Shape and Appearance Context Modeling

Xiaogang Wang
AI Lab, M.I.T.
Cambridge, MA 02139
xgwang@csail.mit.edu

Gianfranco Doretto Thomas Sebastian Jens Rittscher Peter Tu
Visualization and Computer Vision Lab, GE Global Research
Niskayuna, NY 12309
{doretto,sebastia,rittsche,tu}@research.ge.com

Abstract

In this work we develop appearance models for computing the similarity between image regions containing deformable objects of a given class in realtime. We introduce the concept of shape and appearance context. The main idea is to model the spatial distribution of the appearance relative to each of the object parts. Estimating the model entails computing occurrence matrices. We introduce a generalization of the integral image and integral histogram frameworks, and prove that it can be used to dramatically speed up occurrence computation. We demonstrate the ability of this framework to recognize an individual walking across a network of cameras. Finally, we show that the proposed approach outperforms several other methods.

1. Introduction

A typical way of building appearance models is by first computing *local descriptors* of an image, and then by aggregating them using different strategies. *Bag-of-features* approaches [20, 12, 14, 23, 26, 4], represent images, or portions of images, by a distribution/collection of, possibly densely computed, local descriptors, vector-quantized according to a predefined *appearance dictionary* (made of *appearance labels*). These rather simple models perform remarkably well in both specific object (intra-category) and object category (inter-category) recognition tasks, and are robust to occlusions, illumination, and viewpoint variations.

We are interested in the intra-category recognition problem, where we have to describe and match the appearance of deformable objects such as people in video surveillance footage, where geometric deformations and photometric variations induce large changes in appearance. Unfortunately, under these challenging conditions a local descriptor matching approach performs poorly [6], mainly due to the failure of capturing higher-order information, such as the relative spatial distribution of the appearance labels. Some approaches address exactly this issue [11, 27, 21, 19], but they mainly focus on inter-category discrimination, as opposed to recognizing specific objects. We refer to those as *multi-layer* approaches.

In this work we propose a multi-layer appearance modeling framework for extracting highly distinctive descriptors of given image regions containing deformable objects of a known class. We are interested in realtime applications and computational complexity is one of our major concerns. In addition, the model should be robust to illumination, viewpoint changes, as well as object deformations. This is a tall order, and the challenge is to strike a balance between *distinctiveness*, *computational complexity*, and *invariance* of the model.

The first layer of the model densely computes a local description of the image[1]. More precisely, we propose to compute *histograms of oriented gradients*[2] (HOG) in the *Log-RGB* color space [5] (Section 4). The second layer aims at capturing the spatial relations between appearance labels. We explore two approaches, the *appearance context* (Section 5), and the *shape and appearance context* (Section 6). The former model encapsulates information similar to the co-occurrence of appearance labels. The latter model extends the former by using object parts explicitly to improve distinctiveness. Parts identification is done by a modified *shape context* [1] algorithm, which uses a *shape dictionary* learnt a priori. This effectively segments the image into regions that are loosely associated with specific object parts.

The proposed models entail computing several statistics over image subregions. We will first introduce *integral computations* (Section 2), a generalization of the integral image [25] and integral histogram [18] frameworks, and show how to perform fast computations of statistics (e.g. mean and covariance) of multidimensional vector-valued functions over (discrete) domains of arbitrary shape. Based on this framework, we present a fast algorithm to compute occurrence, and co-occurrence (Section 3), which enables realtime performance by providing a dramatic speed up when compared with the state-of-the-art approach [19]. The resulting occurrence matrix will be the descriptor of a given object.

[1] In challenging situations a dense representation has been found outperforming the sparse one also by other authors [6, 24].

[2] Different flavors of HOG's have proven to be successful in several settings [12, 2, 10].

We apply our appearance models to the challenging appearance-based *person reidentification*[3] problem [6], which is the ability to recognize an individual across a network of cameras based on appearance cues. We show that this framework can perform recognition of people appearing from three different viewpoints using a k-nearest neighbor approach, and achieve a significant performance increase over the state-of-the-art approach [6].

Related prior art. In [6] the authors consider a sparse local descriptor matching approach, and a dense approach where body parts are automatically extracted and compared. The dense approach outperforms the sparse one. In [9] the inter-camera brightness transfer functions is learnt to perform people track-linking across multiple cameras. Although it would be beneficial, our work does not use this information. Related work also includes vehicle reidentification [7], tracking [28], and category recognition [17, 13].

In [19] the authors use the local descriptors of [26], learnt to maximize intra-category invariance, and add another layer that captures the co-occurrence between appearance labels. They show an improved inter-category discrimination over [26]. Note that our appearance models are meant to maximize intra-category discrimination rather than invariance.

2. Integral computations

In this section we introduce the unified framework of the *integral computations*, show how it specializes to the *integral image* [25], and *integral histogram* [18], show his power in computing statistics over rectangular domains, and extend its applicability to general, non-simply connected rectangular domains.

Given a function $f(\mathbf{x}) : \mathbb{R}^k \to \mathbb{R}^m$, and a rectangular domain $D = [u_1, v_1] \times \cdots \times [u_k, v_k] \subset \mathbb{R}^k$, if there exists an antiderivative[4] $F(\mathbf{x}) : \mathbb{R}^k \to \mathbb{R}^m$, of $f(\mathbf{x})$, then

$$\int_D f(\mathbf{x})\, \mathrm{d}\mathbf{x} = \sum_{\boldsymbol{\nu} \in \mathbb{B}^k} (-1)^{\boldsymbol{\nu}^T \mathbf{1}} F(\nu_1 u_1 + \bar{\nu}_1 v_1, \cdots, \nu_k u_k + \bar{\nu}_k v_k) \;, \tag{1}$$

where $\boldsymbol{\nu} = (\nu_1, \cdots, \nu_k)^T$, $\boldsymbol{\nu}^T \mathbf{1} = \nu_1 + \cdots + \nu_k$, $\bar{\nu}_i = 1 - \nu_i$, and $\mathbb{B} = \{0, 1\}$. If $k = 1$, then $\int_D f(x)\, \mathrm{d}x = F(v_1) - F(u_1)$, which is the Fundamental Theorem of Calculus. If $k = 2$, then $\int_D f(\mathbf{x})\, \mathrm{d}\mathbf{x} = F(v_1, v_2) - F(v_1, u_2) - F(u_1, v_2) + F(u_1, u_2)$, and so on.

