Automatic Single-Image People Segmentation and Removal for Cultural Heritage Imaging

Marco Manfredi, Costantino Grana, and Rita Cucchiara

Università degli Studi di Modena e Reggio Emilia, Modena MO 41125, Italy

Abstract. In this paper, the problem of automatic people removal from digital photographs is addressed. Removing unintended people from a scene can be very useful to focus further steps of image analysis only on the object of interest, A supervised segmentation algorithm is presented and tested in several scenarios.

Keywords: people removal, segmentation, cultural heritage imaging, inpainting

1 Introduction

Multimedia technologies find fruitful employment into tools and solutions for Cultural Heritage (CH) preservation and exploitation. Traditionally CH documents, reports and publications are created by few (expert) for many (e.g. users,tourists,etc...) based on expensive photographic and documentation campaigns. Instead, there is a huge heritage, especially in Italy and Europe, that is still only partially documented both for lack of economic support, and for terrible environmental changes (e.g. earthquakes), which change their conservation status and external aspects.

In this paper we address the problem of fully-automatic people removal from single images in order to create a privacy-compliant and de-personalized dataset of CH pictures, working on the huge picture footage coming from social networks and photo sharing sites. CHI fosters the development of technologies for digital capture and documentation of the worlds cultural, scientific, and artistic treasures, although most of the related techniques are up-to-now oriented to creating manual or semi-automatic tools both for experts and private users [1].

The "people removal" is a typical problem for digital documentation and consists in deleting humans from pictures to improve the quality or the semantic value of the picture. This is a useful feature of many graphic tools (e.g. GIMP), which is now also being provided for mobile devices[1] or as an online tool[2]. Indeed, people removal is a well-addressed problem on videos or image sequences, where multiple images differences [2, 3] or structure prediction [4] are available, while robust automatic solutions for single images have not been presented yet.

[1] http://www.scalado.com/display/en/Remove
[2] http://www.snapmania.com/info/en/trm

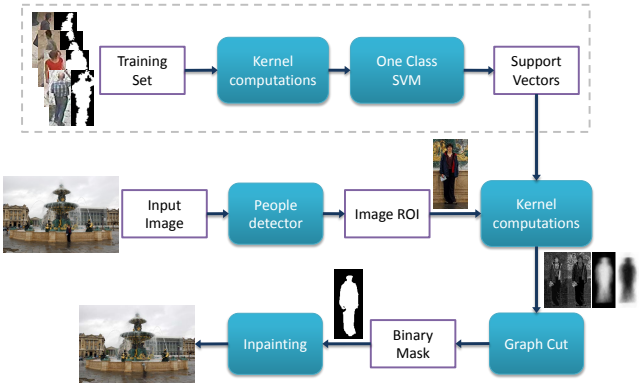

Fig. 1. Overview of the proposed system.

Three steps are required to address the single-image people removal task: automatic target detection, person segmentation and highlighted area inpainting. While an automatic people detector can be used to initialize the system, in our solution the detection are manually supplied, so to correctly initialize the further steps. A supervised segmentation algorithm, able to accurately outline the people thanks to a structured machine learning approach is adopted. This step creates the mask to apply photo enhancing algorithms, such as inpainting [5] or Seam Carving [6], to remove people in the scene without any user input.

In this paper we describe the system workflow, focusing in particular on the structured segmentation algorithm, comparing with other segmentation algorithms and illustrate results on a common segmentation dataset and on web images.

2 Single-image people removal

Taking a clear picture of a cultural heritage site, a church, a statue in the middle of a crowded square can be an undertaking. Unintended people often clutter the scene, ruining the shot. Since inpainting has to add unreal but plausible image data, the less is added, the better it is. So, usually, manual selection of the removal areas should be provided, in order to have a perfectly segmented shape, without under-segmented areas and possibly with very small over segmented ones. In our case the problem of target removal becomes a problem of automatic target detection and segmentation.

Automatic people (or pedestrian) detection is a well-studied problem in computer vision, Enzweiler and Gavrila provide a thorough survey on the topic [7]. All of these approaches return the bounding box where the target is found, without a precise silhouette segmentation, which is a challenging problem, due to the high variability of people pose and scale, different lighting condition, low contrast between the selected person and the background and not uniform color and texture distribution within the shape. The main strength of this work is indeed

the people segmentation step: a graph-cut [9] segmentation is employed, and the unary potentials are learned from a training set, weighting examples based on their similarity to the target image. We exploit One Class SVMs to describe the people class and Joint Kernels between image and mask pairs, to robustly learn an energy function to be minimized with an s/t graph cut framework.

