# Artificial Complex Cells via the Tropical Semiring

Lior Wolf and Moshe Guttmann
School of Computer Science
Tel-Aviv University
{wolf,guttm}@cs.tau.ac.il

## Abstract

*The seminal work of Hubel and Wiesel [14] and the vast amount of work that followed it prove that hierarchies of increasingly complex cells play a central role in cortical computations. Computational models, pioneered by Fukushima [12], suggest that these hierarchies contain feature-building cells ("S-cells") and pooling cells ("C-cells"). More recently, Riesenhuber & Poggio have developed the HMAX model [25], in which S-cells perform linear combinations, while C-cells perform a MAX operation.*

*We note that methods for computing the connectivity of S-cells abound since many algorithms for suggesting informative linear combinations exist. There are, however, only few published methods that are suitable for the construction of C-cells. Here, we build a novel dimensionality reduction algorithm for learning the connectivity of C-cells, using the framework of the max-plus ("tropical") semiring.*

## 1. Introduction

The fast and accurate object recognition achieved by humans is still far beyond machine capabilities. A prominent theory claims that a large part of it is achieved by a feed-forward hierarchical system, which attains viewpoint- and other invariances while maintaining selectivity, *c.f*. [21].

Selectivity is well studied in computer vision. Traditionally, it has been modeled by linear models, with examples ranging from correlation-based template-matching through face-selective eigenfaces to separating hyperplanes obtained via SVM. Mid- and high-level invariances, we claim, are less understood. A modern approach would handle it either by constructing invariant image descriptors (*e.g*. [18]), or by enriching the training set with multiple views of the objects (while still implementing a "flat" learning model), *e.g*. [26]. Both these solutions work well but seem to be in direct conflict with the selectivity requirement.

Standard hierarchical model theories deal with invariances using the concept of pooling cells. For example, in [23], Poggio and Edelman propose to construct one view-invariant unit by pooling from several view-tuned units. This solution was hand-crafted, but how can one learn the connectivity of the pooling cells automatically?

The recent work of Serre *et al*. [28] shows that sampling of random prototypes can be effective (see also [29]). Each prototype is a small fragment from a set of "natural images". This solution is, however, limited to retinotopic maps, where spatial fragments are well defined.

A more computational approach for learning the connectivity of pooling cells is by locating groups of cells that have a strong inclination to exchange activity with each other when observing different views of the same object. For continuously varying sequences, Foldiak's trace rule [10] is a natural choice for identifying such groups. It does not, however, provide a general solution.

To gain insight on the solution we propose here, let us describe the set of input units as nodes in a directed graph. Among all these units several have discriminatory capabilities for a specific object. Let us call these A,B and C. For any specific view of this object, only some of these units may be active, and this set of active units changes rather smoothly. For example, assume that for one view point, only units A and B are active, then the view point changes and units B and C become active, followed by another change in viewpoint, when units C and A become active.

The amount of information in unit B changes abruptly when A in known. They are not independent. If we weigh the edges of the (fully connected) graph by the amount of information change, our subgraph A-B-C above is "enriched". We locate such enriched sub-graphs by extracting *optimal cycles*. Specifically, cycles with the highest mean weight.

The computational framework we use to compute optimal C-cell pooling is the *max-plus semiring*. In this semiring, the equivalent of PCA we derive amounts to finding maximal mean cycles. These cycles mark groups of variables that are highly connected, and are therefore good candidates for the construction of pooling C-cells.

## 2. The tropical semiring

The HMAX model [25] contains S-cells that are linear combination of their input, and C-cells that perform a MAX operation. One can use conventional linear algebra and compute connections of S-cells using algorithms such as PCA and LDA. For C-cells, because of the non-linearity of the MAX operator, a different set of tools is required. Here, we use the max-plus semiring's solid foundations [13] to develop such tools.

The max-plus semiring $\mathbb{R}_{\max}$ is the set $\mathbb{R} \cup \{-\infty\}$, equipped with two operations: $\max$ as addition, and $+$ as multiplication. The addition operator is marked $\oplus$, for example, $2 \oplus 3 = \max(2,3) = 3$. The multiplication operator is marked $\otimes$, *e.g.* $1 \otimes 1 = 1 + 1 = 2$. The neutral element for addition, the zero element, is marked $\mathbf{0}$ and equals $-\infty$, since $\mathbf{0} \oplus a = a \oplus \mathbf{0} = \max(a, -\infty) = a$. The unit element, $\mathbf{1}$, which is the neutral element for multiplication equals 0, since $\mathbf{1} \otimes a = a \otimes \mathbf{1} = 0 + a = a$.

