Correlation Filters with Limited Boundaries

Hamed Kiani Galoogahi
Istituto Italiano di Tecnologia
Genova, Italy
hamed.kiani@iit.it

Terence Sim
National University of Singapore
Singapore
tsim@comp.nus.edu.sg

Simon Lucey
Carnegie Mellon University
Pittsburgh, USA
slucey@cs.cmu.edu

Abstract

Correlation filters take advantage of specific properties in the Fourier domain allowing them to be estimated efficiently: $\mathcal{O}(ND\log D)$ in the frequency domain, versus $\mathcal{O}(D^3 + ND^2)$ spatially where D is signal length, and N is the number of signals. Recent extensions to correlation filters, such as MOSSE, have reignited interest of their use in the vision community due to their robustness and attractive computational properties. In this paper we demonstrate, however, that this computational efficiency comes at a cost. Specifically, we demonstrate that only $\frac{1}{D}$ proportion of shifted examples are unaffected by boundary effects which has a dramatic effect on detection/tracking performance. In this paper, we propose a novel approach to correlation filter estimation that: (i) takes advantage of inherent computational redundancies in the frequency domain, (ii) dramatically reduces boundary effects, and (iii) is able to implicitly exploit all possible patches densely extracted from training examples during learning process. Impressive object tracking and detection results are presented in terms of both accuracy and computational efficiency.

1. Introduction

Correlation between two signals is a standard approach to feature detection/matching. Correlation touches nearly every facet of computer vision from pattern detection to object tracking. Correlation is rarely performed naively in the spatial domain. Instead, the fast Fourier transform (FFT) affords the efficient application of correlating a desired template/filter with a signal.

Correlation filters, developed initially in the seminal work of Hester and Casasent [15], are a method for learning a template/filter in the frequency domain that rose to some prominence in the 80s and 90s. Although many variants have been proposed [15, 18, 20, 19], the approach's central tenet is to learn a filter, that when correlated with a set of training signals, gives a desired response, e.g. Figure 1 (b). Like correlation, one of the central advantages of the approach is that it attempts to learn the filter in the frequency domain due to the efficiency of correlation in that domain.

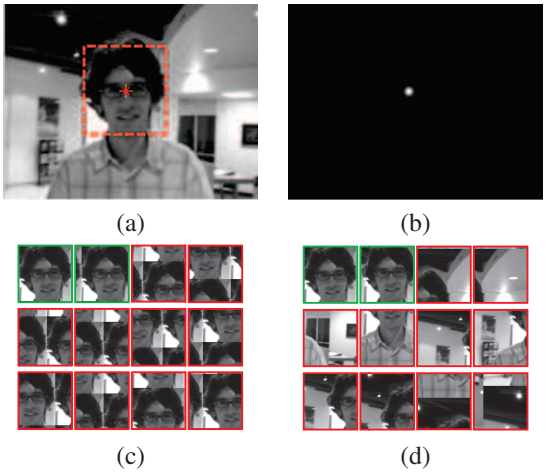


Figure 1. (a) Defines the example of fixed spatial support within the image from which the peak correlation output should occur. (b) The desired output response, based on (a), of the correlation filter when applied to the entire image. (c) A subset of patch examples used in a canonical correlation filter where green denotes a non-zero correlation output, and red denotes a zero correlation output in direct accordance with (b). (d) A subset of patch examples used in our proposed correlation filter. Note that our proposed approach uses all possible patches stemming from different parts of the image, whereas the canonical correlation filter simply employs circular shifted versions of the same single patch. The central dilemma in this paper is how to perform (d) efficiently in the Fourier domain. The two last patches of (d) show that $\frac{D-1}{T}$ patches near the image border are affected by circular shift in our method which can be greatly diminished by choosing $D << T$, where D and T indicate the length of the vectorized face patch in (a) and the whole image in (a), respectively.

Interest in correlation filters has been reignited in the vision world through the recent work of Bolme et al. [5] on Minimum Output Sum of Squared Error (MOSSE) correlation filters for object detection and tracking. Bolme et al.'s work was able to circumvent some of the classical problems

with correlation filters and performed well in tracking under changes in rotation, scale, lighting and partial occlusion. A central strength of the correlation filter is that it is extremely efficient in terms of both memory and computation.

The Problem: An unconventional interpretation of a correlation filter, is that of a discriminative template that has been estimated from an unbalanced set of "real-world" and "synthetic" examples. These synthetic examples are created through the application of a circular shift on the real-world examples, and are supposed to be representative of those examples at different translational shifts. We use the term synthetic, as all these shifted examples are plagued by circular boundary effects and are not truly representative of the shifted example (see Figure 1(c)). As a result, the training set used for learning the template is extremely unbalanced with one real-world example for every $D-1$ synthetic examples (where D is the dimensionality of the examples). These boundary effects can dramatically affect the resulting performance of the estimated template [19]. Fortunately, these effects can be largely removed (see Section 2) if the correlation filter objective is slightly augmented, but has to be now solved in the spatial rather than frequency domains. Unfortunately, this shift to the spatial domain destroys the computational efficiency that make correlation filters so attractive. This is the challenge that this paper addresses.

Contribution: In this paper we make the following contributions:

- We propose a new correlation filter objective that can drastically reduce the number of examples in a correlation filter that are affected by boundary effects.
- We theoretically demonstrate, however, that solving this objective in closed form drastically decreases computational efficiency: $\mathcal{O}(D^3 + ND^2)$ versus $\mathcal{O}(ND \log D)$ for the canonical objective where D is the length of the vectorized image and N is the number of examples.
- We demonstrate how this new objective can be efficiently optimized in an iterative manner through an Augmented Lagrangian Method (ALM) so as to take advantage of inherent redundancies in the frequency domain. The efficiency of this new approach is $\mathcal{O}([N+K]T \log T)$ where K is the number of iterations and T is the size of the search window.
- We show that our approach performs learning with dense sampling, meaning that it is capable of incorporating all possible patches stemmed from different parts of training images with a constant amount of memory regardless to the number of training images and patches (Figure 1 (d)).