This approach allows to cope with the fact that people shapes are found in several poses and postures, and walking pedestrians, chatting groups or other tourists have different aspects. Figure 1 summarizes the steps of the algorithm and describes the full system workflow.

3 Structural Segmentation

Structural prediction through SSVMs [10] proved to be effective in many computer vision tasks, such as object detection [11], tracking [12] and recently also image segmentation [13]. Structured segmentation describes the problem of learning a function $f : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ is the space of samples (images) and $\mathcal{Y}$ is the space of structured labels (binary masks), from a segmented training set. SSVMs learn a scoring function $F(x, y)$ that matches a sample x with a label y, such that maximizing F through the label space gives the correct output label for sample x.

A common approach is to work in the dual formulation using positive definite joint kernels, so that the scoring function $F(x, y)$ can be written as:

$$F(x,y) = \sum_{y' \in \mathcal{W}} \alpha_{y'} \left(\frac{1}{n} \sum_{i=1}^{n} [K((x,y),(x_i,y_i)) - K((x,y),(x_i,y_i'))] \right), \tag{1}$$

where $\mathcal{W}$ is the set of the most violated constraints, α are the weights for the support vectors that are found solving the dual problem, and (x_i, y_i) are the training pairs. Given an input image x, we can find the output label by maximizing $F(x, y)$:

$$y^* = \arg\max_{y \in \mathcal{Y}} F(x,y). \tag{2}$$

This maximization can be done using graph cuts. Unfortunately, this formulation has two relevant performance issues:

- during training we have to construct the set of the most violated constraints $\mathcal{W}$: for each training sample, find k constraints (k depends on the desired accuracy), each with the size of the training set, and solve an inference step (graph cut maximization) for each element;
- during testing we have to compare a sample x with each support vector. A single support vector is composed of all training images and their corresponding most violated constraint (mask), as in (1).

The main idea behind the proposed model is to exploit one-class SVMs in a kernel space to learn a set of support vectors and their relative weights and to delete outliers from the training set, thus reducing the complexity at testing time. This idea was firstly introduced by Lampert *et al.* [14], with the name of Joint Kernel Support Estimation, and applied to object localization and sequence labeling. We want to use $f(x) = \arg\max p(x, y)$ for prediction, assuming that $p(x, y)$ is high only if y is a correct label for x. The support of $p(x, y)$ can be effectively obtained by a one-class support vector machine (OC-SVM), expressing $p(x, y)$ as a linear combination of a suitable joint kernel K that matches two sample-label pairs. The joint kernel can be an arbitrary Mercer kernel [15]. The output of the OC-SVM learning process becomes a linear combination of kernel evaluations with training samples, thus the prediction function can be formulated as:

$$f(x) = \arg\max_{y \in Y} \sum_{i=1}^{n} \alpha_i K\left((x, y), (x_i, y_i)\right). \tag{3}$$

The learning process can be done using standard existing implementations of OC-SVM, with the joint kernel matrix between sample-label pairs.

It is important to point out the difference between our approach and KSSVMs [16]: in the training phase we only have to construct the joint kernel matrix between training samples, and then train a standard non linear OC-SVM, no inference steps are required during training. As a consequence, the training time does not depend on the structure of the output space, but only on the size of the training set.

The similarity kernel for the task of people segmentation was formulated as the product of an image kernel and a mask kernel:

$$K((x_i, y_i), (x_j, y_j)) = \theta(x_i, x_j) \cdot \Omega(x_i, x_j, y_i, y_j), \tag{4}$$

where $\theta(x_i, x_j)$ measures the similarity of the objects depicted in x_i and x_j, and acts as a weight for the mask similarity kernel $\Omega(x_i, x_j, y_i, y_j)$. Consequently, if the two images are very different, the final similarity measure will be low, even if the masks are similar.

3.1 Image Similarity Kernel

The purpose of the image similarity kernel is to return high similarity values between images that contain very similar objects. We adopt a general purpose similarity measure between images, the comparison of HOG descriptors [17]. HOGs can be compared using standard similarity measures like Bhattacharyya distance. Since we are working with images of the same category (people), we employ a Gaussian kernel, capable of distinguishing between different images, due to the parameter σ, optimized for the specific dataset. The image similarity kernel between image x_i and image x_j becomes:

$$\theta(x_i, x_j) = \exp\left(-\frac{\|\rho(x_i) - \rho(x_j)\|^2}{2\sigma^2}\right), \tag{5}$$

where $\rho(x_i)$ is the feature vector extracted from image x_i. For the computation of the HOG descriptors we adopted rectangular HOG (R-HOG) [17], computing gradients on R,G, and B color channels and taking the maximum, then dividing the image with a 5×5 grid of cells (25 cells), and grouping them in 4 partially overlapped blocks of 3×3 cells each. Trilinear interpolation between histogram bins and cells was appropriately applied.

