PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Object detection using feature-based template matching

Simone Bianco, Marco Buzzelli, Raimondo Schettini

Simone Bianco, Marco Buzzelli, Raimondo Schettini, "Object detection using feature-based template matching," Proc. SPIE 8661, Image Processing: Machine Vision Applications VI, 86610C (6 March 2013); doi: 10.1117/12.2006224

Event: IS&T/SPIE Electronic Imaging, 2013, Burlingame, California, United States

SPIE.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 7/5/2018 Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

Object Detection Using Feature-based Template Matching

Simone Bianco, Marco Buzzelli, Raimondo Schettini

DISCo (Dipartimento di Informatica, Sistemistica e Comunicazione)
Università degli Studi di Milano-Bicocca, viale Sarca 336, 20126 Milano, Italy

ABSTRACT

Pattern matching, also known as template matching, is a computationally intensive problem aimed at localizing the instances of a given template within a query image. In this work we present a fast technique for template matching, able to use histogram-based similarity measures on complex descriptors. In particular we will focus on Color Histograms (CH), Histograms of Oriented Gradients (HOG), and Bag of visual Words histograms (BOW). The image is compared with the template via histogram-matching exploiting integral histograms. In order to introduce spatial information, template and candidates are divided into sub-regions, and multiple descriptor sizes are computed. The proposed solution is compared with the Full-Search-equivalent Incremental Dissimilarity Approximations, a state of the art approach, in terms of both accuracy and execution time on different standard datasets.

Keywords: Template matching, Color Histograms, Histograms of Oriented Gradients, Bag of Words, Integral Histogram

1. INTRODUCTION

Pattern matching, also known as template matching, is a computationally intensive problem aimed at localizing the instances of a given template T within a query image I. Pattern matching founds numerous applications in signal processing, computer vision, and image and video processing domains: product quality control,[1] image compression,[2] object detection,[3] block matching in motion estimation,[4] image denoising,[5] and action recognition.[6] The easiest pattern matching algorithm is the Full Search (FS) which relies on calculating at each position of the image I a similarity or dissimilarity score between the template T and the current image sub-window under analysis, also called candidate. Once the similarity has been computed for all candidates, a threshold is adopted to classify the candidates into matching patterns and mismatching ones. Numerous exhaustive Full Search equivalent techniques have been recently proposed to speed-up template matching on grayscale images,[7] and also on multi-channel images.[8] Most of these techniques use pixel-based functions such as L^p norm or correlation as the similarity measure.

In this work we present a fast technique for template matching, able to use histogram-based similarity measures. In particular we will focus on Color Histograms (CH), Histograms of Oriented Gradients (HOG),[9] and Bag of visual Words (BOW).[10] To obtain a BOW description of an image, we compute Scale-Invariant Feature Transform (SIFT)[11] and the resulting descriptors are vector quantized using a k-entry codebook of visual word prototypes. Although with very different meanings, the output of the three representations is an indexed image, where for the case of Color Histograms each index corresponds to a different quantized color in the color space used. For the HOG representation, each index corresponds to a different quantized edge direction. For the BOW representation each index corresponds to a different feature prototype. The performances of our method (applied to different descriptors) are compared with those of the Incremental Dissimilarity Approximation algorithm (IDA),[12] a full-search-equivalent method for template matching based on the L^p norm in terms of both accuracy and execution time on different standard datasets.

2. THE PROPOSED METHOD

The proposed method is structured in two main parts: image and template description, and the actual search of the template in the image.

Image Processing: Machine Vision Applications VI, edited by Philip R. Bingham, Edmund Y. Lam,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 8661, 86610C · © 2013 SPIE-IS&T
CCC code: 0277-786X/13/$18 · doi: 10.1117/12.2006224

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 7/5/2018
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

2.1 Image and Template Description

The first part consists in applying the chosen descriptor (CH, HOG, or BOW) to two dense grids over both the image and the template. The choice of grid-cell size is arbitrary. The size of the second grid-cell size if twice the first one.