Related Work: Bolme et al. [5] recently proposed an extension to traditional correlation filters referred to as Minimum Output Sum of Squared Error (MOSSE) filters. This approach has proven invaluable for many object tracking tasks, outperforming state of the art methods such as [2, 23] at 2010. What made the approach of immediate interest in the vision community was the dramatically faster frame rates than current state of the art (600 fps versus 30 fps). A strongly related method to MOSSE was also proposed by Bolme et al. [6] for object detection/localization referred to as Average of Synthetic Exact Filters (ASEF) which also reported superior performance to state of the art. A full discussion on other correlation filters such as Optimal Tradeoff Filters (OTF) [21], Unconstrained MACE (UMACE) [24] filters, Multi-Channel Correlation Filters (MCCF) [16], Maximum Margin Correlation Filters (MMCF) [22], kernel MOSSE [13], etc. and their applications [17, 14, 4] are outside the scope of this paper. Readers are encouraged to inspect [19] for a full treatment on the topic.

Notation: Vectors are always presented in lower-case bold (e.g., $\mathbf{a}$), Matrices are in upper-case bold (e.g., $\mathbf{A}$) and scalars in italicized (e.g. a or A). $\mathbf{a}(i)$ refers to the ith element of the vector $\mathbf{a}$. All M-mode array signals shall be expressed in vectorized form $\mathbf{a}$. A M-mode convolution operation is represented as the $*$ operator. One can express a M-dimensional discrete circular shift $\Delta\boldsymbol{\tau}$ to a vectorized M-mode matrix $\mathbf{a}$ through the notation $\mathbf{a}[\Delta\boldsymbol{\tau}]$. The matrix $\mathbf{I}$ denotes a $D \times D$ identity matrix and $\mathbf{1}$ denotes a D dimensional vector of ones. A $\hat{}$ applied to any vector denotes the M-mode Discrete Fourier Transform (DFT) of a vectorized M-mode matrix signal $\mathbf{a}$ such that $\hat{\mathbf{a}} \leftarrow \mathcal{F}(\mathbf{a}) = \sqrt{D}\mathbf{F}\mathbf{a}$. Where $\mathcal{F}()$ is the Fourier transforms operator and $\mathbf{F}$ is the orthonormal $D \times D$ matrix of complex basis vectors for mapping to the Fourier domain for any D dimensional vectorized image/signal. Additionally, we take advantage of the fact that $\text{diag}(\hat{\mathbf{h}})\hat{\mathbf{a}} = \hat{\mathbf{h}} \circ \hat{\mathbf{a}}$, where $\circ$ represents the Hadamard product, and diag() is an operator that transforms a D dimensional vector into a $D \times D$ dimensional diagonal matrix. The role of filter $\hat{\mathbf{h}}$ or signal $\hat{\mathbf{a}}$ can be interchanged with this property. Any transpose operator $^\top$ on a complex vector or matrix in this paper additionally takes the complex conjugate in a similar fashion to the Hermitian adjoint [19]. The operator conj($\hat{\mathbf{a}}$) applies the complex conjugate to the complex vector $\hat{\mathbf{a}}$.

2. Correlation Filters

Due to the efficiency of correlation in the frequency domain, correlation filters have canonically been posed in the frequency domain. There is nothing, however, stopping one (other than computational expense) from expressing a correlation filter in the spatial domain. In fact, we argue that viewing a correlation filter in the spatial domain can give us

crucial insights into fundamental problems in current correlation filter methods.

MOSSE correlation filter [5] can be expressed in the spatial domain as solving a ridge regression problem,

$$E(\mathbf{h}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{D} ||\mathbf{y}_i(j) - \mathbf{h}^\top \mathbf{x}_i[\Delta\boldsymbol{\tau}_j]||_2^2 + \frac{\lambda}{2}||\mathbf{h}||_2^2 \quad (1)$$

where $\mathbf{y}_i \in \mathbb{R}^D$ is the desired response for the i-th observation $\mathbf{x}_i \in \mathbb{R}^D$ and λ is a regularization term. $\mathbb{C} = [\Delta\boldsymbol{\tau}_1, \ldots, \Delta\boldsymbol{\tau}_D]$ represents the set of all circular shifts for a signal of length D. Bolme et al. advocated the use of a 2D Gaussian of small variance (2-3 pixels) for $\mathbf{y}_i$ centered at the location of the object (typically the centre of the image patch). The solution to this objective becomes,

$$\mathbf{h} = \mathbf{H}^{-1}\sum_{i=1}^{N}\sum_{j=1}^{D} \mathbf{y}_i(j)\mathbf{x}_i[\Delta\boldsymbol{\tau}_j] \quad (2)$$

where,

$$\mathbf{H} = \lambda\mathbf{I} + \sum_{i=1}^{N}\sum_{j=1}^{D} \mathbf{x}_i[\Delta\boldsymbol{\tau}_j]\mathbf{x}_i[\Delta\boldsymbol{\tau}_j]^\top \quad (3)$$

Solving a correlation filter in the spatial domain quickly becomes intractable as a function of the signal length D, as the cost of solving Equation 2 becomes $\mathcal{O}(D^3 + ND^2)$.

Putting aside, for now, the issue of computational cost, the correlation filter objective described in Equation 1 produces a filter that is particularly sensitive to misalignment in translation. A highly undesirable property when attempting to detect or track an object in terms of translation. This sensitivity is obtained due to the circular shift operator $\mathbf{x}[\Delta\boldsymbol{\tau}]$, where $\Delta\boldsymbol{\tau} = [\Delta x, \Delta y]^\top$ denotes the 2D circular shift in x and y. It has been well noted in correlation filter literature [19] that this circular-shift alone tends to produce filters that do not generalize well to other types of appearance variation (e.g. illumination, viewpoint, scale, rotation, etc.). This generalization issue can be somewhat mitigated through the judicious choice of non-zero regularization parameter λ, and/or through the use of an ensemble $N > 1$ of training observations that are representative of the type of appearance variation one is likely to encounter.