Equation (1) defines a class of operations that we call *integral computations*. They are very attractive for at least three reasons. First, in the discrete domain one specification of $F(\mathbf{x})$ can always be found, e.g.[5] $F(\mathbf{x}) = \sum_{\mathbf{u} \leq \mathbf{x}} f(\mathbf{u})$. Second, $F(\mathbf{x})$ can be computed from a single pass inspection[6] of $f(\mathbf{x})$, i.e. with computational cost $O(N^k)$, where N^k represents the dimension of the discrete domain where $f(\mathbf{x})$ is defined. Finally, we will see that Equation (1) enables the computation of statistics over the rectangular domain D in constant time $O(1)$, regardless of the size of D.

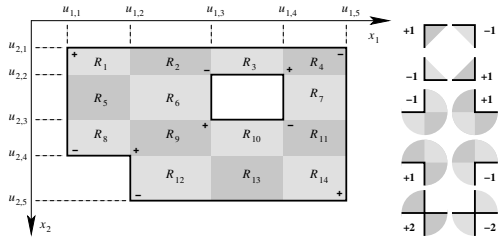

Figure 1. **Generalized rectangular domain and corner types.** Left: Example of a generalized rectangular domain D, partitioned into simple rectangular domains $\{R_i\}$. Right: Function $\alpha_D(\mathbf{x})$. It assumes values different then zero only if $\mathbf{x}$ is a corner of D. The specific value depends on the type of corner. For the planar case there are only 10 types of corner, depicted here along with the corresponding values of α_D.

Specific cases. If $k = 2$ and $m = 1$, and $f(\mathbf{x}) \doteq I(\mathbf{x})$, a grayscale image, then $F(\mathbf{x})$ in Computer Vision is now referred to as the *integral image* of $I(\mathbf{x})$ [25]. If $f(\mathbf{x}) \doteq e \circ q \circ I(\mathbf{x})$, where $q : \mathbb{R} \to \mathcal{A}$ is a quantization (labeling) function, with quantization levels $\mathcal{A} = \{a_1, \cdots, a_m\}$, and $e : \mathcal{A} \to \mathbb{N}^m$ is such that $a_i \mapsto e_i$, where e_i is the unit vector with only the i-th component different then 0, then $F(\mathbf{x})$ is the so called *integral histogram* of $I(\mathbf{x})$ with respect to q [18]. In general, one has the freedom to design $f(\mathbf{x})$ *ad libitum*, in order to take advantage of the properties of the integral computations.

Computing statistics. If $\mathbf{x}$ is a uniformly distributed random variable, and $E[\cdot|D]$ denotes the expectation where $\mathbf{x}$ is constrained to assume values in D, then one can write the expression of simple statistics, such as the mean[7] of $f(\mathbf{x})$ over D

$$E[f(\mathbf{x})|D] = \frac{1}{|D|} \int_D f(\mathbf{x})\, \mathrm{d}\mathbf{x} \;, \tag{2}$$

or the covariance of $f(\mathbf{x})$ over D

$$E[(f(\mathbf{x}) - E[f(\mathbf{x})|D])(f(\mathbf{x}) - E[f(\mathbf{x})|D])^T|D] = \frac{1}{|D|} \int_D g(\mathbf{x})\, \mathrm{d}\mathbf{x} - \frac{1}{|D|^2} \int_D f(\mathbf{x})\, \mathrm{d}\mathbf{x} \int_D f(\mathbf{x})^T\, \mathrm{d}\mathbf{x} \;, \tag{3}$$

where $g(\mathbf{x}) : \mathbb{R}^k \to \mathbb{R}^{m \times m}$ is such that $\mathbf{x} \mapsto f(\mathbf{x}) f(\mathbf{x})^T$. Similarly, higher-order moments can be written in this manner. What those expressions share is the fact that the integral operation can be substituted with the result of Equation (1). Therefore, given $F(\mathbf{x})$, their computation cost is $O(1)$, independent from the size of D.

[3] If we were to apply the model to the people track-linking problem, it would be more appropriate to talk about *person reacquisition*.

[4] If the Fubini's theorem for indefinite integrals holds, then $F(\mathbf{x})$ exists.

[5] Here $\mathbf{u} \leq \mathbf{x}$ is intended as $u_1 \leq x_1, \cdots, u_k \leq x_k$.

[6] The single pass inspection of $f(\mathbf{x})$ is the k-dimensional extension of the 2-dimensional version described in [25, 18].

[7] The operation $|\cdot|$ applied to a domain or a set indicates the area or the cardinality, respectively.

The expressions (2) and (3) assume very different meanings according to the choice of $f(\mathbf{x})$. For instance, for the integral image they represent mean and covariance of the pixel intensities over the region D. On the other hand, for the integral histogram, (2) is the histogram of the pixels of the region D, according to the quantization q. Recently, [22] used (3) as a region descriptor to perform object detection and texture classification, where $f(\mathbf{x})$ was the output of a bank of filters applied to the input image $I(\mathbf{x})$.

Domain generalization. To the best of our knowledge the integral computations have been used only when the region D is rectangular. On the other hand, Equation (1) can be generalized to domains defined as follows (see Figure 1).

Definition 1. *$D \subset \mathbb{R}^k$ is a* generalized rectangular domain *if his boundary ∂D is made of a collection of portions of a finite number of hyperplanes perpendicular to one of the axes of $\mathbb{R}^k$.*

If $\nabla \cdot D$ indicates the set of corners of a generalized rectangular domain D, then the following result holds.

Theorem 1.

$$\int_D f(\mathbf{x})\,\mathrm{d}\mathbf{x} = \sum_{\mathbf{x}\in\nabla\cdot D} \alpha_D(\mathbf{x}) F(\mathbf{x}) \;, \tag{4}$$

where $\alpha_D : \mathbb{R}^k \to \mathbb{Z}$, is a map that depends on k. For $k = 2$ it is such that $\alpha_D(\mathbf{x}) \in \{0, \pm 1, \pm 2\}$, according to which of the 10 types of corners depicted in Figure 1, $\mathbf{x}$ belongs to.

Theorem 1 (proved in the Appendix), says that if D is a generalized rectangular domain, one can still compute the integral of $f(\mathbf{x})$ over D in constant time $O(1)$. This is done by summing up the values of $F(\mathbf{x})$, computed at the corners $\mathbf{x} \in \nabla \cdot D$, and multiplied by $\alpha_D(\mathbf{x})$, which depends on the type of the corner. For the planar case the types of corners are depicted in Figure 1. Therefore, *given any discrete domain D, by simply inspecting the corners to evaluate α_D, one can compute statistics over D in constant time $O(1)$.* This simple and yet powerful result enables designing fast, more flexible and sophisticated region based features, like the one of the following Section.

