Hierarchical Top Down Enhancement of Robust PCA*

Georg Langs[1], Horst Bischof[2], and Walter G. Kropatsch[1]

[1] Pattern Recognition and Image Processing Group 183/2
Institute for Computer Aided Automation, Vienna University of Technology
Favoritenstr. 9, A-1040 Vienna, Austria
{langs,krw}@prip.tuwien.ac.at
[2] Institute for Computer Graphics and Vision, TU Graz
Inffeldgasse 16 2.OG, A-8010 Graz, Austria
bischof@icg.tu-graz.ac.at

Abstract. In this paper we deal with performance improvement of robust PCA algorithms by replacing regular subsampling of images by an irregular image pyramid adapted to the expected image content. The irregular pyramid is a structure built based on knowledge gained from the training set of images. It represents different regions of the image with different level of detail, depending on their importance for reconstruction. This strategy enables us to improve reconstruction results and therefore the recognition significantly. The training algorithm works on the data necessary to perform robust PCA and therefore requires no additional input.

1 Introduction

The human visual system takes advantage of the ability to distinguish between interesting regions and less relevant regions in the field of view. By using this knowledge it is able to improve its performance considerably. [1] and [2] describe two strategies to obtain and apply information about the importance of different regions of an image when simulating the human visual system. *Bottom-up* methods retrieve their features only from the present input image [3]. *Top-down* methods are driven by knowledge which is available before getting the input. Experiments [1] have shown that human vision and particularly the scan paths of the eyes, called saccades, are not only dependent on the input image, but largely on previous knowledge i.e. *top-down* expectations. In this paper we propose a method to incorporate a top-down strategy in a robust PCA algorithm [4] for object recognition. In our approach, instead of performing sequential saccades, we change the initial representation of the images with respect to the relevance of different regions. We demonstrate that by using this top-down knowledge we are able to significantly improve the recognition results.

* This research has been supported by the Austrian Science Fund (FWF) under grant P14445-MAT and P14662-INF

T. Caelli et al. (Eds.): SSPR&SPR 2002, LNCS 2396, pp. 234–243, 2002.

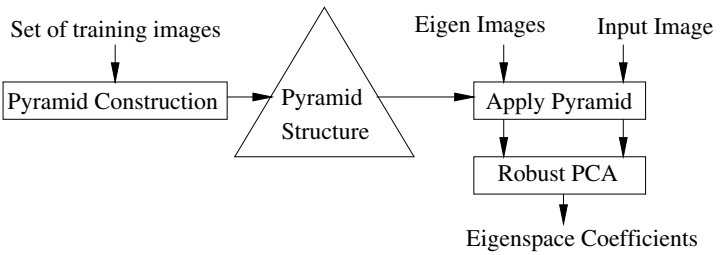


Fig. 1. The basic concept of our algorithm. It is divided into a training- (left) and a recognition phase (right)

The paper is organized as follows: In section 2 an overview of the algorithm is presented. The pyramid structure constructed by the algorithm is presented in section 3. The *training phase* is explained in section 4. Section 5 describes the application of the pyramid during the *reconstruction phase*. Finally we present experimental results in section 6 and give a conclusion in section 7.

2 Our Approach

The approach presented in this paper aims to enhance robust recognition based on eigenimages ([4]). We deal with an input, which can be an image or any other signal, that is represented as a vector of pixel values. Instead of performing sequential saccades we change the initial representation in order to stay abreast of the regions of interest. Each region is represented to an extend that corresponds to its importance for the reconstruction. The modified representation is used as input for robust PCA. Robust PCA represents training images in the eigenspace. To recognize an input image the coefficients of the eigenvectors are determined by solving an overdetermined system of linear equations in a robust manner. Further robustness is achieved by randomly selecting subsets of pixels.

The new representation has the following advantages over the unprocessed image, where all regions are represented to the same extend: Regions with small importance for the reconstruction or recognition are usually similar on different images of the training set, therefore if used for robust recognition they support almost all hypotheses. A large set of irrelevant pixels that are consistent with almost all hypotheses strongly interferes with the method in three ways: (1) It causes huge equation systems in the fitting step, which are numerically unstable (2) it wastes time because useless hypotheses are built and (3) the difference between good and bad hypotheses is likely to become smaller.