3.2 Mask Similarity Kernel

The mask similarity kernel takes into consideration both images and masks to extract knowledge about how comparable two segmentations are. The kernel is composed of a linear combination of three parts:

$$\Omega(x_i, x_j, y_i, y_j) = \sum_{k=1}^{3} \beta_k \Omega_k(x_i, x_j, y_i, y_j). \tag{6}$$

The first kernel $\Omega_1(y_i, y_j)$ only depends on the binary masks, and directly compares the similarity between the two, by counting the number of corresponding pixels:

$$\Omega_1(y_i, y_j) = \frac{1}{P} \sum_{p=1}^{P} \delta(y_{ip}, y_{jp}), \tag{7}$$

where P is the total number of pixels in the image, y_{ip} is the p-th pixel of image y_i, and $\delta(\cdot, \cdot)$ is an indicator function defined as:

$$\delta(y_{ip}, y_{jp}) = \begin{cases} 1 & \text{if} \quad y_{ip} = y_{jp} \\ 0 & \text{if} \quad y_{ip} \neq y_{jp} \end{cases}. \tag{8}$$

The second and the third kernels exploit 3D color histograms computed in the RGB space. Let's define F_i^j and B_i^j as foreground and background histograms extracted from image x_i using mask y_j, and $P_r(x_p|H)$ as the likelihood of pixel x_p to match histogram H. We use negative log-likelihoods to express the penalties to assign a pixel to foreground or to background, as firstly introduced by [18]. Negative log-likelihoods are defined as:

$$L(x_p \,|\, H) = -log(P_r(x_p|H)). \tag{9}$$

We can also define:

$$L(x_p|y_p, F, B) = \begin{cases} L(x_p \,|\, B) & \text{if} \quad y_p = \text{“}obj\text{”} \\ L(x_p \,|\, F) & \text{if} \quad y_p = \text{“}bkg\text{”}. \end{cases} \tag{10}$$

To highlight the mutual agreement of two masks, the second kernel extracts an histogram from image x_i using mask y_j and evaluates it using y_i. Having F_i^j and B_i^j:

$$\Omega_2(x_i, y_i) = \frac{1}{P} \sum_{p=1}^{P} L(x_{ip}|y_{ip}, F_i^j, B_i^j). \tag{11}$$

The third kernel exploits global features extracted from the entire training set to model the expected color distribution of foreground and background pixels. We define F_G and B_G as the global histograms extracted from training samples using their relative masks.

$$\Omega_3(x_i, x_j, y_i, y_j) = \Omega_{3i} \cdot \Omega_{3j} \tag{12}$$

where

$$\begin{array}{l} \Omega_{3i} = \frac{1}{P} \sum_{p=1}^{P} L(x_{ip}|y_{ip}, F_G, B_G) \\ \Omega_{3j} = \frac{1}{P} \sum_{p=1}^{P} L(x_{jp}|y_{jp}, F_G, B_G) \end{array}. \tag{13}$$

The histograms are quantized uniformly over the 3D color space using a fixed number of bins per channel, set at 16 by experimental evaluations (no smoothing is applied).

4 Graph construction

It is worth noting that the previously defined kernels compare two image-mask pairs, while at testing time the test mask is obviously missing. Kernels must thus be reformulated so to return pixel-wise potentials, in order to perform the maximization reported in (3). This maximization is done using s/t graph cuts [18].

The problem can be formulated as a maximum a posterior estimation of a Markov Random Field, minimizing the energy function:

$$E(y) = R(y) + \lambda B(y), \tag{14}$$

where y is a binary vector of pixel labels and $R(y)$ is the unary term expressing the cost of assigning a pixel to the foreground or to the background. $B(y)$ is the smoothness term, formulated as proposed by Rother *et al.* [19]:

$$B(y) = \sum_{p,q \in \mathcal{N}} \delta(y_p, y_q) \frac{1}{dist(p,q)} \exp\left(-\frac{\|x_p - x_q\|^2}{2\sigma^2}\right) \tag{15}$$