Matrix operations are defined naturally: if $a$ and $b$ are vectors of length $n$, $a^\top \otimes b = (a_1 \otimes b_1) \oplus \cdots \oplus (a_n \otimes b_n) = \max_i a_i b_i$. More generally, let $A$ and $B$ be an $l \times m$ and $m \times n$ matrices over the set $\mathbb{R} \cup \{-\infty\}$. $(A \otimes B)_{ij} = \bigoplus_k A(i,k) \otimes B(k,j) = \max_k A(i,k) + B(k,j)$, where, we use $\bigoplus_i$ to denote indexed addition over the semiring, similarly to the use of $\Sigma_i$ in the conventional algebra.

The spectral theory for square matrices in the max-plus semiring is similar to the conventional one. Let $A \in (\mathbb{R}_{\max})^{n \times n}$, eigenvector/eigenvalue relations are defined in the conventional way

$$A \otimes x = \lambda \otimes x \tag{1}$$

where $x \in (\mathbb{R}_{\max})^n \setminus \{\mathbf{0}^n\}$ is the eigenvector, and $\lambda \in \mathbb{R}_{\max}$ is the eigenvalue. In conventional notation the eigenvalue/eigenvector relations amount to

$$\forall i \in 1, \ldots, n, \quad \max_{1 \le j \le n} (A(i,j) + x(j)) = \lambda + x(i) \quad . \tag{2}$$

Directed graphs are defined based on square matrices in the usual manner. The graph $G_A$ associated with $A \in (\mathbb{R}_{\max})^{n \times n}$, has a set of nodes $\mathcal{N} = \{1, \ldots, n\}$, and a set of edges $\mathcal{E} = \{(i,j) | A(i,j) \neq \mathbf{0}\}$. The element $A(i,j)$ denotes the weight of the edge $(i,j)$. A matrix $A$ is called irreducible if $G_A$ is strongly connected.

The spectral problem for a matrix $A$ in the max-plus semiring is tightly connected with the notion of optimal cycle mean of the graph $G_A$. A cycle is a path in the graph such that the first node of the path corresponds to the last, it can be represented by a set of edges $c \subseteq \mathcal{E}$. The cycle mean for a cycle $c$ in $G_A$ is defined as

$$\frac{\sum_{(i,j) \in c} A(i,j)}{\sum_{(i,j) \in c} 1} \quad . \tag{3}$$

The maximal cycle mean is the maximum of the above equation over all cycles in a graph. A classical algorithm for detecting the optimal mean cycle is Karp's algorithm [16].

The following result connects the notion of maximal cycle mean with spectral notions.

**Theorem 1 (Max-plus spectral theorem [2])** *An irreducible matrix $A \in (\mathbb{R}_{\max})^{n \times n}$ has a unique eigenvalue which equals the maximal cycle mean of $G_A$.*

## 3. max-plus PCA

PCA [11] is an extremely popular algorithm for dimensionality reduction, which is used extensively in many research and engineering fields. PCA computes a projection to an ordered set of principle components, such that the projections are uncorrelated, and for every $k$ the first $k$ principle components best reconstruct the original data. In one algebraic formulation, where $A$ is the covariance matrix of the data, PCA finds a set of orthogonal unit vectors $u_i, i = 1..k$ such that $\sum_{i=1}^k u_i^\top A u_i$ is maximized. It is easy to show that for one possible solution (which is in fact the classical solution) the vectors $u_i$ are the eigenvectors of $A$ ordered by the corresponding eigenvalues, *i.e.* $Au_i = \lambda_i u_i$, for some eigenvalues $\lambda_1 > \lambda_2 > \cdots > \lambda_k$. The PCA maximization problem is therefore equivalent to:

$$arg \max_{\{u_i\}_{i=1}^k} \sum_{i=1}^k \lambda_i \;\; s.t. \;\; Au_i = \lambda_i, \;\; u_i^\top u_j = \delta_{ij} \tag{4}$$

Since the matrix $A$ is sufficient for the computation of PCA, a natural kernel extension was suggested for which $A$ can be replaced by any positive definite kernel matrix [27].