Our approach is divided into 2 main phases (Figure 1): During a *training phase* the computer is taught the importance of each pixel position as well as the dependencies between neighbouring pixels i.e. how much their values correlate in the training data. The algorithm builds an irregular *pyramid structure*

that represents a given input image with different levels of detail. This pyramid structure is build by contracting the initial regular image template. Irregular image pyramids represent images as a set of graphs with decreasing number of nodes. During contraction consecutive levels are built by choosing a set of surviving vertices, and assigning them a set of sons, the receptive field. [5] gives detailed explanations of the concept of irregular image pyramids.

During the *recognition phase* different levels of the pyramid structure are applied on the input image as well as on the eigenimages of the database. The resulting vectors are used as input for robust PCA.

3 The Pyramid Structure

The result of our algorithm is a *pyramid structure* that can be applied to any input image of a given size. In 3.1 we describe the structure and give its exact definition, in 3.2 we explain an example.

3.1 Definition of the Pyramid Structure and Its Application

The pixels of the input image can be indexed by $i = 1, ..., N_1$ where N_1 is the number of pixels in the image. To convert a rectangular grid (the image) to a vector we use the transformation

$$i_{vec} = (i_{arr} - 1)n + j_{arr} \tag{1}$$

where i_{vec} indicates the index in the vector and (i_{arr}, j_{arr}) the vertical and horizontal coordinates of a pixel in an $m \times n$ image. Each pyramid level P_k consists of a vector of nodes, each of them representing a set of pixels, its *receptive field*.

$$P_k = \langle n_{1,k}, ..., n_{N_k,k} \rangle \qquad \forall i, k : n_{i,k} \in \mathfrak{P}(\{1, ..., N_1\}) \tag{2}$$

The receptive fields are not overlapping and the union of the receptive fields together with a set r covers the whole image. r is the set of pixels, that have $weight = 0$. They are irrelevant or interfering with the reconstruction and are therefore ignored.

In the first level each node represents one pixel i.e. $n_{i,1} = i$. During contraction a node in the level $k + 1$ takes over the indices from its sons in level k:

$$n_{i,k+1} = \bigcup_{j:n_{j,k} \text{son of } n_{i,k+1}} n_{j,k} \tag{3}$$

The final pyramid structure with L levels consists of L vectors of nodes $P_k, k = 1, ..., L$. Each node represents a receptive field in the base level.

We define the procedure how to apply a pyramid structure P on an image [1]: The structure can be applied to an input image independently for each level, i.e. one can construct a certain level of the pyramid directly from the input image without constructing the levels in between.

[1] Note that an extension to other data representable in vector form is straightforward

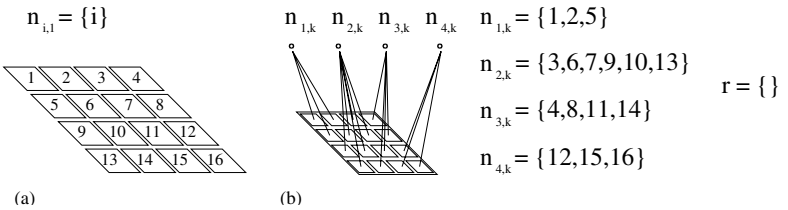


Fig. 2. An example of (a) the base-level and (b) a pyramid level P_k with 4 nodes

Definition 1. *Let B_1 be an image with pixel values $\langle b_{i,1}\rangle_{i=1,...,N_1}$ (in our experiments we used an image size of $N_1 = 128 \times 128 = 16384$ pixels). To calculate the k-th level $B_k = \langle b_{1,k}, ..., b_{N_k,k}\rangle$ of the pyramid, for each node the mean value of the pixel values in the receptive field is calculated:*

$$\forall i : b_{i,k} = \frac{\sum_{j \in n_{i,k}} b_{j,1}}{|n_{i,k}|} \tag{4}$$

Note that P is an 'empty' structure, in the sense that the nodes don't have attributes like gray values assigned to them. Only when calculating B_k of an input-image, gray values are assigned to the nodes of B_k according to P_k.