The original $w \times h$ image is then represented with two grids of d-unit-long descriptors (for example $d = 128$ in the case that the SIFT descriptor is used). The next step is the quantization of the descriptors themselves. For this purpose a vocabulary of word prototypes is needed, which can be created as follows:

- Extraction of the chosen descriptors from a set of suitable images (see Sec. 3.1).
- Clustering of the descriptors. The chosen implementation is based on Euclidean distance, complete linkage. Each cluster is then represented by its median.
- Creation of a data structure for the median descriptors to facilitate the subsequent search. An efficient tool for this purpose is the k-dimensional tree,[13] for which the VLFeat[14] implementation has been used.

After the creation of the vocabulary, which may be executed every time on the searched template, or once and for all on a training set of images, it is possible to find for each descriptor in the image and template the most similar one from the vocabulary, assigning it the corresponding index. This may be seen as the search for the nearest point in a k-dimensional space, and is in fact efficiently performed by applying a nearest neighbor search to the k-dimensional tree created.[15]

2.2 Search of the Template in the Image

The search of the template in the image is done with a sliding window of the same size of the template. For each location (x, y) of the top-left corner of the sliding window over the image (i.e. for each candidate match), the histogram of indexes is calculated and compared with that of the template. In order to add spatial information to histogram-based representations, both the template and candidate matches are partitioned into sub-regions.

Integral histograms[16] have been adopted to increase the computational efficiency of the histograms. An integral histogram IH is the natural extension of the concept of integral image[17] to histograms, and consists in a 3-dimensional structure with height and width equal to those of the original image I, and depth equal to the cardinality of the vocabulary of descriptors used. For each location with row and column indexes (x, y) in the IH structure, the following identity holds along the third dimension:

$$IH(x,y) = \sum_{\substack{x' \leq x \\ y' \leq y}} i(x', y') \tag{1}$$

From the IH structure it is possible to compute the histogram h of any rectangular region with vertices $ABCD$ with a constant-time operation:

$$\sum_{\substack{x_A < x' \leq x_C \\ y_A < y' \leq y_C}} h(x', y') = IH(C) + IH(A) - IH(B) - IH(D) \tag{2}$$

Where $+$ and $-$ are bin-wise operations, and histogram normalization is performed after the algebraic sum of the four histograms.

The population of the IH structure can be done efficiently starting from the histogram of the first pixel $(1, 1)$, and progressively building the nearby histograms by adding one single index to one single bin for each movement:

$$IH\,(x, y) = h(x, y) + IH(x-1, y) + IH(x, y-1) - IH(x-1, y-1) \tag{3}$$

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 7/5/2018
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

Since the descriptors extraction uses two different sizes, two matrices of indexes for every image are created. To manage this extra information we sum sum the integral histograms of the two matrices.

Now, for each position of the sliding window over the image, the portion it covers is split into the specified number of sub-regions, and each i^{th} sub-region is compared with the corresponding sub-region of the template by applying a sum of square differences of the histogram bins. The sum of the distances relative to the sub-regions in the current candidate is then compared with a specified threshold, in order to determine if rejecting or accepting the match. Another possible strategy is to discard a candidate whenever the distance of two corresponding sub-regions exceeds a specified value. The choice made here makes the search more tolerant towards occlusions, by accepting a candidate with some regions very different to the corresponding ones in the template as long as the others are very similar.

3. EXPERIMENTAL RESULTS

The performances of our method are compared with those of the Incremental Dissimilarity Approximation algorithm (IDA),[12] a full-search-equivalent method for template matching based on the L^p norm. IDA involves the use of two dissimilarity measures (or "distances") between template T and candidate region $I_s(x,y)$, and divides them into sub-regions to speed up the search. Let δ be the first distance, which is computationally expensive:

$$\delta_p(x,y) = \|I_s(x,y) - T\|_p^p \tag{4}$$

And let β be the second distance, which is far quicker than the first one:

$$\beta_p(x,y) = \left|\|I_s(x,y)\|_p - \|T\|_p\right|^p \tag{5}$$

where $\|\cdot\|_p$ is the L^p norm:

$$\|x\|_p = \left(\sum |x_i|^p\right)^{\frac{1}{p}} \tag{6}$$

The authors have shown[12] that the less expensive distance β is always a lower bound of the distance δ. So, for each candidate, they first compute the partial distances β' for each pair of corresponding sub-regions, which once summed up result in the value β for the whole template-candidate pair. Then, knowing that $\beta \leq \delta$, if β exceeds a given threshold then the candidate will certainly have to be discarded. If, instead, the distance β does not exceed the threshold, they compute the more expensive partial distance δ' on the first sub-region, and use it in the sum of partial distances along with the less expensive one of the other sub-regions. Iteratively, this value is compared with the given threshold, and if the condition holds (i.e. the distance is lower than the threshold) the expensive distance on the next sub-region is computed and substituted. A candidate is accepted when the expensive distance δ has been calculated on every sub-region and is below the given threshold.

The computational cost of the template search is highly dependent on the nature of the image itself: when false instances are very similar to the template, the rejection of the candidate typically occurs in the latest substitutions thus requiring the computation of the expensive distance on many sub-regions. When false instances are extremely different with respect to the template, they will be discarded during the first substitutions, without the need of computing the expensive distance on many sub-regions.

Three variants of our method (CH, HOG, and BOW), with the number of sub-regions set to three, are compared against the Full-Search-equivalent IDA algorithm for both accuracy and computational time. The tests are performed on the three standard datasets used by Tombari et al.:[18]

1. **Guitar**
   Seven 63 × 63 color templates extracted from a good quality picture are searched for in 10 variants of the scene (640 × 480) shot under different lighting conditions.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 7/5/2018
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

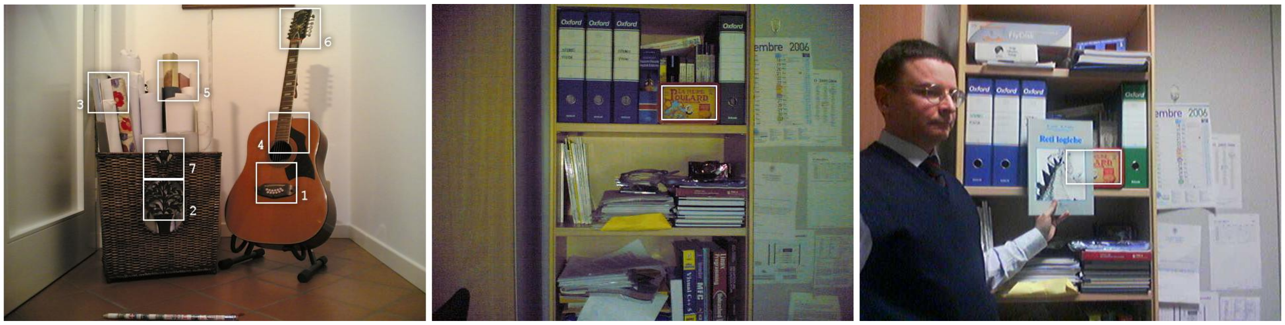

Figure 1. One example image from each dataset, with the searched templates highlighted