2.1. Boundary Effects

A deeper problem with the objective in Equation 1, however, is that the shifted image patches $\mathbf{x}[\Delta\boldsymbol{\tau}]$ at all values of $\Delta\boldsymbol{\tau} \in \mathbb{C}$, except where $\Delta\boldsymbol{\tau} = \mathbf{0}$ (no shift), are not representative of image patches one would encounter in a normal correlation operation (Figure 1(c)). In signal-processing, one often refers to this as the *boundary effect*. One simple way to circumvent this problem spatially is to allow the training signal $\mathbf{x} \in \mathbb{R}^T$ to be a larger size than the filter $\mathbf{h} \in \mathbb{R}^D$ such that $T > D$. Through the use of a $D \times T$ masking matrix $\mathbf{P}$ one can reformulate Equation 1 as,

$$E(\mathbf{h}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{T} ||\mathbf{y}_i(j) - \mathbf{h}^\top \mathbf{P}\mathbf{x}_i[\Delta\boldsymbol{\tau}_j]||_2^2 + \frac{\lambda}{2}||\mathbf{h}||_2^2 \quad (4)$$

The masking matrix $\mathbf{P}$ of ones and zeros encapsulates what part of the signal should be active/inactive, Figure 2. The central benefit of this augmentation in Equation 4 is the dramatic increase in the proportion of examples unaffected by boundary effects ($\frac{T-D+1}{T}$ instead of $\frac{1}{D}$ in canonical correlation filters). From this insight it becomes clear that if one chooses $T >> D$ then boundary effects become greatly diminished (Figure 1(d)). The computational cost $\mathcal{O}(D^3 + NTD)$ of solving this objective is only slightly larger than the cost of Equation 1, as the role of $\mathbf{P}$ in practice can be accomplished efficiently through a lookup table.

It is clear in Equation 4, that boundary effects could be removed completely by summing over only a $T - D + 1$ subset of all the T possible circular shifts. However, as we will see in the following section such a change along with the introduction of $\mathbf{P}$ is not possible if we want to solve this objective efficiently in the frequency domain.

2.2. Efficiency in the Frequency Domain

It is well understood in signal processing that circular convolution in the spatial domain can be expressed as a Hadamard product in the frequency domain [19]. This allows one to express the objective in Equation 1 more succinctly and equivalently as,

$$\begin{aligned} E(\hat{\mathbf{h}}) &= \frac{1}{2}\sum_{i=1}^{N} ||\hat{\mathbf{y}}_i - \hat{\mathbf{x}}_i \circ \text{conj}(\hat{\mathbf{h}})||_2^2 + \frac{\lambda}{2}||\hat{\mathbf{h}}||_2^2 \quad (5) \\ &= \frac{1}{2}\sum_{i=1}^{N} ||\hat{\mathbf{y}}_i - \text{diag}(\hat{\mathbf{x}}_i)^\top \hat{\mathbf{h}}||_2^2 + \frac{\lambda}{2}||\hat{\mathbf{h}}||_2^2 \ . \end{aligned}$$

where $\hat{\mathbf{h}}, \hat{\mathbf{x}}, \hat{\mathbf{y}}$ are the Fourier transforms of $\mathbf{h}, \mathbf{x}, \mathbf{y}$. The complex conjugate of $\hat{\mathbf{h}}$ is employed to ensure the operation is correlation not convolution. The equivalence between Equations 1 and 5 also borrows heavily upon another well known property from signal processing namely, Parseval's theorem which states that $\mathbf{x}_i^\top \mathbf{x}_j = D^{-1}\hat{\mathbf{x}}_i^\top \hat{\mathbf{x}}_j \quad \forall i, j, \quad$ where $\mathbf{x} \in \mathbb{R}^D$. The solution to Equation 5 becomes

$$\begin{aligned} \hat{\mathbf{h}} &= [\text{diag}(\hat{\mathbf{s}}_{xx}) + \lambda\mathbf{I}]^{-1}\sum_{i=1}^{N} \text{diag}(\hat{\mathbf{x}}_i)\hat{\mathbf{y}}_i \quad (6) \\ &= \hat{\mathbf{s}}_{xy} \circ^{-1} (\hat{\mathbf{s}}_{xx} + \lambda\mathbf{1}) \end{aligned}$$

where $\circ^{-1}$ denotes element-wise division, and

$$\hat{\mathbf{s}}_{xx} = \sum_{i=1}^{N} \hat{\mathbf{x}}_i \circ \text{conj}(\hat{\mathbf{x}}_i) \quad \& \quad \hat{\mathbf{s}}_{xy} = \sum_{i=1}^{N} \hat{\mathbf{y}}_i \circ \text{conj}(\hat{\mathbf{x}}_i) \quad (7)$$

are the average auto-spectral and cross-spectral energies respectively of the training observations. The solution for $\hat{\mathbf{h}}$ in Equations 1 and 5 are identical (other than that one is posed in the spatial domain, and the other is in the frequency domain). The power of this method lies in its computational efficiency. In the frequency domain a solution to $\hat{\mathbf{h}}$ can be found with a cost of $\mathcal{O}(ND \log D)$. The primary cost is associated with the DFT on the ensemble of training signals $\{\mathbf{x}_i\}_{i=1}^N$ and desired responses $\{\mathbf{y}_i\}_{i=1}^N$.

3. Our Approach

A problem arises, however, when one attempts to apply the same Fourier insight to the augmented spatial objective in Equation 4 for computational efficiency. Equation 4 can be expressed in the Fourier domain as:

$$E(\mathbf{h}) = \frac{1}{2}\sum_{i=1}^{N} ||\hat{\mathbf{y}}_i - \text{diag}(\hat{\mathbf{x}}_i)^\top \sqrt{D}\mathbf{F}\mathbf{P}^\top \mathbf{h}||_2^2 + \frac{\lambda}{2}||\mathbf{h}||_2^2 \tag{8}$$

Unfortunately, since we are enforcing a spatial constraint $\mathbf{P}^\top$ on $\mathbf{h}$ the efficiency of this objective balloons to $\mathcal{O}(D^3 + ND^2)$ as $\mathbf{h}$ *must* be solved in the spatial domain.