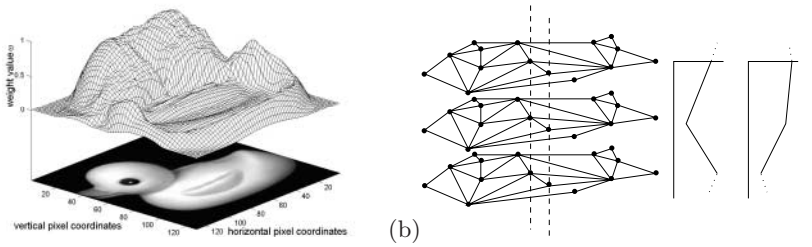
3.2 Example of a Pyramid Structure

To illustrate the pyramid structure we give an example of 2 levels P_1 and P_k. Figure 2(a)shows the base-level: The size of the images in the training set is $4 \times 4 = 16$ and for all nodes in the base-level $n_{i,1} = i$ (calculated according to equation 1) holds. (b) shows a level with 4 nodes, each of them representing a set of pixels in the base-level. The set r is empty in this example.

4 Training Phase

We assume that we are given a set of n images that represent one or more objects with different pose, and are of the same size. All these images are represented in a single eigenspace. From this set of training images we can retrieve the following information:

1. The eigenimages, eigenvalues and the coordinates of the training images in the eigenspace.
2. The variance of the value in each pixel over the training set.
3. The dependencies between pixels or receptive fields over the training set: For a given pixel or node i the pixel values in each training image $\{v_1, ..., v_n\}$ form a vector of values $(v_{i,1}, ..., v_{i,n})$, the *value profile*. In figure 3(b) value profiles of two neighbouring nodes in 3 images are depicted. Each node represents a receptive field. Two value profiles can be compared by calculating



(a) (b)

Fig. 3. (a)Weight map based on variance for a training set, consisting of 36 images of a rotating duck. (b) Value profiles of neighbouring nodes in the training set

their correlation $corr(v_i, v_j)$. By contracting two nodes with highly correlated value profiles the loss of information is expected to be smaller than the loss caused by contracting two pixels with more independent behavior.

4.1 Weight Target Contraction

During the contraction process we represent a given image with decreasing precision. In levels $2, 3, ...$ we deal no longer with individual pixels, but with nodes that represent a set of pixels in the original image. A node $n_{i,l}$ in level l is assigned the weight

$$\omega_{n_{i,l}} = z \cdot \sum_{x_i \in \text{ receptive field of } n_{i,l}} f(\omega_i) \tag{5}$$

We initiate z with $z_o = 1$ and define a *weight target* $1 - \tau$. P_1 is the base level of the pyramid and its neighborhood relation $\hat{N}_1 \subseteq P_1 \times P_1$ is defined according to the input data. If level P_i and $\hat{N}_i \subseteq P_i \times P_i$ have been built then we build level P_{i+1} according to the following rules:

1. Perform stochastic decimation [6] on P_i thus every node is assigned a status *survivor* or *non-survivor*. All nodes with $\omega_{i,j} > (1 - \tau)$ become *survivors*.
2. A *non-survivor* n_{i,j_0} chooses a father from all neighbouring survivors $\{n_{i,j_1}, ..., n_{i,j_k}\}$. $n_{i,j_{father}}$ becomes father if its *value profile* is most correlated with the *value profile* of the *non-survivor* n_{i,j_0} i.e. if the distance $(d_i)_{j_0,j_{father}}$ is minimal. d_i is the *distance map* of level i;
3. If the *weight* of the *non-survivor* $\omega_{n_{i,j_0}}$ and the *weight* of the chosen *father* $\omega_{n_{i,j_{father}}}$ sum up to a value

$$\omega_{n_{j,j_0+1}} = \omega_{n_{i,j_0}} + \omega_{n_{i,j_{father}}} > (1 + \tau) \tag{6}$$

then do not merge and change status n_{i,j_0} to *survivor*;

4. Define the neighborhood relation of the new level according to stochastic decimation [6].
5. If contraction terminates set $z_{new} = z/2$.