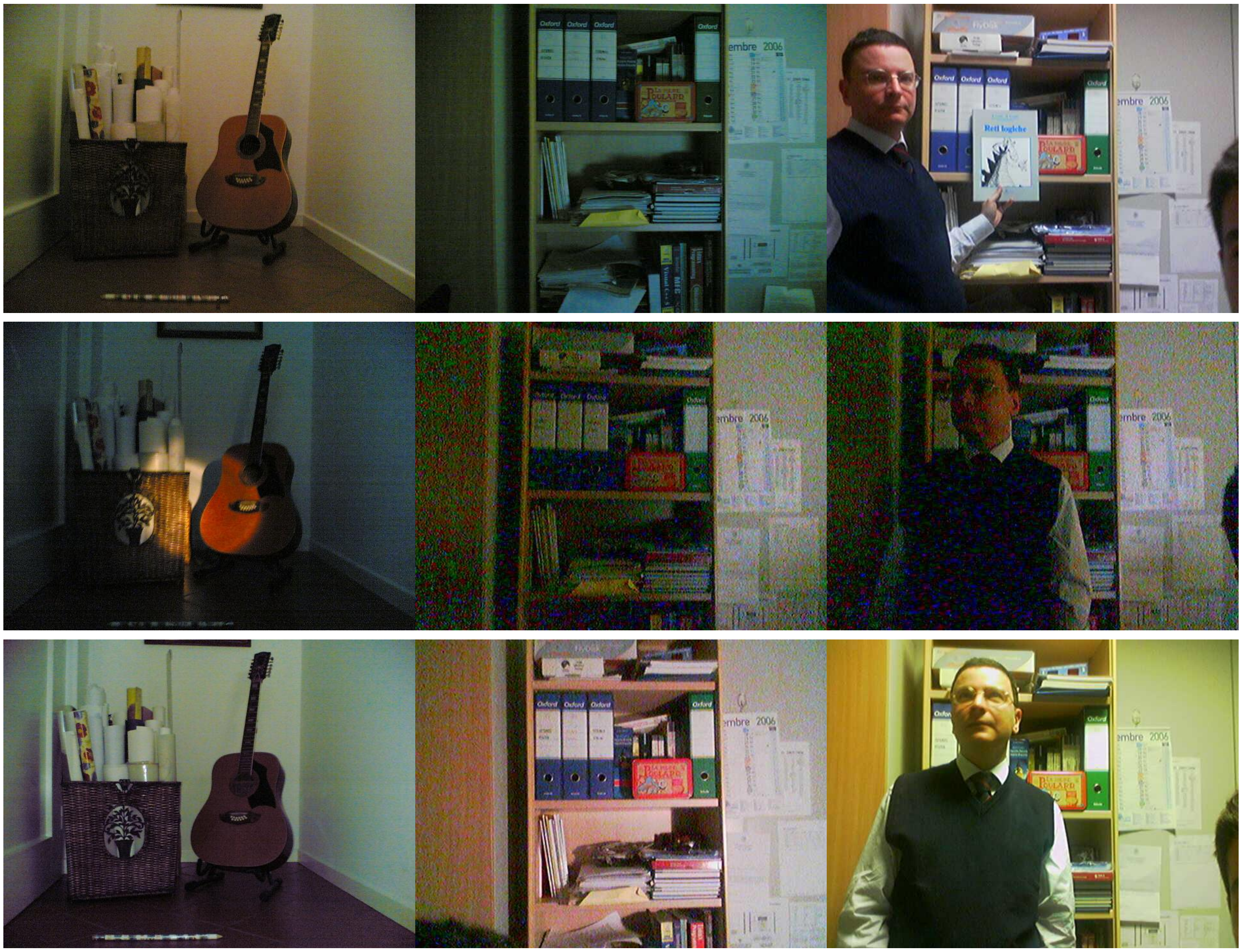

Figure 2. Scene variability within the three datasets

2. **Mere Poulard - Illumination Changes**
   One 85×53 color template is searched for in 12 variants of a scene (640×480) shot under different lighting conditions.

3. **Mere Poulard - Occlusions**
   One 85×53 color template is searched for in 8 variants of the same scene (640×480) with different levels of occlusions of the template instance in the image.

The code was written in MATLAB (release 2012a) and supported with C++ libraries. Tests have been

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 7/5/2018
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

	Guitar	M.P. A	M.P. B	**Total**
IDA	19/70	0/12	0/8	**19/90**
CH	10/70	1/12	2/8	**13/90**
BOW	54/70	5/12	2/8	**61/90**
HOG	34/70	2/12	4/8	**40/90**

	Guitar	M.P. A	M.P. B	**Total**
IDA	19/70	0/12	0/8	**19/90**
CH	14/70	0/12	0/8	**14/90**
BOW	55/70	4/12	5/8	**64/90**
HOG	23/70	2/12	2/8	**27/90**

Table 1. Results with ad-hoc (left) and default (right) vocabularies, for the three datasets

