Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Image and Signal Processing (CIISP 2007)

Towards Image Retrieval by Texture Segmentation with Genetic Programming

Vic Ciesielski and Djaka Kurniawan and Andy Song
School of Computer Science and Information Technology
RMIT University
Melbourne, Australia, 3001
Email: {vc,dkurniaw,asong}@cs.rmit.edu.au

***Abstract*— This paper examines the feasibility of an approach to image retrieval from a heterogeneous collection based on texture. For each texture of interest (T), a T-vs-other classifier is evolved for small $n \times n$ windows using genetic programming. The classifier is then used to segment the images in the collection. If there is a significant contiguous area of T in an image, it is considered to contain that texture for retrieval purposes. We have experimented with sky and grass textures in the Corel Volume 12 image set. Experiments with a single image indicate that classifiers for the two textures can be learned to a high accuracy. Experiments with a test set of 714 Corel images gave a retrieval accuracy of 84% for both sky and grass textures. These results suggest that the use of texture could enhance retrieval accuracy in content based image retrieval systems.**

I. Introduction

Texture is rarely used in image retrieval. One reason for this is that conventional texture processing tends to be computationally expensive. It is expensive because conventional texture classification is a two step process that requires features to be computed for some region of an image and then a classifier applied to the feature vector to determine the actual texture. This computation must then be repeated on many regions of the image.

In previous work we have used genetic programming to evolve one step texture classifiers which were very fast and accurate [1]. These classifiers were then used to perform fast image segmentation. They were tested on synthetic images composed of regions of textures from the Brodatz album [2]. On this set of images, fast, quite accurate texture segmentation was achieved. Figure 1 shows the segmentation of an image constructed from the Brodatz grass texture and one of the Brodatz cloud textures. In figure 1b, the segmented image, the grass area is rendered in black and the cloud area in white. While the segments are not absolutely aligned with the original image, all of the texture regions have been captured. This is a two class problem in which each class is a texture of interest.

Our aim in this paper is to determine whether the the same approach can be effective when one class is the texture of interest and the other class is anything other than this class. Such a facility could add more capability to content based image retrieval systems [3]. In current systems it is not possible to find, for example, images containing waterfalls or surf breaking on a beach. However, waterfalls and surf have a distinct texture which is detected by most humans. Our hypothesis is that it will be possible to learn classifiers for such textures from example images and then use these classifiers for image retrieval in a large heterogeneous image data base.

(a) Original Image (b) Segmented Image

Fig. 1. Segmentation of grass and cloud mosaic

In this paper we use the Corel Volume 12 image data set. This is a collection of about 2,500 images. They are the kinds of photographs a tourist might take while on holiday in another country. There are images from a number of countries including Nepal, France, Sweden and the USA. The image size is 384×256 pixels.

In this paper we investigate the hypothesis for two relatively simple textures, sky and grass. In particular we are interested in:

1) How accurately can we learn classifiers for *sky-vs-other* and *grass-vs-other* for a single image?
2) How accurate is the segmentation based on these classifiers?
3) How do (1) and (2) scale up to a larger number of heterogeneous images?
4) How accurate is image retrieval using the classifiers?

II. Related Work

While there is a considerable history of work on visual texture, the definition of texture is still imprecise. However, it is generally agreed that a texture is spatially homogeneous and contains repeated visual patterns. In synthetic textures, such as horizontal lines, vertical lines or a checkerboard, the basic structure is repeated exactly. In natural textures, such as grass, wood, sand or rocks, there is some random variation in size, shape, intensity or colour in the repetitions of the basic structure. This basic repeating unit is sometimes called a *texel*. Most of the work in the texture field is on grey level images because texture is primarily a spatial intensity property rather than a colour property.

1-4244-0707-9/07/$25.00 ©2007 IEEE

The texture classification problem involves images of single textures and requires a method of distinguishing the different texture classes, for example images of clouds from images of grass. This is conventionally done in two steps, first a feature vector is extracted from each image, and second the feature vector is given to a classifier to determine the class. There are currently many features to choose from, including Haralick features [4], Laws masks [5] and various wavelet transforms [6]–[9]. A major disadvantage of this approach is that computing feature values is expensive. In previous work [1], [10], [11] we have developed a one step approach in which the image pixels are used directly in genetic program classifiers. The classifiers are not quite as accurate as the best conventional approaches, but much faster.