The algorithm proceeds with the following major steps: *Step 1* decides which receptive fields are merged with other receptive fields when constructing the successor of the level. After performing stochastic decimation algorithm [6] we modify the initial partition according to the weight map (This is an opposite strategy to [7]). *Step 2* chooses a neighbouring receptive field to merge with. All *sons* (or *grandchildren* resp.) in the base level P_1 of one *father* in an arbitrary level build its receptive field. If the resulting receptive field does not meet certain requirements defined in step (3) then steps (2) and (3) are canceled. The contraction process proceeds until no further merging is possible (Equation 6). The contraction stops, z is decreased and again contraction is performed until convergence.

While the 1^{st} priority is to merge receptive fields with high correlation (search for father) the 2^{nd} is to merge them only until they reach a certain weight according to (6). This strategy leads to a more balanced distribution of weights compared to a Gaussian pyramid or a pyramid built by plain stochastic decimation.

Experiments (Section 6) show that with $f(\omega_i) = \omega_i^s$ in (5) there is no exponent s that performs best on all resolutions resp. levels. A function defined in equation 8 resulted in the smallest reconstruction error.

Theorem 1. *Let* $(\omega_i)_{i=1,...,N}$ *be a* weight map *with* $0 \leq \omega_i \leq 1$ *for all* $i = 1, ..., N$. *Let* P_i *denote the set of nodes and* $\hat{N}_i \in P_i \times P_i$ *be the neighborhood relation in level* i. *Let* d_i *be* distance maps *i.e. functions from* N_i *in* $[-1, 1]$. *Then method 2 converges to a single node.*

A proof of Theorem 1 is given in [8].

In addition it is possible to estimate the size of a receptive field, if we are given an interval for the weights of the pixels lying in the receptive field. This is helpful during search of a monotonously ascending function $f(\omega_i) : [0, 1] \rightarrow [0, 1]$. Let $n_{l,j}$ be a node with a receptive field $p_i, i = 1, ..., N$ and let τ denote the target tolerance, then

$$\omega_{n_{l,j}} = z \sum_{i=1}^{N} f(\omega_i) \leq (1 + \tau) \tag{7}$$

holds. We assume that $\forall i = 1, ..., N : \omega_i \in [\bar{\omega} - \delta, \bar{\omega} + \delta]$ and get the estimation $N \cdot f(\bar{\omega} - \delta) \leq \sum_{i=1}^{N} f(\omega_i) \leq N \cdot f(\bar{\omega} + \delta)$ and finally $N \leq \frac{\frac{1}{z}(1+\tau)}{f(\bar{\omega}-\delta)}$. Figure 4(b) gives an impression of the expected influence of $f(\omega_i)$ on the size of the receptive fields. The function *logsig* is a modified log-sigmoid transfer function. It is defined by modifying the function $ls(\omega) = \frac{1}{(1+e^{-\omega})}$ in order to get a function $logsig : [0, 1] \rightarrow [0, 1]$

$$logsig(\omega) = \frac{ls(l \cdot (\omega - t)) - ls(-l \cdot t)}{ls(l \cdot (1 - t)) - ls(-l \cdot t)} \tag{8}$$

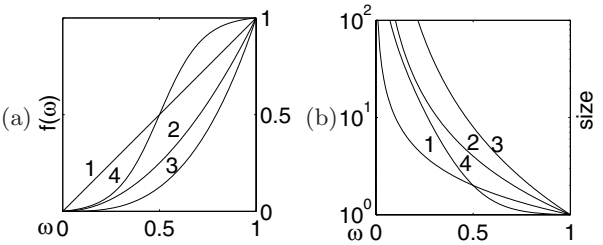


Fig. 4. Some functions (a) $f_1(\omega_i) = \omega$, $f_2(\omega_i) = \omega_i^2$, $f_3(\omega_i) = \omega_i^3$, $f_4(\omega_i) = logsig(\omega_i)$; (b): a comparison of the corresponding sizes of the receptive fields

l and t are parameters. l controls the steepness of the curve, while t shifts the steepest part of the curve along the x-direction.