3.1. Augmented Lagrangian

Our proposed approach for solving Equation 8 involves the introduction of an auxiliary variable $\hat{\mathbf{g}}$. In this case, Equation 8 can be identically expressed as:

$$\begin{aligned} E(\mathbf{h}, \hat{\mathbf{g}}) \quad &= \quad \frac{1}{2}\sum_{i=1}^{N} ||\hat{\mathbf{y}}_i - \text{diag}(\hat{\mathbf{x}}_i)^\top \hat{\mathbf{g}}||_2^2 + \frac{\lambda}{2}||\mathbf{h}||_2^2 \\ &\text{s.t.} \quad \hat{\mathbf{g}} = \sqrt{D}\mathbf{F}\mathbf{P}^\top \mathbf{h} \ . \end{aligned} \tag{9}$$

We propose to handle the introduced equality constraints through an Augmented Lagrangian Method (ALM) [7]. The augmented Lagrangian of our proposed objective can be formed as,

$$\begin{aligned} \mathcal{L}(\hat{\mathbf{g}}, \mathbf{h}, \hat{\boldsymbol{\zeta}}) \quad &= \quad \frac{1}{2}\sum_{i=1}^{N} ||\hat{\mathbf{y}}_i - \text{diag}(\hat{\mathbf{x}}_i)^\top \hat{\mathbf{g}}||_2^2 + \frac{\lambda}{2}||\mathbf{h}||_2^2 \\ &+ \quad \hat{\boldsymbol{\zeta}}^\top (\hat{\mathbf{g}} - \sqrt{D}\mathbf{F}\mathbf{P}^\top \mathbf{h}) \\ &+ \quad \frac{\mu}{2}||\hat{\mathbf{g}} - \sqrt{D}\mathbf{F}\mathbf{P}^\top \mathbf{h}||_2^2 \end{aligned} \tag{10}$$

where μ is the penalty factor that controls the rate of convergence of the ALM, and $\hat{\boldsymbol{\zeta}}$ is the Fourier transform of the Lagrangian vector needed to enforce the newly introduced equality constraint in Equation 9. ALMs are not new to learning and computer vision, and have recently been used to great effect in a number of applications [7, 8]. Specifically, the Alternating Direction Method of Multipliers (ADMMs) has provided a simple but powerful algorithm that is well suited to distributed convex optimization for large learning and vision problems. A full description of ADMMs is outside the scope of this paper (readers are encouraged to inspect [7] for a full treatment and review), but they can be loosely interpreted as applying a Gauss-Seidel optimization strategy to the augmented Lagrangian objective. Such a strategy is advantageous as it often leads to extremely efficient subproblem decompositions. A full description of our proposed algorithm can be seen in Algorithm 1. We detail each of the subproblems as follows:

Subproblem g:

$$\begin{aligned} \hat{\mathbf{g}}^* \quad &= \quad \arg\min \mathcal{L}(\hat{\mathbf{g}}; \hat{\mathbf{h}}, \hat{\boldsymbol{\zeta}}) \\ &= \quad (\hat{\mathbf{s}}_{xy} + \mu\hat{\mathbf{h}} - \hat{\boldsymbol{\zeta}}) \circ^{-1} (\hat{\mathbf{s}}_{xx} + \mu\mathbf{1}) \end{aligned} \tag{11}$$

where $\hat{\mathbf{h}} = \sqrt{D}\mathbf{F}\mathbf{P}^\top \mathbf{h}$. In practice $\hat{\mathbf{h}}$ can be estimated extremely efficiently by applying a FFT to $\mathbf{h}$ padded with zeros implied by the $\mathbf{P}^\top$ masking matrix.

Subproblem h:

$$\begin{aligned} \mathbf{h}^* \quad &= \quad \arg\min \mathcal{L}(\mathbf{h}; \mathbf{g}, l) \\ &= \quad (\mu + \frac{\lambda}{\sqrt{D}})^{-1}(\mu\mathbf{g} + l) \end{aligned} \tag{12}$$

where $\mathbf{g} = \frac{1}{\sqrt{D}}\mathbf{P}\mathbf{F}^\top \hat{\mathbf{g}}$ and $l = \frac{1}{\sqrt{D}}\mathbf{P}\mathbf{F}^\top \hat{\boldsymbol{\zeta}}$. In practice both $\mathbf{g}$ and l can be estimated extremely efficiently by applying an inverse FFT and then applying the lookup table implied by the masking matrix $\mathbf{P}$.

Lagrange Multiplier Update:

$$\hat{\boldsymbol{\zeta}}^{(i+1)} \leftarrow \hat{\boldsymbol{\zeta}}^{(i)} + \mu(\hat{\mathbf{g}}^{(i+1)} - \hat{\mathbf{h}}^{(i+1)}) \tag{13}$$

where $\hat{\mathbf{g}}^{(i+1)}$ and $\hat{\mathbf{h}}^{(i+1)}$ are the current solutions to the above subproblems at iteration $i+1$ within the iterative ADMM.

Choice of μ: A simple and common [7] scheme for selecting μ is the following

$$\mu^{(i+1)} = \min(\mu_{\max}, \beta\mu^{(i)}) \ . \tag{14}$$

We found experimentally $\mu^{(0)} = 10^{-2}$, $\beta = 1.1$ and $\mu_{\max} = 20$ to perform well.

3.2. Learning with Dense Sampling

The advantage of learning detectors with dense sampling strategy has been fully explored in recent approaches [25, 13, 12]. These approaches intend to train a classier/detector by exploiting all possible negative patches which can be extracted from training samples (e.g. 10^4 patches of a 100×100 training image), as an alternative to Hard Negative Mining (HNM). Adopting a dense sampling strategy within an LDA framework, Hariharan et al. [12] demonstrated very competitive detection performance compared to HNM with superior memory usage and computations.

Inspecting Equation 4 one can see that the circular shift operator $\Delta\boldsymbol{\tau}$ returns all shifted versions of the (vectorized) training image $\mathbf{x}_i$, $\{\mathbf{x}_i[\Delta\boldsymbol{\tau}_j]\}_{j=1}^{T}$, where T is the length of $\mathbf{x}_i$. Note that the shifted images in our approach are implicitly generated by the circular shift property of convolution/correlation operation in the Fourier domain, and in practice, we do not need to directly use the shift operator $\Delta\boldsymbol{\tau}$ to generate the shifted images (see Equation 8). By applying the masking matrix $\mathbf{P}$ on each shifted image, $\mathbf{P}\mathbf{x}_i[\Delta\boldsymbol{\tau}_j]$ in Equation 4, we indeed select (crop) a patch (sub image) from $\mathbf{x}_i[\Delta\boldsymbol{\tau}_j]$ whose size is smaller than the size of $\mathbf{x}_i$ and is centered on the j^{th} location of the (vectorized) image/signal. This generates all T possible patches/samples of the training image $\mathbf{x}_i$, Figure 2. Since this dense sampling is embedded in the objective in Equation 4 and we learn correlation filters by optimizing this objective, the proposed approach is a dense sampling based learning technique.