The texture segmentation problem involves images with a number of different texture regions and the task is to identify the different regions as in figure 1. Usually this involves applying a classifier to a large number of smaller regions in the image, which makes conventional approaches very slow. [1] describes an approach using one step classifiers in which the reduced accuracy is compensated by a voting procedure.

There has been some previous work on using texture in image retrieval. Glatard et al. [12] describe a conventional application of texture in image retrieval. A bank of 16 Gabor features is computed for each 8×8 sub image and incorporated into the feature vector representing an image. Similarity for retrieval purposes is measured by Euclidean distance. A rigorous evaluation has not been done, but a number of cases of good performance are given.

Rubner et al. [13] describe an image similarity metric, earth mover's distance, and describe its use in the retrieval of images from a data base of images of animals. There are 500 grey level images in the data base and two examples of good performance in retrieving images of zebras and cheetahs are given. A query is posed by marking a rectangular texture patch in a query image and the system returns images which have a significant area of a similar texture. A number of Gabor features are used to determine texture similarity.

Howarth and Rüger [14] evaluated three texture feature sets for CBIR. Those texture features were co-occurence matrices, Tamura features and Gabor filters. The evaluation and implementation were focussed on query-by-example image retrieval. They concluded that the co-occurence matrices showed solid performance but the other two texture feature sets did not.

III. Texture Classification and Segmentation with Genetic Programming

To achieve the segmentation shown in figure 1 it is first necessary to determine a window size and to evolve a classifier for texture windows of this size. Once the classifier is available, segmentation is then achieved by sweeping the classifier across the image. Algorithms for both of these tasks are given below.

A. Classification

Given any image of a single texture T, any sub image is also an image of texture T as long as it is not too small. For example, any $n \times n$ window cut from the grass area of figure 1 is a grass texture as long as it is not smaller than a blade of grass. In order to evolve a classifier it is necessary to choose a sub image (or window) size. Some judgment is needed in choosing the window size. If the window is too small then the variation in a texture cannot be captured. If it is too large the segmentations tend to have ragged edges. For the image in figure 1 we have used a window size of 16×16.

Once the window size is chosen, $n \times n$ cutouts are randomly extracted to create the training data. This data is then used to evolve a classifier using the configuration shown in table I.

TABLE I
Genetic Programming Configuration for a *T1-vs-T2* classifier.

Parameter	Value
Population Size	200
Crossover Rate	0.90
Mutation Rate	0.0
Elitism Rate	0.10
Max Generations	150
Selection	Proportional to fitness
Termination	150 generations, or 100% fitness
Replacement	Generational replacement
Fitness	Classification accuracy on the training examples
Functions	$+, -, \times, /, \geq, \leq, =, if, between$
Terminals	Individual pixels of an $n \times n$ window, random constants

B. Segmentation

Our texture segmentation method is a supervised, region identifying method. The algorithm is shown in Figure 2. The first step is evolving a population of texture classifiers which can differentiate one texture from another, as described above.

The most accurate evolved classifier is then used for segmentation. It takes an $n \times n$ window of the image as input. The classifier starts from the top left corner of a target image and sweeps through the whole image. At any position during the sweeping, the classifier will determine whether the covered region is Texture 1 or Texture 2, and label the pixels of that region accordingly. At the completion of the sweeping process, most of the pixels in the target images will have multiple labels due to the overlap in the sweeping. The final class of a pixel is determined by voting strategy - the majority of the labels of the pixel.

Figure 1b is the binary image generated by applying this algorithm to figure 1a using a window size of 16, 1300 training examples, and a step size of 1.

IV. Segmentation of a Single Corel Image

In the previous section we established that the proposed approach works reasonably well for a mosaic of two textures. In this next section we address the situation where there are two textures of interest and other non textural regions in an image. The image is shown in figure 3a. The two textures of interest are sky and grass and the building comprises the rest of the image. In order to find the grass segment we adapt the above approach in the following way:

Input:	$T1, T2$	Textures for which example images are available
	I	An image containing only a number of regions of textures T1 and T2
	w,h	width and height of I
	n	sub-image size
	d	step size for moving window, $1 < d \leq n/2$
Output:	O	a two colour image $O_{ij} = colour1$ for texture 1 and $O_{ij} = colour2$ for texture 2