5 Reconstruction

The following steps reconstruct or recognize a given image using a given eigenspace with a base consisting of N eigenimages, a pyramid P and level i. P is an empty structure as it is described in section 3. We calculate B_i according to *definition* 1:

1. For all eigenimages of the eigenspace: calculate level B_i according to the pyramid P. This results in N vectors $\{e_1, ..., e_N\}$
2. Calculate pyramid level $B_i^{inputimage}$ of the input image

The coefficients of the training images and the input image do not change [4]. The resulting vectors $\{e_1, ..., e_N, B_i^{inputimage}\}$ are input to robust PCA. Note that the 1^{st} point is performed during the training phase. During reconstruction only one level based on the input image has to be calculated. Computational expensive steps i.e. the contraction of an image template to a pyramid structure takes place entirely during the training phase.

6 Experiments

Experiments were performed on a dataset of gray level images of different objects. The database (COIL-20 [9]) contains images of 20 objects, each object rotated around its vertical axis with images taken in 5^o steps. Our training set consists of 36 images (i.e. 10^o steps) of one object taken from the database. The size is $128 \times 128 = 16384$ pixels. The test set consists of the same 36 images, each 50% occluded. Target tolerance is $\tau = 0.1$. After calculation of B_i the test input images are reconstructed by unconstraint robust PCA [4]. Figure 5(c) shows a comparison of the mean squared reconstruction error. The horizontal line in figure 5(c) represents the error gained with full resolution i.e. without processing before PCA. The modified *logsig* function ($l = 9$,$t = 0.8$) performs best

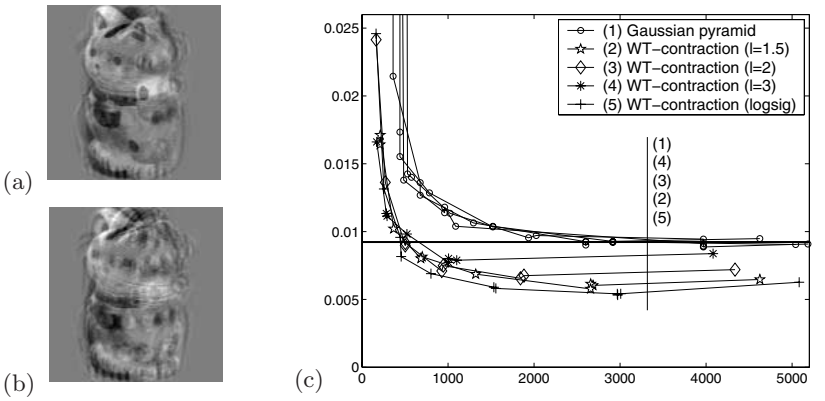


Fig. 5. (a) image of a cat, reconstructed after irregular downsampling and (b) after regular downsampling (c) Mean squared reconstruction error for pyramids constructed using different contraction algorithms

on almost all levels. Note that in figure 4 $f(\omega_i) = logsig(\omega_i)$ provides smallest receptive fields for important regions and the steepest increase of receptive field size for decreasing weight. For $f(\omega_i) = \omega_i^s$ at high resolutions lower values outperform higher values for s. This also corresponds to smaller receptive field sizes at higher weights. In figure 6 each dot represents a pixel. The x-coordinate represents its weight, the y-coordinate the size of the receptive field it lies in. (a) shows the diagram for $s = 2$ (1842 nodes) and (b) for $s = 3$ (1559 nodes). Figure 6(c) shows randomly colored receptive fields constructed by WT-contraction on training images of a rotating duck (weight map in Figure.3). Note the small receptive field size in regions where the head gives most information about the pose. For extremely low resolutions higher s-values slightly outperform lower ones. The reason is the possibility to build larger fields for pixels with low weight. This leaves more nodes for more important regions. $f(\omega_i) = logsig(\omega_i)$ attempts to combine both advantageous features.