3. Occurrence computations

In this section we define the concept of *occurrence*, propose a new algorithm based on Theorem 1 to compute it, analyze the computational complexity of the algorithm, and compare it to the state-of-the-art approach.

Let $S : \Lambda \to \mathcal{S}$, and $A : \Lambda \to \mathcal{A}$ be two functions defined on a discrete domain Λ of dimensions $M \times N$, and assuming values in the label sets $\mathcal{S} = \{s_1, \cdots, s_n\}$ and $\mathcal{A} = \{a_1, \cdots, a_m\}$ respectively. Also, let $\mathcal{P} = \{p_1, \cdots, p_l\}$ be a partition such that $\bigcup_i p_i$ represents the plane, and $p_i \cap p_j = \emptyset$, if $i \neq j$ (see Figure 2 for an example). Given $p \in \mathcal{P}$, and a point on the plane $\mathbf{x}$, we define $p(\mathbf{x}) \doteq \{\mathbf{x} + \mathbf{y} | \mathbf{y} \in p\}$, and we indicate with $h(a, p(\mathbf{x})) \doteq P[A(\mathbf{y}) = a | \mathbf{y} \in p(\mathbf{x})]$ the probability distribution[8] of the labels of A over the region $p(\mathbf{x})$. In other words, for a given A, if we randomly select a point $\mathbf{y} \in p(\mathbf{x})$, the probability that the label at that point will be a is given by $h(a, p(\mathbf{x}))$. If we now set $D_s \doteq \{\mathbf{x} | S(\mathbf{x}) = s\}$, $s \in \mathcal{S}$, we define the *occurrence function* as follows (see Figure 2).

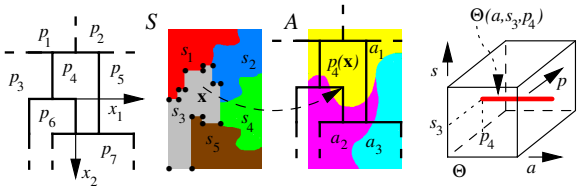

Figure 2. **Partition and occurrence definition.** From left to right. Example of a generic partition $\mathcal{P}$. Example of a function S, and of a function A. Representation of Θ. If $h(a, p_4(\mathbf{x}))$ is the normalized count of the labels of A in $p_4(\mathbf{x})$ (the partition element p_4 translated at $\mathbf{x}$), then by averaging $h(a, p_4(\mathbf{x}))$ over all $\mathbf{x} \in \{\mathbf{y} | S(\mathbf{y}) = s_3\} \doteq D_{s_3}$, we obtain $\Theta(a, s_3, p_4)$ (red line). The dots over S highlight the corner points $\nabla \cdot D_{s_3}$.

Definition 2. *The* occurrence function *$\Theta : \mathcal{A} \times \mathcal{S} \times \mathcal{P} \to \mathbb{R}_+$, is such that the point (a, s, p) maps to*

$$\Theta(a, s, p) = E[h(a, p(\mathbf{x})) | D_s] \;. \tag{5}$$

The meaning of the occurrence function is the following: *Given S and A, if we randomly select a point $\mathbf{x} \in D_s$, the probability distribution of the labels $\mathcal{A}$ over the region $p(\mathbf{x})$ of A is given by $\Theta(\cdot, s, p)$.* One special case is when $\mathcal{S} = \mathcal{A}$, and $S(\mathbf{x}) = A(\mathbf{x})$, and Θ is typically referred to as *co-occurrence*. When Θ is intended as a collection of values corresponding to all the points of the domain $\mathcal{A} \times \mathcal{S} \times \mathcal{P}$, we refer to it as the *occurrence*, or *co-occurrence matrix*.

3.1. Fast occurrence computation

In this section we present a novel result that allows a fast computation of the occurrence function. The derivation is based on the fact that the occurrence is computed over a discrete domain Λ, where every possible sub-domain is a (discrete) generalized rectangular domain, and all the results of Section 2 can be applied.

Theorem 2. *The occurrence function* (5) *is equal to*

$$\Theta(a, s, p) = |D_s|^{-1} |p|^{-1} \sum_{\mathbf{x}\in\nabla\cdot D_s, \mathbf{y}\in\nabla\cdot p} \alpha_{D_s}(\mathbf{x}) \alpha_p(\mathbf{y}) G(a, \mathbf{x}+\mathbf{y}) \;, \tag{6}$$

where

$$G(\cdot, \mathbf{x}) = \int_{-\infty}^{\mathbf{x}} \int_{-\infty}^{\mathbf{u}} e \circ A(\mathbf{v})\,\mathrm{d}\mathbf{v}\,\mathrm{d}\mathbf{u} \;, \tag{7}$$

and[9] $e : \mathcal{A} \to \mathbb{N}^m$ is such that the inner integral is the integral histogram of A.

Theorem 2 (proved in the Appendix) leads to the Algorithm 1. We point out that even though the occurrence has been introduced for S and A defined on a two-dimensional

[8] $P[\cdot]$ indicates a probability measure.

[9] Note that $a \in \mathcal{A}$ is intended to index one of the elements of the m-dimensional vector $G(\cdot, \mathbf{x})$.

Algorithm 1: Fast occurrence computation

Data: Functions A and S
Result: Occurrence matrix Θ

1 **begin**
2 Use (7) to compute G from a single pass inspection of A
// Compute $|D_s|$ α_{Ds} and $\nabla \cdot D_s$
3 **foreach** $\mathbf{x} \in \Lambda$ **do**
4 $|D_{S(\mathbf{x})}| \longleftarrow |D_{S(\mathbf{x})}| + 1$
5 **if** IsCorner $(\mathbf{x})$ **then**
6 Set $\alpha_{D_{S(\mathbf{x})}}(\mathbf{x})$
7 $\nabla \cdot D_{S(\mathbf{x})} \longleftarrow \nabla \cdot D_{S(\mathbf{x})} \bigcup \{\mathbf{x}\}$
// Use (6) to compute Θ
// $|p|$ α_p and $\nabla \cdot p$ known a priori
8 **foreach** $s \in \mathcal{S}$ **do**
9 **foreach** $p \in \mathcal{P}$ **do**
10 **foreach** $\mathbf{x} \in \nabla \cdot D_s$ **do**
11 **foreach** $\mathbf{y} \in \nabla \cdot p$ **do**
12 $\Theta(\cdot, s, p) \longleftarrow \Theta(\cdot, s, p) +$
13 $\alpha_{D_s}(\mathbf{x})\alpha_p(\mathbf{y})G(\cdot, \mathbf{x} + \mathbf{y})$
14 $\Theta(\cdot, s, p) \longleftarrow |D_s|^{-1}|p|^{-1}\Theta(\cdot, s, p)$
15 **return** Θ
16 **end**

domain, the definition can be generalized to any dimension, and Theorem 2 still holds.

