# A Hierarchical Recursive Partial Active Basis Model

Pavel Herrera-Domínguez[1] and Leopoldo Altamirano-Robles[2]

[1] INAOE, Puebla
pavel13@inaoep.mx
[2] INAOE, Puebla
robles@inaoep.mx

**Abstract.** Recognition of occluded objects in computer vision is a very hard problem. In this work we propose an algorithm to construct a structure of a model using learned active basis models, then use it to do inference over the most probable detected parts of an object, to allow partial recognition using the standard sum-max-maps algorithm used for *active basis*. We tested our method and present some improvements on occluded face detection using our algorithm, we also present some experiments with other partially occluded objects.

## 1 Introduction

The problem of occlusion is a very hard problem in computer vision, because we need to fit some model to some image but we do not know all the possible ways the model can appear occluded, therefore we propose an algorithm to split a global model and then fit it by parts without losing the spatial information.

The objective of this work is to propose an algorithm to construct a model that gives the chance to detect occluded parts in a natural way, and to construct it with the less possible user supervision , in the way to avoid the time consuming labeling. In this work we assume that the object is centered and bounded, this can be resolved using local templates for active basis [8] or applying similar algorithms to the proposed by Zhu,L et al. [9].

## 2 Previous and Related Work

There are several works about the occlusion problem, most of them using discriminative models like Viola-Jones *adaboost* [4], some other approaches use grammars in the inference stage, like Wu et al. [6] that give a numerical study about the importance of each stage on the top-down and button-up inference. There are also some works for generative models, like the one from Baker et al. [1] that gives an extension for the AAM models(Cootes et al.[2]) using a more robust metric on the inference stage.

J.A. Carrasco-Ochoa et al. (Eds.): MCPR 2010, LNCS 6256, pp. 1–10, 2010.

There are also works related to construct a hierarchical model like the proposed by Zhu,l et al [9] that gives an unsupervised algorithm to construct models using structures formed by edge-lets. Also a recent work for pedestrian detection that use several part-detectors and merge them to have better detection[5].

The difference of our work with others is that for example we do not try to detect parts independently like Wu et al.[6] and merge them. The difference with the original Sum-Max-Maps[8] used for active basis (the representation we are using) is that we have an intermediate stage merging the parts and deciding if they are present or not, this way we can have a *partial* recognition of the object that could be useful in some applications. In the case of the learning algorithm applied, the difference and contribution is that we use the algorithm proposed by Wu et al. [8], learning algorithm also spatial relations that help in the recognition step.

## 3 Active Basis

The active basis model is a deformable model which consist of a small number of Gabor wavelet elements $B_{x,y,s,\theta}$ at selected locations and orientations. Each element is given by $B_{x,y,s,\theta}(x',y') = G(\hat{x}/s, \hat{y}/s)$ where $G(x,y) = e^{-\frac{(\frac{x}{\sigma_x})^2+(\frac{y}{\sigma_y})^2}{2}} e^{ix}$ and $\hat{x} = (x'-x)cos\alpha - (y'-y)sin\alpha$, $\hat{y} = (x'-x)sin\alpha + (y'-y)cos\alpha$, $s$ is the scale parameter and $\alpha$ is the orientation[7]. Using this elements we can represent the image as follows.

$$I_m = \sum_{i=0}^{n} c_{m,i} B_{m,i} + \epsilon \tag{1}$$

where $\epsilon$ is the residual image and

$$\begin{gathered} B_{m,i} \approx B_i \\ B_i = B_{x_i,y_i,s,\alpha_i} \\ B_{m,i} = B_{x_{m,i},y_{m,i},s,\alpha_{m,i}} \\ x_{m,i} = x_i + d_{m,i}\sin\alpha_i \\ y_{m,i} = y_i + d_{m,i}\cos\alpha_i \\ \alpha_{m,i} = \alpha_i + \delta_{m,i} \\ d_{m,i} \in [-b_1, b_1], \delta_{m,i} \in [-b_2, b_2] \end{gathered} \tag{2}$$

This means that $B_i$ is allowed to shift and rotate in the intervals given, allowing small deformation in the object model.

The full details are explained in the original papers by Zhu et al. [7][8], here we give only a short description of how they are learned and how they are used in recognition and detection.