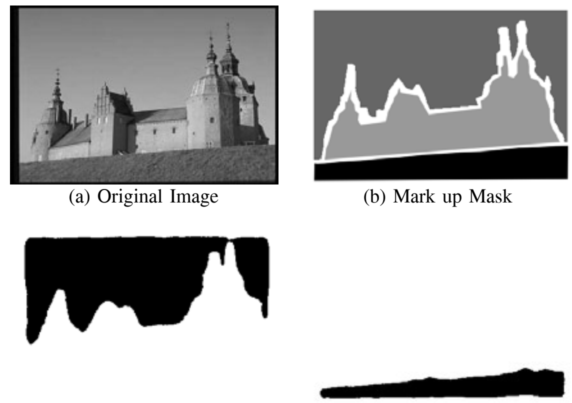
1) Generate a single-step classifier which uses a window of size $n \times n$.
2) Use the generated classifier to sweep the input image I.
    a) Start at the top-left corner of the image.
    b) Pass the $n \times n$ window to the classifier.
    c) Label all the pixels in this window with the class label output of the classifier.
    d) Move the window d pixels right and repeat 2(b) and 2(c), until the window reaches the right boundary of the image.
    e) Reposition the window to the left edge of the image and move down by d pixels, then repeat from step 2(b) until the whole image has been completely sampled.
3) Generate the output image which only contains the regions.
    a) Label each pixel based on voting. For example, if the majority of classifier outputs for a pixel is texture $T1$, then this pixel is labeled as $T1$.
    b) Assign each pixel a color based on its class label.
    c) Output the generated image O.

Fig. 2. Segmentation Algorithm for Two Textures

First, we manually create an image mask as shown in figure 3b. There are four regions, one from which it is permitted to take example cut outs of sky (S), one from which it is permitted to take example cut outs of grass (G), one from which it is permitted to take example cut outs of other (O) and one, the white region, from which no training cutouts are permitted (X). In the white region some cutouts could be ambiguous and using these in training data will result in lower training accuracy.

Second, we evolve two classifiers, *sky-vs-all-else* in which cutouts from region S are used as $T1$ and cutouts from regions $G \cap O$ are used as $T2$. The second is *grass-vs-all-else* in which cutouts from region G are used as $T1$ and cutouts from regions $S \cap O$ are used as $T2$.

Third, we apply the segmentation algorithm using each of the evolved classifiers in turn. The *sky-vs-all-else* classifier gives the segmentation shown in figure 3c. Carrying out the same procedure with *grass-vs-all-else* gives the segmentation shown in figure 3d. The various parameter values used for this are shown in table II. Also given in this table is the accuracy of the evolved classifier on an independent test set. This is perhaps unnecessary, but it does show that there has been some over training.



(a) Original Image (b) Mark up Mask (c) Sky Segmentation (d) Grass Segmentation

Fig. 3. Original Image and Mask for Sky, Grass and Other

TABLE II

PARAMETERS AND RESULTS FOR SINGLE IMAGE SEGMENTATION

Parameter	Sky	Grass
Window Size	16×16	16×16
Number of training cutouts	1,300	1,300
Training Accuracy	99%	99%
Number of test cutouts	168	168
Test Accuracy	95%	91%
Step Size	1	1

The segmentation achieved is very good. However, it needs to be noted that this result is on a single image and the textures of interest are relatively simple and that the other region is relatively uniform and uncluttered. In the next section we investigate how this approach can be extended to a collection of heterogeneous images.

V. RETRIEVAL FROM AN IMAGE COLLECTION

In this section we consider the task of retrieving images containing sky or grass textures from the Corel collection. We have created masks like the ones shown in figure 3b for 884 images. Of these, 170 have been allocated to a training image data set and 714 to a test image data set[1]. We have then evolved a *sky-vs-all-else* classifier and a *grass-vs-all-else* using the procedure described above, but this time with texture cutouts randomly sampled from the all of the training images. The segmentation algorithm of figure 2 has been slightly modified. Rather than using majority voting, a heuristically determined

[1] We now have two training/test data sets. One is the set of cutouts described earlier. The other consists of full images as described here. To avoid confusion we will explicitly refer to cutouts and images in what follows.