		Guitar	Mere P. A	Mere P. B
IDA	vocab	<0.0001s (<0.0001s)	<0.0001s (<0.0001s)	<0.0001s (<0.0001s)
	template	0.0010s (0.0010s)	0.0008s (0.0006s)	0.0006s (0.0001s)
	total	**0.0010s**	**0.0008s**	**0.0006s**
CH	vocab	0.0395s (0.0056s)	0.0438s (<0.0001s)	0.0478s (<0.0001s)
	template	0.0174s (0.0180s)	0.0163s (0.0024s)	0.0154s (0.0007s)
	total	**0.0569s**	**0.0601s**	**0.0632s**
BOW	vocab	0.0621s (0.0034s)	0.0759s (<0.0001s)	0.0776s (<0.0001s)
	template	0.0251s (0.0075s)	0.0253s (0.0018s)	0.0250s (0.0017s)
	total	**0.0872s**	**0.1012s**	**0.1026s**
HOG	vocab	0.0452s (0.0005s)	0.0526s (<0.0001s)	0.0541s (<0.0001s)
	template	0.0164s (0.0026s)	0.0179s (0.0015s)	0.0176s (0.0013s)
	total	**0.0616s**	**0.0705s**	**0.0717 s**

Table 2. Average offline times (and standard deviations in brackets) for the three datasets, expressed in seconds

performed on the following hardware: Operating System: Microsoft Windows 7; CPU: Intel Core2 Duo Processor E6750, 4M Cache, 2.66 GHz, 1333 MHz FSB; RAM: 2.00GB.

3.1 Recognition Accuracy

Tests have been performed with both ad-hoc vocabularies, built on-the-fly from each template, and with a default one, built only once from the whole dataset.

For each template-image pair, the output of every investigated method is counted as valid when at least 50% of the bounding box intersects the ground truth provided.

As can be seen in Table 1, among the three variants proposed, the BOW and HOG ones outperform the IDA method, with the BOW-based variant producing the overall best results. The CH-based variant instead, produces the overall worst results. For what concerns the choice of vocabulary, the main difference lies in the HOG variant, with a total score of 40/90 for the ad-hoc vocabularies against 27/90 with the use of a default one.

3.2 Computational performances

Computational time has been split into offline and online.
For the proposed method, offline time refers to vocabulary generation (supposing the creation of one vocabulary from each template) and template description. For the IDA algorithm offline time is basically null, since little or no precomputation is required. Online time involves image description (when required) and the template search itself.

Offline time shows how the use of a descriptor-based representation introduces an additional computational burden (see Figure 3 and Table 2). On the other hand, online times with the proposed method are on average better than those obtained with the already time-efficient IDA, and more predictable, as can be seen from the standard deviation values in Table 3 and observed the graphs in Figure 4.

For what concerns the time-performances, the new method asks for some precomputation that is not required by the IDA algorithm, but outperforms it with average lower online times (composed of the image transformation and the search of the template in the image). The total time required by the HOG-variant is thus on average 56.7% less that that required by IDA, while that required by the BOW-variant is on average 39.6% less.

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 7/5/2018
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

