Grass Field Detection for TV Picture Quality Enhancement

Bahman Zafarifar¹ and Peter H. N. de With², IEEE Fellow

¹ NXP Semiconductors, ² Eindhoven University of Technology, ³ Philips Consumer Electronics

Abstract—Current TV image enhancement can be improved if the image is analyzed, objects of interest are segmented, and each segment is processed with content-specific enhancement algorithms. In this paper we present an algorithm for segmenting grass areas in video sequences. The system employs multi-scale texture analysis and adaptive color and position models for computing a pixel-based soft segmentation map. Compared to previously reported algorithms, our system shows a clear improvement in the detection result: at 10% false positive rate, the true positive rate of our algorithm yields 91%, vs. 66% and 58% of two existing methods.

I. INTRODUCTION

Image enhancement in current TVs is performed globally, e.g. contrast and brightness, or it is adapted to the local properties of a small pixel neighborhood, e.g. selecting only pixels that are likely to be part of a single object [1]. The latter locally-adaptive method can be improved if the adaptation is extended towards a more elaborate analysis of the image, in order to consider the true nature of the video content. Such content-based adaptation can be realized by analysis and segmentation of objects of interest using a number of detectors, followed by optimized processing of each segmented area. Based on this concept, we previously developed an algorithm for detecting sky areas in TV images [2]. In this paper we extend this concept with another object detector, which reuses the developed techniques, in order to come to a more universal solution.

Grass fields are frequently seen in sports programs and outdoor scenes. This motivates the subjective importance of image enhancements in grass areas. Previously reported work on grass detection for real-time video includes a method based on pixel-level color and texture features [3]. To reduce the variations in the detection result of this pixel-based approach, [4] proposes 8×8 block averaging of a color-only grass detector and a binary classification to grass/no-grass classes. We developed a method that extends these algorithms with better color and texture features, multi-scale image analysis, and modeling of grass areas prior to a pixel-accurate soft segmentation.

II. ALGORITHM DESCRIPTION

We propose a detection system (Fig. 1) that is based on 1) analyzing the image using color and texture features, 2) modeling the object (grass areas) by color and position models, and 3) computing a pixel-accurate soft segmentation map by using the input image and the mentioned models.

...
applications. To address this problem, we propose using this initial grass probability to model the gras areas by color and position models (Fig. 4-b and c), and then to employ the models in re-computing a pixel-accurate final grass segmentation map, using the input image at full resolution. This is described in the following two sections.

B. Modeling

The color model is a spatially varying value, representing the estimated grass color at each image position (Fig. 4-b). The color model is implemented using three small \((h \times w)\) matrices, \(M_f, M_l, \) and \(M_v\) one for each color component. As an example, the luminance color model \(M_f\) is defined as

\[
M_f(r, c) = \frac{\sum_{i=-h}^{h} \sum_{j=-w}^{w} (Y(r+i, c+j) \times P_{\text{initial}}(r+i, c+j) \times W(i, j))}{\sum_{i=-h}^{h} \sum_{j=-w}^{w} (P_{\text{initial}}(r+i, c+j) \times W(i, j))}, \tag{2}
\]

which fits \(M_f\) to the values of the corresponding color component \(Y\) of the input image, using a Gaussian kernel \(W\), weighted by the initial grass probability \(P_{\text{initial}}\). This ensures that the color model is not influenced by parts of the image that are initially not considered as grass.

The position model \(P_{\text{position}}\) is a smooth version of the initial grass probability, obtained by filtering with a Gaussian kernel. This model is implemented as a small \((h \times w)\) matrix of values.

The above model-creation procedure is computationally expensive, but the small resolution of the models \((h \times w\) is 16 times smaller than the input image) reduces the amount of computations. The models are up-scaled to the input image resolution in the computation of the final segmentation.

C. Segmentation

A pixel-accurate soft segmentation-map,

\[
P_{\text{final}} = P_{\text{color final}} \times P_{\text{position}}, \tag{2}
\]

is computed in the segmentation stage, using the full-resolution input image and the color and position models. Here, \(P_{\text{position}}\) denotes the up-scaled version of the position model (Fig 4-c), and \(P_{\text{color final}}\) is the final color probability, computed by a 3D Gaussian function centered at the spatially-varying color given by the up-scaled version of the color model (Fig 4-b).

III. RESULTS AND CONCLUSIONS

We applied the proposed algorithm and the methods from [3] and [4] to a test set of 62 manually annotated images. Fig. 3 compares the ROC curve (true-positive vs. false-positive rates) of the three algorithms. It can be seen that the proposed algorithm yields better results almost along the entire curve. At 10% false positive rate, the true positive rate of our algorithm yields 91%, vs. 66% of [3] and 58% of [4].

Fig. 5-top demonstrates the improved segmentation result by rejecting the trees. This improvement is the result of a more compact representation of the grass color, using PCA analysis. Fig. 5-bottom illustrates the improved segmentation result in sunny and shadow areas.

We conclude that the proposed algorithm outperforms the existing methods in correctly rejecting non-grass, and correctly detecting grass areas under different illumination conditions, due to better color and texture features and the employed modeling. Furthermore, high computational demands have been avoided by performing the modeling in low resolution.

![ROC comparison](image)

Fig. 3. Performance comparison of the proposed and existing algorithms.

![Sample result](image)

Fig. 4. a: input, b: color model, c: position model.

![Sample result](image)

Fig. 5. Sample result: a: input, b: result from [3], c: result proposed algorithm.

REFERENCES