Complexity analysis. Given S and A, the na¨ive approach to compute Θ costs $O(N^4)$ in time (we assume $M \sim N$), which is too much for real-time applications, even if N is not very large. In [8] a dynamic programming approach reduces the cost to $O(N^3)$. In [19] a particular partition $\mathcal{P}$ where every $p \in \mathcal{P}$ is a square ring defined by $|\nabla \cdot p| = 8$ corners enables a computation cost[10] of $O(N^2 l|\nabla \cdot p|) = O(N^2 C_{\mathcal{P}})$, where $C_{\mathcal{P}} \doteq l|\nabla \cdot p|$ represents the total number of corners of $\mathcal{P}$.

We now calculate the computational cost of Algorithm 1. Line 2 can be evaluated by a single pass inspection of A, and has the same computational cost of an integral histogram, i.e. $O(N^2)$. Line 3-7 is another single pass inspection of S with cost $O(N^2)$. Line 12 costs $O(1)$. Line 11 is an average multiplying factor of $C_{\mathcal{P}}/l$, where $C_{\mathcal{P}} \doteq \sum_i |\nabla \cdot p_i|$. Line 10 is an average multiplying factor of C_S/n, where $C_S \doteq \sum_i |\nabla \cdot D_{s_i}|$. Line 8 and 9 are multiplying factors of n and l respectively. Therefore, the cost of 8-13 is $O(C_S C_{\mathcal{P}})$. Finally, the total cost of Algorithm 1 is $O(N^2 + C_S C_{\mathcal{P}})$. In practice we have $C_s C_{\mathcal{P}} \sim N^2$. Therefore, Algorithm 1 has an effective cost of $O(N^2)$, which is $C_{\mathcal{P}}$ (the number of corner points of the partition $\mathcal{P}$) times faster then the state-of-the-art [19]. It is interesting to note that Algorithm 1 is only marginally sensitive to the choice of the partition $\mathcal{P}$, which, as opposed to [19], here is allowed to be arbitrary.

[10] In [19] $|\nabla \cdot p|$ is part of the hidden constants. Here we make the dependency explicit to better compare their approach with ours.

4. Bag-of-features modeling

In this section, as well as in Sections 5 and 6, we are interested in designing a highly distinctive *descriptor* for an image I, belonging to $\mathbb{I}$, the space of all the images defined on a discrete domain Λ of dimensions $M \times N$ pixels. To this end we process the image by applying an operator $\Phi : \mathbb{I} \times \Lambda \to \mathbb{R}^r$, such that $(I, \mathbf{x})$ is mapped to a *local descriptor* $\varphi(\mathbf{x}) \doteq \Phi(I, \mathbf{x})$. The operator Φ could be a bank of linear filters, as well as any other non-linear operation.

Once φ is available, the descriptor is computed in two steps. The first one performs a vector quantization of φ, according to a quantization function $q : \mathbb{R}^r \to \mathcal{A}$, with quantization levels $\mathcal{A} = \{a_1, \cdots, a_m\}$. This produces the *appearance labeled image* $A(\mathbf{x}) \doteq q \circ \varphi(\mathbf{x})$ (see Figure 3 for an example). We refer to $\mathcal{A}$ as the *appearance dictionary*, made of *appearance labels* learnt a priori. The second step computes the histogram of the labels $h : \mathcal{A} \to [0, 1]$, such that the label a maps to

$$h(a) \doteq P[A(\mathbf{x}) = a] \,. \tag{8}$$

The image descriptor is defined to be the histogram h.

HOG Log-RGB operator. In Section 7 we experiment with several operators Φ, such as different color spaces and filter banks, and test their matching performance with the descriptor (8). The best performer operates in the RGB color space, and is such that $\varphi(\mathbf{x}) \doteq (\text{HOG}(\nabla \log(I_R), \mathbf{x}); \text{HOG}(\nabla \log(I_G), \mathbf{x}); \text{HOG}(\nabla \log(I_B), \mathbf{x}))$, where I_R, I_G, I_B, are the R, G, and B channels of I respectively. The operator $\text{HOG}(\cdot, \mathbf{x})$ computes the ℓ bins *histogram of oriented gradients* of the argument, on a region of $w \times w$ pixels around $\mathbf{x}$. The gradient of the Log-RGB space has an effect similar to the homomorphic filtering, and makes the descriptor robust to illumination changes.

5. Appearance context modeling

The main drawback of the bag-of-features model is that images of different objects that share the same appearance label distribution $h(a)$, share also the same descriptor, annihilating the distinctiveness that we are seeking. This is due to the fact that (8) does not incorporate any description of how the object appearance is distributed in space. On the other hand, this information could be captured by computing the spatial co-occurrence between appearance labels.

Appearance context. More precisely, the co-occurrence matrix Θ, computed on the appearance labeled image $A(\mathbf{x})$ with the plane partition $\mathcal{P}$ depicted in Figure 4, will be referred to as the *appearance context* descriptor of I, which is an $m \times m \times l$ matrix.

Appearance context vs. bag-of-features. Note that the information carried by the descriptor (8) is included in the appearance context descriptor. In fact, by using (6) one can show that Θ reduces to (8), in particular, for every $b \in \mathcal{A}$ we have $h(a) = 1/|\Lambda| \sum_{p \in \mathcal{P}} |p| \Theta(a, b, p)$.

Figure 3. **Data set and shape and appearance labeled images.** From left to right. Two samples from the data set of 143 different individuals, recorded from 3 different viewpoints, with corresponding shape labeled image S, and appearance labeled image A. Note that the decomposition into parts performed by S tries to compensate the misalignment induced by pose and viewpoint changes, as well as person bounding box imprecisions.

Plane partition. $\mathcal{P}$ is made of $p_1, \cdots, p_l$, L-shaped regions. Every quadrant is covered by $l/4$ regions. Every L-shape is $4N/l$ and $4M/l$ thick along the x_1 and x_2 directions, respectively. Therefore, the set $\{p_i\}$ can be partitioned into groups of 4 elements forming $l/4$ corresponding concentric square rings.