3.3. Computational Cost

Inspecting Algorithm 1 the dominant cost per iteration of the ADMM optimization process is $\mathcal{O}(T \log T)$ for FFT. There is a per-computation cost (before the iterative component, steps 4 and 5) in the algorithm for estimating the auto- and cross-spectral energy vectors $\hat{\mathbf{s}}_{xx}$ and $\hat{\mathbf{s}}_{xy}$ respectively. This cost is $\mathcal{O}(NT \log T)$ where N refers to the number of training signals. Given that K represents the number of ADMM iterations the overall cost of the algorithm is therefore $\mathcal{O}([N + K]T \log T)$.

3.4. Memory Efficiency

Given N vectorized training images of length T, the memory usage of our approach to learn a correlation filter is $\mathcal{O}(T)$. This is the amount of memory required to compute the auto- and cross-spectral energy vectors $\hat{\mathbf{s}}_{xx}$ and $\hat{\mathbf{s}}_{xy}$ in steps 4 and 5, Equation 7. This means that the memory cost of the proposed approach is constant and independent of both the number of images and ADMM iterations. Moreover, as mentioned above, the dense sampling is embedded in our learning approach, meaning that no extra memory is required to load all possible patches of training images.

We emphasize this advantage by giving a practical example. Suppose that 100,000 100×100 of training images (in double precision) are given to learn a 50×50 object detector (template). In this case, our method uses $1000 \times 100 \times 100 = 10^7$ patches of size 50×50 to train the detector (by dense sampling) and this amounts only 0.02 MB storage to compute $\hat{\mathbf{s}}_{xx}$ and $\hat{\mathbf{s}}_{xy}$. On the other hand, learning a SVM classifier [26], which has been extensively employed for recognition tasks, incurs a memory cost linear to the number of patches [16]. Therefore, using 10^7 of 50×50 patches, storage belows out to an untenable 200 GB.

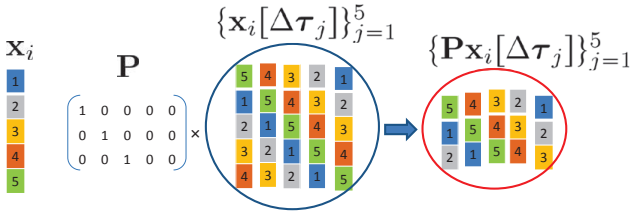


Figure 2. Dense sampling: (left) a vectorized signal $\mathbf{x}_i$ of length $T = 5$, (right) all possible sub-signals of $\mathbf{x}_i$ with length $D = 3$ obtained by multiplying all shifted versions of $\mathbf{x}_i$, $\{\mathbf{x}_i[\Delta\boldsymbol{\tau}_j]\}_{j=1}^{T}$ by mask matrix $\mathbf{P}$, $\{\mathbf{P}\mathbf{x}_i[\Delta\boldsymbol{\tau}_j]\}_{j=1}^{T}$. $\mathbf{P}$ is a $D \times T$ matrix.

Algorithm 1 Our approach using ADMMs

1: Intialize $\mathbf{h}^{(0)}$, $l^{(0)}$.
2: Pad with zeros and apply FFT: $\sqrt{D}\mathbf{F}\mathbf{P}^\top\mathbf{h}^{(0)} \rightarrow \hat{\mathbf{h}}^{(0)}$.
3: Apply FFT: $\sqrt{D}\mathbf{F}l^{(0)} \rightarrow \hat{\boldsymbol{\zeta}}^{(0)}$.
4: Estimate auto-spectral energy $\hat{\mathbf{s}}_{xx}$ using Eqn. (7).
5: Estimate cross-spectral energy $\hat{\mathbf{s}}_{xy}$ using Eqn. (7).
6: $i = 0$
7: **repeat**
8: Solve for $\hat{\mathbf{g}}^{(i+1)}$ using Eqn. (11), $\hat{\mathbf{h}}^{(i)}$ & $\hat{\boldsymbol{\zeta}}^{(i)}$.
9: Inverse FFT then crop: $\frac{1}{\sqrt{D}}\mathbf{P}\mathbf{F}^\top\hat{\mathbf{g}}^{(i+1)} \rightarrow \mathbf{g}^{(i+1)}$.
10: Inverse FFT then crop: $\frac{1}{\sqrt{D}}\mathbf{P}\mathbf{F}^\top\hat{\boldsymbol{\zeta}}^{(i+1)} \rightarrow l^{(i+1)}$.
11: Solve for $\mathbf{h}^{(i+1)}$ using Eqn. (12), $\mathbf{g}^{(i+1)}$ & $l^{(i)}$.
12: Pad and apply FFT: $\sqrt{D}\mathbf{F}\mathbf{P}^\top\mathbf{h}^{(i+1)} \rightarrow \hat{\mathbf{h}}^{(i+1)}$.
13: Update Lagrange multiplier vector Eqn. (13).
14: Update penalty factor Eqn. (14).
15: $i = i + 1$
16: **until** $\hat{\mathbf{g}}, \mathbf{h}, \hat{\boldsymbol{\zeta}}$ has converged

4. Experiments

4.1. Localization Performance

In the first experiment, we evaluated our method on the problem of eye localization, comparing with leading correlation filters in the literature, e.g. OTF [21], MACE [20], UMACE [24], ASEF [6], and MOSSE [5]. The CMU Multi-PIE face database [1] was used for this experiment, containing 900 frontal faces with neutral expression and normal illumination. We randomly selected 400 images for training and the reminder for testing. All images were cropped to have a same size of 128×128 with fixed coordinates of the left and the right eyes. The cropped images were power normalized to have a zero-mean and a standard deviation of 1.0.

