Image Segmentation Evaluation Using an Integrated Framework

Kevin McGuinness*, Gordon Keenan*, Tomasz Adamek† and Noel E. O'Connor†

Centre for Digital Video Processing, Dublin City University, Glasnevin, Dublin 9, Ireland.
*{*kmcguinness, gkeenan*}*@computing.dcu.ie*, †{*adamekt, oconnorn*}*@eeng.dcu.ie*

Keywords: Image Segmentation, Video Signal Processing, Segmentation Evaluation, Software Package

Abstract

In this paper we present a general framework we have developed for running and evaluating automatic image and video segmentation algorithms. This framework was designed to allow effortless integration of existing and forthcoming image segmentation algorithms, and allows researchers to focus more on the development and evaluation of segmentation methods, relying on the framework for encoding/decoding and visualization. We then utilize this framework to automatically evaluate four distinct segmentation algorithms, and present and discuss the results and statistical findings of the experiment.

1 Introduction

Image segmentation is a core tool in numerous image processing, machine vision and content based multimedia retrieval applications and as a result has been the focus of intense research for many years (see [5] for a recent review). Indeed, so numerous are the approaches that have been proposed, that selecting an optimal algorithm for a particular application has become an arduous and very time consuming task. Even when an algorithm has been selected, adapting the method to accept/produce the required image or video format may require significant effort. The framework we have developed is intended to allow simple integration, visualization, evaluation and comparison of many different image and video segmentation algorithms, providing researchers with the means to select an appropriate algorithm for a given task. The usefulness of the framework is validated by applying it in a significant segmentation evaluation task.

The paper is organized as follows. In section 2 we briefly outline the architecture and features of the framework we have developed. In section 3 we give an overview of the segmentation algorithms that have been integrated into to the tool thus far. Section 4 details the evaluation experiment setup and presents an analysis and discussion of the results. Finally we present conclusions in section 5 and suggest future work in 6.

The framework was developed as part of the K-Space[1] project, and includes several state-of-the-art image and video segmentation techniques contributed by the various partners.

2 Segmentation Framework

This section gives a brief overview of the image and video segmentation framework we have developed. For the interested reader, a more detailed description can be found in [12].

2.1 Features and Functionality

The following describes the main features of the platform:

User Interface: The user interface provides automatic decoder selection, concurrent browsing of video frames and segmented images, selected-range segmentation and a simple interface for selecting algorithms and their parameters. Several useful segmentation visualization methods are also included. A screenshot of the interface is shown in figure 1.

Image and Video Formats: The framework was designed to support all common image and video formats transparently. A built-in video decoder capable of seek-able, frame accurate video decoding of various video formats, including MPEG-1, 2, 4, Motion-JPEG, Quicktime and WMF is provided. Also provided is an image decoder capable of decoding both individual images and sequences of key-frames transparently. It is capable of decoding all common image formats, including JPEG, PNG, PNM, GIF and BMP.

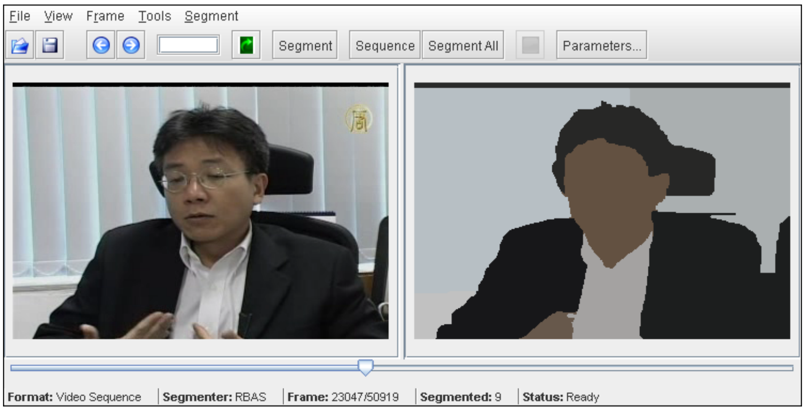
Region-Map Format: The framework encodes region-maps using an efficient, portable format based on a subset of PNG. This allows lossless storage of segmented video sequences with minimal space overhead.

Batch Processing Interface: The batch processing interface allows command line segmentation of large image/video collections. All the parameters that can be selected in the graphical user interface can be input into a parameter file. Files, ranges and increments can be selected for highly configurable segmentation.