Below, we derive a PCA-like method in the max-plus semiring. Unlike the classical PCA, a projection to a principle component in max-plus is non-linear. As we will see, max-plus PCA has several unique characteristics. For example, it does not require the use of positive definite or even symmetric affinity matrices. Max-plus PCA also has the property of producing sparse solutions, *i.e.* for each projection to a principle component only a small subset of the variables participates.

We start with a square affinity matrix $W$. Although we use the word affinity, which is sometimes used in kernel PCA, $W$ does not need to be symmetric. Assume, for example, that $W$ measures a score related to the lack of uncertainty between variables. Our goal is to locate subsets of variables that are highly constrained between themselves. We assume that grouping of such variables can provide a stable measurement even if some of them are not stable. For a group of variables of a given size, we measure the group's affinity as the sum of affinities between all of its elements.

In graph notation, given an $n \times n$ affinity matrix $W$, we consider the induced complete graph $G_W$. We measure the affinity of a subset of the nodes $V \subseteq V(G_W) = \{1, \ldots, n\}$

as the sum of weights in the sub-graph $K_V$ induced by the subset $V$.

$$W(V) = \sum_{e \in E(K_V)} W(e), \tag{5}$$

where for $e = (i, j)$, $W(e) := W(i, j)$. Rephrasing, and assuming that our affinities are finite, the weight of a subset $V$ is exactly the weight of the corresponding clique in $G_w$.

$W(V)$ is not appropriate as a score for the quality of subsets of features since it favors large groups over small ones. In particular, for $W$ with no negative elements the maximal clique is also the maximum weight clique. Therefore, we normalize the weight of the clique by the number of edges in it, and use the mean clique weight. The search for cliques with maximal mean weights can be formalized as the following maximization problem:

$$\max_V \frac{1}{|V|(|V|-1)} \sum_{v_i, v_j \in V} w(v_i, v_j), \tag{6}$$

which is known as the remote-clique problem.

This optimization problem, of finding the maximum mean weight clique in a complete graph is known [5] to be NP-Hard. Although the problem does have a polynomial time approximation algorithm (factor 2) [4] its time complexity is being held by large constants thus making the approximation formidable.

### 3.1. Connections between mean clique weight and mean cycle weight

We suggest using mean weight cycles (Eq. 3) instead of mean cliques. Mean weight cycles can be detected in polynomial time, and even naïve implementations run on graphs containing hundreds of nodes without difficulty. The mean weight cycle and the mean clique weight are related as follows: (1) The optimal mean weight clique is not necessarily a sub-graph of the optimal mean cycle, as can be shown by an example; (2) an optimal clique is always bounded by the optimal mean Hamiltonian cycle on its nodes.

**Theorem 2** *A complete graph contains a Hamiltonian cycle having a mean weight larger than the mean weight of the graph.*

**Proof:** For simplicity we consider undirected graphs. Similar arguments hold for directed graphs as well. Given a complete graph $G = (V, E)$ with a weight function $W : E \rightarrow \mathbb{R}$ its mean weight is:

$$w_k = \frac{2}{|V|(|V|-1)} \sum_{\forall e \in E(G)} w(e) \tag{7}$$

There exists a Hamiltonian cycle C' s.t.

$$\frac{1}{|C'|} \sum_{\forall e \in E(C')} w(e) \geq w_k \tag{8}$$

Assuming such a cycle does not exist, we have that for all Hamiltonian cycles $c_i$ in G:

$$\frac{1}{n} \sum_{e \in c_i} w(e) < w_k \tag{9}$$

Denote the multigraph $G'$: $V(G') = V(G)$ and $E(G') = E(G) \bigcup E'(G)$, where $E'(G)$ is a second copy of every edge in $E(G)F$, *i.e.* every edge is repeated twice. Lemma 3 of [17] states that the number of edge disjoint Hamiltonian cycles in a complete graph is $\lfloor \frac{n-1}{2} \rfloor$. We claim that on $G'$ there are exactly $n-1$ edge disjoint Hamiltonian cycles. This follows from the fact that $G$ has either 0 or $\frac{n-1}{2}$ "leftover" edges which do not participate in the Hamiltonian cycle cover, hence $G'$ has either 0 or $n$ leftover edges.