### 3.1 Learning Active Basis

The active basis model specifies the distribution of the image $I$ as in equation (3).

$$p(I|B) = q(I)\frac{p(C)}{q(C)} = q(I)\frac{p(c_1, ..., c_n)}{qc_1, ..., c_n} \tag{3}$$

where $q(I)$ is the reference distribution. Assuming independence between the Gabor elements we have.

$$p(I|B) = q(I)\prod_{i=1}^{n}\frac{p(c_1)}{q(c_i)} \tag{4}$$

where $p(c_i)$ is parametrized as an exponential family model $p(c_i;\lambda) = \frac{1}{Z(\lambda)}e^{\lambda h(r_i)}q(c_i)$ where $r_i = | < I, B_i > |^2$ is the local energy of Gabor filter response and $h(r_i)$ is a sigmoid transformation function, and $q(c_i)$ is pooled from generic background images in an off-line stage. Replacing the parametrized distribution on equation(4) we have the probability of a single image given the model in equation (5).

$$p(I|B) = q(I)\prod_{i=1}^{n}\frac{1}{Z(\lambda)}e^{\lambda h(r_i)} \tag{5}$$

Now to learn B from a set of training images $I = \{I_1, ..., I_M\}$, we need to maximize the log-likelihood $\log\prod_{m=1}^{M}\frac{p(I_m|B_m)}{q(I_m)}$ over all images

$$\sum_{m=1}^{M} log(\frac{p(I_m|B_m)}{q(I_m)}) = \sum_{m=1}^{M} log(\frac{p(r_m)}{q(r_m)}), \tag{6}$$

when $M \to \infty$ we are maximizing the Kullback-Leibler divergence estimator between $p$ and $q$. This way we learn $B$, this is for each $B_i$ we have five parameters for each element $(x_i, y_i, \alpha_i, \lambda_i, \log Z_i)$, where $(x_i, y_i, \alpha_i)$ the parameters of the Gabor elements, and $(\lambda_i, \log Z_i)$ the parameters of the images responses distribution.

### 3.2 Using the Model

This way of learning $B_i$ also give us a score function to find the model in a new image and sketch the object in the image. To find the model given by $B$ we maximize the equation (8) this means to find the correct parameters $\Theta$ (center scale and position of the model, plus the parameters for each element of $\{B_i\}$)

$$argmax_{\Theta}\frac{P(I_m|B,\Theta)}{q(I_m)} = argmax_{\Theta}log(\frac{P(I_m|B,\Theta)}{q(I_m)}) \tag{7}$$

and

$$log(\frac{P(I_m|B,\Theta)}{q(I_m)}) = \sum_{i=0}^{n}(\lambda_i * h(r_{i,m})) - log(Z_i) \tag{8}$$

The full details of how to do this can be found in the original paper [8].

## 4 Formulation of the Occlusion Problem

Although in the literature exist very effective algorithms to learn models using *active basis* for deformable objects and use these models in recognition tasks, still there are not many advances in the solution of how to solve the occlusion problem, there are a solutions for example using grammars[6], using alpha, beta, gamma processes, but in this case the *active basis* models are used as simple detectors in the leaves of the grammar.

In this work we propose a different approach, let us first comment some key points to be considered.

1. There is no way to learn all possible forms that an object can be occluded.
2. The problem of occlusion seen from the active basis point of view can be interpreted as follows: there are some Gabor elements ($B_i$) not present in the image that contains the object to be recognized, so instead of having all $B_i$ we will have only a subset of them.

The previous points give us the idea to consider the problem of occlusion as finding the most probable subset of basis that are present in the image.

## 5 Recursive Hierarchical Active Basis

In this work we propose an extension of what is proposed by Ying Nian et al.[8] who proposed to use part-templates to form a recursive model to deal with articulated objects, they give an inference algorithm named recursive sum-max maps. What we propose is to construct a hierarchical-graph with the active basis, having two kind of relations, inter-level that is relations of the active basis at the same level of detail, and relations intra-level, this is how more detailed active basis than the original one are related, the figure (1) describes the idea. $R_i$ are the relations inter-level, and they are given by the indexes of the low detailed basis corresponding to the higher detail basis. The inter-level relations are spatial relations given as relative positions of the surrounding squares of each part.

### 5.1 Learning

We have already seen that we can learn a model $B$ given a set of images of one object. Now we will describe how to learn the recursive structure. In algorithm (1) we can see the pseudo-code of how to learn the structure. The basic idea is to learn separate detailed parts of the images guiding the learning algorithm with the basis already learned in a low-detailed more general scale. This way we can construct a graph that contains the relations between the parts maintaining which elements correspond to each new detailed part, and at the same time we have spatial relations between the same level of detail basis. To learn a priori probabilities, that will be used in the inference algorithm, we use the percentage of the object represented by that sub-part learned.

