U.P.B. Sci. Bull., Series C, Vol. 74, Iss. 2, 2012 ISSN 1454-234x

REAL-TIME AUTOMATIC LANDSCAPE IMAGE DETECTION IN DIGITAL STILL CAMERA PREVIEW IMAGES WITH FUZZY LOGIC USER CONTROL

Constantin VERTAN[1], Marta ZAMFIR[2]

În cele ce urmează este prezentat un sistem de analiză automată a scenei, bazat pe imaginile ("preview"-uri) achiziționate de camera foto înainte de captarea fotografiei. În urma analizei, scena este clasificată - fără intervenția utilizatorului - în "peisaj" sau "non-peisaj", scopul acestei clasificări fiind acela de a stabili cele mai potrivite valori ale parametrilor de expunere. Soluția propusă poate fi integrată într-un cadru mai larg de clasificare a scenelor, capabil să distingă între mai multe tipuri de scene. Un astfel de sistem este util mai ales în cazul în care scene de natură diferită se succed intercalat şi repede, astfel încât utilizatorul nu are timpul necesar să modifice manual parametrii de expunere. Clasificarea în „peisaj"/"non-peisaj" poate fi reglată de utilizator prin varierea unui singur parametru (realizată prin inferență fuzzy) de-a lungul unei scale având punctul neutru amplasat la mijlocul ei.

This paper describes a real-type system for the automatic analysis of the imaged scene, via the camera preview image stream, without user intervention, and the classification of that scene into the classes of landscape image/ non-landscape image. This classification is primarily used for an automatic setting of the camera exposure program. The proposed solution can be integrated into a larger image classification framework, able to discriminate between several types of scenery, particularly suited for the cases where scenery and/or illumination conditions alternate in a very fast succession. The landscape detection system is trimmed via a single user-controlled parameter, via fuzzy logic inference, with a "neutral" point approximately at the middle of the scale.

Keywords: image classification, real-time image systems, consumer imaging, landscape detection

1. Introduction

The obvious alternate solution to automatic scene detection is to manually set the exposure program of the camera according to the imaged scene. This is cumbersome if scenery and/or illumination conditions alternate in a very fast succession.

[1] Prof., Applied Electronics and Informatics Engineering Department, University POLITEHNICA of Bucharest, Romania, cvertan@alpha.imag.pub.ro
[2] PhD Lecturer, Applied Electronics and Informatics Engineering Department, University POLITEHNICA of Bucharest, Romania, mzamfir@alpha.imag.pub.ro

Many of the existing techniques of scene classification categorize images using primarily vectors assembled from low-level features (such as colors, edges, textures), which are further fed into a classifier. Szummer and Picard [1] classify images as indoor/outdoor by computing the low-level features on sub-blocks of the image instead of the whole image. They use the color histogram in Ohta color space and texture features computed using MRSAR (multi-resolution simultaneous autoregressive model) and a combination of 2D DFT and DCT coefficients computed over 8x8 pixel blocks.

Serrano et al. [2] introduce a low complexity, low dimensional feature set and use wavelet texture features and a two-stage SVM classifier, achieving a classification rate of 90.2% on a database of 1200 consumer photographs, while Kim et al. [3] divide the image in five block regions whose ECOH descriptors (edge and color orientation histograms) are differently weighted in a SVM classification stage. Their algorithm reduces the false positives (produced by objects having color similar to sky or grass) and improves the computing efficiency.

Vailaya et al. [4] propose a Bayesian framework for the hierarchical classification by MDL-driven vector classifier features of vacation pictures. The landscape class (a subset of “outdoor”) is further refined into specific scenery types.

Sky detection and image orientation detection are closely related to indoor/outdoor classification algorithms.

Luo and Etz [5] perform color classification by a multilayer back-propagation neural network, followed by a region extraction algorithm; they finally validate the sky detection using a physical model of the sky based on the scattering of light by small particles in the air.

Wang and Zhang [6] use low-level, local visual features extracted in $N \times N$ sub-blocks of the image: color moments (chrominance information) and edge directions histogram - EDH (luminance information), which are used in SVM based classifiers for image orientation detection.