MPEG-7 Output Generation: In order to allow interoperability of the produced region descriptions with other systems (e.g. content based information retrieval systems), a module is provided to generate an MPEG-7[1] description of a segmented image or video sequence. In the description each image is made up of a collection of still regions described by a bounding rectangle and a reference to the label mask. The region labeling is considered consistent throughout the video sequence.

In the future we plan to incorporate shot boundary detection into the application and to describe a video as a sequence of shots. This will allow for the formal description of both still and moving regions.

[1] K-Space - Knowledge Space of Semantic Inference for Automatic Annotation and Retrieval of Multimedia Content.



Format: Video Sequence | **Segmenter:** RBAS | **Frame:** 23047/50919 | **Segmented:** 9 | **Status:** Ready

Figure 1: Screenshot of the Application User Interface

2.2 Architecture Overview

The framework is arranged into three main layers. The top-level module, the Application, hosts the user interface, user preferences, batch processing interface and integration logic. The application layer implements all of its encoding, decoding and segmentation via interfaces specified in the module below this, the External API. This API consists of a set of interfaces for plug-in developers, as well as commonly required utilities to simplify development. The bottom layer contains of all the plug-ins, that is, all of the segmentation algorithms and codecs. For a more detailed architecture description, see [12].

3 Segmentation Algorithms

In this section we present an overview of the segmentation algorithms currently integrated into the framework. Figure 2 gives some sample images demonstrating the typical output of these algorithms.

Modified RSST using Syntactic Features
This algorithm [2] is based on the well known Recursive Shortest Spanning Tree [13] method utilizing the more perceptually uniform L*u*v* color model and syntactic visual features [4] to improve the quality of the segmentation. The syntactic features represent geometric properties of regions and their spatial configurations, and in a sense, attempt to model some of the gestalt grouping factors [10] that have been observed in the human visual system. The features used by the algorithm are homogeneity, compactness, regularity and inclusion.

Homogeneity refers to spatial color changes in the image. The algorithm begins by forming an initial over-segmentation using a simple euclidean distance merging predicate, then switches to a more appropriate color model for these larger segments. The latter model accounts for outliers, and controls over segmentation of objects comprised of gradients using the boundary melting approach [17].

The regularity, compactness and inclusion criteria model the spatial and geometric properties of regions, favoring region configurations with smoother and more convex objects, similar to most found in natural scenes.

Spatio-Temporal Region Adjacency Graphs
This method [8] aims to produce segmentations of a moving images with a high degree of temporal coherency by operating iteratively on pairs of frames using a hierarchical merging process. An initial spatial segmentation based on color is formed using a fast Minimum Spanning Tree approach. To compensate for the fact that the initial segmentation may differ significantly for each successive frame, as a frame is processed, the initial segmentation is constrained using an edge detector so that a given frame is over-segmented with respect to the previous frame. Feature points [16] are extracted and tracked for each frame, allowing grouping of temporally connected regions. Grid based space-time merging groups unmatched regions and a subgraph matching stage is used to complete the procedure.

Optimized Mean Shift
Proposed in [3], this algorithm is based on the popular mean shift segmentation algorithm described in [6] with three specific modifications intended to improve performance and temporal stability. The first of these optimizations improves the performance of the algorithm by performing a moderate quantization of the L*u*v* color space prior to segmentation. As observed in [3], this quantization does not significantly effect the quality of the segmentation but does however result in a more sparse feature space in which nearby feature vectors have been effectively grouped by the quantization. The resulting sparse feature space induces a significant performance gain, as less data now has to be considered in the search window when computing the mean shift vector.

The next optimization serves both to improve the performance of the segmentation and to improve it's temporal stability when applied to moving images. When computing the mean shift, the objective is to determine the significant modes in joint spatio-color space. Once these modes, or cluster centers, have been determined in frame F_{i-1} they can be propagated and used as the initial estimates for the modes in frame F_i. Under the assumption of a similar spatio-color space between adjacent frames, these modes will now more quickly converge on the true modes of frame F_i using the mean shift procedure. The algorithm also includes a check for any situation in which the feature spaces of adjacent frames are not similar (after a shot cut, for example) to avoid propagating cluster centers between dissimilar frames.

