The Intermediate Matching Kernel for Image Local Features

Sabri Boughorbel
Imedia Research Group
INRIA Rocquencourt
78153 Le Chesnay
E-mail: Sabri.Boughorbel@inria.fr

Jean Philippe Tarel
ESE Division
LCPC, 58 Bd Lefebvre
75015 Paris
E-mail: Tarel@lcpc.fr

Nozha Boujemaa
Imedia Research Group
INRIA Rocquencourt
78153 Le Chesnay
E-mail: Nozha.Boujemaa@inria.fr

***Abstract*— We introduce the Intermediate Matching (IM) kernel for SVM-based object recognition. The IM kernel operates on a feature space of vector sets where each image is represented by a set of local features. Matching algorithms have proved to be efficient for such types of features. Nevertheless, kernelizing the matching for SVM does not lead to positive definite kernels. The IM kernel overcomes this drawback, as it mimics matching algorithms while being positive definite. The IM kernel introduces an intermediary set of so-called virtual local features. These select the pairs of local features to be matched. Comparisons with the Matching kernel shows that the IM kernels leads to similar performances.**

I. Introduction

Object recognition is one of the pillar disciplines of computer vision. The machine learning community has provided efficient tools which have proved useful for solving many recognition tasks. One of the important steps in designing object recognition systems is the choice of feature representation. Global features are commonly used for image representation [1]. This has the advantage of dealing with simple structures and thus easy algorithms can be designed for feature processing. The main drawback of such a representation is that it does not capture localized information of objects within images. Local representation overcomes this drawback by extracting local features that allow object recognition against cluttered backgrounds.

Salient points are one of the local representations that have been widely used over the last decade. Detected salient points can be characterized in different ways (SIFT [2], Jet [3], [4], Image patch [5], [6]). This representation has the advantage of leading to invariant features with respect to transformations such as translation, rotation and scaling. Furthermore such features are robust to occlusion and distortion. On the other hand, kernel methods are providing successful tools that solves many recognition problems. SVM has proved to be one of the most efficient kernel method. The success of SVM is mainly due to its high generalization ability. Unlike many learning algorithms, SVM leads to good performances without the need to incorporate prior information. Moreover, the use of a positive definite kernel in the SVM can be interpreted as an embedding of the input space into a high dimensional feature space where the classification is carried out without using explicitly this feature space.

The use of SVM classifiers for image recognition with local features raises the need to design new kernels for such type of representations. One of the most important problems that should be tackled in designing kernels for local features is that, unlike components of $\mathbb{R}^d$, there is no structure between the local features. A few kernels have been introduced to handle such local representations. They include the principal angles kernel [7] which compares the two sets of local features by computing principal angles. This consists in finding maximum angles between local features under orthogonality constraints. In [6], Haussdorf kernel is proven to be definite positive in the case where salient points are characterized with image patches. In [8], while Matching kernel is not positive definite, it yielded to good accuracy. In [9], the two sets of local features are mapped into a high dimension space and fitted by a normal distribution. Then Bhattacharyya's distance is used between these two distributions.

In this paper, we introduce a positive definite kernel based on a new matching approach we term Intermediate Matching. The two sets of local features are matched using an intermediary set of the so-called virtual local features. Each virtual local feature defines a pair of vectors to be matched. We recall in the next section the theoretical framework of the Matching kernel. In § 2, we present the construction of the Intermediate Matching kernel and in § 3 the construction of virtual local features is given. Finally, comparisons between the Matching and the Intermediate Matching kernels are presented in § 4.

II. Matching Kernel

The problem of designing a kernel for sets of local features can be viewed as the choice of weighting coefficients w_{ij} and w'_{ij} in the following general kernel expression

$$K_{ww'}(\mathcal{X},\mathcal{Y}) = \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij} k(x_i, y_j) + \sum_{j=1}^{m}\sum_{i=1}^{n} w'_{ij} k(x_i, y_j) \tag{1}$$

$$w_{ij} \geq 0, w'_{ij} \geq 0$$

where $K_{ww'}$ is the kernel of the two sets of vectors $\mathcal{X} = \{x_1, \ldots, x_n\}$ and $\mathcal{Y} = \{y_1, \ldots, y_m\}$, $x_i, y_i \in \mathbb{R}^d$ and k is the local kernel which operates on the d-dimensional vectors. The two terms in $K_{ww'}$ are requested for a symmetrical matching.

Indeed, the weights w_{ij} and w'_{ij} are respectively normalized by:

$$\sum_{j=1}^{m} w_{ij} = 1 \ \forall i = 1, \dots, n \tag{2}$$

$$\sum_{i=1}^{n} w'_{ij} = 1 \ \forall j = 1, \dots, m \tag{3}$$

We make the assumption that the optimal kernel K^* from the family $K_{ww'}$ is the one that selects the weights which maximize the similarity between the two sets $\mathcal{X}$ and $\mathcal{Y}$.

$$K^*(\mathcal{X}, \mathcal{Y}) = \max K_{ww'}(\mathcal{X}, \mathcal{Y})$$
$$\text{s.t.} \sum_{j=1}^{m} w_{ij} = 1, \sum_{i=1}^{n} w'_{ij} = 1$$

We define the Matching kernel as follows

$$\widetilde{K}(\mathcal{X}, \mathcal{Y}) = \sum_{i=1}^{n} \max_{j=1,\dots,m} k(x_i, y_j) + \sum_{j=1}^{m} \max_{i=1,\dots,n} k(x_i, x_j)$$

$\widetilde{K}$ associates each vector from $\mathcal{X}$ to its nearest neighborhood from $\mathcal{Y}$. Moreover, each vector from $\mathcal{Y}$ is matched with its nearest neighborhood from $\mathcal{X}$. Hence $\widetilde{K}$ is a symmetric kernel. $\widetilde{K}$ can be viewed as the optimal choice within the family $K_{ww'}$ under the assumption of similarity maximization previously described.

Proof: for any $w_{ij} \geq 0$, s.t. $\sum_{j=1}^{m} w_{ij} = 1$

$$\begin{aligned}
k(x_i, y_j) &\leq \max_{j=1,\dots,m} k(x_i, y_j) \\
w_{ij} k(x_i, y_j) &\leq w_{ij} \max_{j=1,\dots,m} k(x_i, y_j) \\
\sum_{j=1}^{m} w_{ij} k(x_i, y_j) &\leq \underbrace{\sum_{j=1}^{m} w_{ij}}_{1} \max_{j=1,\dots,m} k(x_i, y_j) \\
\sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} k(x_i, y_j) &\leq \sum_{i=1}^{n} \max_{j=1,\dots,m} k(x_i, y_j)
\end{aligned}$$

The symmetric inequality can be obtained similarly for w'_{ij}

$$\sum_{j=1}^{m} \sum_{i=1}^{n} w'_{ij} k(x_i, y_j) \leq \sum_{j=1}^{m} \max_{i=1,\dots,n} k(x_i, y_j)$$

So we deduce $K_{ww'}(\mathcal{X}, \mathcal{Y}) \leq \widetilde{K}(\mathcal{X}, \mathcal{Y})$. The equality is achieved for the particular choice of weights: $w_{i,j_i} = 1$ and $w_{ij} = 0$ elsewhere, where j_i is defined by $k(x_i, y_{j_i}) = \max_{j=1,\dots,m} k(x_i, y_j)$, and $w'_{i_j,j} = 1$ and $w'_{ij} = 0$ elsewhere, where i_j is defined by $k(x_{i_j}, y_j) = \max_{i=1,\dots,n} k(x_i, y_j)$. Therefore $\widetilde{K} = K^*$. ∎

The Matching kernel $\widetilde{K}$ has been previously used for local image features [8]. Nevertheless, its positive definiteness is not obvious. Unfortunately, the simple proof in [8] of positive definiteness of the Matching kernel is not correct. Indeed, coefficients w_{ij} and w'_{ij} depend on $\mathcal{X}$ and $\mathcal{Y}$ as shown previously. Thus $\widetilde{K}$ cannot be viewed as a sum of positive definite kernels. Counter examples can be found to prove that such kernels are actually not positive definite, see [10]. Although Matching kernels have been successfully applied for recognitions tasks [8], [10], their use is risky since we no longer insure that the SVM maximizes the margin in some space. Moreover, there is always a potential risk that SVM does not converge. We also lose the nice property of solution uniqueness when L_2 regularization is used [11]. To overcome these drawbacks, we introduce the Intermediate Matching kernel. The main advantage of this kernel is that it mimics Matching kernels while being positive definite.

III. The Intermediate Matching Kernel

The Intermediate Matching kernel is designed such that it is positive definite. A set of virtual local features $\mathcal{V}$ is introduced. The role of each virtual local feature in $\mathcal{V}$ is to select the pairs of local features from $\mathcal{X}$ and $\mathcal{Y}$ to be matched. The positive definiteness of the Intermediate Matching kernel is given by the following proposition

***Proposition** 1:* We define the following mapping function from $\mathcal{P}(\mathbb{R}^d)$ (the set of parts of $\mathbb{R}^d$) to $\mathbb{R}^d$ which associates for every set $\mathcal{X}$ the nearest neighborhood elements from $\mathcal{X}$ to $\boldsymbol{v} \in \mathbb{R}^d$.

$$\begin{aligned}
\Phi_{\boldsymbol{v}} : \mathcal{P}(\mathbb{R}^d) &\rightarrow \mathbb{R}^d \\
\mathcal{X} &\rightarrow \boldsymbol{x}^* = arg \min_{\boldsymbol{x} \in \mathcal{X}} \|\boldsymbol{x} - \boldsymbol{v}\|
\end{aligned}$$

Using the mapping $\Phi_{\boldsymbol{v}}$, we define the Intermediate Matching Kernel as follows

$$K_{\mathcal{V}}(\mathcal{X}, \mathcal{Y}) = \sum_{\boldsymbol{v}_i \in \mathcal{V}} e^{-\frac{1}{2\sigma^2} \|\Phi_{\boldsymbol{v}_i}(\mathcal{X}) - \Phi_{\boldsymbol{v}_i}(\mathcal{Y})\|^2} \tag{4}$$

where $\mathcal{V} = \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_p\}$, $\boldsymbol{v}_i \in \mathbb{R}^d$ is the set of virtual local features. We claim that the Intermediate Matching kernel $K_{\mathcal{V}}$ is positive definite.

Proof: To prove the positive definiteness of the Intermediate Matching kernel, we use the positive definiteness of the RBF kernel:

$$K_{RBF}(\boldsymbol{x}, \boldsymbol{y}) = e^{-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|^2}{2\sigma^2}}, \ \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$$

Indeed, there exists a mapping $\Phi_{RBF} : \mathbb{R}^d \rightarrow \mathcal{H}$ such that $K_{RBF}(\boldsymbol{x}, \boldsymbol{y}) = \Phi_{RBF}(\boldsymbol{x}) \cdot \Phi_{RBF}(\boldsymbol{y})$ where $\mathcal{H}$ is the feature space associated with RBF kernel. We set $K_{\boldsymbol{v}}(\mathcal{X}, \mathcal{Y}) = e^{-\frac{1}{2\sigma^2} \|\Phi_{\boldsymbol{v}}(\mathcal{X}) - \Phi_{\boldsymbol{v}}(\mathcal{Y})\|^2}$. This last kernel can be written as:

$$\begin{aligned}
K_{\boldsymbol{v}}(\mathcal{X}, \mathcal{Y}) &= K_{RBF}\left(\Phi_{\boldsymbol{v}}(\mathcal{X}), \Phi_{\boldsymbol{v}}(\mathcal{Y})\right) \\
&= \Phi_{RBF}(\Phi_{\boldsymbol{v}}(\mathcal{X})) \cdot \Phi_{RBF}(\Phi_{\boldsymbol{v}}(\mathcal{Y})) \\
&= \tilde{\Phi}_{\boldsymbol{v}}(\mathcal{X}) \cdot \tilde{\Phi}_{\boldsymbol{v}}(\mathcal{Y})
\end{aligned}$$

$$\text{where} \ \ \tilde{\Phi}_{\boldsymbol{v}} : \mathcal{P}(\mathbb{R}^d) \longrightarrow \mathcal{H}$$
$$\mathcal{X} \longmapsto \tilde{\Phi}_{RBF}(\Phi_{\boldsymbol{v}}(\mathcal{X}))$$

$K_{\boldsymbol{v}}(\mathcal{X}, \mathcal{Y})$ is thus positive definite. The Intermediate Matching kernel can be written as:

$$K_{\mathcal{V}}(\mathcal{X}, \mathcal{Y}) = \sum_{\boldsymbol{v}_i \in \mathcal{V}} K_{\boldsymbol{v}_i}(\mathcal{X}, \mathcal{Y})$$

and is thus positive definite as a sum of positive definite kernels. ■

It is important to recall that the positive definiteness of the Intermediate Matching kernel is insured when the process of building virtual local features is *not related* to the two sets $\mathcal{X}$ and $\mathcal{Y}$. Moreover, the mapping $\Phi_{\boldsymbol{v}}$ can be chosen differently from that presented in proposition 1. For instance, we can choose a mapping Φ_v which defines for any set of local features $\mathcal{X}$ the mean vector the k-nearest neighborhood of a vector $\boldsymbol{v}$. Hence a wide variety of Intermediate Matching kernels can be derived.

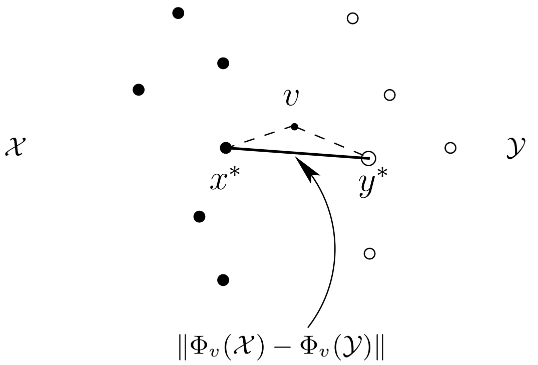

Fig. 1. For each element v of $\mathcal{V}$, the nearest elements $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$ are chosen from $\mathcal{X}$ and $\mathcal{Y}$, respectively. The distance between the two points is then computed and used in the computation of the Intermediate Matching kernel.

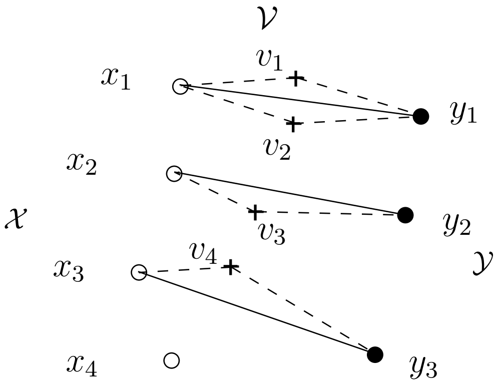

Fig. 2. Complete example of the computation of the Intermediate Matching kernel. The two points $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ match the sample pair $(\boldsymbol{x}_1, \boldsymbol{y}_1)$ twice. The point x_4 is far from $\mathcal{V}$, so it is not used during kernel computation.

Fig. 1 shows the construction of the mapping $\Phi_{\boldsymbol{v}}$. For the virtual local features $\boldsymbol{v} \in \mathcal{V}$, the nearest neighborhood vectors $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$ from sets $\mathcal{X}$ and $\mathcal{Y}$ are selected, then the distance $\|\boldsymbol{x}^* - \boldsymbol{y}^*\|$ is used for the kernel computation. We notice that $\boldsymbol{y}$ does not appear explicitly in the distance expression. It is used only to define the pair of vectors $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ to be matched. Fig. 2 presents a detailed example of Intermediate Matching kernel construction. We notice that virtual local feature $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ select the same pair $(\boldsymbol{x}_1,\boldsymbol{y}_1)$, so that the contribution of this pair counts twice in the kernel computation. The virtual local feature $\boldsymbol{v}_3$ selects the pair $(\boldsymbol{x}_2,\boldsymbol{y}_2)$ and $\boldsymbol{v}_4$ selects the pair $(\boldsymbol{x}_3,\boldsymbol{y}_3)$. If a matching algorithm was used in this example, the pair $(\boldsymbol{x}_4, \boldsymbol{y}_3)$ would be matched, since they are the nearest, but due to the fact that $\boldsymbol{v}_3$ is closer to $\boldsymbol{x}_3$ than $\boldsymbol{x}_4$ we did not obtain the result of the classical matching. We also notice that $\boldsymbol{x}_4$ was involved in the expression of the Intermediate Matching, therefore it can be discarded from the set $\mathcal{X}$, without changing the resulting value. In a way, virtual local features play the role of a feature selectors. The previous example demonstrates the influence of the choice of virtual local features on the Intermediate Matching process which can be summarized as follows:

Algorithm 1 Intermediate Matching Kernel

```
Input: $\mathcal{X} = \{\boldsymbol{x_1}, \ldots, \boldsymbol{x}_n\}$, $\mathcal{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m\}$, $\mathcal{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_p\}$, $\sigma$
Initialize: K=0
  for $\boldsymbol{v}_i \in \mathcal{V}$ do
    $\boldsymbol{x}^* = \arg\min_k \|\boldsymbol{x}_k - \boldsymbol{v}_i\|$
    $\boldsymbol{y}^* = \arg\min_k \|\boldsymbol{y}_k - \boldsymbol{v}_i\|$
    $K = K + e^{-\frac{\|\boldsymbol{x}^* - \boldsymbol{y}^*\|^2}{2\sigma^2}}$
  end for
```

As described previously, the computational cost of the Intermediate Matching kernel may appear high. First, the cost of Intermediate Matching kernel is lower than the cost of Matching kernels whenever the number of virtual local features does not exceed the number of vectors of the two sets $\mathcal{X}$ and $\mathcal{Y}$. Second, the computation of the Intermediate Matching kernel can be easily speeded-up by optimizing the search for the nearest vectors.

To sum up, we described the Intermediate Matching kernel and derived its positive definiteness. In the following, we will focus on building the virtual local features.

IV. Construction of the Virtual Local Features

We distinguish between two approaches for constructing the virtual local features $\mathcal{V}$. In the first approach, we consider a continuous and compact subset $\mathbb{R}^d$. In the second approach, $\mathbb{R}^d$ is finite and discrete, and training examples are used to obtain the virtual local features. The description of the first approach will be limited to the theoretical point of view. The second one will be described in more details. Validations and tests for the second approach are given in the experimental section.

A. Continuous case

The set $\mathcal{V}$ is chosen to be a continuous and compact subset of the feature space $\mathbb{R}^d$. The summation in the expression of the Intermediate Matching kernel (4) now turns into an integral:

$$K_{\mathcal{V}}(\mathcal{X}, \mathcal{Y}) \;=\; \frac{1}{|\mathcal{V}|} \int_{v \in \mathcal{V}} e^{-\frac{1}{2\sigma^2} \|\Phi_v(\mathcal{X}) - \Phi_v(\mathcal{Y})\|^2} da$$

The term $e^{-\frac{1}{2\sigma^2} \|\Phi_v(\mathcal{X}) - \Phi_v(\mathcal{Y})\|^2}$ is constant when v belongs to the intersection of the two cells of the multi-dimensional

Voronoi diagrams built from $\mathcal{X}$ and $\mathcal{Y}$. Thus, this integration over v turns into a volume intersection calculus. Let us denote by S_{ij} the volume intersection of the x_i and y_j Voronoi cells:

$$\begin{aligned} K_{\mathcal{V}}(\mathcal{X},\mathcal{Y}) &= \frac{1}{|\mathcal{V}|} \sum_{x_i\in\mathcal{X},y_j\in\mathcal{Y}} S_{ij} e^{-\frac{1}{2\sigma^2}\|x_i-y_j\|^2} \qquad (5) \\ &= \sum_{x_i\in\mathcal{X},y_j\in\mathcal{Y}} w_{ij} e^{-\frac{1}{2\sigma^2}\|x_i-y_j\|^2} \end{aligned}$$

Rewritten with weights w_{ij} as in the previous equation, we recognize the general kernel expression (1), up to the normalization. Notice that, even if w_{ij} depends on $\mathcal{X}$ and $\mathcal{Y}$, the kernel $K_{\mathcal{V}}$ is positive definite.

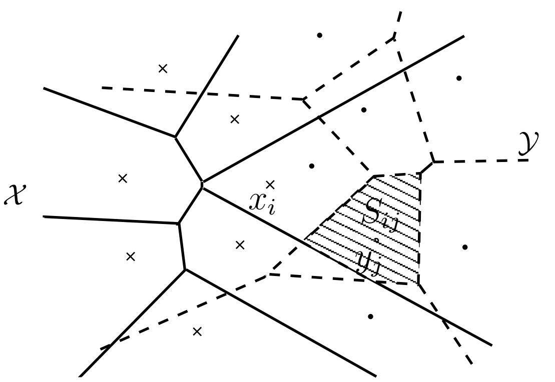

Fig. 3. For both sets $\mathcal{X}$ and $\mathcal{Y}$, Voronoi diagrams are constructed. The surface of the intersection between the two cells associated with x_i and y_j is termed S_{ij}.

To obtain the weights w_{ij}, we need to compute the volume of the intersection between the cells of the two Voronoi diagrams. This may be difficult when the number of dimensions d is large. Fig. 3 presents a 2D example of two Voronoi diagrams intersecting. For the pair (x_i,y_j) the volume of the two intersecting cells is shown with dashed lines and denoted by S_{ij} as in (5). The approach in the continuous case seems to be promising, and we plan to investigate it more deeply in a further study.

B. Discrete case

Within this approach, the set $\mathcal{V}$ is chosen to be finite and discrete. Each virtual local feature in $\mathcal{V}$ will select a vector in $\mathcal{X}$ and a vector in $\mathcal{Y}$. Therefore, virtual local features have to be chosen in highly informative regions of $\mathbb{R}^d$. To do so, a database of training examples is used. The regions of $\mathbb{R}^d$ with a high density of local features are interesting for choosing virtual features. To do that, all local features of all the training images are gathered and then a clustering is performed. In our case a simple FCM clustering algorithm is applied, but other clustering algorithms can be used with equal success. The cluster centers are then taken as the virtual local features. FCM clustering algorithm is summarized in Algorithm 2. The choice of the number of classes can be under-estimated without difficulty. The consequence is only a possible reduction in the number of matched local features.

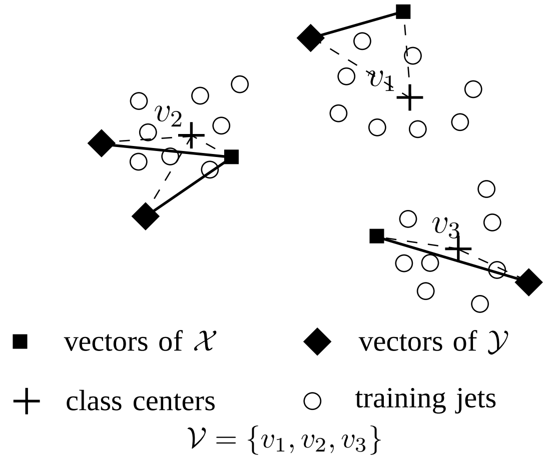
v_1
v_2
v_3
■ vectors of $\mathcal{X}$
◆ vectors of $\mathcal{Y}$
+ class centers
○ training jets
$\mathcal{V} = \{v_1, v_2, v_3\}$

Fig. 4. The virtual local features are the class centers of the clustering of whole local features extracted from training images.

Algorithm 2 Fuzzy C-Mean clustering algorithm [12]

```
Input: $\{\boldsymbol{x}_i\}, i = 1, \dots, q$
Initialize: Set the number of classes $p$ and the weighting factor $f$ so that $1 < f < \infty$
repeat
    Update class center's $\boldsymbol{v}_i = \frac{\sum_{k=1}^{q}(u_{ik})^f \boldsymbol{x}_k}{\sum_{k=1}^{q}(u_{ij})^f}$, $i = 1, \dots, p$
    Update memberships $u_{ik} = \frac{1}{\sum_{j=1}^{p}\left(\frac{\|\boldsymbol{v}_i - x_k\|}{\|\boldsymbol{v}_j - x_k\|}\right)^{\frac{2}{f-1}}}$ for all $i = 1, \dots, p,\ k = 1, \dots, q$
until $\boldsymbol{v}_i$ and $u_{ik}$ are stable.
```

Moreover, a small number p of classes increases the speed of the kernel computation. Indeed, the computational complexity of the Intermediate Matching kernel is $O(p(n+m))$. Compared to the computational complexity of the Matching kernel, which is $O(nm)$, the Intermediate Matching kernel is faster as soon as the number of virtual local features is smaller than n and m the numbers of local features in the two vectors sets.

V. Experiments

For the experiments we used an image database containing 4 objects as shown in Fig. 5. For each object, about 120 images with a size of 640× 480 are taken with different backgrounds and illumination conditions (about 30 different backgrounds, outdoor, indoor).

Salient points are detected using Harris detector [3], as shown in Fig. 6. Jets are chosen to be the invariant to rotation color differential features [4].

Virtual local features are obtained by clustering local features obtained from training images. To do so, there are two possibilities: clustering positive and negative examples separately or together. A couple of comparisons we made shows that separate clustering leads to better performance. This can be explained by the fact that introducing prior information can improve the performances. A k-fold cross-validation test is used to evaluate the recognition rate (k=4). For the sake of simplicity, we evaluate the one-vs-the others recognition task.

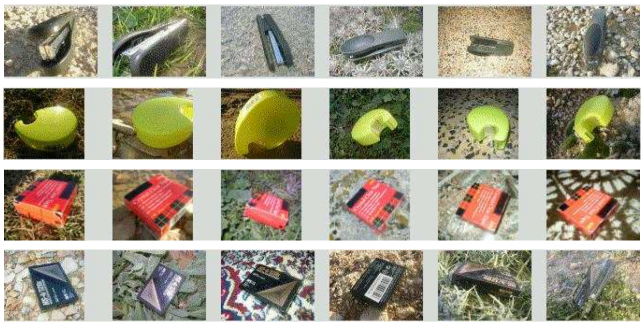

Fig. 5. Images used for experiments, each row presents respectively objects A, B, C and D.

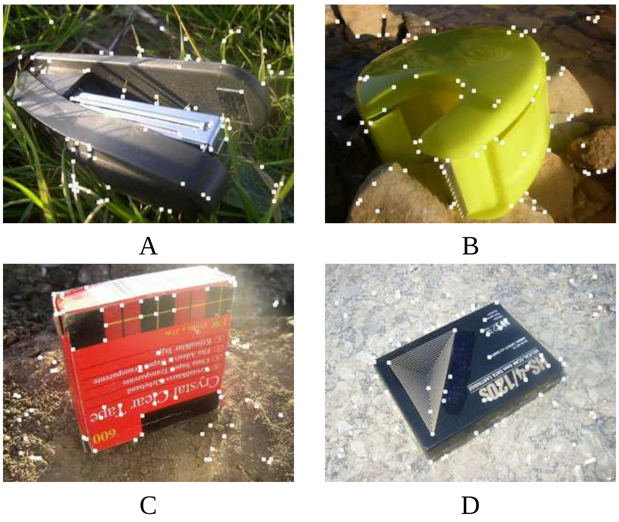

Fig. 6. Some test images of the four objects A, B, C, and D, with detected salient points.

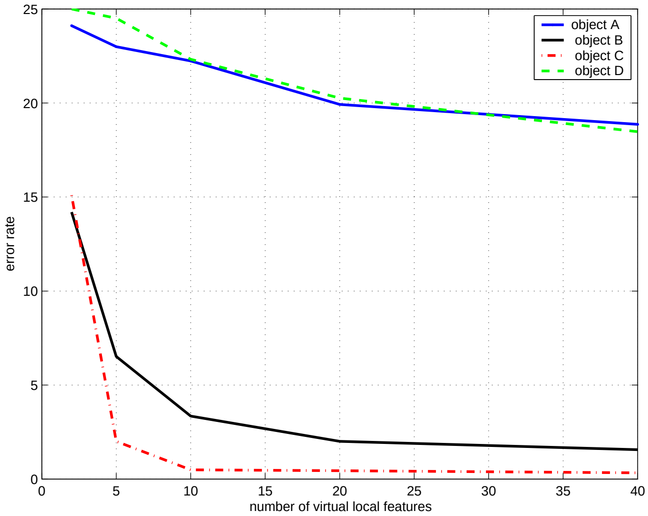

Fig. 7. Error rate with respect to the number of virtual local features (number p of cluster centers). Error rate is rapidly saturated.

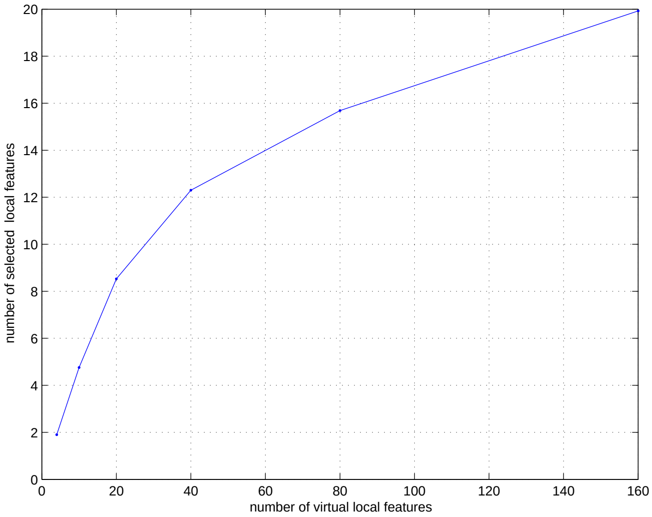

Fig. 8. The number of selected local features vs the number of virtual local features shows that a small portion of local features are involved in the computation of Intermediate Matching kernel.

Fig. 7 presents the error rate with respect to the number of virtual local features for the different objects. The number of virtual local features ranges from 2 to 80, and increasing their number improves the performance. However, the error rate is rapidly saturated, which means that a high number of virtual local features is not necessary to achieve best performances. For a given set of virtual local features, a small number of local features is involved in the Intermediate Matching kernel. This fact can be viewed as an online feature selection. Fig. 8 shows the evolution of the number of selected local features with respect to the number of virtual local features. In Fig. 9, error rate comparisons between the Intermediate Matching and Matching kernel is presented with respect to the scale parameter σ in (4) for objects A, B, C, and D. Performances of the Intermediate Matching kernel are good. Tab. I summarizes the optimal error rate obtained from the cross validation procedure. The Matching kernel is better than the Intermediate Matching kernel by about 1% for the recognition of objects A and D. While the latter's performances are better on objects C and D by about 0.5%. As a consequence, using 80 virtual local features, Intermediate Matching and Matching kernels are similar in performances, while only the Intermediate Matching kernel is sure to be definite positive. The performances of the Intermediate Matching kernel can be improved by increasing slightly more the number of virtual local features. We noticed that the obtained error rates for objects A and D are around 20% while error rates for the objects B and C are less than 1 %. This can be explained by the high similarity between objects A and D which increases possible confusions during the learning stage. One possible way to overcome such problem would be to take the spatial configuration of salient points into account so as to enhance recognition with respect to object shapes.

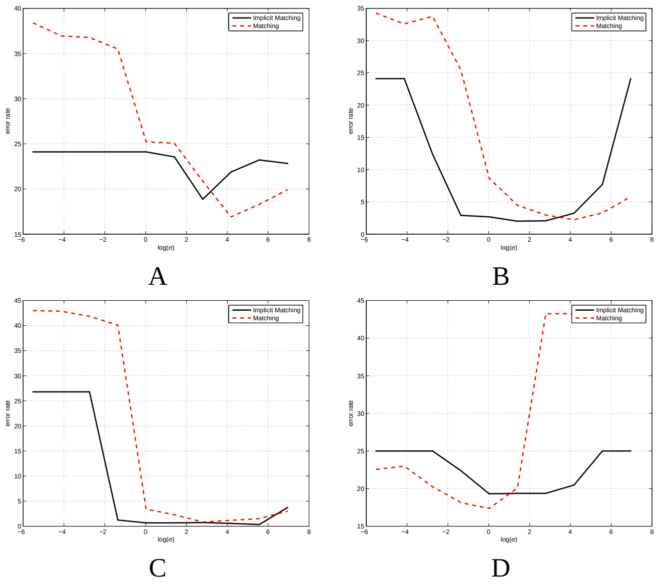

Fig. 9. Error rate comparisons between Intermediate Matching and Matching kernels with respect to the hyper-parameter σ in (4), for objects A, B, C, and D. The Intermediate Matching kernel uses 80 virtual local features obtained by clustering training local features.

objects	A	B	C	D
IM kernel	18.86%	**1.56%**	**0.33%**	18.47 %
σ	0.1	1	0.1	10
C	16.38	4.09	262.14	65.53
M kernel	**16.90%**	2.23%	0.83%	**17.24%**
σ	65.53	65.53	16.38	262.14
C	0.1	0.1	0.1	1000

TABLE I

BEST ERROR RATE OBTAINED FOR THE INTERMEDIATE MATCHING KERNEL (IM) AND MATCHING KERNEL (M). THE IM KERNEL IS TRAINED WITH 80 VIRTUAL LOCAL FEATURES.

VI. CONCLUSION

The Matching kernel for sets of vectors has been recently introduced and has been proved to be efficient for SVM-based image recognition with local features. However, this kernel is not positive definite and therefore the convergence of the SVM algorithm to a unique solution is not guaranteed. Matching kernel no longer insures that SVM algorithm converges. Moreover, The new Intermediate Matching kernel overcomes these drawbacks as it mimics the matching kernel while being positive definite. An intermediate set of virtual local features is introduced for matching. For each virtual local feature, the nearest pair of vectors to the virtual local feature is matched. In our experiments, with a reduced number of virtual local features, the new Intermediate Matching kernel performs as well as the Matching kernel. The Intermediate Matching kernel can be easily extended by changing the mapping Φ_v. This leads to a larger family of positive definite kernels for local features.

REFERENCES

[1] O. Chapelle, P. Haffner, and V. Vapnik, "Support vector machines for histogram-based image classification," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, 1999.

[2] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[3] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 530–535, 1997.

[4] V. Gouet, P. Montesinos, and D. Pelé, "Stereo matching of color images using differential invariants," in *Proceedings of the* IEEE *International Conference on Image Processing*, Chicago, Etats-Unis, 1998.

[5] Z. Zhang, R. Deriche, O. Faugeras, and Q. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," *Artificial Intelligence*, vol. 78, no. 1-2, pp. 87–119, 1995.

[6] A. Barla, F. Odone, and A. Verri, "Hausdorff kernel for 3D object acquisition and detection," in *In Proceedings of the European Conference on Computer Vision, LNCS 2353*, 2002, pp. 20–33.

[7] L. Wolf and A. Shashua, "Learning over sets using kernel principal angles," *Journal of Machine Learning Research (JMLR)*, vol. 4, no. 10, pp. 913–931, 2003.

[8] C. Wallraven, B. Caputo, and A. Graf, "Recognition with local features: the kernel recipe," in *ICCV*, 2003, pp. 257–264.

[9] R. Kondor and T. Jebara, "A kernel between sets of vectors," in *Proceedings of the ICML*, 2003.

[10] S. Boughorbel, J.-P. Tarel, and F. Fleuret, "Non-mercer kernels for SVM object recognition," in *Proceedings of British Machine Vision Conference (BMVC'04)*, London, England, 2004, pp. 137 – 146, http://www-rocq.inria.fr/ tarel/bmvc04.html.

[11] Christopher J. C. Burges and David J. Crisp, "Uniqueness of the SVM solution.," in *NIPS*, 1999, pp. 223–229.

[12] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.