MULTI-VIEW OBJECT DETECTION BY CLASSIFIER INTERPOLATION

Xiaobai Liu†,‡, Haifeng Gong†,§, Shuicheng Yan‡, Hai Jin†
†Huazhong University of Science and Technology, Wuhan, China
‡Lotushill Research Institute, EZhou, China
§University of California Los Angeles, Los Angeles, USA
¶National University of Singapore, Singapore

ABSTRACT

In this paper, we propose a novel solution for multi-view object detection. Given a set of training examples at different views, we select examples at a few key views and train one classifier for each of them. Then classifiers for more intermediate views can be interpolated from key views. The interpolation is conducted on the weights and positions of features, under the assumption that they can all be expressed as functions of view angle. Finally, the learned and interpolated classifiers are combined into a boosting framework to construct a multi-view classifier to further validate the effectiveness of the interpolation. Experiments of interpolated single view classifier and combined multi-view classifier are conducted on car data sets and their performances are compared to corresponding learned classifiers. The results illustrate that the interpolated classifiers give comparable performance as classifiers learned from data, and that the combined classifiers give similar results as their learned counterparts.

Index Terms— Classifier Interpolation; Multi View; Object Template; Active Basis

1. INTRODUCTION

The appearances of objects are heavily affected by view angles. In the literature, multi-view recognition tasks is mainly handled in three ways - CAD model, explicit mixture model and implicit mixture model. i) CAD-based methods. An explicit 3-D model of a target is generated and subsequently used in target matching. The basic idea is to estimate the pose of the CAD model, which are then used to matches with the queried image. These algorithms are usually limited to the poor ability of processing the flexible shape and dimension changes of objects. Therefore, recent literature prefers ‘divide-and-conquer’ strategy to avoid explicit 3D modeling: several object models are built, each describes objects in a range of view. This strategy leads to mixture appearance models. ii) Explicit mixture appearance model[4, 3]. These methods usually first cluster multi-view/pose training data into different categories, and then combine the separately trained binary classifiers into a multi-class classifier. iii) Implicit mixture appearance model. In these methods [6, 2], the training samples are not provided with the view knowledge. Thus, training classifiers requires an additional clustering procedure for discovering the view labels. Besides the three categories, there are a few works considering combination of appearance models with rough 3D information. For example, [1] introduce an approach to accurately detect and segment cars in various views. In training, they exploited a rough 3D object model to learn physically localized plane appearances. [8] described a approach to automatically match the target based on a view morphine database constructed by our multi-view morphine algorithm.

In this paper, we propose a novel framework for multi-view object detection. See Figure 1, given a set of training examples at different views, we select examples at a few key views and train one classifier for each of them. Then classifiers for more intermediate views can be interpolated from key views. The interpolation is conducted on the weights and positions of features, under the assumption that they can all be expressed as functions of view angle. Finally, the learned and interpolated classifiers are combined into a boosting framework to construct a multi-view classifier to further validate the effectiveness of the interpolation.

The contribution of this work lies in two aspects. 1) The classifier interpolation framework which can predict classifiers for unseen object view. 2) The Active Haar features which can produce more intuitive classifier.
2. ACTIVE HAAR CLASSIFIER

A classifier is commonly composed of several features, weak classifiers or support vectors. To interpolate new classifiers from known classifiers, an important requirement is that the features and parameters learned at different cases should be compatible, or it is not feasible to interpolate on them. Although classifier interpolation is not tricky idea, as far as our best knowledge, it has not appeared in the literature. The reason lies in the lack of training algorithms that can produce very meaningful features which correspond to object parts exactly.

Recently, Wu et al. in [7] developed an active sketch algorithm which can learn object templates with very intuitive features from training images of various categories. In their generative model, the deformable template consists of a set of Gabor wavelet elements at different locations and orientations. These elements are allowed to slightly perturb their locations and orientations before they are linearly combined to generate each individual training or testing example. This active basis model can be learned from training image patches by the shared pursuit algorithm, which sequentially selects the elements of the active basis from a large dictionary of Gabor wavelets.

![Active Basis](image1.png)

(a) Active Basis  (b) Active Haar

Fig. 2. Active Haar can produce very meaningful templates.

They provided many variants of their algorithm, and in this work, we adopt the maximum correlation variant. Let \( I_m, m = 1, ..., M \), be the \( m \)-th training image, \( B_{i,\bar{x},\theta} \) be the feature selected from a feature bank \( \Omega \), with type \( i \), position \( \bar{x} \) and orientation \( \theta \). If \( B_{i,\bar{x},\theta} \) and \( B_{i,\bar{x}',\theta'} \) are features of same type with a little position and orientation perturbation, we denote them as \((\bar{x}, \theta) \approx (\bar{x}', \theta')\). Let \( < I_m, B_{l,\bar{x},\theta} > \) be the filter response, and \( |I_m, B_{l,\bar{x},\theta}| = \max_{(\bar{x}',\theta')\approx(\bar{x},\theta)} < I_m, B_{l,\bar{x}',\theta'} > \) be the shifted maximum of the responses. If \( N \) features are selected, in which the type, position and orientation of \( i \)-th feature are \( l_i, \bar{x}_i, \theta_i \) respectively, then the active feature presents classifier in the following form

\[
h(I) = \text{sign} \left( \sum_{i=1}^{N} w_i r_i(I) - w_0 \right) \tag{1}\]

where \( I \) is the image, \( r_i(I) = |I, B_{i,\bar{x}_i,\theta_i}| \) is response of the \( i \)-th feature on image \( I \), \( w_i \) is the weight of the \( i \)-th feature, and \( w_0 \) is the threshold.

The features are selected by the following procedure:

(0) for \( m = 1, ..., M \), and for each \( B_{i,\bar{x}_m,\theta_m} \in \Omega \), compute \( r_n(I_m) = |I_m, B_{i,\bar{x}_m,\theta_m}| \). Set \( i \leftarrow 1 \).

(1) for each candidate \( B_{i,\bar{x},\theta} \in \Omega \), do

- for \( m = 1, ..., M \), choose the optimal \( B_{m,i} \) that maximizes \( |I_m, B_{m,i}| \) among all possible \( B_{m,i} \approx B_i \).
- choose that particular candidate \( B_i \) with the maximum corresponding \( \sum_m |I_m, B_{m,i}|^{1/2} \).
- set \( w_i = \sum_m |I_m, B_{m,i}|^{1/2} / M \).

(2) for \( m = 1, ..., M \), for each \( B \approx B_{m,i} \), set \( |I_m, B| = 0 \), to enforce approximate non-overlapping constraint.

(3) if \( i = N \), normalize \( w_i \) so that \( \sum w_i^2 = 1 \), then stop.

Otherwise let \( i \leftarrow i + 1 \), and go to (1)

In order to make the feature linear-addictable, a whitening transformation is introduced \( r(I) = -\log F_i([I, B, \bar{x}, \theta]) \), where \( F_i() \) is the tailed accumulation distribution of the responses of feature of \( i \)-th type on negative examples. The tailed accumulation distribution function is discretized by the top ratio histogram. The process will ensure heterogeneous features are well-calibrated and comparable, sharing the same distribution on natural image ensemble. To accelerate the computation, the whitening transformation can be fitted by a cubic spline.

In order to make the learnt template more intuitive and more localized for view interpolating, we replace the Gabor feature with Haar-like feature [5] and set the number of orientations to 15. For simplification, we only use edge Haar, and have not used ridge or blob Haar. Figure 2 shows the comparison of the classifier learned by the original active basis and our variant with Haar-like feature. From the Figure 2 one can see that the template trained using Haar-like feature is more meaningful than Gabor.