Invariance. It is known that the co-occurrence matrix is translation invariant, rotation invariant if the $\{p_i\}$ are concentric circular rings, and shows robustness to affine and pose changes [8, 19]. It is not invariant with respect to the size of I. In order to have this property, before computing the appearance context we will normalize the size of I. The morphology of the partition $\mathcal{P}$ makes the appearance context non-rotation invariant. In our application this is a desired property as it increases distinctiveness. In fact, lack of rotation invariance allows us to distinguish between a person wearing a white T-shirt and black pants vs. a person wearing a black T-shirt and white pants.

6. Shape and appearance context modeling

Let's now consider a toy example to highlight an important weakness of the appearance context descriptor, and examine the appearance labeled image of a person depicted in Figure 4 (right). We indicate with D_f and D_h the face and hand regions respectively, and notice that these are the only regions that have been assigned the label $a_1 \in \mathcal{A}$. Using the notation introduced in Section 3, we also have $D_{a_1} = D_f \cup D_h$. The region of the torso, arm, and hair has been assigned the label $a_2 \in \mathcal{A}$. For a_1 and a_2 the appearance context can be rewritten as $\Theta(a_2, a_1, p) = \frac{D_f}{D_{a_1}} E[h(a_2, p)|D_f] + \frac{D_h}{D_{a_1}} E[h(a_2, p)|D_h] \doteq h_f(p) + h_h(p)$. Note that $h_f(p)$, $(h_h(p))$, is the occurrence of a_2 at a given distance and orientation from D_f, (D_h), defined by p. Figure 4 sketches $h_f(p)$, $h_h(p)$, and their sum. $h_f(p)$ highlights that a_2 is mostly present in the blue and yellow quadrants of the partition $\mathcal{P}$. $h_h(p)$ highlights that a_2 is mostly present in the red and green quadrants of the partition $\mathcal{P}$. $h_f(p) + h_h(p)$ shows that a_2 is uniformly distributed over all the quadrants.

We point out that averaging h_f and h_h has caused a loss of information, and therefore descriptive power. From an information theoretic point of view, two unimodal distributions have been merged to obtain an almost uniform distribution, with a significant increase of entropy. To prevent such a situation, this toy example suggests that *if we could identify the parts of a given object, it would be more descriptive to capture the spatial occurrence of the appearance labels with respect to each of the parts rather then to each of the appearance labels*.

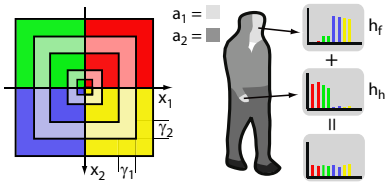

Figure 4. **L-shaped partition and appearance context averaging effect.** From left to right. Sketch of the L-shaped plane partition used in Section 5 ($\gamma_1 = 4N/l$, $\gamma_2 = 4M/l$), and 6 ($\gamma_1 = 4Nd/t$, $\gamma_2 = 4Md/t$). Illustration of the averaging effect when appearance context descriptors are pooled from the entire object region.

Shape and appearance context. Given I containing an object of a given class, let A be its appearance labeled image, and let S (defined over Λ) be its *shape labeled image*, where pixel labels are meant to identify regions of I occupied by specific parts of the object (see Figure 3 for examples). We define the *shape and appearance context* descriptor of I, the occurrence matrix Θ, computed over S and A, which is an $m \times n \times l$ matrix. Similarly to the appearance context descriptor, the information carried by the descriptor (8) is included in the shape and appearance context descriptor.

Shape labeled image. We propose to compute the shape labeled image with a procedure that is inspired by the idea of shape context [1, 16]. Given the image $I \in \mathbb{I}$, we process it according to an operator $\Omega : \mathbb{I} \times \Lambda \rightarrow \mathbb{R}^d$, such that I is mapped to $\omega(\mathbf{x}) \doteq \Omega(I, \mathbf{x})$. In Section 7 we consider different operators Ω. A fast and reliable choice is such that $\omega(\mathbf{x}) = \text{HOG}(\nabla I_L, \mathbf{x})$, where I_L is the L channel of the Lab color space of I. From ω, at every pixel we compute a form of *shape context descriptor* ψ, defined as $\psi(\mathbf{x}) \doteq (E[\omega|p_1(\mathbf{x})]; \cdots; E[\omega|p_{t/d}(\mathbf{x})]) \in \mathbb{R}^t$. Here $\{p_1, \cdots, p_{t/d}\}$ indicates a plane partition of the same kind used for the appearance context descriptor, but with t/d L-shaped regions rather then l (see Figure 4). Once ψ is available, we vector quantize it according to a quantization (labeling) function $q : \mathbb{R}^t \rightarrow \mathcal{S}$, with quantization levels defined by a *shape dictionary* $\mathcal{S} = \{s_1, \cdots, s_n\}$, made of *shape labels* learnt a priori. This produces the *shape labeled image* $S(\mathbf{x}) \doteq q \circ \psi(\mathbf{x})$.

Neighboring pixels have similar shape context descriptors. Therefore, the quantization process produces a "piecewise-like" segmentation of the image into regions $D_{s_i} = \{\mathbf{x}|S(\mathbf{x}) = s_i\}$, which are meant to always identify the same region/part of the object of interest (see Figure 3).

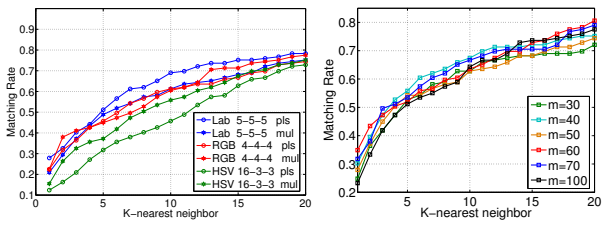

Figure 5. **Bag-of-features: Color spaces and linear filters.** Matching comparison using different color spaces and quantization schemes (left). Matching comparison of a linear filter bank against appearance dictionary size (right).

As a welcome side effect, the occurrence computation takes great advantage of this segmentation, as C_S ends up being very small.

Invariance. The shape and appearance context enjoys the same invariance properties of the appearance context. However, it should be noted that translation invariance is lost if the same parts of an object are labeled differently in different images, e.g. an arm labeled as a leg. This means that using a fixed mask to identify object parts under pose and viewpoint changes, as well as object bounding box imprecisions, significantly decreases the performance of the descriptor. The decomposition into parts with the shape labeled image tries to compensate exactly those variations (see Figure 3).

7. Experiments

Data set. The data set is composed by the one in [6], containing 44 different individuals recorded from three different non-overlapping camera views (see Figure 3), to which we added new images of 99 individuals, recorded from three very similar viewpoints. Every person in each view is represented by two to four images, of about 80×170 pixels in size. In a video-surveillance system such images would be cropped out of the full frames by a person detector, or tracker module.

Matching. As in [6], for every pair person/camera view, we compute the k-nearest people in the other views. Since every pair is represented by a number of images, the distance between two pairs is the minimum distance between two images (that do not belong to the same pair), measured according to a given distance between descriptors. Identification results are reported in terms of cumulative match characteristic (CMC) curves [15], which tell the rate at which the correct match is within the k-nearest neighbors, with k that varies from 1 to 20.

Training and testing. The appearance and shape dictionaries are learnt using simple k-means clustering after computing φ and ψ on the training set, respectively. For training, we used the images of 30% of the individuals randomly selected. The rest of the data set was used for testing.

Distances. Appearance labels are learned and assigned using L_1 norm. Shape labels are learned and assigned using

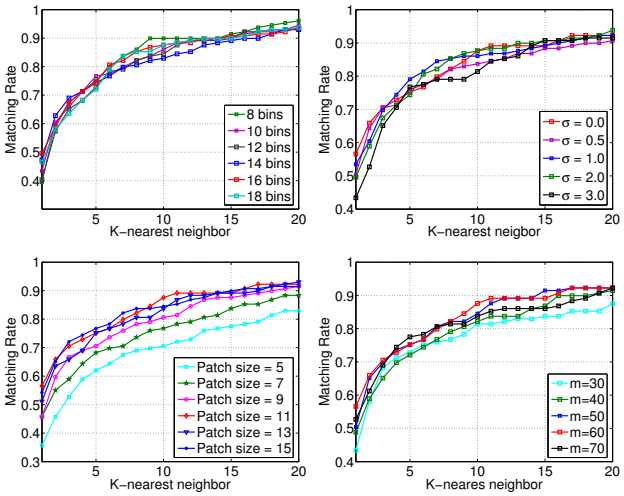

Figure 6. **Bag-of-features: HOG Log-RGB.** Matching comparison by varying one of the operator parameters at a time, such as the quantizing orientations ℓ (top-left), the image smoothing σ (top-right), the patch size w (bottom-left), and the appearance dictionary dimensionality m (bottom-right).

the χ^2 distance[11]. Descriptor matching using (8) is done with the intersection distance (used also in [6]). Descriptor matching using appearance, and shape and appearance context is done with L_1 norm.

Bag-of-features: Color spaces and linear filters. Figure 5 (left) shows results where Φ is a color transformation and quantization. We tried the Lab, RGB, and HSV color spaces, quantized along the three axes according to the number of bins reported in the figure. The suffixes "pls" and "mul" indicate whether channel quantization is performed independently, or jointly. The Lab and the RGB color spaces seem to perform better. In Figure 5 (right), Φ is the linear filter (LF) bank used in [26]. It shows that linear filtering improves versus simple color quantization, and that there is an optimal dimensionality for $\mathcal{A}$, say around $m = 60$.

Bag-of-features: HOG Log-RGB. Figure 6 summarizes our search for optimal settings for computing HOG's. We optimize four parameters (one at a time): the number of quantizing orientations ℓ, the prior Gaussian smoothing of the image σ, the patch size w, and the appearance dictionary dimensionality m. We found a good operating point[12] with $\ell = 16$, $\sigma = 0$, $w = 11$, and $m = 60$.

Shape descriptors. For computing ω, the HOG uses $d = 8$ quantizing orientations, a patch size $w = 11$, and a shape dictionary of dimension $n = 18$. Note that there might be other ways to design the operator[13] Ω.

[11] Note that ψ is a concatenation of histograms.

[12] We tested the HOG with other color spaces. The L channel and the Log-RGB color space are the best options. They double the performance with respect to the linear filter bank (CMC curves included in [3]).

[13] We experimented a Canny edge detector, where ψ does an edge pixels count on the partition regions. This approach compares well with the HOG,

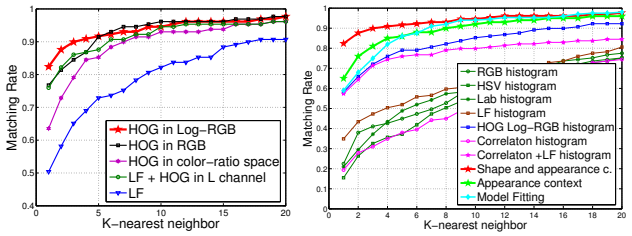

Figure 7. **Shape and appearance context.** Left: Matching comparison between several approaches for computing appearance labels. Approaches using HOG outperform the ones that do not. Right: Matching comparison summary of several appearance models. The shape and appearance context outperforms all others, including the state-of-the-art model fitting [6]. Approaches that capture spatial relationships outperform those that do not.

Shape and appearance context: Appearance comparison. Figure 7 (left) shows the results corresponding to different choices of Φ. We tested the HOG in three different color spaces. We also tested the linear filter bank (LF) used in [26], and a combination of the LF and the HOG of the L channel of the Lab color space. It is noticeable how the configurations that include the HOG significantly outperform the LF bank. The HOG in the Log-RGB color space is the best performer.

Comparison summary. Figure 7 (right) compares the matching performance of several approaches. Besides appearance and shape and appearance context, we have included simple bag-of-features approaches where Φ is a color transformation (to spaces such as RGB, HSV, Lab), but also more sophisticated ones, where Φ is a LF bank [26], or the HOG in the Log-RGB space. Results using the histogram of correlatons,[14] and his concatenation with the histogram (8), where Φ is the LF bank [19] are also included.

The results indicate that the shape and appearance context is the best performing algorithm. Further, approaches that capture the spatial relationships among appearance labels significantly outperform the approaches that do not, such as bag-of-features. Finally, the comparison with the model fitting algorithm [6], shows that the matching rate of the first neighbor is 59%, whereas for the shape and appearance context it is 82%, which is a significant improvement (see Figure 7).

8. Conclusions

We propose the shape and appearance context, an appearance model that captures the spatial relationships among appearance labels. This outperforms other methods, especially in very challenging situations, such as the appearance-based person reidentification problem. In this scenario, our extensive testing to design several operators has led to a method that achieves a matching rate of 82% on a large data set.

The new algorithm to compute occurrence, which cuts the computational complexity down to $O(N^2)$, enables the computation of the shape and appearance context in real-time,[15] which is not possible using existing methods. The computation of the occurrence is based on the proposed integral computations framework, which generalizes the ideas of integral image and integral histogram to multidimensional vector-valued functions, and allows computing statistics over discrete domains of arbitrary shape.

The appearance variation of our testing data set shows that our approach is robust to a great deal of viewpoint, illumination, and pose changes, not to mention background contamination. Future investigation will include methods to remove this contamination, and testing the approach in other scenarios, e.g. multiple hypothesis tracking.

Appendix

In this Appendix we first introduce some notation and then give the proofs of Theorem1, and 2. The variable $\boldsymbol{\nu}$ can be interpreted as partitioning $\mathbb{R}^k$ into his 2^k open orthants $O_{\boldsymbol{\nu}} \doteq \{\mathbf{x} \in \mathbb{R}^k | x_i > 0 \text{ if } \nu_i = 1, \text{ or } x_i < 0 \text{ if } \nu_i = 0, i = 1, \cdots, k\}$. We introduce the notation $O_{\boldsymbol{\nu}}(\mathbf{x}) \doteq \{\mathbf{x}+\mathbf{y} | \mathbf{y} \in O_{\boldsymbol{\nu}}\}$. We also introduce a function $\beta_{\mathbf{x}}(\boldsymbol{\nu}) : \mathbb{B}^k \rightarrow \mathbb{B}$, such that: i) $\beta_{\mathbf{x}}(\boldsymbol{\nu}) = 1$ if $\mathbf{x}$ is an adherent point[16] for the open set $D \cap O_{\boldsymbol{\nu}}(\mathbf{x})$; ii) $\beta_{\mathbf{x}}(\boldsymbol{\nu}) = 0$ otherwise. It is trivial to prove that: I) $\mathbf{x} \in D \setminus \partial D \iff \beta_{\mathbf{x}}(\boldsymbol{\nu}) = 1 \; \forall \boldsymbol{\nu}$; II) $\mathbf{x} \notin D \iff \beta_{\mathbf{x}}(\boldsymbol{\nu}) = 0 \; \forall \boldsymbol{\nu}$. Finally, if $\boldsymbol{\nu}_j$ represents j out of the k components of $\boldsymbol{\nu}$, and if $\boldsymbol{\nu}_{k-j}$ represents the remaining $k-j$ components, then we define *edges* and *corners* of the boundary ∂D as follows: A point $\mathbf{x} \in \partial D$ lays on an *edge* if there exist j components of $\boldsymbol{\nu}$, with $1 \leq j \leq k-1$, such that $\beta_{\mathbf{x}}(\boldsymbol{\nu})$ does not depend on $\boldsymbol{\nu}_{k-j}$, i.e. $\beta_{\mathbf{x}}(\boldsymbol{\nu}) = \beta_{\mathbf{x}}(\boldsymbol{\nu}_j)$, $\forall \boldsymbol{\nu}$. If $\mathbf{x}$ does not lay on an edge, it is a *corner*. We indicate the set of corners with $\nabla \cdot D$.

Proof of Theorem 1. Let $\{u_{i,1}, u_{i,2}, \cdots | u_{i,j} \in \mathbb{R}, u_{i,j} < u_{i,j+1}, i = 1, \cdots, k\}$, be the set of points along $\{x_i\}$, such that D is made of portions of hyperplanes passing through these points. Figure 1 illustrates an example for $k = 2$. The intersection of the hyperplanes with D defines a partition $D \doteq \bigcup_i R_i$ into rectangular regions $\{R_i\}$, which allows us to write $\int_D f(\mathbf{x}) \mathrm{d}\,\mathbf{x} = \sum_i \int_{R_i} f(\mathbf{x}) \,\mathrm{d}\mathbf{x}$, and apply Equation (1) to each term of the summation. By rearranging the terms, and using the function $\beta_{\mathbf{x}}(\boldsymbol{\nu})$, the integral can be rewritten as $\sum_{\mathbf{x} \in \mathcal{D}} \alpha_D(\mathbf{x}) F(\mathbf{x})$, where $\mathcal{D}$ is the set of all the corner points of the regions $\{R_i\}$ (note that $\nabla \cdot D \subseteq \mathcal{D}$),

with a slight advantage to the latter (CMC curves included in [3]).

[14] We have used our own implementation of the algorithm [19]. Learning was done with k-means only. We warn the reader that their approach was originally designed to solve the inter-category recognition problem, and here we have tested it outside his natural domain.

[15] Our C++ implementation of the algorithm can run at 10fps on an image region of 250×100 pixels.

[16] A point $\mathbf{x}$ is an adherent point for an open set B, if every open set containing $\mathbf{x}$ contains at least one point of B. A point $\mathbf{x}$ is an adherent point for B if and only if $\mathbf{x}$ is in the closure of B.

and $\alpha_D(\mathbf{x}) \doteq \sum_{\boldsymbol{\nu}} (-1)^{\boldsymbol{\nu}^T \mathbf{1}} \beta_{\mathbf{x}}(\boldsymbol{\nu})$. Now we recall that if $\mathbf{x} \in \mathcal{D} \setminus \partial D$, then $\beta_{\mathbf{x}}(\boldsymbol{\nu}) = 1$, and this implies $\alpha_D(\mathbf{x}) = 0$. On the other hand, if $\mathbf{x}$ is on an edge, then we can write $\alpha_D(\mathbf{x}) = \sum_{\boldsymbol{\nu}_j} (-1)^{\boldsymbol{\nu}_j^T \mathbf{1}} \beta_{\mathbf{x}}(\boldsymbol{\nu}_j) \sum_{\boldsymbol{\nu}_{k-j}} (-1)^{\boldsymbol{\nu}_{k-j}^T \mathbf{1}} = 0$, and Equation (4) is valid. When $\mathbf{x}$ is a corner described by $\beta_{\mathbf{x}}$, one should proceed with a direct computation of $\alpha_D(\mathbf{x})$. For $k = 2$, $\alpha_D(\mathbf{x})$ is different then zero only for the 10 cases depicted in Figure 1, in which it assumes the values indicated. □

Proof of Theorem 2. According to (5), Θ can be computed by using Equation (2), and subsequently applying Theorem 1, giving

$$\Theta(a, s, p) = |D_s|^{-1} \sum_{\mathbf{x} \in \nabla \cdot D_s} \alpha_{D_s}(\mathbf{x}) H(a, p(\mathbf{x})) \ , \qquad (9)$$

where $$H(a, p(\mathbf{x})) = \int_{-\infty}^{\mathbf{x}} h(a, p(\mathbf{u}))\, \mathrm{d}\mathbf{u} \ . \qquad (10)$$