Several similar solutions are implemented either as features in software suites or already embedded in cameras produced by important manufacturers (Casio, Sony-Ericsson, Samsung, Kodak, Panasonic etc.). The aim of the named solutions is to recognize particular types of scenes (beach, snow, sunset, day/night portrait, backlight, foliage, aquarium, fireworks, day/night landscape, indoor, party, museum, kids & pets, sports, text etc.) and to set accordingly specific imaging parameters.

The proposed solution offers fast landscape/non-landscape image classification, based on the analysis of preview images. In the case of a landscape image, a simple, numerical definition cannot be provided; instead, a set of natural language descriptive statements can help in collecting and testing Landscape images:

- Outside may be primarily characterized by the presence of green and blue colors;
- The scene must contain a very distant (at infinity) contour that separates the sky from: a land, a mountain, a forest etc.;
- The scene should NOT contain big objects in foreground, close to the camera;
- The scene should NOT contain portraits (big faces);
- The percent of manmade objects in the scene must be small (no more than 20-30%);
- The scene should NOT contain long vertical lines (which are present, in general, at close buildings, poles etc.);
- The scene should NOT contain beach or seascapes, lands covered with yellow crops or grass, yellow/red autumn forests etc.;
- The scene must be well illuminated (daylight) so that colors are not altered or attenuated.

2. Proposed implementation

The analysis is performed at the available preview resolution on the digital still camera, but no higher than 320 by 240 pixels, in the native YCbCr color representation. Real-time, on-camera implementation requires in most cases even the subsampling of the preview image with factors from 2 to 4. Within the image are measured two groups of descriptors: color descriptors and texture descriptors.

The color descriptors are based mainly on the statistics computed for the blue and green pixels within the image. Blue and green are defined in terms of their YCC values, only for pixels that are bright enough in order to have a clear color. The simplest descriptors are limited to the percentage of blue pixels and the percentage of green pixels within the image. The descriptors can be extended by using more colors (gray, yellow etc.) or other, advanced statistics that embed some information on the spatial distribution of the colors (such as the MPEG-7 Color Structure Descriptor, or the Color Coherence description, or by the separate accounting of colors within individual image regions).

The texture descriptors are based mainly on the statistics computed from the gradient orientations. Gradients are computed for the luminance component of the pixels that are bright enough. The orientation is quantized into 4 values (horizontal, vertical, first diagonal, second diagonal). The resulting data is statistically described, using one of several statistical measures. These measures range from simple occurrence probabilities of the various orientations within the image to orientation probability density function, statistical moments of the orientations, or spatially–related statistical measures (such as MPEG-7 unhomogeneous texture description, statistical description of edge runlengths etc.).

Computational speedup can be achieved in two ways. The first approach is to downsample the preview image and to check every fourth or eighth or sixteenth pixel. The second approach reduces by half the number of convolutions needed for the computation of edge orientations via a look-up-table that stores the quantized edge orientation (including the coherent/ non-coherent information) indexed by the quantized values of the horizontal and vertical image gradients. The basic idea of the speed-up is to obtain the orientation of the local edge using only two convolutions for the gradient computation (e.g. only the vertical/ horizontal gradients). The two values of the computed gradients, properly scaled, are used as indexes for a two-dimensional LUT that stores for each pair of gradient values the final integer value describing the edge/non-edge and the orientation class (numbers between 1 and 5). The LUT has a size of 96 by 96 bytes, needing less than 9KB of memory space and is synthetically presented in Fig.1.

Fig. 1. LUT orientation table storing vertical, horizontal and diagonal orientation codes according to vertical and horizontal gradient magnitudes

The classification of an image as landscape is decided upon the verification of a set of simple tests. The tests involve the verification of minimal acceptance conditions for the colors and for the texture pattern from the image, conditions that mimic the assumed lexical description of the landscape. The conditions are based on sets of pre-defined thresholds, obtained by heuristics and extended experimental testing on a large labeled test image database.

The conditions that are finally checked involve the separation of the landscape/non-landscape images according to the total blue and green color content (as shown in Fig.2 and implemented by the first condition in algorithmic step 5 below) and according to the textural content, measured by the vertical/horizontal gradient ratios and the standard deviation of all gradient orientations. Fig.3 shows the separation of landscape/non-landscape images according to these features; it follows that this separation can be approximated by a simple line, as implemented in the second condition from the algorithmic step 5.