The final optimization of the original mean-shift algorithm occurs in the post-processing step. Instead of clustering small regions to larger ones in scan-line order, as originally proposed, the improved algorithm groups regions starting with

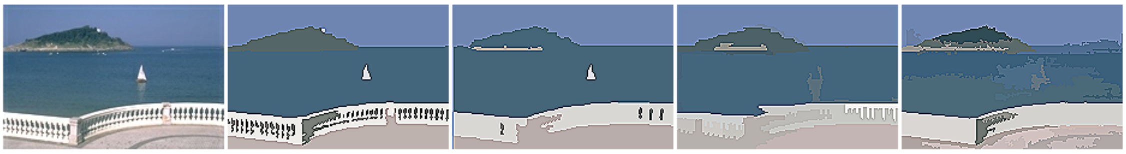


Figure 2: Sample output of some of the segmentation algorithms. From left to right: Original, M-RSST, SRG, ST-RAG, MSHIFT

the smallest, making the algorithm independent of raster order thus improving its stability. The aforementioned color space quantization is also exploited in the post-processing step to improve performance when re-assigning pixels that have been placed in incorrect clusters.

Statistical Region Merging
The statistical region merging algorithm, proposed in [14] is based upon the graph formulation of image processing. In their paper, Nielson and Nock formulate image segmentation as a statistical inference problem, and derive a simple merging order and merging predicate that can achieve, with high probability, a low error in segmentation.

If an image is viewed as a graph, where each pixel denotes a node in the graph, and each node is connected via a link to every other node in it's 4-neighborhood, then a segmentation can be achieved by successively merging connected nodes. With this formulation, it suffices to specify a predicate to determine whether or not to merge two nodes, and the order in which this predicate should be tested, in order to completely specify the segmentation algorithm. The test to determine whether or not two adjacent regions should be merged is known as *merging predicate*, and the order in which these tests are carried out is called the *merging order*.

Nock and Nielson define the merging predicate so as to provide a quantitative bound on the segmentation error as follows. Given a region R in an image I, let

$$b(R) = g\sqrt{\frac{1}{2Q|R|}\ln\left(\frac{1+|R|}{\delta}\right)\min\left(g,|R|\right)} \qquad (1)$$

where g is the range of the color band (usually $g = 256$) and $\delta = 1/(6|I|^2)$. The merging predicate is then defined as:

$$\mathscr{P}(R,R') \Leftrightarrow |\bar{R}' - \bar{R}| \le \sqrt{b^2(R) + b^2(R')} \qquad (2)$$

where $\bar{R}$ is a the value of color band (ex. red), and the predicate $\mathscr{P}(R,R')$ must be true for all color bands.

Given the merging predicate, the merging order is given as an invariant A which is defined as follows: If a test between any two true regions occurs, it implies that all tests between pairs of regions contained within these regions has occurred previously. For RGB color images the invariant is realized by sorting the links to be tested according to a weight W_{ij} equal the maximum absolute difference between each of the color bands, i.e.,

$$W_{ij} = \max(|R_i - R_j|, |G_i - G_j|, |B_i - B_j|) \qquad (3)$$

and testing the merging predicate on links in the implied order.

The SRM algorithm then proceeds as follows. Links are first formed between pixels in given a 4-neighborhood system, and weighted as W_{ij} defined above. These links are then sorted according to their edge weight. Then a single pass through the links is performed, merging the corresponding regions if the merging predicate is satisfied. For RGB colors $W_{ij} \in [0..255] \in \mathbb{Z}$, and so the sorting procedure can be performed in $O(n)$ using the bucket sort algorithm [7], and thus the entire algorithm runs in linear time.

4 Evaluation

In this section we detail the evaluation experiment setup and present an analysis and discussion of the results. This serves as a validation of the utility of the platform, as well as relating results that are intended to be a practical resource for the research community in general.

4.1 Dataset and Ground Truth

The experiments were carried out using the images and ground truth segmentations in the Berkeley segmentation dataset [11]. This excellent dataset is comprised of 300 images, consisting of a test set of 100 images and a training set of 200 images. For each image, the dataset contains an average of ≈ 11 corresponding human generated ground truth segmentations.