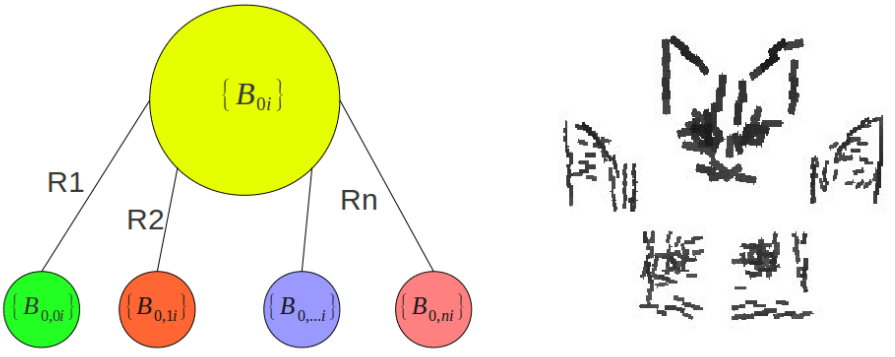



**Fig. 1.** Illustrative idea of the split model and its parts with the spatial relations to the global model

**Algorithm 1.** Learn-recursive-model

**Require:** *Itrain*
  $\{B_i\}$ ←learn from Itrain at scale Scale
  **if** $level = max\_level$ **then**
    $output \leftarrow \{B, \emptyset\}$
  **else**
    $Basis\_parts \leftarrow Split\{B_i\}$ using spatial relations
    **for** $part \in Basis\_Parts$ **do**
      **for** $img \in Itrain$ **do**
        $nTrain \leftarrow \{nItrain\} \cup subimage(img, part.subWindow)$
      **end for**
      $\{part.B_i\} \leftarrow$ Learn-recursive-model using nItrain
      $\{part.P(this|Parts)\} \leftarrow$ Probability of this part given that is a subpart
    **end for**
    $output \leftarrow \{B, Basis\_parts\}$
  **end if**

## 6 Using the Hierarchy

We modify the algorithm proposed by Wu et al. [8] to improve the recognition even when the object is partially occluded. The idea is based on the following simple observations.

1. If we need to maximize the log-likelihood, it is clear from the equation (8) that we should take the values which are greater than some threshold, that should represent the basis that are present as part of the object.
2. If we take the previous observation as a method to see the presence or not of and object, we need to merge them, because it is not enough taking them alone.

**Algorithm 2.** Computing SUM2MAP for incomplete basis

**for** $x = 1$ to W **do**
  **for** $y = 1$ to H **do**
    **for** $t = 1$ to NumberOfBasis **do**
      **if** $\lambda_t * MAX1MAP(x + B_{t,x}, y + B_{t,y}, B_{t,\theta}) - \log Z_t > 0$ **then**
        $H = H \cup \{t\}$
      **end if**
    **end for**
    $SUM2MAP(x, y) \leftarrow solve(H)$ this is Algorithm (3)
    $PresentBasis(x, y) \leftarrow H$
  **end for**
**end for**

**Algorithm 3.** Solve the inference problem button up using Dynamic Programming given the Hypothesis

Initialize $Table$ to $\epsilon$
$level \leftarrow lowest\_level$
**for** $t \in H$ **do**
  $Table[level][t] \leftarrow 1$
**end for**
$level \leftarrow level - 1$
**while** $level \geq 0$ **do**
  **for** $part \in Parts[level]$ **do**
    $prob \leftarrow 0$
    **for** $subpart \in part$ **do**
      $prob \leftarrow prob + Table[level + 1][subpart] * P(subpart|part)$
    **end for**
    $Table[level][part] \leftarrow prob$
    **if** $Table[level][part] < \epsilon_{part}$ **then**
      Retrieve info to H, this is when the part is not present just for sketching
    **end if**
  **end for**
  $level \leftarrow level - 1$
**end while**
$output \leftarrow Table[0][0]$

With this observations the SUM-MAX-MAPS algorithm [8] can be used to partially find an object, adding the relations learned in the training gives us an easy and natural way to merge the sub-parts.