The simplest software implementation is briefly described below.

1. Acquire the preview image, perform necessary image subsampling, if needed, obtain YCC color description.
2. Set descriptor computation subsampling step (decide the pixel step).
3. Compute the image color descriptors: for each inspected pixel obtain the color label (blue, green, etc. ...) as a result of comparing the YCbCr components of the pixel to pre-determined thresholds ("the color definition"). Accumulate the same-labeled pixels into separate counts, according to the desired descriptor. When using the simplest decision model, accumulate all green and blue pixels into a single count that will be normalized with respect to the total number of inspected pixels (TOTALBG).
4. Compute the image texture descriptors by gradient orientations: for each inspected pixel compute the four gradients and obtain the orientation of the strongest gradient (or compute only the horizontal and vertical gradients and apply LUT); select only the gradients that are strong enough with respect to a fixed coherence threshold THRESH_COHERENCE. Accumulate an orientation-indexed pixel count that is used as primary data for orientation statistics. The simplest computed statistic is the standard deviation (STDEV) of the four edge orientation probabilities of occurrence (PROBH, PROBV, PROBD1, PROBD2).
5. After the scan of the entire preview image, the simplest landscape classification decision is taken by checking that the image contain a sufficient amount of blue and green colored pixels (i.e. TOTALBG>THRESH_COLOR_PERC) and that the orientation is not dominantly horizontal or vertical (i.e. PLANESEP_MULTIPLIER*(PROBH + PROBV - THRESH_PLANESEP_PERC) > STDEV).

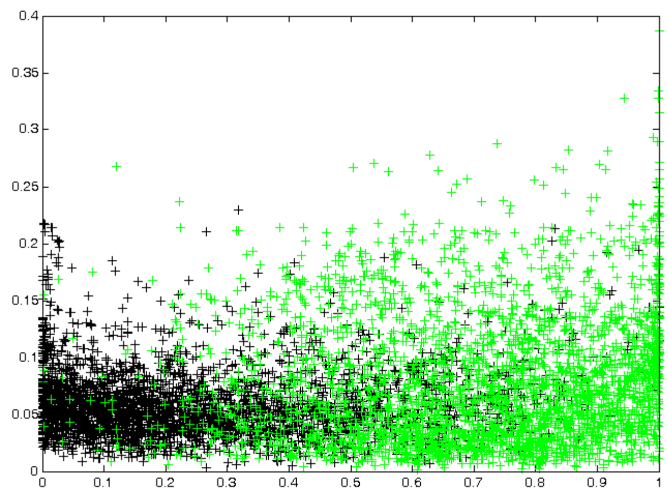

Fig. 2. Standard deviation of the orientations (vertical axis) vs. total green and blue image content (horizontal axis) for landscape (green) and non-landscape (black) images

3. Results and discussions

The proposed technique was tested on a test database containing 10425 images (3694 landscape images, 3120 indoor images and 3611 city/outdoor images). The initial image labeling was performed manually, by the consensus of several observers, establishing a realistic ground truth. The system performance is 90.99% correct landscape classification and 10.34% false positives. There are 333 missed landscapes and 696 indoor/outdoor images classified as landscape (538 outdoor – mild error and 158 indoor – significant error).

There are three classes of algorithm failure:

1. classification failure due to the average luminance of the image (images that are too dark)
2. classification failure due to unrecognized colors
3. classification failure due to intrinsic algorithm limitations

Class 1 failures are seldom and not that difficult to deal-with. Relaxing the luminance threshold used for the definition of colors can solve some of the problems. The decrease of the luminance threshold by 8-unit decrements adds some 1% to the correct landscape recognition rate (increasing the false positive rate by the same amount). Tests were performed to luminance within the (16, 40) range.