For each machine segmented image, we evaluate against all available ground-truths, presenting both the best-match and mean results. The best-match results can be interpreted as accuracy when we consider a segmentation to be correct if it matches any human generated segmentation. On the other hand, the mean result is less sensitive to any particular human segmenter, and can be interpreted as a measure of how much a segmentation corresponds to the type of segmentations that are typically produced by humans.

4.2 Algorithms Evaluated

We evaluated all four region based segmentation algorithms, referred to to as follows:

- *M-RSST*: Modified RSST using Syntactic Features.
- *ST-RAG*: Spatio-Temporal Region Adjacency Graphs
- *MSHIFT*: Optimized Mean Shift
- *SRM*: Statistical Region Merging

Also, in order to form a "worst-case" baseline, we also carried out the evaluation on mismatched segmentations, denoted algorithm MM. That is, compatible pairs (in terms of size) of image segmentations derived from different scenes were randomly selected, and we computed the evaluation measures on these mismatched pairs. For all of the above algorithms the default/recommended parametrization was used. Because of this, none of the algorithms mentioned requires any special training data, thus, for the experiments, we also separately evaluated against the training data set.

Table 1: GCE/LCE mean and variance for the various algorithms (given as percentages for clarity).

		GCE				**LCE**			
		Best		**Mean**		**Best**		**Mean**	
Algorithm	**Data**	μ	σ^2	μ	σ^2	μ	σ^2	μ	σ^2
M-RSST	*Test*	11.32 ± 1.60	0.65	18.10 ± 1.81	0.83	7.90 ± 0.99	0.25	12.20 ± 1.29	0.42
M-RSST	*Train*	10.03 ± 0.95	0.46	16.81 ± 1.15	0.67	6.98 ± 0.63	0.20	10.63 ± 0.73	0.27
ST-RAG	*Test*	14.60 ± 1.75	0.78	22.64 ± 1.94	0.96	10.66 ± 1.24	0.39	15.24 ± 1.40	0.50
ST-RAG	*Train*	12.20 ± 1.16	0.69	20.00 ± 1.28	0.84	9.13 ± 0.79	0.32	13.35 ± 0.84	0.36
SRM	*Test*	16.14 ± 1.73	0.76	24.47 ± 1.99	1.01	10.73 ± 1.22	0.38	15.52 ± 1.47	0.55
SRM	*Train*	14.62 ± 1.22	0.76	22.80 ± 1.32	0.89	10.17 ± 0.90	0.41	14.77 ± 1.00	0.51
MSHIFT	*Test*	25.74 ± 2.48	1.56	35.43 ± 2.58	1.69	19.89 ± 1.91	0.93	27.03 ± 2.19	1.22
MSHIFT	*Train*	24.68 ± 1.64	1.38	33.58 ± 1.81	1.67	19.31 ± 1.43	1.04	25.75 ± 1.64	1.37

4.3 Evaluation Methodology

Each algorithm is evaluated using two separate segmentation evaluation methodologies suited to region based image segmentation.

The first methodology, proposed in [11], is designed such that if one segmentation is a refinement of another, that is, if it is approximately the same except that it has a higher level of detail, then the error measure produced should be very small. The method produces two such measures of segmentation error, known as local and global consistency error, based on a definition of local refinement error. Given two segmentations of the same image: S_1 and S_2, if $R(S, p_i)$ corresponds to the set of pixels containing pixel p_i in S then the local refinement error is defined as:

$$E(S_1, S_2, p_i) = \frac{|R(S_1, p_i) \setminus R(S_2, p_i)|}{|R(S_1, p_i)|} \tag{4}$$

Using the above definition, the global (GCE) and local (LCE) consistency error are defined as:

$$\text{GCE} = \frac{1}{A} \min \left\{ \sum_{\text{all pixels } p_i} E(S_1, S_2, p_i), \sum_{\text{all pixels } p_i} E(S_2, S_1, p_i) \right\} \tag{5}$$

$$\text{LCE} = \frac{1}{A} \sum_{\text{all pixels } p_i} \min\{E(S_1, S_2, p_i), E(S_2, S_1, p_i)\} \tag{6}$$

where A is the area of the image, in pixels. Both measures are in the range $[0..1]$, where values closer to zero denote a better segmentation. It has been observed that these measures, in general, correspond well with human perception and usually produce very low values when used to compare different human segmentations of the same scene. However, care should be taken in interpreting the measures as due to their tolerance of refinement, they are not sensitive to over- and under-segmentation. Indeed, in the extreme case of over-segmentation, where each pixel is considered a separate region, $GCE = LCE = 0$.