If there are 0 left over edges then there are $n-1$ cycles in the cover. In the case where there are $n$ leftover edges, another Hamiltonian cycle can be formed. This follows from the fact that after removing $2\frac{n-2}{2}$ Hamiltonian cycles, the degree of each vertex is exactly 2. Since the degree is even and equals 2, we have an Euler cycle which is also a Hamiltonian cycle. Resulting in:

$$\sum_{i=1}^{n} \sum_{e \in c_i} w(e) < n(n-1)w_k \tag{10}$$

However, the Hamiltonian cycles cover all of the edges of $G'$ and therefore the left hand side must equal the right. □

The relations between the mean cycle weight and the mean clique weight suggest that it is possible to find the maximum mean weight clique by looking for maximum mean weight cycles without overlooking any optimal mean clique. However, not all of the optimal cycles suggest the existence of an optimal clique. To overcome this, we can inspect the optimal cycles and prune those that do not indicate optimal mean cliques. In practice, we view those cases as pathological, hence skipping the pruning process.

When looking for more than one clique we may add the requirement that the sets of vertices of each clique are disjoint. This way each group of features (remember that each node in the graph corresponds to a variable) has no elements in common with other groups. Below we define the max-plus variant of PCA, where the requirement for disjoint cycles is translated to orthogonal principle components.

### 3.2. max-plus PCA formalization

The max-plus principle component problem, defined below, is designed to find a set of disjoint cycles such that the sum of their mean weight is maximized. The max-plus formalization is based on Theorem 1 above, which states the connection between an eigenvalue of a matrix $W$‘ and the max mean cycle in $G_{W‘}$.

**Problem 1 (max-plus PCA)** *Given a matrix $W$, find $k$ vectors $u_1, \ldots, u_k$ such that the criterion below is maximized subject to the given constraints.*

$$\max_{\{u_i\}} \sum_{i=1}^{k} \lambda_i \ , \quad \textit{subject to}: \tag{11}$$

$$\forall i \quad [W \otimes u_i]_{F(u_i)} = [\lambda_i \otimes u_i]_{F(u_i)} \tag{12}$$

$$\forall i \quad u_i^{\top} \otimes u_i = \boldsymbol{1} \tag{13}$$

$$\forall i \neq j \quad u_i^{\top} \otimes u_j = \boldsymbol{0} \ , \tag{14}$$

*where $F(u)$ is the set of the finite elements' indices in $u$,* i.e. $F(u) = \{j | u(j) > -\infty\}$*, and $[v]_S$ denotes the vector which contains the elements of $v$ with indices in the set $S$.*

The max-plus PCA problem is very similar in form to the classical PCA problem given in Eq. 4 above. The difference is that the vectors $u_i$ are not eigenvectors of $W$, rather, they are eigenvectors of the matrices one gets by considering only the rows and columns in the sets $F(u_i)$. As a result, if one eliminates from $W$ all rows and columns $j$ for which $u_i(j) = -\infty$, then one gets a matrix $W_i$ for which $[u_i]_{F(u_i)}$ is an eigenvector with an eigenvalue $\lambda_i$. This can be readily verified from Eq. 12, which does not constrain the rows of $W$ which are not in $F(u_i)$. Also, in the product of Eq. 12 every column $j$ of $W$ for which $u_i = \mathbf{0} = -\infty$ is being annihilated in the matrix product since it is max-plus-multiplied by the absorbing $\mathbf{0}$. Therefore, Eq. 12 implies $W_i \otimes [u_i]_{F(u_i)} = \lambda_i \otimes [u_i]_{F(u_i)}$. Hence, by Theorem 1, $\lambda_i$ is the maximal mean cycle on $W_i$.

The two other constrains of the max-plus PCA optimization problem, Eq. 13 and 14, are analog to the conventional PCA requirements for which each principle component has a norm of 1 and is orthogonal to the other principle components. The max-plus unit norm constraint (Eq. 13) states that the maximum element of each $u_i$ is zero. Without this constraint the optimization problem would be unbounded. The orthogonality constraint (Eq. 14) states that for every $i \neq j$ the conventional algebra sum of $u_i$ and $u_j$ is a vector of all $-\infty$. Since each $u_i$ is associated with a maximum mean-cycle on the set of nodes $F(u_i)$, orthogonality means that the set of all such cycles are vertex-disjoint.