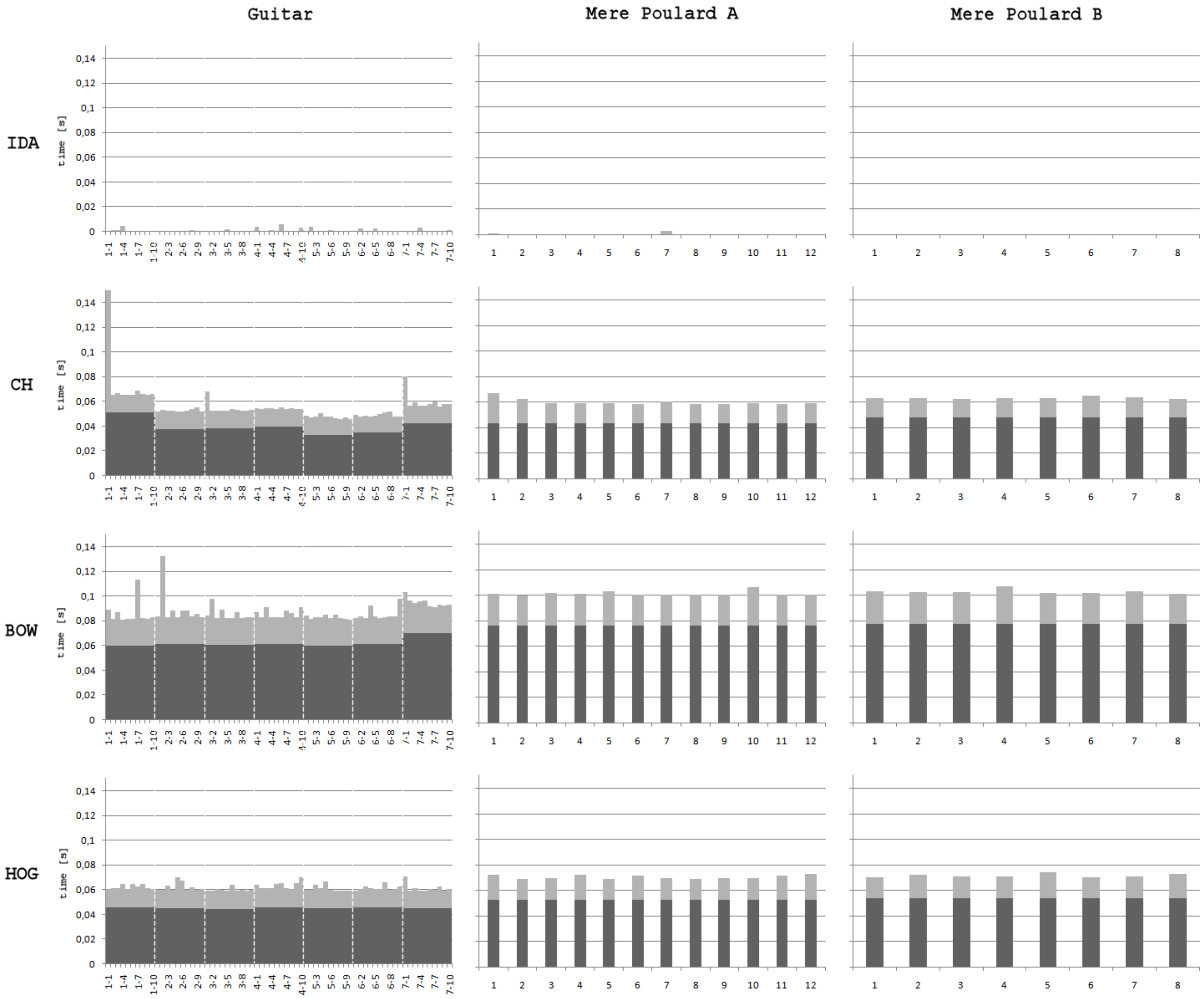

Figure 3. Offline times for the three datasets

		Guitar	Mere P. A	Mere P. B
IDA	image	0.3005s (0.0309s)	0.3234s (0.0565s)	0.2989s (0.0201s)
	search	6.1291s (3.5575s)	4.8632s (2.7816s)	4.0519s (2.7730s)
	total	**6.4296s**	**5.1866s**	**4.3508s**
CH	image	1.1076s (0.1202s)	1.0673s (0.0224s)	1.0489s (0.0188s)
	search	0.5648s (0.0123s)	0.5852s (0.1326s)	0.5453s (0.0039s)
	total	**1.6724s**	**1.6524s**	**1.5943s**
BOW	image	2.5548s (0.0857s)	2.6101s (0.0534s)	2.5365s (0.0364s)
	search	0.5549s (0.0121s)	0.5469s (0.0100s)	0.5424s (0.0036s)
	total	**3.1097s**	**3.1571s**	**3.0790s**
HOG	image	1.6523s (0.0294s)	1.7075s (0.1951s)	1.6817s (0.1555s)
	search	0.5723s (0.0716s)	0.5497s (0.0073s)	0.5464s (0.0055s)
	total	**2.2246s**	**2.2572s**	**2.2281s**

Table 3. Average online times (and standard deviations in brackets) for the three datasets, expressed in seconds