In order to also consider refinement error, we employ another performance measure based on the Hamming distance between non-maximally intersecting regions proposed in [9]. Let S and T be two segmentations of the same image, and $S = \{S_1, \ldots, S_m\}$ and $T = \{T_1, \ldots, T_n\}$ where S_i corresponds to the set of pixels in region i from segmentation S. We associate with each region S_i a region T_k such that $S_i \cap T_k$ is maximal. The Hamming distance between two segmentations is defined as:

$$\mathrm{D}_H(T \Rightarrow S) = \sum_{S_i \in S} \sum_{T_j \neq T_k} |S_i \cap T_j| \tag{7}$$

which corresponds to the sum of areas of intersection for all non-maximally intersecting regions. Given this definition, the performance measure is given by:

$$p = 1 - \frac{\mathrm{D}_H(T \Rightarrow S) + \mathrm{D}_H(S \Rightarrow T)}{2A} \tag{8}$$

We will denote this the Huang-Dom measure (HD). Again, the measure is in the range $[0..1]$, where this time values closer to one denote a better segmentation.

4.4 Results and Analysis

In total, evaluating 4 distinct algorithms on a database of 300 images, each having ≈ 11 ground truth images resulted in 13,070 individual comparative evaluations. Table 1 and 2 tabulate the means and variances of the performance metrics computed with respect to the evaluated algorithms over both collections. The given mean confidence intervals are computed assuming a normal distribution and using a confidence coefficient $\alpha = 95\%$. Table 3 contains the means and variances computed using the mismatched segmentations, as mentioned previously, and can be interpreted as a baseline. The values are given as percentages, so as to make them more readable. Lower values for GCE and LCE are favorable, whereas a higher Huang-Dom measure is preferred. As mentioned earlier, we compute the performance measures on both the best match ground truth segmentation available and the mean value over all ground truths for each image. These values are tabulated under the *Best* and *Mean* columns in the tables.

In terms of mean GCE and LCE, the M-RSST algorithm appears to be the best performing of the four with the ST-RAG algorithm in second place marginally outperforming the SRG algorithm. According to the Huang-Dom measure, the M-RSST, ST-RAG and SRG algorithms appear comparable, but MSHIFT is heavily penalized for its tendency to over-segment images. However, it should be considered that some of the tested algorithms, in particular MSHIFT and ST-RAG, are optimized specifically for moving image segmentation, and it is thus expected that a performance evaluation that considers the temporal consistency of a segmentation algorithm would likely

Table 2: Huang-Dom mean and variance for the algorithms

		Huang-Dom			
		Best		Mean	
	Data	μ	σ^2	μ	σ^2
M-RSST	*Test*	78.3 ± 2.1	1.15	74.1 ± 2.2	1.17
M-RSST	*Train*	80.6 ± 1.5	1.09	76.3 ± 1.5	1.07
ST-RAG	*Test*	78.4 ± 1.7	0.74	73.7 ± 1.8	0.86
ST-RAG	*Train*	79.2 ± 1.2	0.79	75.1 ± 1.3	0.82
SRM	*Test*	78.8 ± 1.5	0.58	73.7 ± 1.6	0.66
SRM	*Train*	79.2 ± 1.6	0.67	74.7 ± 1.2	0.75
MSHIFT	*Test*	66.9 ± 2.0	1.03	62.7 ± 2.0	1.02
MSHIFT	*Train*	68.5 ± 1.5	1.17	64.5 ± 1.5	1.19

Table 3: Baseline mean and variance on mismatched frames

	GCE		LCE		HD	
	Test	**Train**	**Test**	**Train**	**Test**	**Train**
μ	35.65	34.61	27.95	34.61	58.52	59.17
σ^2	2.30	2.24	1.39	2.24	0.69	0.79

assign these methods a much higher performance measure. Plots of GCE and Huang-Dom performance measures for the test set, ranked best to worst from left to right are shown in figures 3 and 4.