3. CLASSIFIER INTERPOLATION

The proposed classifier can be described by a set of parameters \( C = (w_0; (w_1, l_1, \bar{x}_1, \theta_1), (w_2, l_2, \bar{x}_2, \theta_2), \ldots, (w_k, l_k, \bar{x}_k, \theta_k)) \). After sorting the features and add zero weight to make each \( i \) reserve for the fixed type of feature \( l_i \), we can remove the discrete label \( l_i \) and assume that \( C = (w_0; (w_1, \bar{x}_1, \theta_1), (w_2, \bar{x}_2, \theta_2), \ldots, (w_k, \bar{x}_k, \theta_k)) \) is on a manifold \( \Gamma \). For multi-view classifier manifold, it can be parameterized by view angle \( \rho \), then \( C = C(\rho) = (w_0(\rho); (w_1(\rho), \bar{x}_1(\rho), \theta_1(\rho)), (w_2(\rho), \bar{x}_2(\rho), \theta_2(\rho)), \ldots, (w_k(\rho), \bar{x}_k(\rho), \theta_k(\rho)))) \). Thus, we can divide the full view span into several ranges and train a classifier for each view range. Actually, the more ranges are divided, more accurate the final classifier is. But in fact, it is quite labor and error-prone to manually partition the examples to so many views and to train each classifier for each view. Therefore,
we instead train classifiers for a few key views, and interpo-
late classifiers for new views from the function $C = C(\rho)$. 
Formally, for each view $\rho$, we have classifier
\[
    h(I; \rho) = \text{sign} \left( \sum_{i=1}^{k} w_i(\rho) r_i(I; \rho) - w_0(\rho) \right)
\]
where $r_i(I; \rho) = -\log F_i(I, B_i(\rho), \vec{x}_i(\rho), \theta_i(\rho))$ is the re-
sponse of $i$-th feature at view $\rho$. In implementation, the
interpolation are carried out in the following steps.

Pre-learning: We first collect training examples for four 
key views — front, back, left, right, or front-left, front-right,
back-left, back-right, and then, learn classifiers for each view
independently.

Feature Aligning: Because the automatic learned tem-
plates may be not compatible with each other, minor refine-
ment and adjusting is need to make them compatible. Mean-
while, as active features can produce meaningful models, we
can adjust it manually and register each feature in one view
with another. Now the features of all given views are in corre-
spondences and share a common template, with invisible fea-
tures at certain view assigned zero weights and interpolated
positions. This procedure can also be automated, but here for
simplification, we do it manually. After the refinement, we
need recalculate the weights using adjusted feature positions
like training step.

Interpolation: Given two classifiers at view angle $\rho_0$
and $\rho_1$, their parameters $w_i(\rho_0)$, $\vec{x}_i(\rho_0)$, $\theta_i(\rho_0)$ and $w_i(\rho_1)$,
$\vec{x}_i(\rho_1)$, $\theta_i(\rho_1)$ in correspondences, and our task is for a new 
view $\rho_0 \leq \rho \leq \rho_1$, to interpolate the classifiers’ parameters
$w_i(\rho)$, $\vec{x}_i(\rho)$, $\theta_i(\rho)$.

For object can roughly be approximated by ellipsoid or
generalized cylinder, e.g. pedestrian, we use simple linear
interpolation to obtain feature positions and weights, i.e.
\[
    \vec{x}_i(\rho) = \frac{(\rho_1 - \rho) \vec{x}_i(\rho_0) + (\rho - \rho_0) \vec{x}_i(\rho_1)}{\rho_1 - \rho_0}
\]
\[
    w_i(\rho) = \frac{(\rho_1 - \rho) w_i(\rho_0) + (\rho - \rho_0) w_i(\rho_1)}{\rho_1 - \rho_0}
\]

One object can be roughly approximated by cube or poly-
hedron and thus we should take into count the occlusion and
scaling caused by 3D projection. From two given views $\rho_0$
and $\rho_1$, we select the features visible at view $\rho$ according to
3D occlusion knowledge. Then linear interpolation are ap-
plied to get the weight and threshold at view angle $\rho$. The
positions are interpolated linearly followed by a size scaling
to compensate the 3D shearing. For example, see Figure 3, in
order to interpolate the back view, with the given two views of
the car classifiers, back-left and back-right, we can select fea-
tures by following Eq.(3), and scale the horizontal positions
of features by $\sqrt{2}$ to compensate the 3D shrinking in given
views.