We trained a 64×64 filter of the right eye using 64×64 cropped patches (centered upon the right eye) for the other methods, and full face images for our method ($T = 128 \times$

[1] http://www.multipie.org/

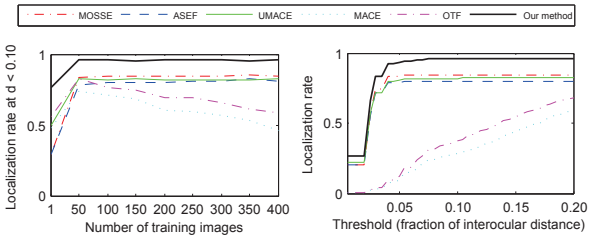
128 and $D = 64 \times 64$). Similar to ASEF and MOSSE, we defined the desired response as a 2D Gaussian function with an spatial variance of $s = 2$. Eye localization was performed by correlating the filters over the testing images followed by selecting the peak of the output as the predicted eye location. The eye localization was evaluated by the distance between the predicted and desired eye locations normalized by inter-ocular distance [6], $d = \frac{\|\mathbf{p}_r - \mathbf{m}_r\|_2}{\|\mathbf{m}_l - \mathbf{m}_r\|_2}$, where $\mathbf{m}_r$ and $\mathbf{m}_l$ respectively indicate the true coordinates of the right and left eye, and $\mathbf{p}_r$ is the predicted location of the right eye. A localization with normalized distance $d < th$ was considered as a successful localization. The threshold *th* was set to a fraction of inter-ocular distance.

The average of evaluations across 10 runs are depicted in Figure 3, where our method outperforms the other approaches across all thresholds and training set sizes. The accuracy of OTF and MACE declines by increasing the number of training images due to the over-fitting. During the experiment, we observed that the low performance of the UMACE, ASEF and MOSSE was mainly caused by wrong localizations of the left eye and the nose. This was much less in our method, since our filter was trained using all negative patches (dense sampling) collected form full face images. A visual depiction of the filters and their outputs can be seen in Figure 4. The Peak-to-Sidelobe Ratio (PSR) [5] values show that our method returns stronger correlation responses than the other filters.

Moreover, we examined the influence of T (the size of training images) on the performance of eye localization. for this, we used cropped patches of the right eye with varying sizes of $T = \{D, 1.5D, 2D, 2.5D, 3D, 3.5D, 4D\}$ to train filters of size $D = 32 \times 32$. In the case of $T = D$, training images are just the right eye patches, while in $T = 4D$ the training images are full face images. The result is illustrated in Figure 5(a), showing that the lowest performance obtained when $T = D$ and the localization rate improved by increasing the size of the training patches with respect to the filter size. The highest localization rate was obtained when $T = 4D$. The reason is that when $T = 4D$ (i) the portion of patches unaffected by boundary effects ($\frac{T-D+1}{T}$) is remarkably increases, and (ii) a huge set of negative patches (from nose, mouth, the left eye, etc.) are used for filter training that makes the filter fairly robust against wrong detections of background patches.

4.2. Runtime Performance

This experiment demonstrates the advantage of our approach to other iterative methods. Specifically, we compared our proposed approach against other methods in literature for learning filters efficiently using iterative methods. We compared our convergence performance with a steepest descent method [27] for optimizing our same objective. Results in Figure 6 represent: (a) time to converge as a function of the filter size and the number of training images, and (b) the number of iterations required to optimize the objective in Equation 8. In (a) one notices impressively how convergence performance is largely independent to the filter size and the number of images used during training. This can largely be attributed to the per-computation of the auto- and cross-spectral energies. As a result, iterations of the ADMM do not need to re-touch the training set, allowing our approach to dramatically outperform more naive iterative approaches such as [27] that needs to re-compute a set of convolutions in the spatial domain over each iteration of the training process. Similarly, in (b) one notices how relatively few iterations are required to achieve good convergence.



(a) (b)

Figure 3. Eye localization performance as a function of (a) number of training images, and (b) localization thresholds.

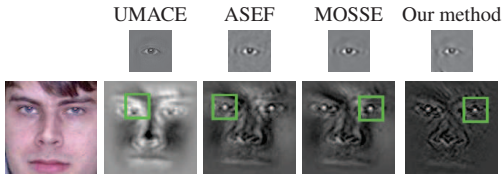
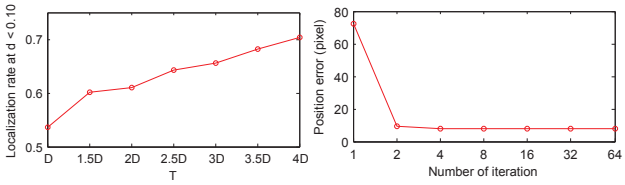
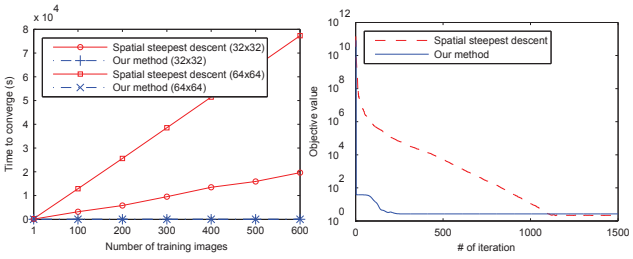


Figure 4. An example of eye localization. The outputs (bottom) are produced using 64×64 correlation filters (top). The green box represents the approximated location of the right eye. The peak strength (PSR) shows the sharpness of the output peak.



(a) (b)

Figure 5. (a) Localization rate as a function of training images size (T), the size of filter is $D = 32 \times 32$. (b) Tracking position error versus the number of ADMM iterations. We selected 2 iterations as a tradeoff between tracking performance and computation.



(a) (b)

Figure 6. Runtime performance of our method versus steepest descent method [27]. Our approach shows superior performance in terms of: (a) convergence speed to train two filters with different sizes and (b) the number of iterations required to converge.