Next we describe the inherent ambiguity in the computation of $u_i$ and the projections to the principle components. Then, on section 3.4 we describe some unorthodox constraints on the structure of the affinity matrix $W$. The discussion on the optimization of the max-plus PCA problem is deferred to section 3.5.

### 3.3. Ambiguity in the selection of $u_i$

Although the eigenvalue of a fully connected affinity matrix $W$ is unique, there are several possible eigenvectors, even for the case where only one optimal cycle exists. They can be characterized in the following manner:

**Theorem 3 (Eigenvector basis)** *Let $\lambda$ be the eigenvalue of $W$, and define $N := W - \lambda$. Assume that there is only one optimal mean cycle. Let $c$ be its set of vertices. Choose one reference node $r \in c$, and define the vector $x$ with elements*

$$x(i) = \textit{max weight of all simple paths in } N \textit{ from } r \textit{ to } i. \tag{15}$$

*$x$ is an eigenvector of $W$. Moreover, the set of all possible $x$ of this form is a basis of the eigenvector space of $W$.*

**Proof:** This is a less general version of [20] Theorem 1, [13] Theorem 13. □

Therefore, for each $\lambda_i$, there are many vectors $u_i$ that satisfy Eq. 12. One can handle this ambiguity in several ways. One option is to construct an over-complete set of principle components by replacing $u_i$ with the set of all possible vectors that satisfy Eq. 12. This makes sense, since the number of pooling cells in biological systems is typically large.

In this work we choose to deal with the inherent ambiguity by selecting one specific principle component. This principle component is associated with the term *head variable*. Recall that C-cell type features are motivated by the need to have stable features, which are constructed from a subset of the given variables. Within this subset we define the *head variable* ($r$) as the variable with the maximal out-weight, *i.e.* the one for which the sum of weights of the out-edges is maximal. We consider this variable to be the most easily verifiable variable in the subset, since the values of others depend on it. This makes the *head variable* a suitable core for a construction of a stable feature.

The projection vector $u_i$ is constructed as follows: for all entries of vertices which are not included in the optimal cycle a $\mathbf{0}$ value is given. Next, we define the *dominant variable* as the element with the highest value in $u_i$. It corresponds to the node (feature) with the longest (normalized) path starting from $r$, *i.e.* it is the one that most nodes (features) depend on. The entry which corresponds to the *dominant variable* is given a value of 0, which is the maximal value of $u_i$. The rest of the values are given according to the eigenvector $x$, which is obtained by calculating the maximal path lengths in the normalized matrix $N$ to the *head variable*, *i.e.* the vector $x$ defined in Eq. 15 where the reference variable $r$ is taken to be the *head variable*.

From the construction of the matrtix $N$ of Theorem 3, the elements of vector $x$ hold the optimal sum of affinities on a path to the *head variable* minus the length of the path times $\lambda$. Therefore, they serve as normalized measures of connectivity between variables.

The projection of an input vector $x$ to the principle component $u_i$ is given by the inner product $x \otimes u_i$. It boils down to a different inhibition of each value of $x$ . Variables not in $F(u_i)$ are completely inhibited. The *dominant variable* is not inhibited at all, and the rest are inhibited by the measures of the best-path affinity to the *head variable*. After the

inhibition step takes place, the maximum over the variables is taken as the resulting pooled feature.

### 3.4. Asymmetric affinity matrices

Consider a set of cycles on a given $n \times n$ matrix $W$. The diagonal elements of $W$ define cycles of length one. These trivial cycles are not of much interest to us. We therefore eliminate these cycles by modifying $W$ such that $W(i,i) = \mathbf{0}$, $i = 1..n$.

Assuming further that $W$ is symmetric, it can be readily verified that the optimal mean-cycles of $G_W$ are cycles of length two, for which $W(i,j) = W(j,i)$ is maximal across all the edge weights. These cycles have a mean of $W(i,j)$, and since no other edge has a higher weight every other mean cycle weight is lower. We therefore conclude that the max-plus PCA problem is most interesting when $W$ is asymmetric.

While there are many similarity functions that are asymmetric, the use of asymmetric affinity matrices is unorthodox. We could not find any reference for using asymmetric affinity matrices for dimensionality reduction. The underlying reason is the numerical advantages that symmetric matrices hold in the conventional algebra over $\mathbb{R}$, such as having a real eigenvalue. In the max-plus semiring all eigenvalues are real. Moreover, they are equal for $W$ and for $W^\top$.

