CHAPTER X

OBJECT TAGGING FOR HUMAN-ROBOT INTERACTION BY RECOLORIZATION USING GAUSSIAN MIXTURE MODELS

M. GONZÁLEZ-FIERRO[1,2], M. A. MALDONADO[2], J. G. VICTORES[1], S. MORANTE[1,2] and C. BALAGUER[1]

[1]Robotics Lab, Universidad Carlos III de Madrid; [2]Samsamia Technologies S.L. [1]{mgpalaci,jcgvicto,smorante,balaguer}@ing.uc3m.es; [2]{miguelgfierro, miguelmaldonado, santimorante}@samsamia.com

In this work, we present a method to tag objects by applying a color model learned from another source object. We learn the statistical color model of objects using Gaussian Mixture Models and Expectation-Maximization algorithm. The source model is transferred to the target object to be tagged by matching the Gaussian distribution that best describe the color structure. This makes the target gain the color model of the source while maintaining its initial appearance. This algorithm can be used in Human-Robot Interaction to visually tag objects for selection, targeting or discrimination. We perform some experiments to test our proposed method.

1 Introduction

To allow robots to share their living space with humans, they must be able to understand the environment and act intelligently. One of the first steps to accomplish this behavior is enabling robots to identify objects or people. However, in many cases, this is a complex task for the robot alone. Humans can help the robot to understand the environment by helping it to select and tag targets with which to perform a desired task. This can be done by teleoperation (Pierro, 2009), interaction through gestures (Bueno, 2012) or Learning from Demonstration (Argall, 2008).

Object tagging is an area of research that has many applications in social networks and in the Internet in general, and it is widely addressed in computer vision. In (Bergman, 2011), an automatic method for object tagging is presented, where objects like skin, the sky or foliage are automatically tagged. Another popular method is the “bag of words” (Csurka, 2004), where a bag of features treated like words is computed, and then classified to visually categorize objects.

Despite these efforts, there is still a huge gap between what a human is capable of tagging and automatic selection, as (Pavlidis, 2009) defends. There is a growing interest on relying on humans to solve this difficult computer vision tasks (Sigala, 2004). A widely known method is the reCAPTCHA of Google (Von Ahn, 2008), which aims to digitalize old texts with the help of millions of users throughout the web. The first of the two words that appears in a reCAPTCHA is used for security reasons, to find out if the user is a human or a machine. The second one is a word that Google is not capable of recognizing using OCR methods, and is thus presented to the user for human identification.

In this paper we propose a supervised method to tag objects using color substitution. In a first step a color model of the source object is learned using Gaussian Mixture Models (GMM). This color is then applied to the target object, but maintaining the shape and visual structure of the target. As a result, the object changes its color but it is still identifiable by the human. Some examples of color substitution are (Pitie, 2005), that performs histogram matching, or (Tai, 2005), that makes parametric matching. Our work is based on (Huang, 2009), that proposed a method of image recoloring to help colorblind people to recognize objects. The authors swap colors that people with color vision deficiencies may have difficulties in perceiving with colors that they may identify more easily. The final color model application is performed through the method described in (Saphira, 2009).

The aim of this work is to make it easier for a human to interact with a robot in a cluttered environment. Usually, object tagging is performed using a bounding box that surrounds the target object, and at times an attached text. Our proposed method allows the recoloring of objects from a determined source, avoiding the use of occluding bounding boxes. Additionally, different classes of objects may be tagged with user friendly and easily recognizable patterns. Fig. 1 presents a side-by-side view of the bounding box method and the proposed method.

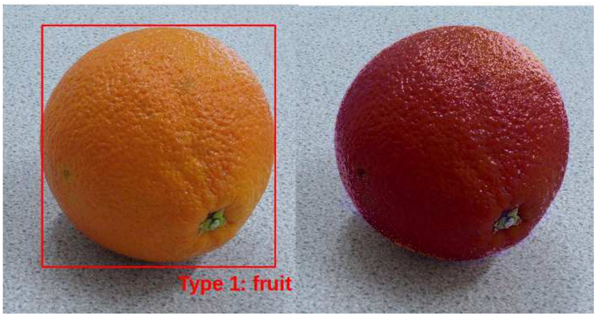

Figure 1: On the left side: the usual tagging system, the object is tagged by a bounding box and a text where the type is expressed. On the right side: our proposal, the object is tagged using a determined source color model, in this case red. As it can be seen, occlusion of the environment by the bounding box (left) is avoided through the use of the presented recoloring mechanism.

The document is ordered as follows: Section 2 outlines the basic mathematical used tools, Section 3 explains how the objects are tagged by color substitution, Section 4 presents the experiments, and Section 5 provides several conclusions.