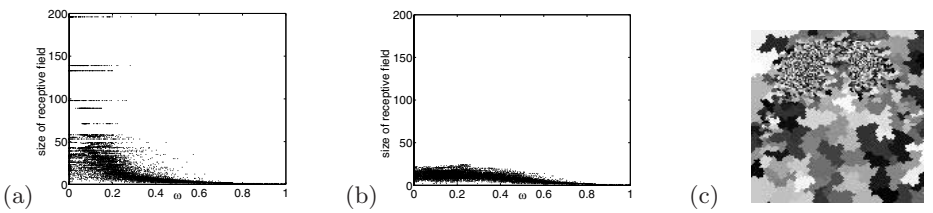


Fig. 6. (a,b) Plot of pixel weights vs. size of receptive fields; (c) randomly colored receptive fields, level of a pyramid based on the weightmap shown in Figure 3

Extensive experiments on all images of the COIL-20 database show that WT-contraction is able to significantly improve for 55% of the objects the reconstruction error (on average to 81% of the error with full resolution). Compared to a Gaussian pyramid the reconstruction error was improved to 61% of the error achieved on Gaussian pyramid levels with similar resolution. Experiments showed that contracting an input image by our algorithm to a number of ~ 3000 nodes ($\sim 18\%$ of full resolution) can decrease the mean squared reconstruction error down to $\sim 53\%$ of the error achieved with full resolution (~ 16384 pixels). For extremely low numbers of nodes few remaining small receptive fields allow stabilization: with less than $\sim 3\%$ of initial 16384 nodes the error is $\sim 2\%$ of the error achieved when the image is contracted with a regular Gaussian pyramid.

7 Conclusion

We present an approach to enhance robust PCA for object recognition and reconstruction. The algorithm simulates human vision, in particular: top-down processing and saccades by building irregular pyramid structures during a training phase. This structures are applied to an input image before robust PCA is performed. During our experiments we decreased the reconstruction error of robust PCA significantly. To represent regions of an image according to their relevance turns out to be crucial, not only to save computation time but also to improve and stabilize reconstruction and recognition results. The presented algorithm is able to meet this goal without a need for additional input. Future work will include optimization of $f(\omega_i)$ to specific tasks and a study of connection between the *distance*- and the *weight* map.

References

1. Lawrence W. Stark and Claudio M. Privitera. Top-down and bottom-up image processing. In *Int. Conf. On Neural Networks*, volume 4, pages 2294 –2299, 1997. 234
2. D.A. Chernyak and L.W. Stark. Top-down guided eye movements. *SMC-B*, 31(4):514–522, August 2001. 234
3. Claudio M. Privitera and Lawrence W. Stark. Algorithms for defining visual regions-of-interest: Comparison with eye fixations. *IEEE Trans. on PAMI*, 22(9), 2000. 234
4. Aleš Leonardis and Horst Bischof. Robust recognition using eigenimages. *CVIU*, 78:99–118, 2000. 234, 235, 240
5. Walter G. Kropatsch. Irregular pyramids. Technical Report PRIP-TR-5, Institute for Automation, Pattern Recognition and Image Processing Group, University of Technology, Vienna. 236
6. Peter Meer. Stochastic image pyramids. *CVGIP*, 45:269 – 294, 1989. 238, 239
7. Jean-Michel Jolion. Data driven decimation of graphs. In *Proc. Of GbR'01, 3rd IAPR Int. Workshop on Graph Based Representations*, pages 105–114, 2001. 239
8. Georg Langs, Horst Bischof, and Walter G. Kropatsch. Irregular image pyramids and robust appearance-based object recognition. Technical Report PRIP-TR-67, Institute for Automation, Pattern Recognition and Image Processing Group, University of Technology, Vienna. 239

9. S.A. Nene, S.K. Nayar, and H. Murase. Columbia object image library (COIL20). Technical Report CUCS-005-96, Columbia University, New York, 1996. 240