**Proposition 1** *In the max-plus semiring, the right eigenvalue and left eigenvalue of a fully connected matrix $W$ are the same real number.*

**Proof:** Recall from Theorem 1 that a fully connected adjacency matrix $W$ has a unique real eigenvalue which equals the maximum mean cycle weight. Since the maximum mean cycle of $G_W$ and $G_{W^\top}$ are the same up to a flip of directions, the right and left eigenvalues are the same. □

The specific asymmetric affinity matrix measure we use in our experiments is a classical score used to measure the lack of uncertainty between two random variables. It is called "coefficients of uncertainty" [24] (or "coefficients of constrains" [8]). For two random variables, $X$ and $Y$, it is defined as

$$C_{XY} := \frac{I(X;Y)}{H(X)}, \tag{16}$$

where $I(X;Y)$ is the mutual information of $X$ and $Y$, and $H(X)$ is the entropy of $X$.

In our experiments (section 4), the $W(j,i)$ is an estimation of the coefficient of uncertainty of variables $i$ and $j$ (this way dependencies follow the arrow directions as in conventional graphical models). Given two such variable, $x^{(i)}$ and $x^{(j)}$, each as a series of $N$ measurements, we estimate the mutual information and the entropy based on their contingency table [24].

### 3.5. A greedy algorithm for max-plus PCA

Unlike the conventional PCA problem, the max-plus PCA problem above is NP-complete since it can be reduced to the problem of maximal weight clique cover [6]. This is no surprise, considering its similarity to the NP-complete clustering problem [3]. We therefore suggest the use of an efficient greedy algorithm.

**Algorithm 1 (Greedy solution for max-plus PCA)** *The following algorithm is used to approximate Problem 1. Given a matrix $W$, perform the following steps, for i = 1..k:*

1. *Using a variation of Karp's [16] algorithm obtain: $\lambda_i$, the mean weight of the max mean weight cycle in $G_A$, and the list of its vertices $c = (i, \ldots, j, i)$.*
2. *Identify the dominant variable $r \in c$ (section 3.3), as the one with the max out-weight in $W$.*
3. *run a variation of the FloydWarshall algorithm [20] on the subgraph of $W - \lambda$ induced by the vertices of $c$ to obtain the max normalized path weights vector $x$ (Eq. 15).*
4. *Define $u_i$ in respect to vector $x$ (section 3.3) on the entries with indices in $c$, **0** elsewhere.*
5. *Remove the rows and columns that correspond to the vertices of $c$ from $W$.*
6. *If the size of $W$ is smaller than 2 (or if $i == k$) return.*

### 3.6. Sparsity

After the principal components have been calculated we represent each input vector $x$ by its correlation (measured in max-plus) to the principal components. Similarly to the conventional PCA, this is achieved by inner products of the form $u_i^\top \otimes x$.

A sparse principle component is defined as a principle component with a small number of non-zero elements. The same definition holds naturally in the conventional algebra over $\mathbb{R}$ and in the max-plus semiring (with $\mathbf{0}$). For conventional PCA, attempts have been made to modify it in order to compute a set of sparse principle components *e.g.* [15, 31]. The max-plus PCA, however, is sparse by nature, because it is hard to build long cycles with high mean weights.

## 4. Experiments

We evaluated the performance of max-plus PCA on several data-sets. In our experiments we used a three-layered hierarchical model that can be seen as a variant of the HMAX model [25]. It contains the input layer, a layer of

S-cells that performs linear combinations and a top C-layer which is based on the MAX operator.

Each data-set was first reduced to a dimensionality of 100 using conventional PCA. The projections to the resulting principle components were normalized to have a variance of 1 (note that max-plus PCA is not scale invariant). The resulting features formed the S-layer. Then, the coefficients of uncertainty (section 3.4) were computed between the variables of the S-layer, and the resulting affinity matrix was used to compute the max-plus PCA. The (max-plus) projection to the set of max-plus principle components formed the top C-layer.

In order to avoid having to select the total number of max-plus principle components, we decided to take the maximum number, *i.e*. run the greedy algorithm until all variables are included in principle components. Since the sets of finite elements in each principle component $u_i$ are disjoint, this number is bounded. For $100$ input features, this procedure typically resulted in around $25$ principle components.