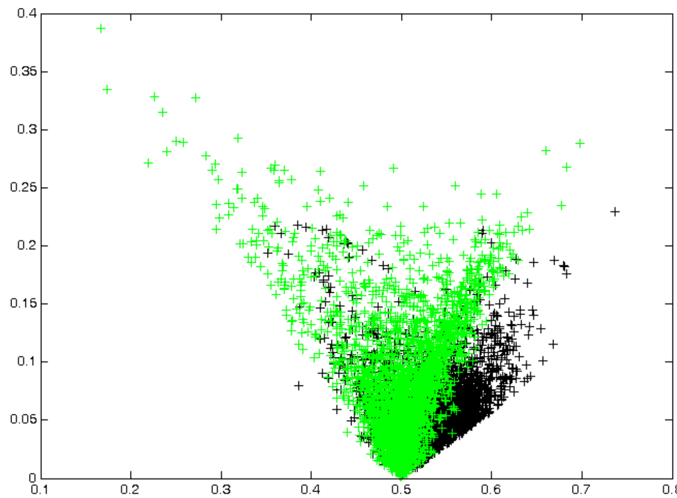

Fig. 3. Standard deviation of the orientation (vertical axis) vs. horizontal and vertical variation (horizontal axis) for landscape (green) and non-landscape (black) images

Class 2 failures are due to the way of color definition. For instance, green is sought to have G>R and G>B. Still, there are green tones with G slightly smaller than R and G >B (yellowish green) that are not recognized as such. The yellow tones are not counted at all in the current implementation (this generates failures for crop fields or beaches). Such failure examples are presented in figures 4 to 6, below. The solution is to change the color definitions, by allowing wider intervals of green and blue and by introducing the class of yellow.

4. Fuzzy user control

There are means of integrating the set of thresholds associated to the tested conditions into a single, user-adjustable form, by providing a single tuning parameter that can shift the limits of landscape acceptance. This adjustment can be done either manually by the user, or can be integrated as an adaptive procedure, that adjusts camera behavior to changing environments. The system can be trimmed via a single parameter, taking values in the [0, 1] range, with a "neutral" point approximately at the middle of the scale. The variation is non-linear on the two parts of the scale; at the "neutral" point, the system exhibits approximately the same performance as the classical (binary) decision. This control system is based on the fuzzyfication of the color, texture and blue margin measures presented before.

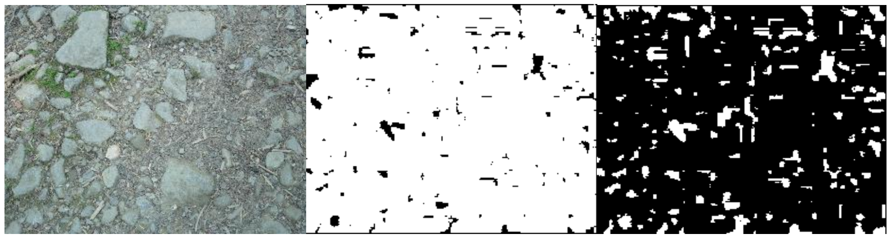

Fig. 4. Original image, green pixels (false colored in white), blue pixels (false colored in white). Green and blue pixels are 99% of image content (as strange as it may seem, the grays in the image have a G component slightly bigger than the R and B components). Edges have a uniform orientation distribution, thus the image fits in the “landscape” class. The green color definition is to blame.

Fig. 5. Original image, green pixels (false colored in white), blue pixels (false colored in white). Green and blue pixels are 50% of image content. The edge orientation distribution is not that uniform, thus the image does not fit in the “landscape” class. The distance with respect to the landscape class boundary is very small; the problem comes from the estimation of edge direction for edges located in areas darker than the lower luminance threshold used for color estimation. These edges (which may be wrong) contribute in the particular case of this image to the non-uniformity of the edge direction distribution.

Fig. 6: Original image, green pixels (false colored in white), blue pixels (false colored in white). Green and blue pixels are 75% of image content. Edges have a uniform orientation distribution, thus the image fits in the “landscape” class.

The color and blue margin measures, due to their similar nature, are fuzzyfied in the same way, measuring the degree in which one passes through the transition domain between non-landscape and definitely landscape images. The fuzzy measure is linearly increasing on the transition range of the color and blue on margin ratios from non-landscape to landscape. The texture measure is fuzzyfied in the inverse manner, since the texture content (measured in the image by the horizontal and vertical gradient ratio and the standard deviation of the oriented gradient ratios) is decreasing from non-landscape to landscape. The three fuzzy measures are aggregated via the classical product operator and the final landscape detection block issues a “landscape recognition” decision by comparing the aggregated fuzzy measure with the user-set threshold (control parameter).

The user sets the acceptance parameter THRESHOLD (in the range [0, 1]). The preferred value of the parameter is 0.6. Higher values of the parameter reduce the correct detection ratio and the false alarms; lower values of the parameter increase the correct detection ratio and the false alarms (based on the evaluation performed on the current labeled test database). Fig.7 shows the variation of system performance with respect to the modification of the acceptance parameter THRESHOLD (user control).

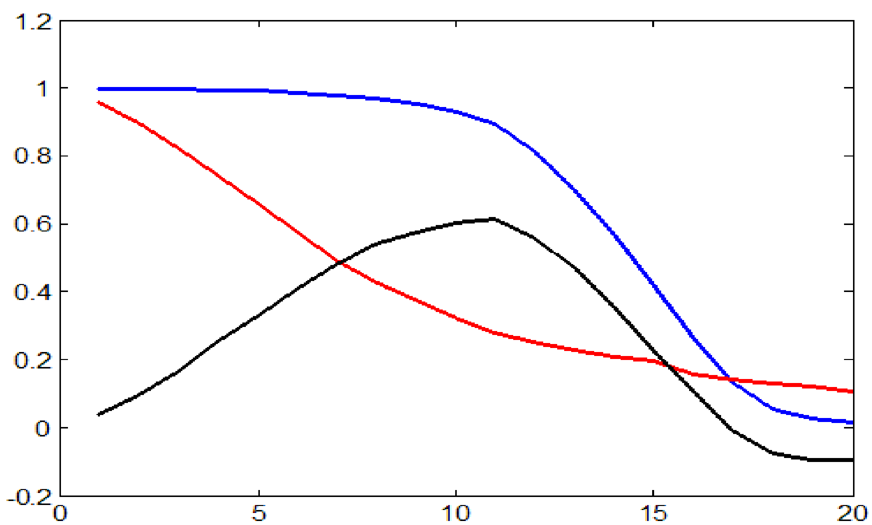

Fig. 7. Fuzzy tuning system performance: tuning parameter on the horizontal axis, graded in increments (20 increments = value 1); percentages on the vertical axis: correct classification (blue line), false positives (red line) and overall system performance (correct classification – false positives, black line).

3. Conclusions

This paper describes a real-type system for the automatic landscape detection in digital camera preview images. This classification is primarily used for an automatic setting of the camera exposure program. The landscape detection system is based on simple image features (color and texture) and linear decision rules. The system performance (correct landscape detection vs. false positives) can be trimmed via a single user-controlled parameter integrated via fuzzy logic inference.

R E F E R E N C E S

[1] *M. Szummer*, *R.W. Picard*, “Indoor-Outdoor Image Classification”, IEEE Intl. Workshop on Content-based Access of Image and Video Databases, ICCV98, Bombay, India, Jan. 1998

[2] *N. Serrano*, *A. Savakis*, *J. Luo*, “A Computationally Efficient Approach to Indoor/Outdoor Scene Classification”, in Proc. of the 16-th International Conference on Pattern Recognition (ICPR'02)", Quebec, Canada, Aug. 2002

[3] *W. Kim*, *J. Park*, *C. Kim*, “A novel method for efficient indoor-outdoor image classification”, Journal of Signal Processing Systems, **vol. 61**, issue 3, pp. 251-258, Feb. 2010

[4] *A. Vailaya*, *M. Figueiredo*, *A. Jain*, *H.J. Zhang*, “Image Classification for Content-Based Indexing”, IEEE Transactions on Image Processing, **vol. 10**, no. 1, pp. 117-130, 2001

[5] *J. Luo*, *S. Etz*, “A physical model-based approach to sky detection in photographic images”, IEEE Transactions on Image Processing, **vol. 11**, no. 3, pp. 201-212, 2002

[6] *Y. Wang*, *H. Zhang*, “Detecting image orientation based on low-level visual content”, Computer Vision and Image Understanding, **vol. 93**, no. 3, pp. 328-346, 2004

[7] *Seungji Yang*, *Yong Man Ro*, “Two-layered Photo Classification Based on Semantic and Syntactic Features”, Dec. 2006.