2 Basic tools

This section reviews some of the most relevant algorithms that the authors use to perform the presented segmentation and tagging recoloring process.

2.1 GrabCut Algorithm

We make use of the GrabCut algorithm (Rother, 2004) to segment the image. GrabCut uses Gaussian Mixture Models and Expectation Maximization to find globally optimal segmented solutions.

The Grabcut algorithm includes two parts, hard segmentation and border matting. In the hard segmentation phase, the algorithm estimates the foreground and the background of the image by using an iterative version of graph-cut optimization (Boykov, 2001). Then, in the border matting phase, alpha values are obtained in a narrow region in the surroundings of the segmentation boundary.

2.2 Gaussian Mixture Model

A Gaussian Mixture Model (GMM) is a parametric probabilistic model for representing subpopulations in training datasets of points. It can be expressed as a weighted sum of **M** multivariate Gaussian distributions. As explained in (Reynolds, 2008), this model can be expressed as:

$$p(x|\lambda) = \sum_{i=1}^{M} w_i g(x|\mu_i, \Sigma_i)$$

Where *x* is a D-dimensional data vector, w_i are the mixture weights, and *g* is the multivariate Gaussian probabilistic density function. This density function is defined by:

$$g(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2} \left|\Sigma_i\right|^{1/2}} exp\left\{-\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)\right\}$$

Where μ is the mean vector and Σ is the covariance matrix. This is a general model that can express several specific scenarios, i.e. a single Gaussian model (where **M**=1), a univariate Gaussian model (where the mean and covariance are actually scalars), or a case where the information among the axes is non-correlated (resulting in a diagonal covariance matrix instead of a full one).

A number of algorithms exist to determine the numerical values of the parameters of a GMM such that it correctly predicts the values of a training dataset *x*. The quality of this prediction is equal to the likelihood of the dataset *x* given the parameters λ . This is,

$$L(\lambda : x) = p(x|\lambda) = \prod_{t=1}^{T} p(x_t|\lambda)$$

Where the parameters λ for a GMM are M, *w*, μ and Σ. Thus, the problem of determining the best predictor is equivalent to finding the parameters that optimize the likelihood. This gives name to the family of Maximum Likelihood Estimation (MLE) algorithms. While MLE for simple distributions is trivial, the GMM case presents a non-linear function of the parameters. The MLE of a GMM may solved through iterative methods, such as the Expectation-Maximization (EM) algorithm.

The EM algorithm is initialized with a set of *a priori* parameters λ . At each iteration, the algorithm looks for a set parameters λ' such that $p(x|\lambda') \geq p(x|\lambda)$. This process is repeated until convergence within a specified threshold reference. The following set of equations guarantee a monotonic increase of the likelihood thus enabling the advance towards an optimal model. They update the expectation of the Gaussian moments and thus compose the Expectation (E step) of the EM algorithm:

Mixture Weights:

$$w_i' = \frac{1}{T} \sum_{t=1}^{T} p(i|x_t, \lambda)$$

Means:

$$\mu_i' = \frac{\sum_{t=1}^{T} p(i|x_t, \lambda) x_t}{\sum_{t=1}^{T} p(i|x_t, \lambda)}$$

Variances:

$$\sigma_i'^2 = \frac{\sum_{t=1}^{T} p(i|x_t, \lambda) x_t^2}{\sum_{t=1}^{T} p(i|x_t, \lambda)} - \mu_i'^2$$

Where w_i' , μ_i' , and σ_i' refer to arbitrary elements of their respective vectors. The Maximization (M step) of the EM algorithm is performed by computing the *a posteriori* distribution. For component *i*, this is given by:

$$p(i|x_t, \lambda) = \frac{w_i g(x_t|\mu_i, \Sigma_i)}{\sum_{k=1}^{M} w_i g(\mu_k, \Sigma_k)}$$

2.3 Kullback–Leibler divergence

The Kullback–Leibler divergence (KL) is a method for determining the similarity between two probabilistic distributions (Kullback, 1951). Usually, one of the distributions is the real data (P) and the other (Q) is the distribution model that you want to use as an interpretation of your data. KL is a measure of how much information your model gives you about the data you are modeling. Formally, for discrete systems, KL is defined as:

$$D_{KL}(P||Q) = \sum_{i} ln(\frac{P(i)}{Q(i)})P(i)$$

Which can be described as the sum of the likelihood of observing one data with the distribution P if the particular model Q actually generates the data. The lower the KL distance is, the more similar the distributions are. In other words, lower KL results indicate that the statistical model Q, assumed for interpret the real data P, is good explaining it. Notice that this measure is distinct when talking about $D_{\mathrm{KL}}(P||Q)$ and $D_{\mathrm{KL}}(Q||P)$. This is the reason why it is considered a "non-symmetric" distance.