where $\mathcal{N}$ is the set of neighboring pixels (8-connected), $\delta(\cdot,\cdot)$ is the indicator function defined in (8), $dist(p,q)$ is the distance between pixels and σ is the expectation of the euclidean distance in color space $\|x_p - x_q\|^2$. At classification time we have to compute the foreground and background potentials P_f and P_b corresponding to the unary term in the graph cut framework. They are the result of a linear combination of potentials P_{f_i} and P_{b_i} obtained from the comparison of the testing image x_j with each support vector (x_i, y_i), weighted by the corresponding α_i. The potentials at position p are:

$$\begin{array}{l} P_{f\,ip} = \theta(x_i, x_j)(\beta_1 P_{f\,ip}^{\,1} + \beta_2 P_{f\,ip}^{\,2} + \beta_3 P_{f\,ip}^{\,3}) \\ P_{bip} = \theta(x_i, x_j)(\beta_1 P_{bip}^{1} + \beta_2 P_{bip}^{2} + \beta_3 P_{bip}^{3}) \end{array} \tag{16}$$

where $\theta(x_i, x_j)$ is the image similarity kernel defined in (5). The first kernel is strictly related to the mask y_i:

$$\begin{aligned} P_{f_{ip}}^{1} &= y_{ip} \\ P_{b_{ip}}^{1} &= 1 - y_{ip} \end{aligned} \tag{17}$$

The second kernel expresses the cost of assigning a pixel to foreground or to background, according to the histograms F_j^i and B_j^i, defined in Sec. 3.2:

$$\begin{aligned} P_{f_{ip}}^{2} &= L(x_{jp} \,|\, B_j^i) \\ P_{b_{ip}}^{2} &= L(x_{jp} \,|\, F_j^i) \end{aligned} \tag{18}$$

The third kernel expresses the cost of assigning a pixel to foreground or to background, given the global histograms F_G , B_G calculated on the training set:

$$\begin{aligned} P_{f_{ip}}^{3} &= L(x_{jp}|B_G) \cdot \tfrac{1}{P}\gamma(x_i, y_i) \\ P_{b_{ip}}^{3} &= L(x_{jp}|F_G) \cdot \tfrac{1}{P}\gamma(x_i, y_i) \end{aligned} \tag{19}$$

where

$$\gamma(x_i, y_i) = \sum_{p=1}^{P} L(x_{ip}|y_{ip}, F_G, B_G). \tag{20}$$

5 Experimental Evaluation

The experimental setup is thought to evaluate both the segmentation accuracy and the inpainting effectiveness, comparing the proposed approach with other segmentation algorithms. To test the segmentation accuracy we chose the Weizmann horse dataset [20], publicly available and commonly used in the segmentation community. We compared our results with GrabCut, initialized using the bounding boxes coming from a part based detector [21, 22], or using the average of the masks of the k nearest image found with the object similarity of Eq. 5. We also report the results obtained by Bertelli *et al.* [16]. Their approach exploits Kernelized Structural SVMs, and reaches the best results (Table 2). The accuracy gap between our solution and the one proposed in [16] is noticeable, but To

Table 1. Image reconstruction results with different initializations.

Inpainting Init.	MSE	SSIM
Our approach	57.24	0.469
GrabCut	67.55	0.391
Bounding Box	110.3	0.123

test the impact that different segmentations have on the inpainting procedure, we collected some images from the web, depicting CH sites with unintended

Fig. 2. Visual comparison of the segmentation results. First column contains the original image with the detected people. The other columns show the area to be inpainted using the bounding box, the GrabCut and the proposed approach respectively.

people cluttering the scene. A qualitative comparison between our approach and GrabCut shows that in the 70% of cases we perform better

We created a dataset to test the impact that different segmentations have on the inpainting procedure. The dataset is composed of 60 photographs of churches and monuments, with several people cluttering the scene. For each photograph, a background image is provided. The inpainting algorithm [5], is initialized in three different ways: directly using the bounding box of the people detection, using the binary mask provided by GrabCut and using the proposed approach. To evaluate the quality of the image reconstruction step we use Structural Similarity Index (SSIM). SSIM is commonly used for image quality assessment, and measures the changes in structural information given a couple of images. SSIM is chosen because more consistent human eye perception than MSE or PSNR.

Results are reported in Table 1. As expected, initializing the inpainting algorithm with the bounding boxes leads to the worst results, in particular, the accuracy of the reconstruction quickly worsen as the person size increases. Het-

erogeneous background is very difficult to reconstruct, but the precision of the segmentation mask considerably affect the reconstruction process. Some qualitative results are reported in Figure 2.

Some parameters of the proposed solution must be optimized over a validation set. These are the kernel weights $\beta_1, \beta_2, \beta_3$ and the parameter ν of the One Class SVM. As for training, the optimization step is done over the 3DPeS dataset, by means of a grid search over the parameter space.