In algorithm (2) it is shown how the algorithm sum-max-maps [8] is modified in the stage SUM2 to apply the idea proposed in this work. Here what we do is to take just the elements that are over some threshold and take them as the most simple hypothesis known, then merge them and compose the object to detect it.

### 6.1 Merging the Elements

Merging the found elements can be done by computing the probability as in equation (9).

$$P(Obj|\theta) = \sum_{SubPart \in Obj} P(SubPart|Obj)P(SubParts|\theta_{subPart}) \quad (9)$$

This can be accomplished by using dynamic programming, like is shown in algorithm (3), where each subproblem is to compute $P(Obj|SubParts)$, this means given the subparts estimate the *percentage* of the object *present*, then use $Obj$ and $P(Obj|SubParts)$ as sub-part of a more general object, the leaves of the dynamic programming tree are the elements of Active Basis that satisfy the threshold mentioned in the MAX1 maps [8].

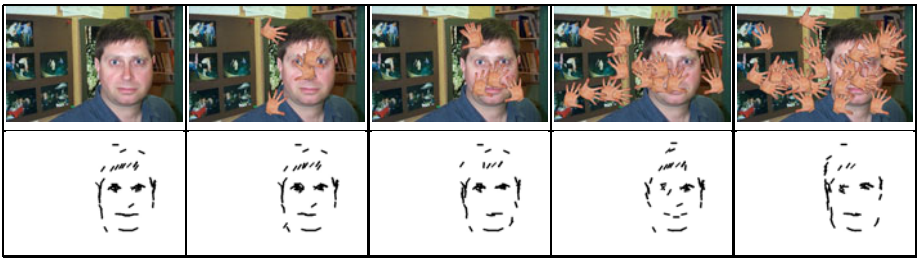
## 7 Experiments

We carried out some experiments with the inference algorithm and the structure learned in face detection task, to see its behavior with occlusion. The images were took from the Caltech 101 data base subcategory Faces[3]. We constructed several artificial images from this database generating some random occlusions in the image to know the level of occlusion it supports. The images contain random objects over all the image. We notice that if we only generate occlusion over the object of interest, the edges generated by these objects tend to form the face edges and help the matching score. To generate the occlusions we place the objects and count the number of pixels in the image that were occluded by the random objects until they satisfy a threshold that depends on the level assigned. Table (1) has examples of images under different levels of occlusion. To train the model we took the faces centered and bounded without any occlusion, the model had 100 elements for the model used with all the experiments.

The localization accuracy is measured by predicting object bounding boxes. For detection to be correct the area $A(rect)$ of overlap between a predicted bounding box $B^c$ and a ground truth bounding box $B^{gt}$ must be more than half the union of both areas:

$$B^c \; is \; correct \; \Leftrightarrow \; \frac{A(B^c \cup B^{gt})}{A(B^c \cup B^{gt})} \geq \frac{1}{2} \quad (10)$$

**Table 1.** Examples of the increasing level of occlusion and the sketches in one image of the data set. The size of all images is around 500 x 350 pixels.



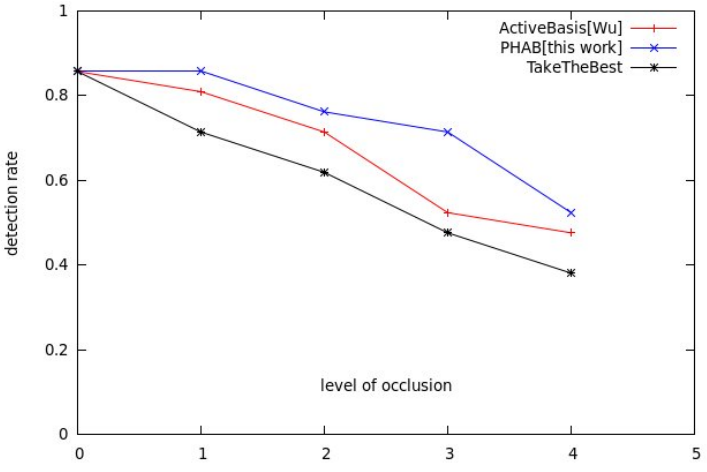



**Fig. 2.** Plotting level of occlusion in axis X vs detection rate in axis Y. The red plot (the curve in the middle) denotes the recognition rates using Sum-max-maps using the equation (8), the one in black (the curve in the bottom) plot was produced using the equation (8) but with the elements used in the blue one (the curve on the top). And finally the blue one is the proposed algorithm. Note how the proposed algorithm overcomes the original one.