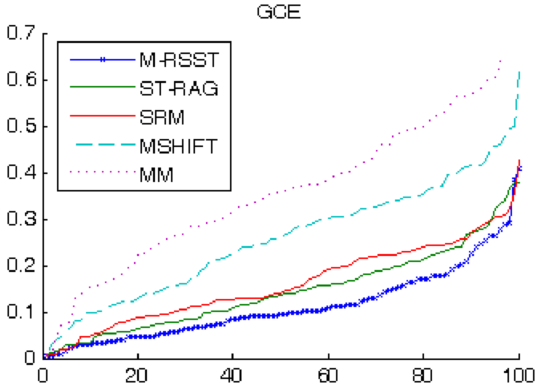


Figure 3: Comparison of GCE values

It is worth noting that the typical mean GCE and LCE error rates for humans, as noted in [11], are 11% and 7% respectively. This figures can be directly compared with the mean (not best) values in table 1. It is clear from the comparison that none of the algorithms achieves segmentation as accurate as a human.

We also examined, over the entire set of experiments, the correlation between GCE, LCE and Huang-Dom measures. It was observed that the GCE and LCE measures are especially highly correlated, with a correlation coefficient of 86.7%, a fact not reported in [11]. This implies that it may not be necessary to use both measures when performing evaluation. The choice of measure should depend on whether mutual refinement with respect to a ground-truth is acceptable, in which case LCE is appropriate. Tolerance of mutual refinement is a task dependant consideration. However, different humans tend to produce segmentations that are mutual refinements of one another, thus LCE is probably more appropriate for perception

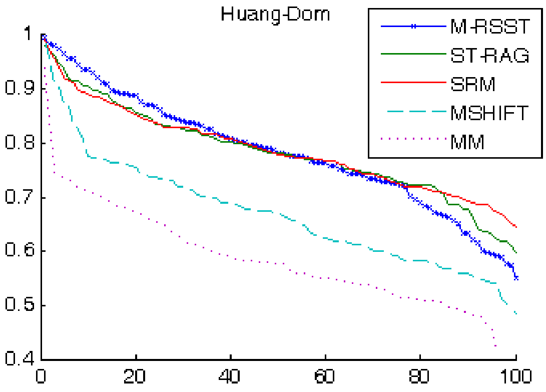


Figure 4: Comparison of Huang-Dom values

based evaluation. A plot of the GCE vs LCE values for a typical experiment (ST-RAG on test collection) is shown in figure 5.

In contrast, the Huang-Dom and GCE/LCE correlation coefficient was computed to be -47.9% and -58.7% respectively, supporting the hypothesis that the measure conveys important information not contained in the GCE and LCE measures, specifically by being more sensitive to under- and over-segmentation, and thus is effective as a complementry performance measure.

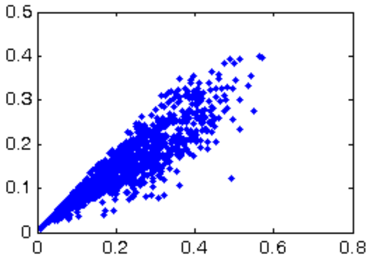


Figure 5: Correlation of GCE and LCE values

It is also interesting to consider the histogram distributions of the performance measures. A plot of the Huang-Dom performance measure histogram for several segmentation algorithms run on the test data set is shown in figure 6. As can be seen, the distributions appear approximately Gaussian in nature, and there seems to be a clear separation between correct and mismatched segmentations.

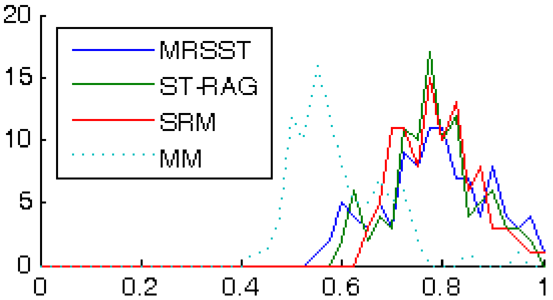


Figure 6: Huang-Dom distribution for several algorithms

5 Conclusions

We have developed a generic framework for automatic image and video segmentation. Several state-of-the-art algorithms have been integrated, and this was exploited in performing an extensive automatic image segmentation evaluation experiment. Two ground-truth based evaluation methodologies were chosen for the experiment, and the complementary nature of the methods was demonstrated in the results. High correlation between GCE and LCE values was made evident, implying that both measures my not be necessary for gauging performance. The statistics presented clearly demonstrate and contrast the effectiveness of each segmentation algorithm, it is hoped that such results will aid other researchers in selecting the most appropriate segmentation method for a given application.