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 7/5/2018
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

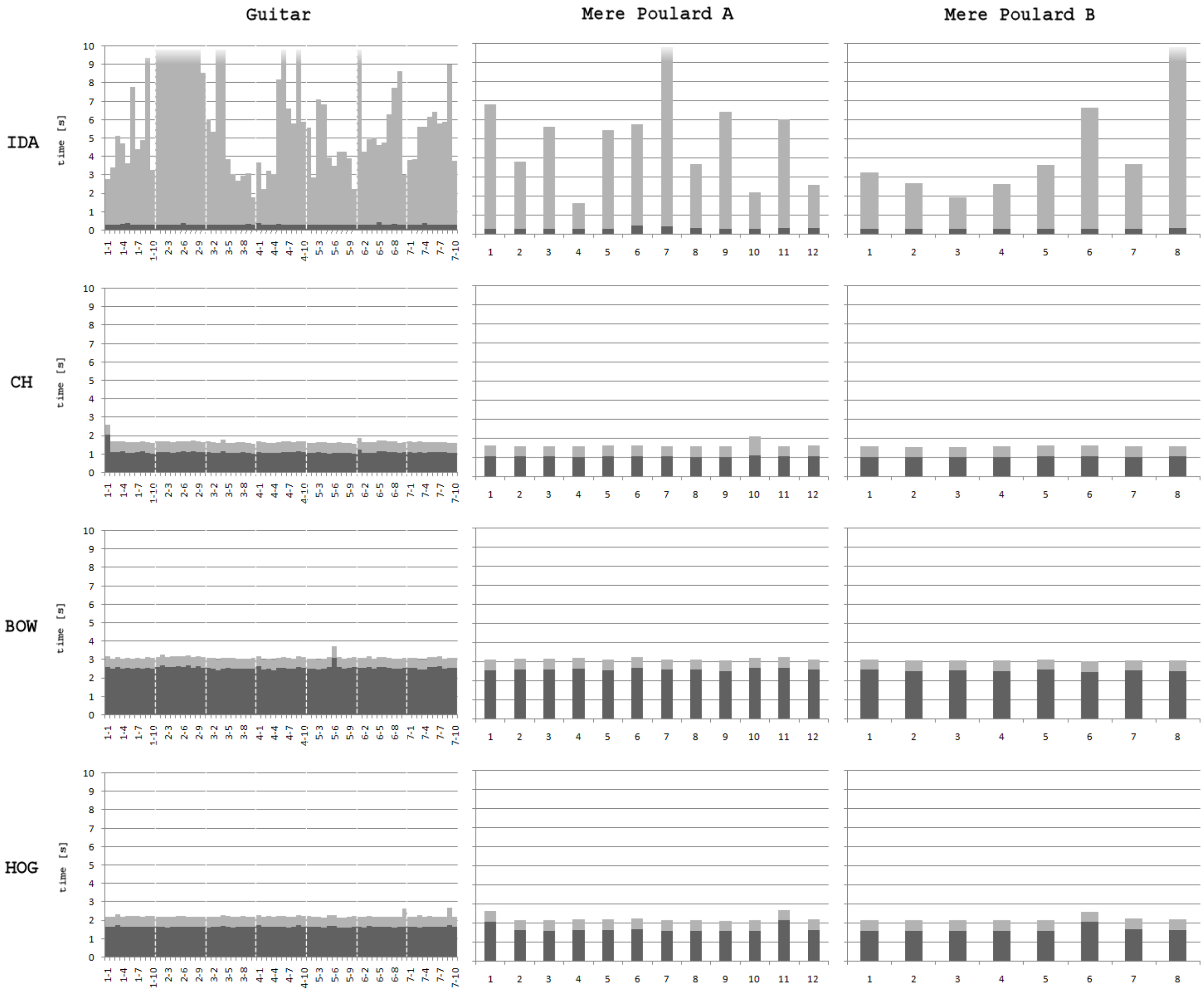

Figure 4. Online times for the three datasets

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 7/5/2018
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use

4. CONCLUSIONS

In this work we have presented a novel technique for template matching, able to use histogram-based similarity measures. Among the possible histogram descriptors, we have focused on Color Histograms (CH), Histograms of Oriented Gradients (HOG), and Bag of visual Words histograms (BOW). The proposed method has been compared with a fast Full-Search equivalent method based on the L_p norm[7] on different datasets. The experimental results showed that the proposed method using HOG and BOW descriptors proves to be more robust and efficient than the IDA algorithm.