So the localization rate is:

$$rate = \frac{positives}{positives + negatives} \tag{11}$$

where the positives are the hypothesis or bounding boxes that holds with equation (10).

The resulted model and the split sub-parts are shown in figure (3). The figure (2) shows the rates under several levels of occlusion; in the plot level 0 is the image without any occlusion. The second row of table (1) shows the partial sketches.

We made another experiment to see the performance on objects with *natural* occlusion. Using a set of cat faces we train the model showed in figure (1) and applied to a cat image taken from the Internet, the result is shown in figure (4). Also we trained a model for cars and tested on a image with some persons in the front, the results are presented in figure (4) also.

### 7.1 Implementation Details

We took the original code for active basis implemented in $C++$[1], then we modify and added the algorithm we propose. The parameters of *deformation* mentioned in equation (2) were the same for all the experiments, for training and testing:

[1] `http://www.stat.ucla.edu/~ywu/AB/active_basis_cpp.html`

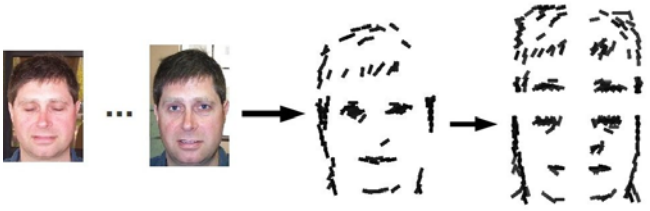


**Fig. 3.** Learned model and its parts, from a trainig set

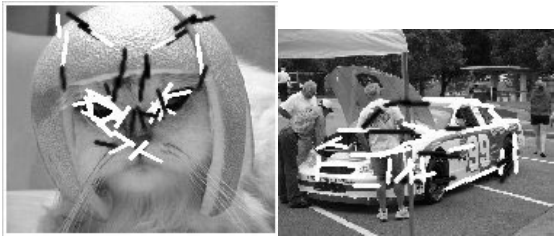


**Fig. 4.** Two diffetent examples to test the algorithm, the left one is a occluded cat an in the right a car with partial occlusion, the white lines denote the basis founded and the *black* ones the predicted ones

number of different orientations 15, angle deformation $\alpha_i = 3$ and the spatial deformation $d_i = 3$.

## 8 Conclusions

We have shown that is possible to detect and sketch the object even when there is partial occlusion. We have improved the detection rates compared with sum-max-maps[8], the detection can be increased even more if we use a better way to construct the model. We showed that by using the model separated on parts the detection rates can be increase under partial occlusion conditions.

Right now we are working splitting the model in the recognition stage, this modification is expected to work better on occlusion scenarios. It is worth to comment that algorithm behaves poor when the scale of the object changes considerably. As future work we will use these models to detect other kind of objects where natural occlusion occurs, like pedestrians, or cars on a parking lot.

## Acknowledgment

The work reported in this paper has been supported by CONACyT scholarship 271666.

## References

1. Baker, S., Matthews, I., Xiao, J., Gross, R., Kanade, T., Ishikawa, T.: Real-time non-rigid driver head tracking for driver mental state estimation. In: 11th World Congress on Intelligent Transportation Systems, Citeseer (2004)
2. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(6), 681–685 (2001)
3. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(4), 594–611 (2006)
4. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple. In: Proc. IEEE CVPR 2001 (2001)
5. Wu, B., Nevatia, R.: Detection and segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses. International Journal of Computer Vision 82(2), 185–204 (2009)
6. Wu, T., Zhu, S.C.: A Numerical Study of the Bottom-up and Top-down Inference Processes in And-Or Graphs. In Review (2009)
7. Wu, Y.N., Si, Z., Fleming, C., Zhu, S.C., Ucla, L.A.: Deformable Template As Active Basis. In: IEEE 11th International Conference on Computer Vision, ICCV 2007, pp. 1–8 (2007)
8. Wu, Y.N., Si, Z., Gong, H., Zhu, S.C.: Learning active basis model for object detection and recognition. International Journal of Computer Vision, 1–38 (2009)
9. Zhu, L., Lin, C., Huang, H., Chen, Y., Yuille, A.: Unsupervised structure learning: hierarchical recursive composition, suspicious coincidence and competitive exclusion. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 759–773. Springer, Heidelberg (2008)