixel: $i_o \in I_o$ **do**
    if $io_m \in \{The\ boundary\ of\ any\ objects\}$ **then**
      $ie_m ++$;
    end if
  end for
  if $Var(NUM(T-50) : NUM(T)) \leq Threshold$ **then**
    break;
  end if
  $k++, T++$;
until $k < K_{max}$

Name:
Yong Hao

Affiliation:
Department of Computer Science and Engineering, Nagoya Institute of Technology

Address:
Gokiso-cho, Showa-ku, Nagoya, Aichi 466-8555, Japan

Brief Biographical History:
2002-2006 B.E., Xi'an University of Technology, China
2008-2010 M.E., Naogya Institute of Technology, Japan
2010- D.E., Naogya Institute of Technology, Japan

Name:
Lifeng He

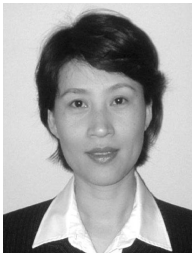
Affiliation:
Graduate School of Information Science and Technology, Aichi Prefectural University

Address:
Nagakute-cho, Aichi 480-1198, Japan

Brief Biographical History:
1997 The Ph.D. degrees in AI and Computer Science from Nagoya Institute of Technology, Japan
1999- Associate Professor, Aichi Prefectural University, Japan
2006-2007 Research Associate, the University of Chicago, USA

Main Works:
• "Fast Connected-Component Labeling, Pattern Recognition," Vol.42, pp. 1977-1987, 2009.
• "A Run-based Two-Scan Labeling Algorithm," IEEE Trans. on Image Processing, Vol.17, No.5, pp. 749-756, 2008.

Membership in Academic Societies:
• Associate for Automated Reasoning (AAR)



Name:
Yuyan Chao

Affiliation:
Graduate School of Environmental Management, Nagoya Sangyo University

Address:
Owariasahi-city, Aichi 488-8711, Japan

Brief Biographical History:
2000 The Ph.D degrees from Nagoya University, Japan
2000-2002 Special Foreign Researcher of Japan Society for the Promotion of Science in Nagoya Institute of Technology
2004-2009 Associate Professor in Nagoya Sangyo University, Japan

Main Works:
• "Fast Connected-Component Labeling," Pattern Recognition, Vol.42, pp. 1977-1987, 2009.
• "A Run-based Two-Scan Labeling Algorithm," IEEE Trans. on Image Processing, Vol.17, No.5, pp. 749-756, 2008.

Membership in Academic Societies:
• Japanese Society for Artificial Intelligence (JSAI)

Name:
Tsuyoshi Nakamura

Affiliation:
Associate Professor, Department of Computer Science and Engineering, Graduate School of Engineering, Nagoya Institute of Technology

Address:
Gokiso-cho, Showa-ku, Nagoya, Aichi 466-8555, Japan

Brief Biographical History:
1998 Ph.D., Nagoya Institute of Technology
1998- Research Associate, Nagoya Institute of Technology
2003- Associate Professor, Nagoya Institute of Technology

Main Works:
• "Color transfer based on hierarchical image-region matching with interactive evolutionary computation," Kansei Engineering Int., Vol.5, No.4, pp. 1-10, 2006.

Membership in Academic Societies:
• The Institute of Electrical and Electronics Engineering (IEEE)
• Association for Computing Machinery (ACM)
• The Institute of Electronics, Information and Communication Engineers (IEICE)
• Japan Society for Fuzzy Theory and Intelligent Informatics (SOFT)
• Japan Society of Kansei Engineering (JSKE)

Name:
Hidenori Itoh

Affiliation:
Professor, Department of Computer Science and Engineering, Graduate School of Engineering, Nagoya Institute of Technology

Address:
Gokiso-cho, Showa-ku, Nagoya, Aichi 466-8555, Japan

Brief Biographical History:
1974- Joined Nippon Telephone and Telegraph Laboratories
1985- Joined The Institute for New Generation Computer Technology
1989- Professor, Nagoya Institute of Technology

Main Works:
• "Rough Set Based FCM Algorithm for Image Segmentation," Int. J. of Computational Science, Vol.1, No.1, pp. 58-68, 2007.