3 Tagging objects by color substitution

The full process of target object recoloring is depicted in Fig. 2.

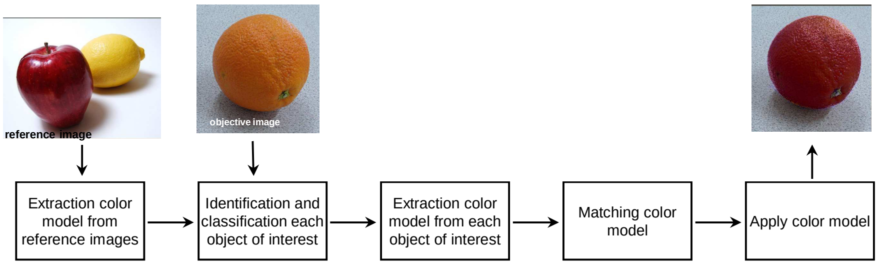
reference image
objective image
Extraction color model from reference images
Identification and classification each object of interest
Extraction color model from each object of interest
Matching color model
Apply color model

Figure 2: Full process.

It basically consists in the following 5 distinct steps.

1) We extract the color model of the reference images; each of them represents a category. This process is actually three-fold:

- Selection of the images from which to extract the color model.
- Segmentation of each object using the GrabCut algorithm, as explained in Section 2.1.
- Estimation of its GMM color model using the EM algorithm, both explained in Section 2.2.

2) In the target image, we perform a supervised identification of objects of interest. For this purpose we select the area of each object of interest and classify them within the categories available.

3) We segment each of the identified objects and extract their color model.

4) Using the KL distance described in Section 2.3, we match the Gaussians of the target object with that of the source object.

5) We apply the color model to tag the object (recoloring) through the method described in (Saphira, 2009).

As a result, we obtain a target image tagged with the source color. The next step would be to perform a task like monitoring, searching, tracking or targeting.

4 Experiments

Some results of our proposal are shown in Fig. 3, Fig. 4 and Fig. 5.

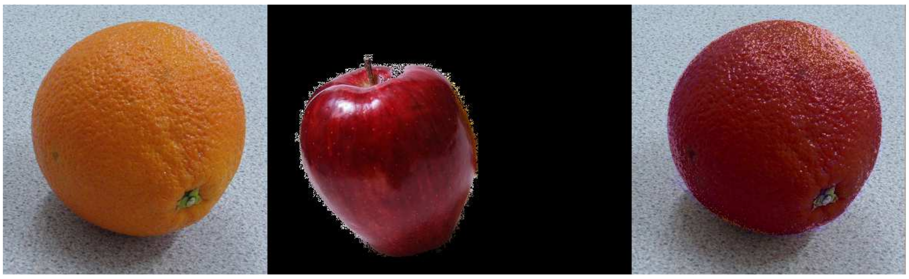

Figure 3: On the left side: an orange. In the center: an apple. On the right side: an orange recolored like an apple.

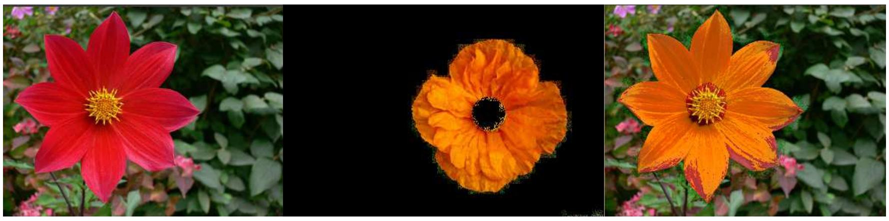

Figure 4: Process of image color substitution: The target object is selected (left image), extraction of color model of source object (center image), application of color model to the target object (right object).

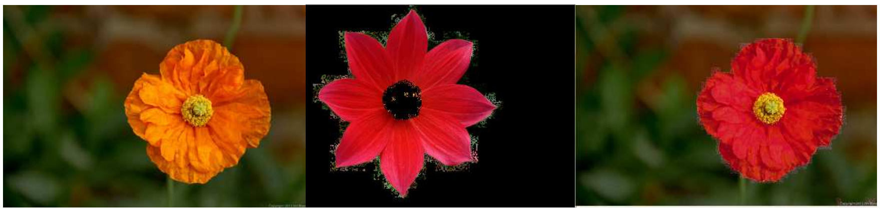

Figure 5: Process of image color substitution, swapping the source and the target.

Initially we select GMM with 3 components to define RGB values, one component to describe each channel. However, instead of choosing three components, we could have actually chosen any number of Gaussian components. The results with several different choices are shown in Figure 6.