Recently, Wolf *et al*. [30], showed that classifying using just the top layer of the hierarchy is sub-optimal. We therefore use all the features of the hierarchy, and not just the top layer. We did not, however, try the different combination strategies suggested in [30]. To combine the features we only used a simple concatenation of the layers' features.

### 4.1. Caltech image recognition data sets

The first data-sets we used were the Caltech image data sets [9]. The data sets: Airplanes, Cars, Faces, Leaves and Motorbikes, as well as the background images were downloaded from `http://www.vision.caltech.edu/`. In each experiment the task is to distinguish between images containing an object and background images that do not contain the object. We used the predefined splits (available to all the data sets but the Leaves data set). For Leaves, we used a random split of 50% training and 50% testing.

As features we used the bag-of-keypoints method of Csurka *et al*., which uses Lowe's SIFT features [18]. First, we used Lowe's binaries, available at `http://www.cs.ubc.ca/~lowe/keypoints/` to find keypoints and compute their SIFT descriptors. We then clustered all the descriptors of the keypoints from the training images to $1000$ clusters, using k-means. Given an image I, we assigned each one of its keypoints to the cluster with the nearest mean. Each image was represented by the histogram of these cluster assignments, *i.e*. by a vector $x \in \mathbb{R}^{1000}$.

We ran PCA on the training examples and computed the first 100 principle components. The projection of $x$ to these principle components formed the S-layer of our hierarchy. We applied max-plus PCA to the S-layer of the training images and obrained max-plus principle components. The max-plus projection to these formed the C-layer.

| Algorithm | Planes | Cars | Faces | Leaves | Mbikes |
|---|---|---|---|---|---|
| Base | 20.10 | 6.88 | 26.04 | 26.84 | 13.46 |
| Base + PCA | 18.98 | 6.88 | 25.81 | 27.89 | 16.19 |
| All three | 14.83 | 6.50 | 17.51 | 18.42 | 11.54 |

Table 1. Error rates (percents) on 5 Caltech data sets. Results are reported for linear SVM classification applied to (1) Bag of SIFT features ;(2) an hierarchy of bag of SIFT features and PCA ;and (3) to a hierarchy of Bag of SIFT features, PCA and max-plus PCA.

In all of our experiments we used a linear SVM. For the testing and training splits reported in [9] we achieved an error rate as reported in table 1. Having a two layer hierarchy (with conventional PCA only) does not improve performance. However, performance is improved significantly when max-plus PCA is added.

### 4.2. FERET face recognition

We performed a set of face recognition experiments using a partial replica of the CSU Face Identification Evaluation System [7], which implements the FERET test [22] for semi-automatic face recognition algorithms with a few minor modifications. The CSU system preprocesses the images by registering eye coordinates, cropping an elliptical mask to exclude non-face regions and then equalizing the histogram of gray level intensities. For each face descriptor tested, a distance matrix is generated that contains a measure of the similarity between all pairs of images in the dataset. These distance matrices are then used to test different probe and gallery image sets to evaluate the performance of various algorithms.

In order to allow for fast testing, we reimplemented a part of the CSU system called "the permutation tool" in Matlab. We have not run the other tests in the CSU system yet. The permutation tool uses a subset of FERET that contains 160 unique subjects, each with 4 images. On each iteration of the test, one probe image and one gallery image is chosen for each of the 160 subjects, and the result is marked correct if the shortest distance from a given probe is to the gallery image of the same subject. The permutation test runs for 10,000 iteration and returns the mean recognition rate and the standard deviation.

Three face representations were compared. The first one is composed out of the preprocessed gray level images. The second was obtained by convolving the images with Gabor filters at four orientations and three different scales. The third representation was Local Binary Patterns (LBP) [19], which were previously shown to be effective face recognition descriptors [1]. In all three experiments Euclidean distances between the image representations were used. The results are shown in Table 2. As can be seen max-plus PCA on top of PCA improves performance, whereas PCA by itself improves the results only slightly.

| Features | Base | Base + PCA | Base + PCA + max-plus PCA |
|---|---|---|---|
| Graylevel image | 63.5% | 64.9% | 65.7% |
| Gabor filters | 70.9% | 71.0% | 73.4% |
| Local Binary Patterns | 75.1% | 75.8% | 78.1% |

Table 2. Success rate for several face representations on the FERET data-set, using the permutation tool of the CSU system. For each face representation three scores are presented: the one of the original feature, the score of the original + PCA, and the results of the original + PCA + max-plus PCA. The standard deviation was 2-3% for all experiments, since 10,000 repetitions were performed, the differences are significant.