Table 2. Performance comparison on the three datasets.

Horses Dataset	$S_a(\%)$	$S_o(\%)$
KSSVM + Hog feature	93.9	77.9
Our method	**91.04**	**73.28**
GrabCut init. with BB	69.53	50.39
GrabCut init. with 1-NN mask	85.66	62.34
GrabCut init. with 5-NN masks	86.93	63.83
GrabCut init. with 10-NN masks	86.46	63.20

6 Conclusions

We proposed a novel segmentation approach based on one-class SVMs and joint kernels between image-mask pairs. The method exploits the ability of OC-SVMs to identify and ignore outliers in the training set, while reducing the number of kernel computations needed at classification time. The characteristics of this generative learning algorithm allow to deal with very large datasets, otherwise intractable using discriminative approaches like KSSVMs and increase the robustness to mislabeled or incomplete ground truths.

References

1. Mudge, M., Ashley, M., Schroer, C.: A digital future for cultural heritage. In: Proceedings of the 21st CIPA symposium. Volume XXXVI-5/C53 of ISPRS Archives. (October 2007)
2. Uchiyama, H., Deguchi, D., Takahashi, T., Ide, I., Murase, H.: Removal of moving objects from a street-view image by fusing multiple image sequences. In: Proceedings of the 20th International Conference on Pattern Recognition. (August 2010) 3456–3459
3. Flores, A., Belongie, S.: Removing pedestrians from google street view images. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on. (June 2010) 53–58
4. Wexler, Y., Shechtman, E., Irani, M.: Space-time completion of video. Pattern Analysis and Machine Intelligence, IEEE Transactions on **29**(3) (March 2007) 463–476

5. Criminisi, A., Perez, P., Toyama, K.: Object removal by exemplar-based inpainting. In: Proceedings of 16th IEEE Conference on Computer Vision and Pattern Recognition. Volume 2. (June 2003) 721–728
6. Avidan, S., Shamir, A.: Seam carving for content-aware image resizing. ACM Trans. Graph. **26**(3) (July 2007)
7. Enzweiler, M., Gavrila, D.: Monocular pedestrian detection: Survey and experiments. Pattern Analysis and Machine Intelligence, IEEE Transactions on **31**(12) (dec. 2009) 2179 –2195
8. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. (june 2009) 304 –311
9. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(9) (September 2004) 1124–1137
10. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. Journal of Machine Learning Research **6** (2005) 1453–1484
11. Girshick, R.B., Felzenszwalb, P.F., McAllester, D.A.: Object Detection with Grammar Models. In: Neural Information Processing Systems. (2011) 442–450
12. Hare, S., Saffari, A., Torr, P.: Struck: Structured output tracking with kernels. In: Proceedings of the 13th International Conference on Computer Vision. (November 2011) 263–270
13. Nowozin, S., Gehler, P., Lampert, C.: On Parameter Learning in CRF-Based Approaches to Object Class Image Segmentation. In: Proceedings of the 10th European Conference on Computer Vision. Volume 6316 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2010) 98–111
14. Lampert, C., Blaschko, M.: Structured prediction by joint kernel support estimation. Machine Learning **77** (2009) 249–269
15. Vapnik, V.: Statistical learning theory. Wiley (1998)
16. Bertelli, L., Yu, T., Vu, D., Gokturk, B.: Kernelized structural SVM learning for supervised object segmentation. In: Proceedings of the 24th IEEE Conference on Computer Vision and Pattern Recognition. (June 2011) 2153–2160
17. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of the 18th IEEE Conference on Computer Vision and Pattern Recognition. Volume 1. (June 2005) 886–893
18. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In: Proceedings of the 8th IEEE International Conference on Computer Vision. Volume 1. (2001) 105–112
19. Rother, C., Kolmogorov, V., Blake, A.: "GrabCut": interactive foreground extraction using iterated graph cuts. In: ACM SIGGRAPH 2004 Papers. SIGGRAPH '04, ACM (2004) 309–314
20. Borenstein, E., Sharon, E., Ullman, S.: Combining top-down and bottom-up segmentation. In: Proceedings of the 17th IEEE Conference on Computer Vision and Pattern Recognition Workshops. (June 2004) 46
21. Girshick, R.B., Felzenszwalb, P.F., McAllester, D.: "Discriminatively Trained Deformable Part Models, Release 5". http://people.cs.uchicago.edu/ rbg/latent-release5/
22. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(9) (2010) 1627–1645