6 Future Work

In this paper we did not explore standalone segmentation evaluation, that is, performance indicators that measure the *empirical goodness* [19] of an algorithm without using ground-truth. In the future, we plan to implement several of these standalone measures (for instance [18]) and compare the statistical distributions of the methods with those of the ground-truth based evaluation paradigms.

We have also recently integrated a semi-automatic image segmentation algorithm based on the binary partition tree approach proposed in [15] into the tool. This particular segmentation algorithm is object based, and as such, produces a “foreground-background” segmentation. In the future, we would like to integrate semi-automatic segmentation algorithms capable of producing multiple regions, allowing fast generation of ground-truth for experiments.

Acknowledgements

This material is based upon work supported by by the European Commission under contract FP6-027026, K-Space: Knowledge Space of semantic inference for automatic annotation and retrieval of multimedia content.

We would also like to acknowledge the contribution of segmentation algorithm implementations from the various K-Space partners, specifically Joanneum Research and Eurecom.

References

[1] Mpeg-7. multimedia content description interface. Standard No. ISO/IEC n.15938, 2001.

[2] T. Adamek, N. O’Connor, and N. Murphy. Region-based segmentation of images using syntactic visual features. In *6th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS’05)*, 2005.

[3] W. Bailer, P. Schallauer, H. B. Haraldsson, and H. Rehatschek. Optimized mean shift algorithm for color segmentation in image sequences. *Proc. of SPIE*, 5685:522–529, March 2005.

[4] C. F. Bennstrom and J. R. Casas. Binary-partition-tree creation using a quasi-inclusion criterion. In *Proceedings of the Eighth International Conference on Information Visualization (IV)*. IEEE Computer Society Press, 2004.

[5] H. D. Cheng, X. H. Jiang, Y. Sun, and J. Wang. Color image segmentation: Advances and prospects. *Pattern Recognition*, 34(12):2259–228, December 2001.

[6] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. on Pat. Anal. and Mach. Intel.*, 24(5):603–619, 2002.

[7] T. H. Cormen, S. Clifford, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, 2 edition, 2001.

[8] E. Galmar and B. Huet. Graph-based spatio-temporal region extraction. In *3rd International Conference on Image Analysis and Recognition (ICIAR’06)*, pages 18–20, Sep 2006.

[9] Q. Huang and B. Dom. Quantitative methods of evaluating image segmentation. In *International Conference on Image Processing (ICIP’95)*, volume 3, pages 53–56, Los Alamitos, CA, USA, October 1995.

[10] K. Koffka. *Principles of Gestalt Psychology*. Harcourt, New York, 1935.

[11] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the 8th International Conference Computer Vision (ICCV’01)*, volume 2, pages 416–423, Los Alamitos, CA, USA, July 2001. IEEE Computer Society.

[12] K. McGuinness, G. Keenan, T. Adamek, and N. O’Connor. A framework for integrating and evaluating automatic region-based segmentation algorithms. In *Proceedings of The First International Conference on Semantics And Digital Media Technology (SAMT’06)*, 2006.

[13] O. Morris, M. Lee, and A. Constantinides. Graph theory for image analysis: an approach based on the shortest spanning tree. In *IEE Proceedings F. Communications, Radar and Signal Processing*, volume 133, pages 146–152, April 1986.

[14] R. Nock and F. Nielsen. Statistical region merging. *IEEE Trans. on Pat. Anal. and Mach. Intel.*, 26(11):1452–1458, November 2004.

[15] P. Salembier and L. Garrido. Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. *IEEE Transactions on Image Processing*, 9:561–576, 2000.

[16] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR’94)*, pages 593–600, Seattle, WA, USA, June 1994.

[17] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. PWS, 2nd edition, 1998.

[18] H. Zhang, J. E. Fritts, and S. A. Goldman. An entropy-based objective evaluation method for image segmentation. In *Proceedings of IS&T/SPIE’s 16th Annual Symposium on Electronic Imaging Conference on Storage and Retrieval Methods and Applications for Multimedia, SPIE*, volume 6307, January 2004.

[19] Y. J. Zhang. A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29(8):1335–1346, August 1996.