Further development may include the ability to better generalize the search with the use of multiple templates depicting the same object, together with the design of a strategy to make the proposed method more robust with respect to scale and rotation changes.

REFERENCES

[1] Aksoy, M., Torkul, O., and Cedimoglu, I., "An industrial visual inspection system that uses inductive learning," *Journal of Intelligent Manufacturing* **15**(4), 569–574 (2004).

[2] Luczak, T. and Szpankowski, W., "A suboptimal lossy data compression based on approximate pattern matching," *Information Theory, IEEE Transactions on* **43**(5), 1439–1451 (1997).

[3] Dufour, R., Miller, E., and Galatsanos, N., "Template matching based object recognition with unknown geometric parameters," *Image Processing, IEEE Transactions on* **11**(12), 1385–1396 (2002).

[4] Moshe, Y. and Hel-Or, H., "Video block motion estimation based on gray-code kernels," *Image Processing, IEEE Transactions on* **18**(10), 2243–2254 (2009).

[5] Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K., "Image denoising by sparse 3d transform-domain collaborative filtering," *Image Processing, IEEE Transactions on* **18**(10), 2080–2095 (2007).

[6] Wu, X., *Template-Based Action Recognition: Classifying Hockey Players Movement*, PhD thesis, The University of British Columbia (2005).

[7] Ouyang, W., Tombari, F., Mattoccia, S., Di Stefano, L., and Cham, W., "Performance evaluation of full search equivalent pattern matching algorithms," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **34**(1), 127–143 (2012).

[8] Mattoccia, S., Tombari, F., and Di Stefano, L., "Efficient template matching for multi-channel images," *Pattern Recognition Letters* **32**(5), 694–700 (2011).

[9] Dalal, N. and Triggs, B., "Histograms of oriented gradients for human detection," in [*Computer Vision and Pattern Recognition*], 886–893, IEEE (2005).

[10] Sivic, J. and Zisserman, A., "Video google: A text retrieval approach to object matching in videos," in [*International Conference on Computer Vision*], 1470 – 1477, IEEE (2003).

[11] Lowe, D., "Distinctive image features from scale-invariant keypoints," *International journal of computer vision* **60**(2), 91–110 (2004).

[12] Tombari, F., Mattoccia, S., and Di Stefano, L., "Template matching based on the l_p norm using sufficient conditions with incremental approximations," in [*Video and Signal Based Surveillance, 2006. AVSS'06. IEEE International Conference on*], IEEE (2006).

[13] Friedman, J. H., Bentley, J. L., and Finkel, R. A., "An algorithm for finding best matches in logarithmic expected time," *ACM Transactions on Mathematics Software* **3**(3), 209–226 (1977).

[14] Vedaldi, A. and Fulkerson, B., "Vlfeat: An open and portable library of computer vision algorithms," in [*Proceedings of the international conference on Multimedia*], 1469–1472, ACM (2010).

[15] Muja, M. and Lowe, D. G., "Fast approximate nearest neighbors with automatic algorithm configuration," in [*International Conference on Computer Vision Theory and Application VISSAPP'09)*], 331–340, INSTICC Press (2009).

[16] Porikli, F., "Integral histogram: A fast way to extract histograms in cartesian spaces," in [*Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*], **1**, 829–836, IEEE (2005).

[17] Crow, F., "Summed-area tables for texture mapping," *Computer Graphics* **18**(3), 207–212 (1984).

[18] Tombari, F., Di Stefano, L., Mattoccia, S., and Galanti, A., "Performance evaluation of robust matching measures," in [*Proc. 3rd International Conference on Computer Vision Theory and Applications (VISAPP 2008)*], (2008).

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 7/5/2018
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use