## 5. Future work

Experimentally, we wish to test our methods on hierarchies with more than two complex layers. Computationally, we wish to go beyond the problem of unsupervised dimensionality reduction, and develop max-plus methods for supervised learning, clustering and density estimation.

## Acknowledgments

LW is supported by the Israel Science Foundation (grants No. 1440/06, 1214/06), and by the Colton Foundation.

## References

[1] T. Ahonen, A. Hadid, and M. Pietikainen. Face recognition with local binary patterns. In *ECCV*, 2004.

[2] F. Baccelli, G. Cohen, G. Olsder, and J. Quadrat. *Synchronization and Linearity*. Wiley, 1992.

[3] J. Barthelemy and B. Leclerc. The median procedure for partitions. *DIMACS Series in Discrete Mathematics*, 1995.

[4] B. Birnbaum and K. Goldman. An improved analysis for a greedy remote-clique algorithm using factor-revealing lps. Wucse -2006-3, Dep. CS and Eng Washington U. in St. Louis, 2006.

[5] B. Birnbaum, J. Turner, and K. Goldman. The remote-clique problem revisited. Master's thesis, Dep. of CS and Eng Washington U. in St. Louis, 2006.

[6] B.Manthey. On approximating restricted cycle covers. In *Workshop on Approximation and Online Algorithms (WAOA)*, 2005.

[7] D. Bolme, J. Beveridge, M. Teixeira, and B. Draper. The csu face identification evaluation system: Its purpose, features and structure. In *Int. Conf. on Vision Systems*, 2003.

[8] C. Coombs, R. M. Dawes, and A. Tversky. *Mathematical Psychology: An Elementary Introduction*. Prentice-Hall, Englewood Cliffs, NJ, 1970.

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

[10] P. Földiák. Learning invariance from transformation sequences. *Neural Comp.*, 3:194–200, 1991.

[11] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Computer Science and Scientific Computing. Academic Press, Inc., 1990.

[12] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cyb.*, 36, 1980.

[13] S. Gaubert and M. Plus. Methods and applications of (max,+) linear algebra. In *STACS'97*, 1997.

[14] D. H. Hubel and T. N. Wiesel. Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat. *J. Neurophys.*, 28:229–289, 1965.

[15] I. Jolliffe and M. Uddin. A modifed principal component technique based on the lasso. *Journal of Computational and Graphical Statistics*, 12:531–547, 2003.

[16] R. Karp. A characterization of the minimum mean-cycle in a digraph. *Discrete Maths.*, 23:309–311, 1978.

[17] G. Li and C. Chen. Disjoint hamiltonian cycles in graphs. *Australas. J. Combin.*, 19:83–89, 1999.

[18] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[19] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29:51–59, 1996.

[20] G. Olsder, K. Roos, and R. van Egmond. An efficient algorithm for critical circuits and finite eigenvectors in the max-plus algebra. *Linear Algebra and its Applications*, 1999.

[21] D. Perrett and M. Oram. Neurophysiology of shape processing. *Img.Vis.Comput.*, 11:317–333, 1993.

[22] P. Phillips et-al. The feret evaluation methodology for face recognition algorithms. *PAMI*, 2002.

[23] T. Poggio and S. Edelman. A network that learns to recognize 3D objects. *Nature*, 343:263–266, 1990.

[24] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Combridge University Press, NY, 1992.

[25] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nat. Neurosci.*, 2:1019–1025, 1999.

[26] H. Schneiderman and T. Kanade. A statistical method for 3D object detection applied to faces and cars. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 746–751, 2000.

[27] B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods—Support Vector Learning*, pages 327–352. MIT Press, Cambridge, MA, 1999.

[28] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *CVPR*, 2005.

[29] S. Ullman and E. Sali. Object classification using a fragment-based representation. In *Proc. of Biologically Motivated Computer Vision*, pages 73–87, Seoul, Korea, 2000.

[30] L. Wolf, S. Bileschi, and E. Meyers. Perception strategies in hierarchical vision systems. In *CVPR*, 2006.

[31] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. Technical report, Statistics Department, Stanford University, 2004.