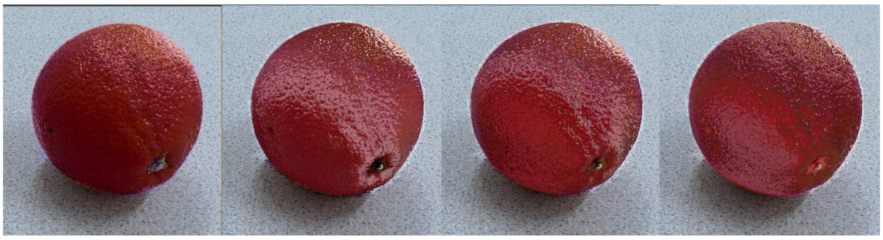

Figure 6: From left to right, object recoloring using 3, 6, 9 and 12 Gaussians.

One of the drawbacks of our application is that only a uniform color model can be learned at the same time. To use 2 or more colors for tagging, the process should be repeated for every color.

4 Conclusions

This paper proposes a tagging system for objects to be used in human-robot interaction for selection, discrimination or targeting. Target objects are tagged by color recoloring instead of the classical bounding box mechanism, thus avoiding environmental occlusion (especially relevant in cluttered environments) and the possibility of applying user-friendly color patterns for labeling.

Acknowledgements

The research leading to these results has received funding from the ARCADIA project DPI2010-21047-C02-01 funded by CICYT project grant on behalf of Spanish Ministry of Economy and Competitiveness.

References

Argall, B. D., Chernova, S., Veloso, M., & Browning, B. (2009). A survey of robot learning from demonstration. Robotics and Autonomous Systems, 57(5), 469-483.

Bergman, R., & Nachlieli, H. (2011). Perceptual segmentation: combining image segmentation with object tagging. Image Processing, IEEE Transactions on,20(6), 1668-1681.

Boykov, Yuri Y., and M-P. Jolly. "Interactive graph cuts for optimal boundary & region segmentation of objects in ND images." Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on. Vol. 1. IEEE, 2001.

J. G. Bueno; M.G.Fierro; L.Moreno; C.Balaguer. Facial Gesture Recognition using Active Appearance Models based on Neural Evolution . 2012 Conference on Human-Robot Interaction (HRI 2012). Boston. USA. Mar, 2012.

Csurka, G., Dance, C., Fan, L., Willamowski, J., & Bray, C. (2004, May). Visual categorization with bags of keypoints. In Workshop on statistical learning in computer vision, ECCV (Vol. 1, p. 22).

Huang, J. B., Chen, C. S., Jen, T. C., & Wang, S. J. (2009, April). Image recolorization for the colorblind. In Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on (pp. 1161-1164). IEEE.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. The Annals of Mathematical Statistics, 22(1), 79-86.

Pavlidis, T. (2009, June). Why meaningful automatic tagging of images is very hard. In Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on (pp. 1432-1435). IEEE.

P.Pierro; D.Hernandez; M.G.Fierro; C.Balaguer. L. Blasi; A. Milani. A human-humanoid interface for collaborative tasks. Second workshop for young researchers on Human-friendly robotics. Sestri Levante. Italy. Dec, 2009.

Pitie, Francois, Anil C. Kokaram, and Rozenn Dahyot. "N-dimensional probability density function transfer and its application to color transfer." Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on. Vol. 2. IEEE, 2005.

Reynolds, D. (2008). Gaussian mixture models. Encyclopedia of Biometric Recognition, 2(17.36), 14-68.

Rother, Carsten, Vladimir Kolmogorov, and Andrew Blake. "Grabcut: Interactive foreground extraction using iterated graph cuts." ACM Transactions on Graphics (TOG). Vol. 23. No. 3. ACM, 2004.

Shapira, L., Shamir, A., & Cohen□Or, D. (2009, April). Image Appearance Exploration by Model□Based Navigation. In Computer Graphics Forum (Vol. 28, No. 2, pp. 629-638). Blackwell Publishing Ltd.

Sigala, M. (2008). Web 2.0, social marketing strategies and distribution channels for city destinations: enhancing the participatory role of travelers and exploiting their collective intelligence. Information communication technologies and city marketing: Digital opportunities for cities around the world, 220-244.

Tai, Yu-Wing, Jiaya Jia, and Chi-Keung Tang. "Local color transfer via probabilistic segmentation by expectation-maximization." Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE, 2005.

Von Ahn, L., Maurer, B., McMillen, C., Abraham, D., & Blum, M. (2008). reCAPTCHA: Human-based character recognition via web security measures.Science, 321(5895), 1465-1468.