4.3. Tracking Performance

Finally, we evaluated the proposed method for the task of real-time tracking on a sequence of commonly used test videos [23], comparing with the leading trackers in the literature [3, 11, 9, 10, 1, 23, 5, 13]. All of these methods were tuned by the parameters proposed in their reference papers. The desired response for a $m \times n$ target was defined as a 2D Gaussian with a variance of $\mathbf{s} = \sqrt{mn}/16$ [13]. The regularization parameter λ was set to 10^{-2}. We evaluated our method with different number of ADMM iterations $\{1, 2, 4, 8, 16, 32, 64\}$, as shown in Figure 5(b), and eventually selected two iterations (a tradeoff between precision and tracking speed) for our tracker. A track initialization process was employed for our approach and MOSSE, where eight random affine perturbations were used to initialize the first filter. We borrowed the online adaption from MOSSE [5] to adapt our filter at i^{th} frame using averaged auto-spectral and cross-spectral energies:

$$\begin{aligned} (\hat{\mathbf{s}}_{xx})^i &= \eta(\hat{\mathbf{x}}_i \circ \text{conj}(\hat{\mathbf{x}}_i)) + (1-\eta)(\hat{\mathbf{s}}_{xx})^{i-1} \\ (\hat{\mathbf{s}}_{xy})^i &= \eta(\hat{\mathbf{y}}_i \circ \text{conj}(\hat{\mathbf{x}}_i)) + (1-\eta)(\hat{\mathbf{s}}_{xy})^{i-1} \end{aligned} \quad (15)$$

where, η is the adaption rate. We practically found that $\eta = 0.025$ is appropriate for our method to quickly be adapted against pose change, scale, illumination, etc.

The tracking results are evaluated in Table 1 based on (i) percentage of frames where the predicted position is within 20 pixels of the ground truth (precision), (ii) average localization error in pixels, and (iii) tracking speed (*fps*), which are standard measures in tracking papers [3] [11] [9]. Our method averagely achieved the highest precision and the lowest localization error, followed by STRUCK. The reason is that our method incorporates visual information from all possible foreground (target) and background (non-target) patches to train the tracker (dense sampling, e.g. 10^4 patches from a 100×100 frame). While, due to computational constrains, the non-filter approaches such as STRUCK and MILTrack employ a handful of target and non-target patches (e.g. 10-20 patches) randomly collected around the estimated position of the object of interest [13].

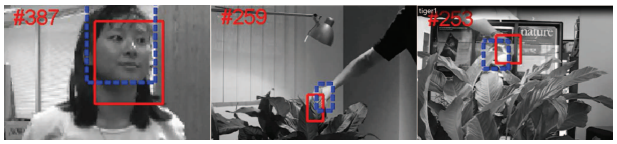


Figure 7. Failure examples of "Girl", "Coke Can" and "Tiger1" videos caused by severe pose changing and full occlusion.

Moreover, the accuracy of MOSSE and kernel-MOSSE is much less than our tracker due to the boundary effects, Figure 1(c). Besides, they do not use any background patch for filter learning. In terms of tracking speed, MOSSE outperformed the other methods by 600 *fps*. Our method obtained lower *fps* than MOSSE, due to its iterative manner. However, the speed of our tracker, 100 *fps*, is still appropriate for real-time tracking. The tracking results for some selected videos is shown in Figures 8, where our method shows higher precision for almost all thresholds in (a) and less drift per frames in (b). Figure 9 depicts some qualitative results in some frames. The extension of Figures 8 and 9 can be found in the supplemental materials.

Please note that the goal of this experiment is *not* showing the superiority of our approach over all the test videos. For instance, we obtained lower precision on "Tiger1" video (79%) compared to MILTrack (94%) and STRUCK (%95). This implies that same as the other correlation filter based trackers, our approach does not perform well on videos such as "Tiger1" with full occlusion and severe appearance changes, Figure 7. Indeed, we aimed to show that very competitive results can be achieved by our simple and fairly fast tracker, compared to the complicated and slow techniques were particularly tailored for object tracking.

5. Conclusions

A method for estimating a correlation filter is presented here that dramatically limits circular boundary effects while preserving many of the computational advantages of canonical frequency domain correlation filters. Moreover, we showed that the proposed approach implicitly learns correlation filters over an embedded dense sampling strategy which is inherited from the shift circular property of the convolution operation in the Frequency domain. This allows one to learn an effective detector/filter by exploiting a huge set of negative examples with very efficient memory cost which was shown to be independent of the number of training images and sampled patches. Our approach demonstrated superior empirical results for both object detection and real-time tracking compared to current state of the arts.

	MOSSE [5]	KMOSSE [13]	MILTrack [3]	STRUCK [11]	OAB(1) [9]	SemiBoost [10]	FragTrack [1]	Our method
FaceOcc1	{**1.00**, 7}	{**1.00**, **5**}	{0.75, 17}	{0.97, 8}	{0.22, 43}	{0.97, 7}	{0.94, 7}	{**1.00**, 8}
FaceOcc2	{0.74, 13}	{0.95, 8}	{0.42, 31}	{0.93, **7**}	{0.61, 21}	{0.60, 23}	{0.59, 27}	{ **0.97**, **7**}
Girl	{0.82, 14}	{0.44, 35}	{0.37, 29}	{**0.94**, **10**}	-	-	{0.53, 27}	{0.90, 12}
Sylv	{0.87, 7}	{**1.00**, 6}	{0.96, 8}	{0.95, 9}	{0.64, 25}	{0.69, 16}	{0.74, 25}	{**1.00**, **4**}
Tiger1	{0.61, 25}	{0.62, 25}	{0.94, 9}	{**0.95**, **9**}	{0.48, 35}	{0.44, 42}	{0.36, 39}	{0.79, 18}
David	{0.56, 14}	{0.50, 16}	{0.54, 18}	{0.93, 9}	{0.16, 49}	{0.46, 39}	{0.28, 72}	{**1.00**, **7**}
Cliffbar	{0.88, 8}	{0.97, 6}	{0.85, 12}	{0.44, 46}	{0.76, -}	-	{0.22, 39}	{ **1.00**, **5**}
Coke Can	{0.96, **7**}	{**1.00**, **7**}	{0.58, 17}	{0.97, **7**}	{0.45, 25}	{0.78, 13}	{0.15, 66}	{0.97, **7**}
Dollar	{**1.00**, **4**}	{**1.00**, **4**}	{**1.00**, 7}	{**1.00**, 13}	{0.67, 25}	{0.37, 67}	{0.40, 55}	{ **1.00**, 6}
Twinings	{0.48, 16}	{0.89, 11}	{0.76, 15}	{**0.99**, **7**}	{0.74, -}	-	{0.82, 14}	{**0.99**, 9}
mean	{0.80, 11}	{0.84, 12}	{0.72, 16}	{0.91, 12}	{0.53, 31}	{0.62, 29}	{0.51, 37}	{**0.97**, **8**}
fps	**600**	100	25	11	25	25	2	100

Table 1. The tracking performance is shown as a tuple of {*precision within 20 pixels*, *average position error in pixels*}, where our method achieved the best performance over 8 of 10 videos. The best *fps* was obtained by MOSSE. Our method obtained a real-time tacking speed of 100 *fps* using two iterations of ADMM. The best result for each video is highlighted in bold.

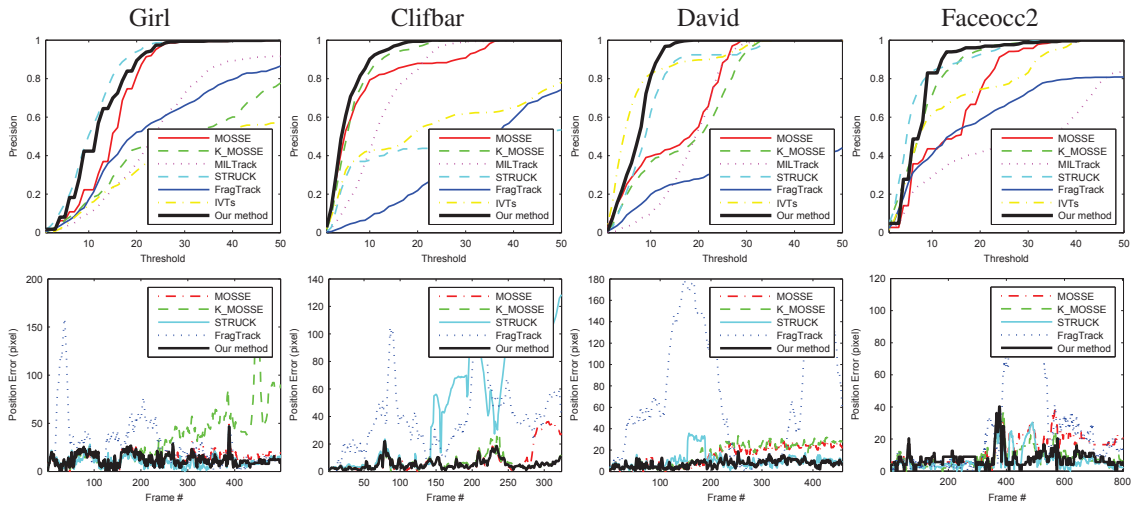


Figure 8. Tracking results for selected videos, (a) precision versus the thresholds, and (b) position error per frame.

Figure 9. Tracking results of our method over two videos with challenging variations of pose, scale, illumination and partial occlusion. The blue (dashed) and red boxes respectively represent the ground truth and the positions predicted by our method. For each frame, we illustrate the target, trained filter and correlation output.

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments based tracking using the integral histogram. In *CVPR*, 2006.

[2] B. Babenko, M. H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, 2009.

[3] B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *PAMI*, 33(8):1619–1632, 2011.

[4] V. N. Boddeti, T. Kanade, and B. Kumar. Correlation filters for object alignment. In *CVPR, 2013*, pages 2291–2298. IEEE, 2013.

[5] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, 2010.

[6] D. S. Bolme, B. A. Draper, and J. R. Beveridge. Average of synthetic exact filters. In *CVPR*, 2009.

[7] S. Boyd. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, 3:1–122, 2010.

[8] A. Del Bue, J. Xavier, L. Agapito, and M. Paladini. Bilinear modelling via augmented lagrange multipliers (BALM). *PAMI*, 34(8):1–14, Dec. 2011.

[9] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via on-line boosting. In *BMVC*, 2006.

[10] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV*. Springer, 2008.

[11] S. Hare, A. Saffari, and P. H. Torr. Struck: Structured output tracking with kernels. In *ICCV*, 2011.

[12] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *ECCV 2012*, pages 459–472. Springer, 2012.

[13] J. F. Henriques, R. Caseiro, P. Martines, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *ECCV*, 2012.

[14] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *PAMI*, 2014.

[15] C. F. Hester and D. Casasent. Multivariant technique for multiclass pattern recognition. *Appl. Opt.*, 19(11):1758–1761, 1980.

[16] H. Kiani, T. Sim, and S. Lucey. Multi-channel correlation filters. In *ICCV*, 2013.

[17] H. Kiani, T. Sim, and S. Lucey. Multi-channel correlation filters for human action recognition. In *ICIP*, 2014.

[18] B. V. K. V. Kumar. Minimum-variance synthetic discriminant functions. *J. Opt. Soc. Am. A*, 3(10):1579–1584, 1986.

[19] B. V. K. V. Kumar, A. Mahalanobis, and R. D. Juday. *Correlation Pattern Recognition*. Cambridge University Press, 2005.

[20] A. Mahalanobis, B. V. K. V. Kumar, and D. Casasent. Minimum average correlation energy filters. *Appl. Opt.*, 26(17):3633–3640, 1987.

[21] P. Refregier. Optimal trade-off filters for noise robustness, sharpness of the correlation peak, and horner efficiency. *Optics Letters*, 16:829–832, 1991.

[22] A. Rodriguez, V. N. Boddeti, B. V. Kumar, and A. Mahalanobis. Maximum margin correlation filter: A new approach for localization and classification. *TIP*, 22(2):631–643, 2013.

[23] D. Ross, J. Lim, R. Lin, and M. Yang. Incremental learning for robust visual tracking. *IJCV*, 77(1):125–141, 2008.

[24] M. Savvides and B. V. K. V. Kumar. Efficient design of advanced correlation filters for robust distortion-tolerant face recognition. In *AVSS*, pages 45–52, 2003.

[25] J. Valmadre, S. Sridharan, and S. Lucey. Learning detectors quickly with stationary statistics. In *ACCV*, 2014.

[26] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[27] M. Zeiler, D. Krishnan, and G. Taylor. Deconvolutional networks. *CVPR*, 2010.