For the feature orientation, we also linearly interpolate it

on the loop manifold of angle.
\[
    v_x(\rho) = \frac{(\rho_1 - \rho) \cos[2\theta(\rho_0)] + (\rho - \rho_0) \cos[2\theta(\rho_1)]}{\rho_1 - \rho_0}
\]
\[
    v_y(\rho) = \frac{(\rho_1 - \rho) \sin[2\theta(\rho_0)] + (\rho - \rho_0) \sin[2\theta(\rho_1)]}{\rho_1 - \rho_0}
\]
\[
    \cos[2\theta(\rho)] = \frac{v_x(\rho)}{\sqrt{v_x(\rho)^2 + v_y(\rho)^2}}
\]
\[
    \sin[2\theta(\rho)] = \frac{v_y(\rho)}{\sqrt{v_x(\rho)^2 + v_y(\rho)^2}}
\]

Classifier Combination As stated in the literature[2, 4],
there are many strategy to combine the classifiers of each
view. We considered the simple one — using classifier of
each view as weak classifier to construct a boosting classifier.
\[
    H(I) = \text{sign} \left[ \sum_{\rho} \alpha(\rho) h(I, \rho) \right]
\]
The weight $\alpha(\rho)$ is trained on all available training examples.

4. EXPERIMENTS

We evaluate the proposed solution for multi-view object
recognition task as follows. First, we compare the ROCs of
the interpolated classifiers with the learned classifiers.
Then, to further validate the effectiveness of the interpolated
classifiers, we also compare the ROCs of boosted classifiers
constructed on learned classifiers and boosted classifiers con-
bstructed on key view classifiers and interpolated classifiers.

The dataset consists of 753 training examples and 1124
testing examples. We manually divide the data set into 8 view
ranges, among which our ranges are used as key views to train
classifiers and the other four views are used as groundtruth for
evaluating the performance of the interpolation.

4.1. Interpolated Classifiers

We compare interpolated classifier to original active bases
model learned from data. Figure 4 shows the templates of
car used in the experiments. Figure 5 shows the ROC com-
parison. From the results, one can observe that, although the
interpolated classifiers are not as effective as learned model,
their performances are quite acceptable. Taking into account
that the interpolated model seen no data on the view, just pre-
diction, those results are quite convincing.
4.2. Boosted Classifiers

Figure 6 shows the comparison of ROCs of the following classifiers on car data: I. MixTrain: Active basis classifier trained directly on all examples; II. Auto-Learn-4Model: Boosted classifier built on 4 key view classifiers learned by active basis; III. Auto-Learn-8Model: Boosted classifier built on 8 view classifiers learned by active basis; and IV. Learn-Interp-8Model: Boosted classifier built on learned 4 key view classifiers and 4 interpolated classifiers. From the results, one can see that Method-I and Method-II are very poor, and Method-III and Method-IV are similar. This implies that in the boosting framework, the interpolated classifiers can help to improve the performance just like the learned classifiers. Method-IV achieves better accuracy than Method-III, since the interpolated classifiers less trend to over-fitting.

5. CONCLUSION AND FUTURE WORK

Our work concentrates on predicting classifier for unseen view angle. Our interpolation is conducted in the space of classifiers, not on image directly. Experiments of interpolated single view classifier and combined multi-view classifier are conducted on pedestrian and car data sets. The results verified the effectiveness of the interpolation framework. Although we only demonstrated on-pane view rotation, we believe that our method can be easily extended to off-plane view rotation, which is part of our future work.

6. ACKNOWLEDGEMENT

This research is done for CSIDM Project No. CSIDM-200803, which is partially funded by a grant from the National Research Foundation (NRF) and administered by the Media Development Authority (MDA) of Singapore, and the National High Technology Research and Development Program of China (863 Program) under grant No.2006AA01A115.

References