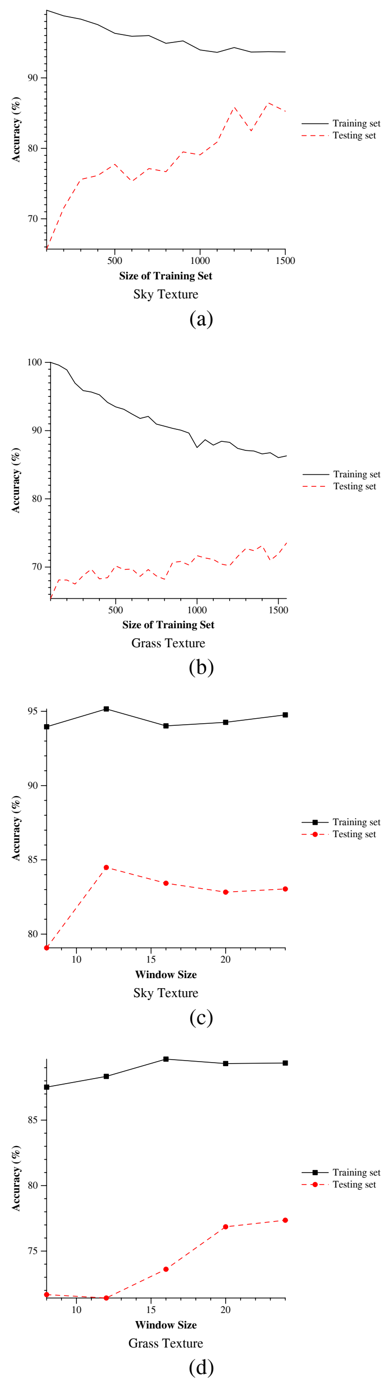


Fig. 4. Effect of Training Set Size (a) and (b) an Window Size (c)and (d)

voting threshold value is used. This is necessary because of the imbalance between the texture and other classes, and also due to the increased variability in class $T2$, since $T2$ is no longer a single texture. For example, most of the images in the Corel collection do not contain grass, and of the ones that do only a small fraction of the pixels are grass pixels. The results of training and test runs for different training set sizes and different window sizes are shown in figure 4. Figures 4a and 4b suggest that there is considerable over training and that for both textures the optimal training set size has not yet been reached. Figure 4c suggests that 12 is the best window (texel) size for sky and that the window size for grass should be at least 20.

From the perspective of image retrieval it is not necessary to generate a full segmentation of an image, it is only necessary to determine whether an image contains a region of the texture of interest. To determine this we apply the segmentation procedure described above to an image and if there is a large enough contiguous area of pixels labelled T we say that the image contains the texture T. Currently this is determined by a heuristic *segmentation threshold*. If the number of contiguous pixels is greater than the threshold we say that the texture is present in the image.

Figure 5 shows a number of examples of varying quality achieved by the procedure described above. For image 174002 both the sky and the grass segmentation have been accurately performed. In both cases the number of pixels in the largest contiguous black region exceeds the segmentation threshold and this image would be correctly retrieved for a sky texture or grass texture query. For image 174000 the sky has been correctly segmented, but there are false positive regions in the grass segmentation. The largest of these exceeds the segmentation threshold so this image would be incorrectly returned in a grass texture query. For image 174001 there is a false positive segment in the sky segmentation which does not exceed the segmentation threshold so this image would not be (correctly) returned in for a sky texture query. The grass segmentation is more interesting. It is not totally clear whether the original image contains grass. Some of the black segments are false positives, but not necessarily all. This image would be retrieved in a grass texture query, but it is not clear whether this is correct or not. For image 174005 both segmentations are clearly false positives and this is not a particularly good result.

TABLE III

PARAMETERS AND RESULTS FOR SEGMENTATION OF TRAINING IMAGES

Parameter	Sky	Grass
Window Size	12×12	16×16
Number of training images	714	714
Number of training cutouts	1,500	1,550
Accuracy on training cutouts	95%	87%
Number of test images	170	170
Number of test cutouts	168	168
Accuracy on test cutouts	89%	80%
Step Size	1	1
Voting Threshold	72	180

TABLE IV

RETRIEVAL PERFORMANCE FOR SKY

		Human	
		Sky	Not Sky
GP Based System	Sky	167 (23%)	58 (8%)
	Not Sky	51 (7%)	437 (61%)

TABLE V

RETRIEVAL PERFORMANCE FOR GRASS

		Human	
		Grass	Not Grass
GP Based System	Grass	3 (0.4%)	89 (12%)
	Not Grass	32 (4%)	619 (83%)

The results of applying the procedure to the test image data base are shown in tables IV and V. The segmentation thresholds were 3,500 and 12,500 for sky and grass respectively. These values were determined empirically, but without great effort to find optimal values. The retrieval performance for sky is quite good with a total of 604/714 (84%) of the images correctly retrieved. The errors are evenly divided between false positives and false negatives. The retrieval accuracy for grass is also around 84% but there are three times more false positives than false negatives. However, this result is somewhat biased by the low number of grass images in the test set (35/714).

