

The 2012 Iberoamerican Conference on Electronics Engineering and Computer Science

## An FPGA Implementation for Image Interpretation Based on Adaptive Boosting Algorithm in the Real-Time Systems

Mario-Alberto Ibarra-Manzano<sup>a</sup>, Dora-Luz Almanza-Ojeda<sup>b</sup>

<sup>a</sup> *Digital Signal Processing Laboratory; Electronics Department; DICIS; University of Guanajuato  
Carretera Salamanca-Valle de Santiago Km.3.5+1.8 Km, Comunidad Palo Blanco, 36885, Salamanca, Guanajuato, Mexico*

<sup>b</sup> *Departamento de Ingeniería Robótica; Universidad Politécnica de Guanajuato  
Av. Universidad Norte SN, Comunidad Juan Alonso, 38483, Cortazar, Guanajuato, Mexico*

---

### Abstract

This paper presents an FPGA architecture for objects classification based on Adaptive Boosting algorithm. The architecture uses the color and texture features as input attributes to discriminate the objects in a scene. Moreover, the architecture design takes into account the requirements of real-time processing. To this end, it was optimized for reusing the texture feature modules, giving, in this way, a more complete model for each object and becoming easier the object-discrimination process. The reuses technique allows to increase the information of the object model without decrease the performance or drastically increase the area used on the FPGA. The architecture classifies 30 dense images per second of size  $640 \times 480$  pixels. Both, classification architecture and optimization technique, are described and compared with others architectures founded in the literature. The conclusions and perspectives are given at the end of this document.

© 2012 Published by Elsevier Ltd.

*Keywords:* Adaptive Boosting Algorithm, Texture and Color Features, FPGA Architecture, Real-Time Systems

---

### 1. Introduction

One of the most used technique on the obstacles detection consist in modeling the objects based on appearance (color, texture,...) and/or geometric attributes. This model must deliver a features vector, by means of this vector it is possible to separate an image in different regions. The feature vector is used as input of the classification algorithm during the learning process. Furthermore, the classification algorithm requires a training set which is used to discriminate the classes by dividing the image in regions or objects in according to an attributes vector. In this case, the number of dimensions  $n$  depends on the number of attributes to be analyzed. It is important to points out that in this methodology, the algorithm to calculate the attributes plays an important role during the final classification, since the performance of detection is directly proportional to the information contained in the attributes vector. On the other hand, the classification algorithm is also important because it allows to obtain the best trace off among performance, training time and classification time.

---

*Email addresses:* [ibarram@ugto.mx](mailto:ibarram@ugto.mx) (Mario-Alberto Ibarra-Manzano), [luzdora@ieee.org](mailto:luzdora@ieee.org) (Dora-Luz Almanza-Ojeda)

Color and texture features are the most frequently used attributes for classifying objects in indoor scenes, because they allow to discriminate several objects. Also, color feature is an attribute that contains sufficient information about the objects, in addition, it does not require much computational power. However, if the environment contains several and different kind of objects, the the color feature is not enough, and it is necessary to complement our information by other feature such as texture. However, the texture attribute implies a disadvantage: texture attributes calculation requires a high computational cost. To overcome this inconvenient, two possible alternatives have been proposed : (1) make a sparse analysis and (2) use dedicated systems. First alternative consist in dividing the image into several regions in which only few points are analyzed to determine the class on the region, thus the calculation time is highly reduced and the real-time performance could be reached. Second alternative proposes to carry out the features analysis on a dedicated system such as a DSP, GPU or FPGA, with the aim of increasing the performance and, unlike first alternative, achieving a dense analysis of the image, that is pixel by pixel.

In the literature, there are different that uses texture analysis on dedicated systems. One of the first solutions to this issue employing an heterogeneous embedded system was proposed by Ibarra-Pico et al [1]. The system consist of a DSP and an FPGA and it is based on a primary variant of the algorithm of sums and differences of histograms (SDH) proposed by Unser [2]. In [1], the authors use as input the image for calculating the histograms, from such the texture attributes are obtained straightforward. However, these new attributes do not hold some characteristics unlike the SDH algorithm proposed by Unser, as a consequence, the performance during classification is reduced. A another solution based on the algorithm of concurrence matrices, was proposed by Tahir [3] in which the performance in classification was improved. The solution proposed consist in obtaining the texture attributes using an FPGA, since this computation requires more computing power. On the other hand, the classification process is carried out on a PC reducing the overall performance in system. In this case, the implementation on a PC cannot be avoided because most of the FPGA resources are already consumed by the texture analysis. This system is used for processing medical images in order to detect cancer. To overcome the problem of the system performance, the image is divided into several regions and each region is classified taking into account all the pixels. This image division allows to reduce the output image with respect to the input image, mainly due to the classification process is performed out to the feature analysis process.

In Lopez-Estrada et. al [4], the authors provide a solution to reduce the consumed area in the texture analysis. In this work, the architecture calculates the texture attributes as in [3] by taking the modules only for the texture features that consume the lowest quantity of resources on the FPGA. Thus, Lopez-Estrada could implement the classification process on an FPGA. However, as this done by a decision tree, the performance of the classification is limited. Furthermore, this strategy forces the entire image to have the "predominant" class resulted, i.e, the system provides a class for each image but it ignores the presence of another classes and the position of such resulting class in the image. After in 2010, Ibarra-Manzano et al. [5, 6] present an adaptation of the SDH algorithm for texture feature analysis allowing to calculate dense attributes for each of the pixels in the image. In spite of a significant reduction in the arithmetic computation of the SDH algorithm, the architecture shows some modules not optimized, moreover most of them are not reused even if that could improve the global performance of the system. In order to overcome this problems, same authors propose in 2011 an extension of their work creating an optimal architecture for a greater amount of image information, from which it is possible to increase the number of attributes [7].

In this work, it is used an analysis optimized of textures for providing a most complete model of the objects and measurably improving the classification results. At this point, it is necessary to consider the case of distinguishing a particular object among several objects of the same kind but with different color. Usually, we found that this particular situation requires more than only texture features information. For this reason, a color model is included to the texture classification architecture with the aim of providing more information to discriminate singular objects. Once texture and color features have been analyzed and the corresponding attributes values found, it is necessary to select a classification method to increase the efficiency and performance of the global strategy. Therefore, we found the AdaBoost learning advanced techniques as the most convenient method due to it is easier to be implemented on an FPGA but essentially because it adaptively updates the distribution of samples by increasing the weights that were poorly represented by the previous classification. Furthermore, AdaBoost algorithm can be very effective with high-dimensional data

متن کامل مقاله

دریافت فوری ←

**ISI**Articles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات