RelCom: Relational Combinatorics Features for Rapid Object Detection

Vijay Venkataraman*
Oklahoma State University
Stillwater, OK
vvenka@okstate.edu

Fatih Porikli
Mitsubishi Electric Research Laboratories
Cambridge, MA
fatih@merl.com

Abstract

We present a simple yet elegant feature, RelCom, and a boosted selection method to achieve a very low complexity object detector. We generate combinations of low-level feature coefficients and apply relational operators such as margin based similarity rule over each possible pair of these combinations to construct a proposition space. From this space we define combinatorial functions of Boolean operators to form complex hypotheses that model any logical proposition. In case these coefficients are associated with the pixel coordinates, they encapsulate higher order spatial structure within the object window. Our results on benchmark datasets prove that the boosted RelCom features can match the performance of HOG features on SVM-RBF while providing 5× speed up and significantly outperform SVM-linear while reducing the false alarm rate 5×∼20×. In case of intensity features the improvement in false alarm rate over SVM-RBF is 14× with a 128× speed up. We also demonstrate that RelCom based on pixel features is very suitable and efficient for small object detection tasks.

1. Introduction

Small object detection still remains one of the most fundamental and challenging tasks in computer vision. On the core, it requires salient region descriptors that can accurately model object appearance and competent classifiers that can distinguish the large pool of object appearances from every possible background and clutter. Detection in infrared images is especially challenging due to the low spatial resolution of the object region. Variable thermal signatures, movable parts, combined with external illumination and pose variations, contribute to the complexity of the problem. Since detectors often form the first stage of the consecutive tracking and recognition tasks it is vitally important the detector be both accurate and fast.

Typically, the entire input image is scanned by a small moving window to compute the corresponding features and evaluating a learned classifier of the object model for each window. Haar wavelets have become popular due to their efficient computation [19]. More recently, histogram-based representations of image gradients in spatial context, including the histogram of oriented gradients (HOG) [6], the scale-invariant feature transform [14], shape context [1], were shown to yield more distinctive descriptors. In [21] a region was represented by the covariance matrix of image attributes in addition to histograms. The list of common descriptors can be extended to Gabor filters, appearance templates, local binary patterns, etc. The explosion of available features has led to the application of data mining approaches [7, 24] for feature selection.

One recent trend in detection algorithms is the assembling of object parts according to spatial relationships in probabilistic frameworks [9], by generative [16] and discriminative models [18], or via matching shapes [2]. Part based approaches are in general more robust towards partial occlusions; however, they can only detect sufficiently large objects. Most leading holistic approaches are classifier methods including k-nearest neighbors, neural networks (NN), support vector machines (SVM), and boosting. Even though boosting enables correlating each weak classifier with a single region in the detection window, it does not encapsulate pair-wise and group-wise relations between two or more regions in the window, which would establish a stronger spatial structure.

Initial attempts to capture such relations can be dated back to the n*-tuple* concept proposed Beldose and Browning in 1959 [3]. The term n-tuple refers to an ordered set of n pixel index values corresponding to distinct pixels on the image plane. Here the feature characterized is the intensity values of the pixels. Earlier explanations regarded it as a simple perceptron in a multilayer neural network [15] and random forest of tree classifiers [13]. However, these approaches strictly make use of the intensity (or binary) values and do not encode the comparative relation between the pixels. More recently, the sparse feature concept has

[*] This work was completed when the author was an intern at MERL and supported in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under grant W911NF-08-1-0293.

978-1-4244-7030-3/10/$26.00 ©2010 IEEE

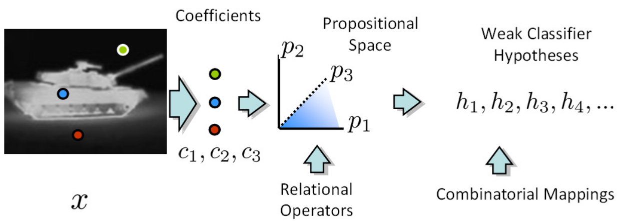


Figure 1. RelCom: After a number of coefficients are selected from the input image (or feature vector), a set of relational operators are imposed to generate a discrete proposition space, from which hypotheses are constructed by applying combinations of Boolean operators (conjunction, disjunction, etc.).

reemerged in the form of a finite number of quadrangular features called granules [11, 20, 8]. In such a granular space, a sparse feature is represented as a linear combination of several weighted granules. These features have certain advantages over Haar wavelets; they are highly scalable, do not require multiple memory accesses, and they partition the feature space into finer granularity [23]. However, each sparse and associated pairing comparison feature is either directly encoded into a scalar term almost like Haar wavelets, or requires both color and gradient attributes, or is defined only between a pair of granule sets.

Many of the existing works [12, 17] on infrared data, on the other hand, depend on simple, pixel level morphological operators and suppression of background clutter for detection of small targets. In [4] a set of range gate features is computed for vehicle detection. In [25] it is shown that feature descriptors developed primarily for visible spectrum can be adopted in infrared in the case of larger targets like pedestrians.

All of the above representations, irrespective of their complexity, are undeniably based on pixel values of the image. This leads us to explore the possibility of extending the sparse pixel features known as n-tuples into more competent forms for detection tasks. Here we introduce the relational combinatorics features *RelCom*. We first generate combinations of low-level attribute coefficients, which may directly correspond to pixel coordinates in the target window or feature vector coefficients representing the window itself, up to a prescribed size n (pairs, triplets, quadruples, etc). We then apply relational operators such as margin based similarity rule over each possible pair of these operands. The space of relations constitutes a proposition space that divides the original feature space into discrete regions. From this space we define combinatorial functions of Boolean operators to form complex hypotheses as shown in Fig. 1. Therefore, we can produce any relational rule over the operands, in other words, any logical proposition over the low-level descriptor coefficients. In case these coefficients are associated with pixel coordinates, we encapsulate higher order spatial structure information within the object window. Using a descriptor vector instead of pixel values, we effectively impose feature selection without any computationally prohibitive basis transformations such as PCA. In addition to proposing a simple methodology to encode the relations between n pixels on an image (or n vector coefficients), we employ boosting to iteratively select a set of weak classifiers from these relations to perform faster target detection.

RelCom is significantly different from the body of work developed around n-tuples, as we explicitly use logical operators with a learned similarity threshold as opposed to raw intensity (or gradient) values. Unlike the sparse features and associated pairings, it extends the combinations of the low-level attributes to multiples of operands to gain better object structure imposition on the classifier. Instead of mining compositional features [24], which can split the feature space only along the dimensions as k-trees, RelCom partitions the space into margin regions along the hyperplanes and constructs higher level hypotheses, thus, it can provide much better granularity using the same number of primitive classification rules.

2. RelCom Features

Consider a dataset $\mathscr{D}_N = \{\mathbf{x}_t, c_t\}_{t=1}^N$ with N training samples where each sample is characterized by its feature vector $\mathbf{x}_t \in \mathbb{R}^d$ and has an associated binary class label $c_t \in \{-1, 1\}$. The traditional classification problem is to find a classifier function $\mathbf{g}(.) : \mathbf{x} \rightarrow c$ that provides a mapping between the feature space and class labels. $\mathbf{g}(.)$ is usually determined by minimizing the classification error over a representative training set. Instead of a direct mapping from the feature space to class labels, we define a binary valued propositional feature space $\{\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_K\}$ where each $\mathbf{f}_k : \mathbf{x} \rightarrow \{0, 1\}$. In effect this is a transformation from the continuous valued scalar space to a binary valued space and possibly a reduction in dimension if $K < d$. The mapping function $\mathbf{f}_k$ can take on a multitude of forms such as a simple decision stump in a single dimension, a multidimensional hyperplane, a threshold based match filter etc. For any given classification problem there are a plethora of possible feature representations of the objects involved. Therefore, the choice of $\mathbf{f}_k$ will be dependent on the semantic meaning of the features $\mathbf{x}$ and the problem at hand.

After obtaining the K-bit binary string $\mathbf{F} = \{\mathbf{f_1}, \mathbf{f_2}, \cdots, \mathbf{f_K}\}$ by choosing an appropriate mapping function, it is easy to see that there are 2^{2^K} possible ways to assign binary class labels to any given test sample $\mathbf{x}$. An example for the case of $K = 3$ is shown in Tab.1 where the left column represents all possible binary string patterns and each hypothesis column $h_i(\mathbf{F})$ on the right represents one possible class label assignment pattern. Though the number of possible hypothesis increases greatly with K we have found in our experiments $K = 2, 3$ was adequate to meet the detection challenge. The value of h_i indicate whether a sample is classified as positive (1) or negative (-1) for a given propositional binary pattern.

$\mathbf{f}_1$	$\mathbf{f}_2$	$\mathbf{f}_3$	$h_1(\mathbf{F})$	$h_2(\mathbf{F})$	$h_3(\mathbf{F})$	$\cdots$	$h_i(\mathbf{F})$	$\cdots$	$h_{256}(\mathbf{F})$
0	0	0	-1	-1	-1	$\cdots$	1	$\cdots$	1
0	0	1	-1	-1	-1	$\cdots$	1	$\cdots$	1
0	1	0	-1	-1	-1	$\cdots$	1	$\cdots$	1
0	1	1	-1	-1	-1	$\cdots$	-1	$\cdots$	1
1	0	0	-1	-1	-1	$\cdots$	-1	$\cdots$	1
1	0	1	-1	-1	-1	$\cdots$	1	$\cdots$	1
1	1	0	-1	-1	1	$\cdots$	-1	$\cdots$	1
1	1	1	-1	1	-1	$\cdots$	1	$\cdots$	1

Table 1. Illustration of the 2^{2^K} possible class label assignments for a propositional binary string of length $K = 3$.

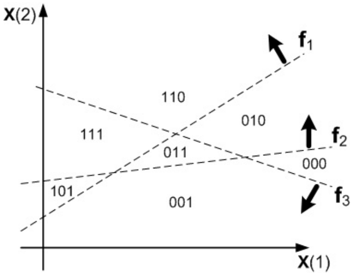


Figure 2. Illustrative mapping from feature space to the propositional space spanned by a 3-bit binary string. The dotted lines represent decision stumps. Data points that lie on the positive normal side (represented by dark arrows) of a decision stump map to a binary 1 in the propositional space.

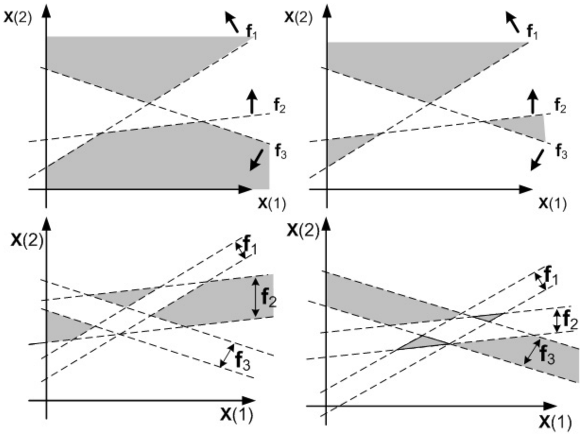


Figure 3. Illustration of possible complex decision boundaries using combinatorial features. Data points that lie in the shaded regions are classified as positives. Propositional mapping using Top: simple decision stumps and Bottom: margin based similarity rule.

To illustrate further the concept of combinatorial features consider a d dimensional feature descriptor $\mathbf{x} = [\mathbf{x}(1) \quad \mathbf{x}(2) \cdots \quad \mathbf{x}(d)]^T$ and an associated 3-bit propositional mapping $\{\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3\}$ using simple decision stumps. Fig.2 represents a hypothetical projection of these decision stumps along the first two dimensions of the feature space. We observe that the entire feature space has been divided into a small number of discrete region each with a binary string label. Fig.3 represents the decision boundaries corresponding to a few possible hypothesis from Tab.1 where data samples falling within the shaded region are classified as positives. It is observed that simple logical operations in the propositional space form complex decision boundaries in the original feature space. Thus we can define complex decision boundaries by combining the results of individual simple decision stumps in a multitude of combinations. From Fig.3 it is easy to perceive that the decision regions resulting from the combinations are more likely to be beneficial in classification problems than those of any individual decision stumps. However, the regions of the individual decision stumps are a subset of the larger set of all possible combinations. Though combinational features allow for complex decision boundaries we still consider each of these to be a weak classifier and perform boosting to select an informative subset from these combinations.

Some of the hypotheses in Tab.1 are degenerate and are logically invalid such as the first and last columns. Half of the remaining are complements of a different column and need not be evaluated explicitly. Based on the definition of $\mathbf{f}_k$ it is possible that some of the patterns in the left column never occur and this further reduces the number hypothesis to be evaluated. An example of this is seen in Fig.2 where the string 100 is not a possibility. Thus, when we search within the hypotheses it is not necessary to evaluate all of 2^{2^K} possibilities.

Fast target detection invariably requires the computational load imposed by features and the propositional mapping to be minimized. In this paper we primarily consider the simplest possible feature - raw image pixel values. The feature vector $\mathbf{x}$ is taken to be a raster scan of the pixel values making its dimension d equal to the number of the pixels in the target window. Experiments with other feature descriptors computed in a target window, e.g. HOG feature are also considered.

Inspired by the n-tuple classifier and other recent works [11, 8] that capture pairwise feature variations in a small subset of the entire feature space, we define our propositional mapping function to be a simple margin based similarity rule that operates on two feature dimensions chosen from a set of n randomly sampled feature dimensions. For a given d dimensional feature vector $\mathbf{x}$, we randomly select n, $(n < d)$, of the possible dimensions and represent it by $P_n = \{p_1, p_2, \cdots, p_n\}$ where each p_k is unique and $p_k \in \{1, 2, \cdots, d\}$. Given an arbitrary n-tuple P_n, for each unique pair (p_i, p_j), $p_i, p_j \in P_n$ we can define a propositional mapping $\mathbf{f}_k$ of the form

$$\mathbf{f}_k(\mathbf{x}) = \begin{cases} 1 & |\mathbf{x}(p_i) - \mathbf{x}(p_j)| \le \tau_k \\ 0 & otherwise, \end{cases} \tag{1}$$

where $\mathbf{x}(p_i)$ represents the value of the feature along the p_i'th dimension. The margin value τ_k indicates the acceptable level of variation and it can be chosen so as to maximize the classification performance of a particular hypotheses if prior knowledge of the feature space is available. Given an n-tuple and the definition in Eq.1 the number of unique propositional mappings $\mathbf{f}_k,\ k \in \{1, \cdots, K\}$ that can be defined is limited to $K = \binom{n}{2}$ corresponding to the number of possible unique pairs (p_i, p_j). We denote the resulting binary string by $\mathbf{F}(\mathbf{x}) = \{\mathbf{f_1}, \cdots, \mathbf{f_K}\}$.

When the propositional mapping in Eq.1 is applied to raw image pixel values with n-tuples, we are effectively analyzing the intensity variation patterns over the windowed image region n pixels at a time. This draws attention to the extremely large number of n-tuples that can be selected for any given image window vector of dimension d, $T_n = d!/(d-n)!$. In this work we mostly deal with the cases of $n = 2, 3$ which we refer to as 'doublet' and 'triplet' respectively. For the case of even a 10×10 template and triplets there exists ≈970k unique choices of triplets. A second point of concern is the selection of K different threshold τ_k, a continuous variable, that is difficult to optimize without prior knowledge of the feature space. The vastness of this parameter space comprising of n-tuples and thresholds makes determining optimal values for either of them impossible. However, each n-tuple, threshold pair can be thought of as a weak classifier and these can be combined by boosting to produce a strong classifier. Since we explore different sparse combinations of the feature space using relational operators we refer to our feature as 'RelCom'.

2.1. Boosting

To select the most discriminative RelCom features from a large pool of candidates we use the discrete AdaBoost algorithm [10]. Since the output of each RelCom hypothesis is binary it can easily be adapted into the discrete AdaBoost framework. AdaBoost works iteratively to combine a number of weak classifiers linearly to produce a strong classifier with acceptable classification performance. In each iteration a single weak classifier is selected from a pool such that it minimizes the weighted error over the training set. The weights of the misclassified samples are increased (and the weights of each correctly classified example are decreased), so that in the next iteration the new weak classifier focuses more on the misclassified examples. It has been shown [10] that for a binary classification problem the error of the final hypothesis decreases exponentially with the number of boosting rounds (i.e additional weak classifier).

Our adaptation of the discrete AdaBoost for RelCom features (shown in Fig. 4) is similar to original AdaBoost, except differences at the level of weak learners. In this case, domain of the weak learners is in the combinatorial n-tuple hypotheses and threshold space. In each iteration random samples of a set of n-tuples P_n^s and associated thresholds $\mathscr{T}^s$ are drawn for more efficient spanning of the enormous search space. These define the mapping from the input feature space $\mathbf{x}_t$ to the propositional space $\mathbf{F}_s^t(\mathbf{x}_t)$. Next we identify the n-tuple pattern, associated threshold and the hypothesis pattern that minimizes the weighted error on the

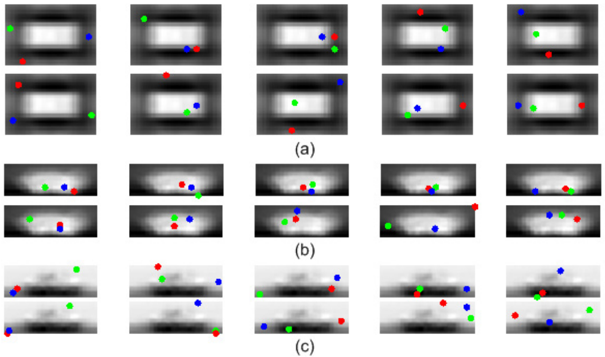
```
Given:
∗ Training dataset with feature vectors, class labels
  $\mathscr{D} : \{(\mathbf{x}_1, c_1), \cdots, (\mathbf{x}_N, c_N)\}$, where $c_t = \pm 1$ indicates the class label.
∗ $N_c$ the required number of weak classifiers.
∗ $S$ weak classifiers pool size.
Initialize:
∗ Sample weights $W_1(t) = \frac{1}{2N^+}, \frac{1}{2N^-}$ for $c_t = 1, -1$ respectively, where $N^+$ and $N^-$ are the number of positive and negative samples.
AdaBoost:
∗ For $i = 1, \cdots, N_c$
    • Randomly sample $S$ $n$-tuples $P_n^1, \cdots, P_n^S$. For each $P_n^s$ also sample threshold values $\mathscr{T}^s = \{\tau_k^s\}$, $k = \{1, \cdots, \binom{n}{2}\}$.
    • For each of the $S$ $n$-tuples compute the propositional mapping $\mathbf{F}_1^t, \mathbf{F}_2^t, \cdots, \mathbf{F}_S^t$ over the training samples $t = \{1, \cdots, N\}$ using Eq.1.
    • For each of the $S$ $n$-tuples compute error for all valid hypothesis in the set $h_j(\mathbf{F})$ $j = \{1, \cdots, 2^{2^K}\}$ as $\lambda_s^j = \sum_{t=1}^{N} W_i(t)[c_t \neq h_j(\mathbf{F}_s^t)]$.
    • Set $P_{sel,n}^i = P_n^{smin}$, $\mathscr{T}_{sel}^i = \mathscr{T}^{smin}$, $h_{sel}^i(\mathbf{F})$=$h_{jmin}(\mathbf{F})$ and $\epsilon_i = \lambda_{smin}^{jmin}$. Where $smin$ and $jmin$ are indices : $\lambda_{smin}^{jmin} < \lambda_s^j \, \forall s \neq smin, j \neq jmin$.
    • Calculate $\alpha_i = \frac{1}{2} \cdot \ln\left[\frac{1-\epsilon_i}{\epsilon_i}\right]$.
    • Update the sample weights for $t = \{1, \cdots, N\}$
      $W_{i+1}(t) = W_i(t) \exp[-\alpha_i c_t h_{sel}^i(\mathbf{F}_{smin}^t)]$.
    • Normalize the weights $\sum_{t=1}^{N} W_{i+1}(t) = 1$.
Output:
∗ Selected $n$-tuples $P_{sel,n}^i$, threshold $\mathscr{T}_{sel}^i$, hypothesis $h_{sel}^i$ and classifier weight $\alpha_i$ for $i = \{1, \cdots, N_c\}$.
```

Figure 4. Training RelCom features with discrete AdaBoost.

```
Given:
∗ A single sample feature vector $\mathbf{x}_{test}$
∗ RelCom classifier consisting $P_{sel,n}^i$, $\mathscr{T}_{sel}^i$, $h_{sel}^i$ and $\alpha_i$ for $i = \{1, \cdots, N_c\}$
Testing:
∗ Identify $\mathbf{F}_1$,$\mathbf{F}_2, \cdots$,$\mathbf{F}_{N_c}$ using $\mathbf{x}_{test}$,$P_{sel,n}^i$ and $\mathscr{T}_{sel}^i$.
∗ final classifier $H(\mathbf{x}_{test})$ =sign $[\sum_{i=1}^{N_c} \alpha_i h_{sel}^i(\mathbf{F}_i)]$.
```

Figure 5. Testing with RelCom features.

training set and update the training sample weights. The identified patterns, threshold and hypothesis are added to the classifier pool with associated weight α_i. Once trained, given a test feature vector, the propositional mapping can be identified from a lookup table as we use a margin based similarity rule. The hypothesis corresponding to the propositional binary pattern is pre-stored in a second lookup table and also requires no computation. The output of the strong classifier is the sign of the sum of the weighted RelCom feature responses as shown in Fig. 5.


(a)

(b)

(c)

Figure 6. Top ten RelCom triplet feature locations shown on the mean positive images for (a) CSUAV, (b) SENSIAC night time and (c) SENSIAC day time datasets. The features identified compare the object with its background aiming to distinguish the silhouette.

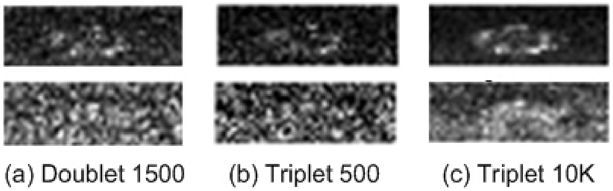

(a) Doublet 1500 (b) Triplet 500 (c) Triplet 10K

Figure 7. Map of normalized weighted location of RelCom features for the SENSIAC dataset. Top: Night time and Bottom: Day time. The target edges, as expected, are found to be more salient by RelCom features.

To illustrate the competence of the boosted RelCom features, we analyze the classifier learnt for detection on two infrared datasets 1)CSUAV (civilian vehicles from aerial view) and 2) SENSIAC (military targets from planar view) using raw pixel values as the feature vector. Figure 6 shows the position of the top 10 RelCom features for the case of triplets. Interestingly, the features are distributed on the target and background to gather clues about the target shape. Figure 7 presents the map of the weighted locations of the RelCom features on the SENSIAC dataset when the number of weak learners is varied. As indicated by brighter intensity, many of the selected features are concentrated along the target edges as they are more discriminative. Increasing the number of features helps to concentrate attention on the salient regions of the target especially in the night time images. For the day time images the effect is less pronounced due to the presence of considerable clutter.

Algortihm	Computational	INRIA	
	complexity	∼ Operations	% of FA
SVM-Linear	$10d$	21,000	4.58
SVM-RBF	$13dN_{sv} + 47N_{sv}$	20,700,000	0.38
RelCom Doublet	$12N_c$	18000	0.22
RelCom Triplet - 1	$16N_c$	**8000**	0.23
RelCom Triplet - 2	$16N_c$	160,000	**0.02**

Table 2. Computational complexity and performance of different algorithms given a input vector of dimension d, the number of: learnt support vectors N_{sv} and weak classifiers N_c. The relative costs of processor operations are measured against the cost of memory access taken to be unity. The above expressions assume the cost of an addition to be 3, multiplication to be 5 and an exponential to be 35. For the INRIA dataset using 64×32 intensity images, $d = 2048$ and $N_{sv} = 776$. We set $N_c = 1500, 500, 10k$ weak learners for the RelCom doublet, triplet-1 and triplet-2 respectively.

2.2. Computational Load

Note that the operator used to map from the feature space to propositional space has a simple margin based distance form. Therefore, it is possible to construct a 2D lookup table to determine the propositional binary string given the n-tuples. This can be achieved without loss of information for intensity features, and an insignificant adaptive quantization loss for other low-level features. Particularly, this lets us masterfully trade in the computational load with the memory imprint of the algorithm, which itself is relatively small (as many 100×100 or 256×256 binary tables as the number of features). In case of 500 triplets, the memory for the lookup tables is approximately 100MB. After obtaining the propositional binary string a secondary lookup table of the hypothesis is used to identify the binary class label. We can then multiply these labels with their corresponding weak classifiers' weights and aggregate the sum to determine the response. In other words, in the testing stage we only need to employ array access operations instead of complex arithmetic operations, which results in a very fast detector. Due to vector multiplications, neither SVM radial basis functions nor linear kernels can be implemented in such a manner.

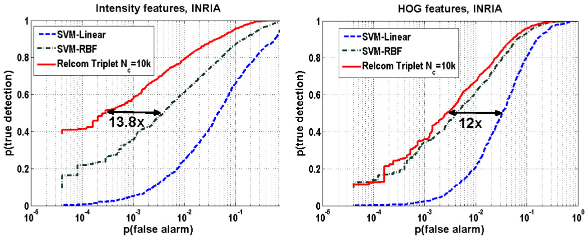
The computational load and the performance of several classifiers including the boosted RelCom doublet and triplets are compared in Table 2. As shown, RelCom boosting provides one of the fastest classifiers whose complexity only depends on the number of weak classifiers even without a cascade implementation. It easily outperforms SVM-RBF and requires only a fraction of the load (∼128× speed up for the INRIA dataset).

Further, in the context of boosted classifiers it is possible to implement a rejection cascade that significantly reduces the computational load in scanning window based detection. As an example, for Haar wavelet based face detection the classifier becomes 750× faster [22] by decreasing the effective number of features to be tested from 6000 (in the original boosted strong classifier) to a mere 8 on average! In other words, a cascade implementation of boosted RelCom has every potential to further speed up detection.

3. Experiments

To demonstrate the capability of the proposed RelCom features, detection is performed on three different datasets. 1) INRIA person[1] dataset - human detection in visual images, 2) SENSIAC ATR[2] dataset - military vehicle detection in midwave infrared (MWIR) images and 3) CSUAV[3] dataset- car detection in MWIR images taken from an UAV. We compare the performance of three different algorithms including SVM-Linear, SVM-RBF and RelCom triplets. The SVM parameters were set to maximize cross validation accuracy on the training set. LibSVM toolbox [5] was used for training and testing. The basis of comparison are the ROC curves that plot the probability of true detection *vs* probability of false alarms and visual detection results.

From the standard INRIA dataset we obtained 2416 pictures of mirrored and centered images and a further 12180 samples of random backgrounds. Of these only a fifth of the positive samples and a tenth of the negative samples were used for training. For testing purposes 24360 random background images and 1126 positives we used. All images were of size 32×64. We tested the algorithm performance for two different feature types: greyscale intensity values and HOG features. For HOG feature calculation, we used a [-1 1] filter in orthogonal directions and adopted integral histograms for fast evaluation. HOG features were computed for 8 directions in non-overlapping blocks of size 16×8 resulting in a $8 \times 4 \times 4 = 128$ dimensional feature vector.

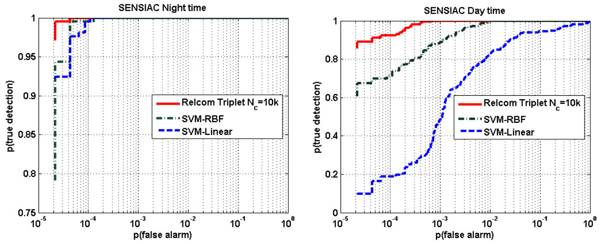


(a) (b)

Figure 8. Comparative ROC curves for INRIA dataset (a) pixel intensity features (b) HOG features.

Figure 8(a) shows the detection performance curves for INRIA dataset when using intensity features. The boosted RelCom triplets with 10k classifiers significantly outperforms SVM-RBF to our surprise by a factor of 13.8 at the 50% true detection level. At the same time it outperforms SVM-Linear by almost a factor of 25. Figure 8(b) presents the ROC curves in the case the HOG feature. Performance of the RelCom is at par with the SVM-RBF and 12 times better than the SVM-Linear. We are able to achieve performance comparable to SVM-RBF at significantly lower computational cost. This illustrates that the proposed method is applicable to any given feature and not limited to intensity features.

The SENSIAC dataset consists of MWIR sequences acquired both during day and night times for eight different targets. The targets are imaged at multiple distances and poses. We select 3 of those targets (Pickup, BTR70 and BRDM2) at a distance of 2000 meters to create a training set of 60 positive samples, 5900 negative samples and the testing set consisted of 200 positive samples, 45800 negative samples each for day and night time images. The images were histogram equalized before extracting sample templates of size 15×45. We train two separate classifiers for day and night time data on greylevel intensity features. The ROC performance curves are shown in Fig.9 for both day and night time detection. For the night time data the performance of all the algorithms is very similar as the target is distinctly visible. The advantage of RelCom features is clearly emphasized in the day time images where is significantly outperforms SVM-RBF even in the presence of significant background clutter. In addition results of detection in three different scenarios are shown in Fig.10, Fig.11 and Fig.12. In a given image, detection is performed by scanning over the entire image with a small target window. Each windowed region is passed as input to the RelCom classifier which quickly identifies it as either target or clutter. Note that in each scenario either the target or the imaging distance is previously unseen (not present in training set). The boosted RelCom classifier is able to clearly detect the target even when other methods fail entirely or result in excessive false alarms.



(a) (b)

Figure 9. ROC curves for SENSIAC dataset (a) Night (b) Day.

The CSUAV dataset contains MWIR images acquired from an UAV flying over a civilian locality. From this dataset we selected 1050 positive and 13000 negative samples for training. For testing purposes 1050 postives and 65000 negatives were used. The template size was 20×30. Here again it was found that the RelCom triplet greatly outperforms the SVMs in detection performance. Fig 13 shows

[1] http://pascal.inrialpes.fr/data/human/

[2] https://www.sensiac.org/external/index.jsf

[3] https://www.sdms.afrl.af.mil/datasets/csuav/

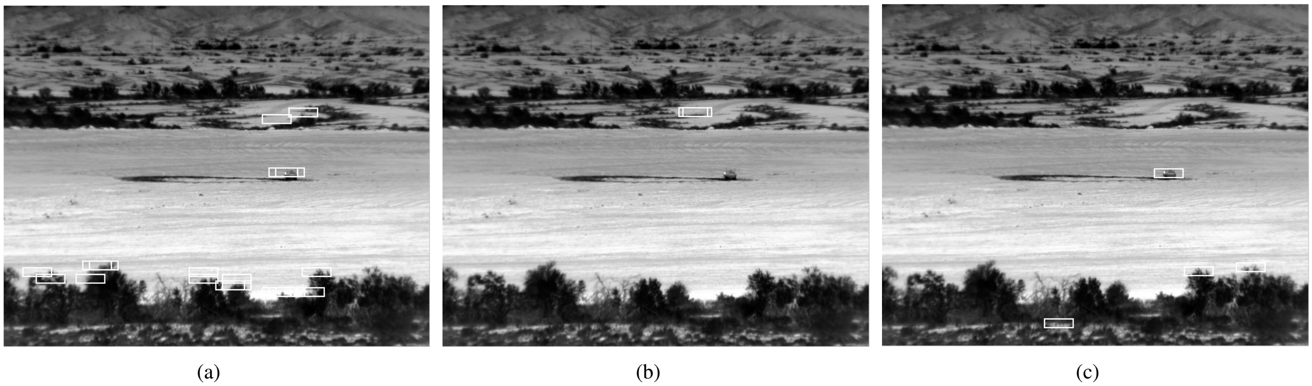


Figure 10. Sample detection results at FA rate of 10^{-4} on Day time SENSIAC data at Distance:1500m (unseen) and Target: BRDM2 (seen) for (a) SVM-Linear (b) SVM-RBF and (c) RelCom triplet.

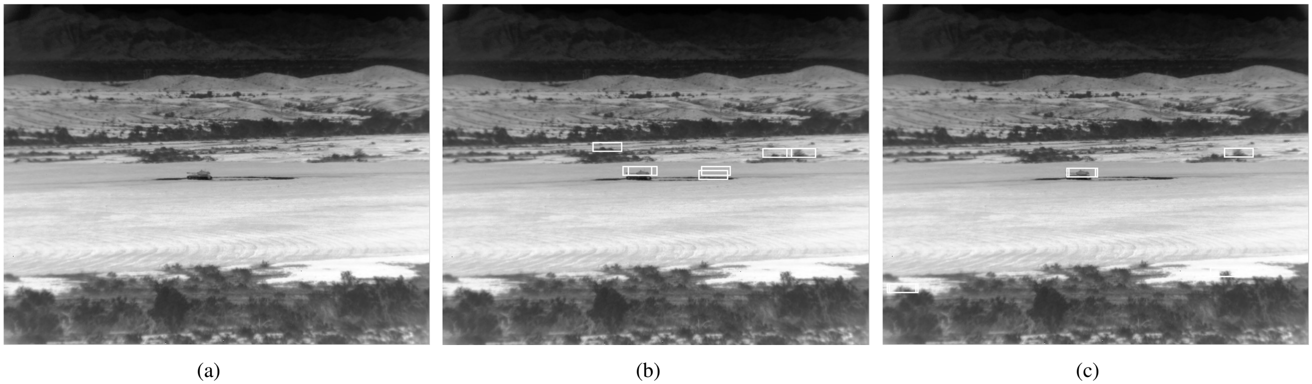


Figure 11. Sample detection results at FA rate of 10^{-4} on Day time SENSIAC data at Distance:2000m (seen) and Target: ZSU23 (unseen) for (a) SVM-Linear (b) SVM-RBF and (c) RelCom triplet.

the result of using RelCom triplets in three different scenarios. Since we trained the classifier with a general template irrespective of orientation, during the detection phase the image was scanned for targets at orientations of 0^o, 45^o and 90^o to detect targets oriented along different directions. The detection results were then combined using non-maximum suppression. We see that majority of the vehicles including those in shade and near trees where correctly detected even though some roof tops were falsely detected. These experiments establish the competence of the RelCom detector for small target detection using intensity features in infrared images.

4. Conclusion and Future Work

We show that high-level combinations of basic relational features can be used in a boosting framework to construct very fast classifiers that are as competitive as SVM-RBF while requiring only a fraction of the computational load. To summarize the advantages of our method:

- RelCom can speed up detection several orders of magnitude because it does not require any complex computations thanks to the two-layer lookup tables.
- It can accommodate both basic features including pixel intensities and other complex descriptor vectors computed within the object window.
- It utilizes simple relational operators to capture the spatial structure within the object window effectively.
- It can be applied to very small object windows unlike HOG features.

As future work, we will improve the feature selection using more sophisticated feature mining strategies [7] in addition to the sampling approach we adopted in AdaBoost. A second improvement will be in the form of a rejection cascade classifier to further speed up detection and extension to multi-class classification.

References

[1] S. Belongie, J. Malik, J. Puzicha, “Shape matching and object recognition using shape contexts”, *IEEE PAMI*, 24(4):509–522, 2002. 1

[2] A. Berg, T. Berg, J. Malik, “Shape matching and object recognition using low distortion correspondence”, CVPR, 2005. 1

[3] W. Bledsoe, I. Browning, “Pattern recognition and reading by machine”, Proc. Eastern Joint Computer Conf., Boston, Mass., 1959. 1

[4] A. L. Chan, S. Z. Der, N. M. Nasrabadi, “Multistage infrared target detection”, *Opt. Eng*, 42(9):2746–2754, 2003. 2

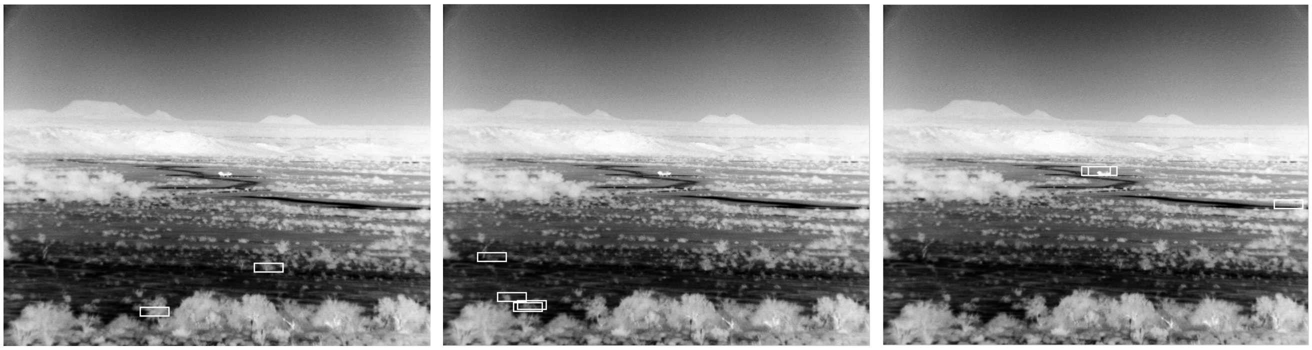


Figure 12. Sample detection results at FA rate of 10^{-4} on Night time SENSIAC data at Distance:4000m (unseen) and Target: BMP2 (unseen) for (a) SVM-Linear (b) SVM-RBF and (c) RelCom triplet.

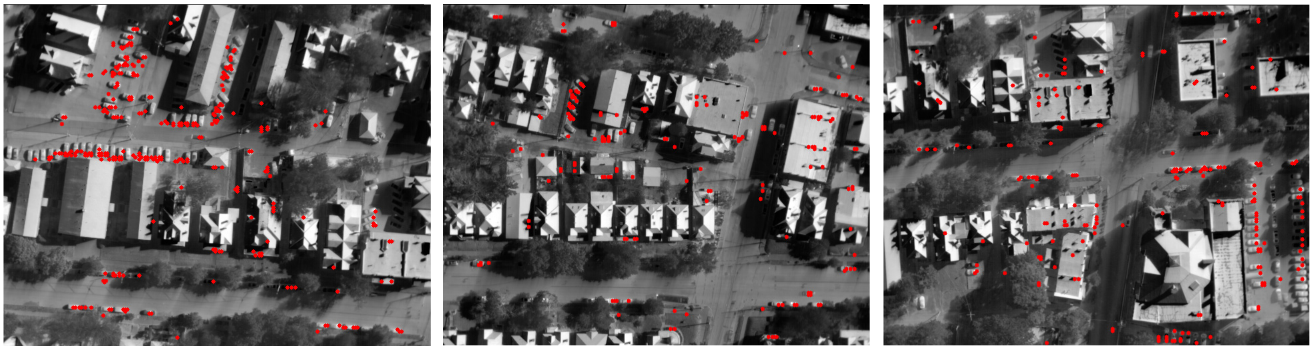


Figure 13. Sample detection results of RelCom triplet at FA rate of 10^{-4} on three different scenarios from the CSUAV dataset.

[5] C. Chang, C. Lin, "LIBSVM: a library for support vector machines," Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm, 2001. 6

[6] N. Dalal, B. Triggs, "Histograms of oriented gradients for human detection", CVPR, 2005. 1

[7] P. Dollár, Z. Tu, H. Tao, S. Belongie, "Feature mining for image classification", CVPR, 2007. 1, 7

[8] G. Duan, C. Huang, H. Ai, S. Lao, "Boosting associated pairing comparison features for pedestrian detection", Workshop on Visual Surveillance, 2009. 2, 3

[9] R. Fergus, P. Perona, A. Zisserman, "Object class recognition by unsupervised scale-invariant learning", CVPR, 2003. 1

[10] Y. Freund, R. E. Schapire, "A decision-theoretic generalization of online learning and an application to boosting", *Jrnl. of computer and system sciences*, 55(1):119–139, 1997. 4

[11] C. Huang, H. Ai, Y. Li, S. Lao, "Learning sparse features in granular space for multiview face detection", FGR, 2006. 2, 3

[12] J. F. Khan, M. S. Alam, "Target detection in cluttered forward-looking infrared imagery", *Opt. Eng*, 44(7), 2005. 2

[13] A. Kotcz, X. Sun, J. Kalita, "Efficient handling of high-dimensional feature spaces by randomised classifier ensembles", ACM SIGKDD, 2002. 1

[14] D. Lowe, "Distinctive image features from scale-invariant keypoints", *IJCV*, 60(2):91–110, 2004. 1

[15] S. Lucas, A. Amiri, "Statistical syntactic methods for high-performance OCR", *IEE Proc. Vision, Image and Signal Processing*, 143(1):23–30, 1996. 1

[16] K. Mikolajczyk, B. Leibe, B. Schiele, "Multiple object class detection with a generative model", CVPR, 2006. 1

[17] U. Braga-Neto, M. Choudhary, J. Goutsias, "Automatic target detection and tracking in forward-looking infrared image sequences using morphological connected operators, *Journal of Electronic Imaging*, 13(4):802–813, 2004. 2

[18] A. Opelt, M. Fussenegger, A. Pinz, P. Auer, "Weak hypotheses and boosting for generic object detection and recognition", ECCV, 2004. 1

[19] P. Papageorgiou, T. Poggio, "A trainable system for object detection", *IJCV*, 38(1):15–33, 2000. 1

[20] H. Sagha, S. Kasei, E. Enayati, M. Dehghani, "Finding sparse features in face detection using genetic algorithms", IEEE Intl conf. Computational Cybernetics, 2008. 2

[21] O. Tuzel, F. Porikli, P. Meer, "Pedestrian Detection via Classification on Riemannian Manifolds", *IEEE PAMI*, 30(10):1713–1727, 2008. 1

[22] P. Viola, M. Jones, "Rapid object detection using a boosted cascade of simple features", CVPR, 2001. 5

[23] B. Wu, H. Ai, C. Huang, S. Lao, "Fast rotation invariant multi-view face detection based on real AdaBoost", FGR, 2004. 2

[24] J. Yuan, J. Luo, Y. Wu, "Mining compositional features for boosting", CVPR, 2008. 1, 2

[25] L. Zhang, B. Wu, R. Nevatia, "Pedestrian Detection in Infrared Images based on Local Shape Features," OTCBVS, 2007. 2