VI. CONCLUSIONS

Our main aim in this work was to determine whether image retrieval from a large heterogeneous collection could be feasible using a segmentation algorithm based on a classifier evolved by genetic programming. We found that for two textures, sky and grass, a retrieval accuracy of around 84% could be achieved on a test set of 714 images. Considering that there is considerable variability in the images and that they have been taken under a wide range of lighting conditions and with a variety of cameras, these results are quite good. While we have experimented with only two textures, one of which, the sky texture, was relatively simple, the results are very promising and suggest that texture could be used to enhance the performance of content based image retrieval.

With respect to our specific research questions we have the following conclusions: *How accurately can we learn classifiers for* sky-vs-other *and* grass-vs-other *for a single image?* These classifiers quite accurate, with a test accuracy of 95% and 91% for sky and grass respectively. *How accurate is the segmentation based on these classifiers* The segmentation of the single image is excellent. The segment boundaries are closely aligned with the corresponding texture regions in the image. There are no false positive regions. *How do the previous two results scale up to a larger number of heterogeneous images?* As expected the training and test accuracies are lower and the segmentations are of varying quality. Some images have been segmented very well, others quite badly. The sky segmentations were generally better than the grass ones, as might be expected. *How accurate is image retrieval using the classifiers?* Despite the lower quality of the segmentation, image retrieval is quite good for the two textures of interest.

A drawback of our approach is that it relies on two thresholds which currently need to be determined manually. Further work is needed on automated ways of finding optimal values or on ways of eliminating them.

A rather surprising finding was the difficulty in the human mark up of the images. Even something as seemingly straightforward as sky requires some difficult judgments, for example, should clear sky be the same as cloudy sky? How much cloud could be permitted in an image before it could no longer be clear sky? Further work is needed on such issues if texture is to be used successfully in image retrieval.

ACKNOWLEDGMENT

We would like to thank Dr. Saied Tahaghoghi for his advice on CBIR systems and assistance with the Corel images.

REFERENCES

[1] Andy Song and Vic Ciesielski. Fast texture segmentation using genetic programming. In Sarker R. et al., editor, *Proceedings of the 2003 Congress on Evolutionary Computation (CEC2003)*, volume 3, pages 2126–2133. IEEE Press, December 2003.

[2] Phil Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover, NY, 1966.

[3] M. L. Kherfi, D. Ziou, and A. Bernardi. Image retrieval from the world wide web: Issues, techniques, and systems. *ACM Comput. Surv.*, 36(1):35–67, 2004.

[4] Robert M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, May 1979.

[5] K. Laws. Rapid texture identification. In *Proceedings of SPIE Conference on Image Processing for Missile Guidance*, volume 238, pages 376–380, 1980.

[6] S. Livens, P. Scheunders, G. van de Wouwer, and D. Van Dyck. Wavelets for texture analysis, an overview. In *Sixth International Conference on Image Processing and Its Applications*, volume 2, pages 581–585, 1997.

[7] S. Arivazhagan and L. Ganesan. Texture classification using wavelet transform. *Pattern Recognition Lettters*, 24(9-10):1513–1521, 2003.

[8] Inna Stainvas and David Lowe. A generative probabilistic oriented wavelet model for texture segmentation. *Neural Processing Letters*, 17(3):217–238, 2003.

[9] S.E. Grigorescu, N. Petkov, and P. Kruizinga. Comparison of texture features based on Gabor filters. *IEEE Transactions on Image Processing*, 11(10):1160–1167, October 2002.

[10] Andy Song. *Texture Classification: A Genetic Programming Approach*. PhD Thesis, RMIT, Department of Computer Science, 2003. http://www.cs.rmit.edu.au/~asong/thesis.pdf.

[11] Andy Song and Vic Ciesielski. Texture classifiers generated by genetic programming. In Xin Yao, editor, *Proceedings of the 2002 Congress on Evolutionary Computation*, volume 1, pages 243–248, Honolulu, May 2002. IEEE.