Now we note that $h(a, p(\mathbf{u}))$ can be computed through the integral histogram of A, $F(a, \mathbf{z}) = \int_{-\infty}^{\mathbf{z}} e \circ A(\mathbf{v})\, \mathrm{d}\mathbf{v}$, and by applying Theorem 1, resulting in $h(a, p(\mathbf{u})) = |p(\mathbf{u})|^{-1} \sum_{\mathbf{z} \in \nabla \cdot p(\mathbf{u})} \alpha_{p(\mathbf{u})}(\mathbf{z}) F(a, \mathbf{z})$. This equation, combined with Equations (10) gives

$$H(a, p(\mathbf{x})) = \int_{-\infty}^{\mathbf{x}} |p(\mathbf{u})|^{-1} \sum_{\mathbf{z} \in \nabla \cdot p(\mathbf{u})} \alpha_{p(\mathbf{u})}(\mathbf{z}) F(a, \mathbf{z})\, \mathrm{d}\mathbf{u} \ . \qquad (11)$$

From the definition of $p(\mathbf{u})$, it follows that $|p(\mathbf{u})| = |p|$, and $\nabla \cdot p(\mathbf{u}) = \{\mathbf{u} + \mathbf{y} | \mathbf{y} \in \nabla \cdot p\}$, and also that $\alpha_{p(\mathbf{u})}(\mathbf{u} + \mathbf{y}) = \alpha_p(\mathbf{y})$. Therefore, after the change of variable $\mathbf{z} = \mathbf{u} + \mathbf{y}$, in Equation (11) it is possible to switch the order between the integral and the summation, yielding $H(a, p(\mathbf{x})) = |p|^{-1} \sum_{\mathbf{y} \in \nabla \cdot p} \alpha_p(\mathbf{y}) \int_{-\infty}^{\mathbf{x}} F(a, \mathbf{u} + \mathbf{y})\, \mathrm{d}\mathbf{u}$. Substituting this expression in (9), and taking into account the expression of $F(a, \mathbf{z})$, proves Equations (6) and (7). □

Acknowledgements

This report was prepared by GE GRC as an account of work sponsored by Lockheed Martin Corporation. Information contained in this report constitutes technical information which is the property of Lockheed Martin Corporation. Neither GE nor Lockheed Martin Corporation, nor any person acting on behalf of either; a. Makes any warranty or representation, expressed or implied, with respect to the use of any information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or b. Assume any liabilities with respect to the use of, or for damages resulting from the use of, any information, apparatus, method or process disclosed in this report.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE TPAMI*, 24:509–522, 2002.

[2] N. Dalai and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893, June 20–25, 2005.

[3] G. Doretto, X. Wang, T. B. Sebastian, J. Rittscher, and P. Tu. Shape and appearance context modeling: A fast framework for matching the appearance of people. Technical Report 2007GRC594, GE Global Research, Niskayuna, NY, USA, 2007.

[4] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *CVPR*, volume 2, pages 524–531, June 20–25, 2005.

[5] B. V. Funt and G. D. Finlayson. Color constant color indexing. *IEEE TPAMI*, 17:522–529, 1995.

[6] N. Gheissari, T. B. Sebastian, P. H. Tu, J. Rittscher, and R. Hartley. Person reidentification using spatiotemporal appearance. In *CVPR*, volume 2, pages 1528–1535, 2006.

[7] Y. Guo, S. Hsu, Y. Shan, H. Sawhney, and R. Kumar. Vehicle fingerprinting for reacquisition & tracking in videos. In *CVPR*, volume 2, pages 761–768, June 20–25, 2005.

[8] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih. Image indexing using color correlograms. In *CVPR*, pages 762–768, San Juan, June 17–19, 1997.

[9] O. Javed, K. Shafique, and M. Shah. Appearance modeling for tracking in multiple non-overlapping cameras. In *CVPR*, volume 2, pages 26–33, June 20–25, 2005.

[10] S. Kumar and M. Hebert. Discriminative random fields. *IJCV*, 68:179–201, 2006.

[11] S. Lazebnik, C. Schmid, and J. Ponce. Affine-invariant local descriptors and neighborhood statistics for texture recognition. In *ICCV*, pages 649–655, 2003.

[12] D. Lowe. Distinctive image features from scale-invariant key points. *IJCV*, 60:91–110, 2004.

[13] X. Ma and W. E. L. Grimson. Edge-based rich representation for vehicle classification. In *CVPR*, volume 2, pages 1185–1192, Oct. 17–21, 2005.

[14] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE TPAMI*, 27:1615–1630, 2005.

[15] H. Moon and P. J. Phillips. Computational and performance aspects of PCA-based face-recognition algorithms. *Perception*, 30(3):3003–321, 2001.

[16] G. Mori and J. Malik. Recovering 3d human body configurations using shape contexts. *IEEE TPAMI*, 28(7):1052–1062, July 2006.

[17] O. C. Ozcanli, A. Tamrakar, B. B. Kimia, and J. L. Mundy. Augmenting shape with appearance in vehicle category recognition. In *CVPR*, volume 1, pages 935–942, New York, NY, USA, 2006.

[18] F. Porikli. Integral histogram: a fast way to extract histograms in cartesian spaces. In *CVPR*, volume 1, pages 829–836, June 20–25, 2005.

[19] S. Savarese, J. Winn, and A. Criminisi. Discriminative object class models of appearance and shape by correlatons. In *CVPR*, volume 2, pages 2033–2040, 2006.

[20] B. Schiele and J. L. Crowley. Recognition without correspondence using multidimensional receptive field histograms. *IJCV*, 36(1):31–50, 2000.

[21] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, pages 1–15, 2006.

[22] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *ECCV*, pages 589–600, 2006.

[23] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *IJCV*, 62:61–81, 2005.

[24] A. Vedaldi and S. Soatto. Local features, all grown up. In *CVPR*, volume 2, pages 1753–1760, 2006.

[25] P. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57:137–154, 2004.

[26] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *ICCV*, volume 2, pages 1800–1807, Oct. 17–21, 2005.

[27] L. Wolf and S. Bileschi. A critical view of context. *IJCV*, 69(2):251–261, Aug. 2006.

[28] Q. Zhao and H. Tao. Object tracking using color correlogram. In *VS-PETS*, pages 263–270, Oct. 15–16, 2005.