[12] Tristan Glatard, Johan Montagnat, and Isabelle E. Magnin. Texture based medical image indexing and retrieval: application to cardiac imaging. In *MIR '04: Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pages 135–142, New York, NY, USA, 2004. ACM Press.

[13] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vision*, 40(2):99–121, 2000.

[14] Peter Howarth and Stefan M. Rüger. Evaluation of texture features for content-based image retrieval. In *CIVR*, volume 3115 of *Lecture Notes in Computer Science*, pages 326–334. Springer, 2004.

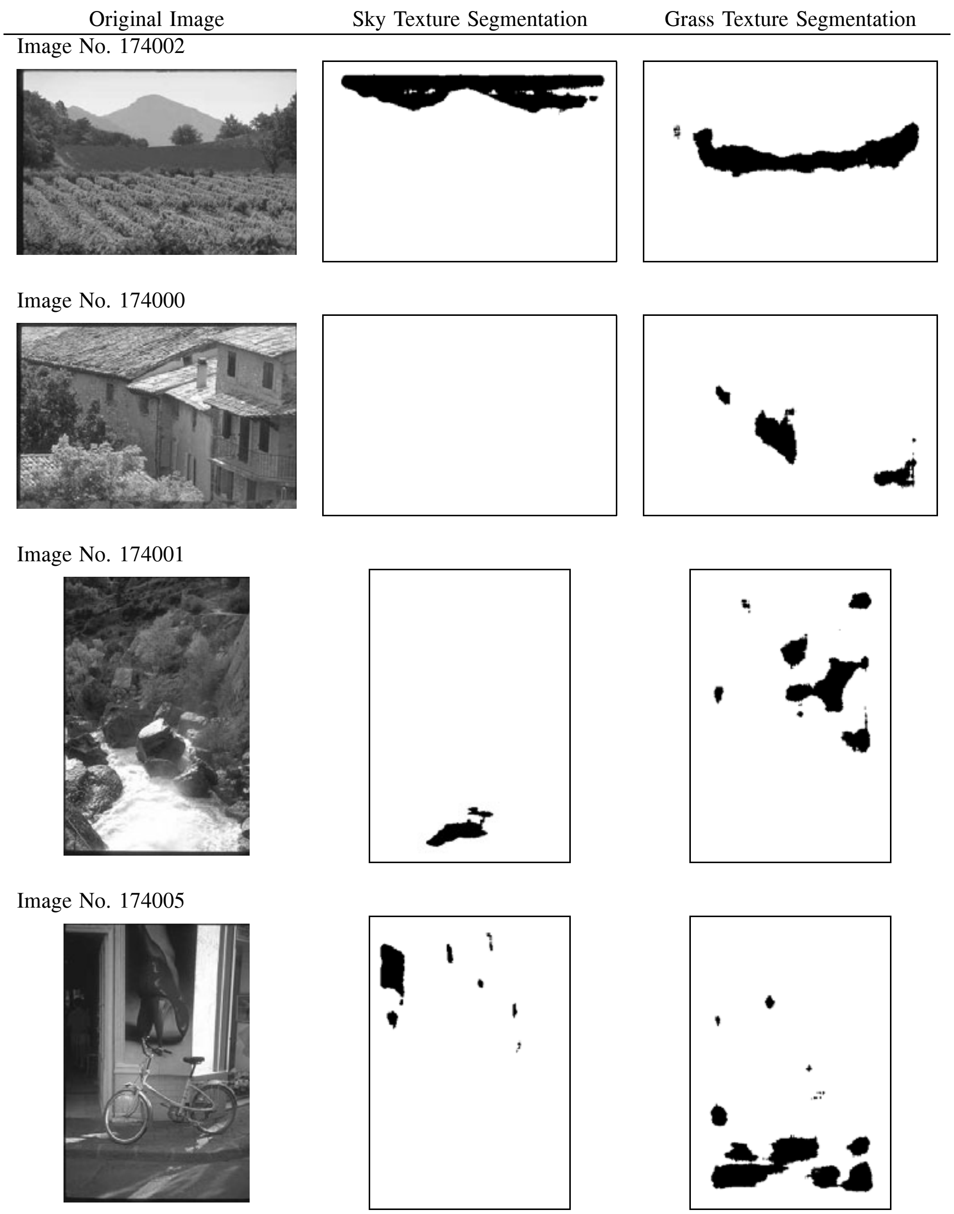


Fig. 5. Example segmentations on the full Corel image set