Image Recognition With Occlusions

Tyng-Luh Liu	Davi Geiger	Robert Hummel
Courant Institute	Courant Institute	Courant Institute
New York University	New York University	New York University
liutyng@cs.nyu.edu	geiger@cs.nyu.edu	hummel@cs.nyu.edu

September 28, 1995

Abstract

We study the problem of how to detect "interesting objects" appeared in a given image, I. Our approach is to treat it as a function approximation problem based on an over-redundant basis. We first establish a template library, say $\mathcal{L}$, then "optimally" decompose I into a linear combination of templates $\tau_j's \in \mathcal{L}$ after applying some suitable transformation A_i to each selected τ_j. We can write this as follows

$$I = \sum_{j=1}^{M} \sum_{i=1}^{N} c_{ij} A_i(\tau_j)\,,$$

where $c_{ij}'s$ are the decomposition coefficients to be found. Since the template library is over-redundant, there are infinite possible sets of $\{c_{ij}\}$ that can "decompose" the image I. To select the "best" decomposition c_{ij} we first propose a global optimization procedure that considers the "L^p norm" $(\sum_j \sum_i |c_{ij}|^p)$ with $p \leq 1$. This concave cost function selects as few coefficients as possible (sparse representation of the image) and handle occlusions, however it contains multiple local minima. We identify all local minima so that a global optimization is possible by visiting all of them (in the case of $p = 1$ linear programming can be applied). Secondly, because of the number of local minima grows exponetially with the number of templates, we investigate a greedy and iterative "L^p Matching Pursuit" strategy. At each stage, say stage n, we select the template τ_j, an (affine) transformation A_i, and coefficient c_{ij} that minimizes the image residue $||R^n I||_{L^p} = ||R^{n-1} I - c_{ij} A_i(\tau_j)||_{L^p}$. The image I itself is the image residue at the first stage, i.e. $R^0 I = I$. Because of occlusions, special attention is devoted to the overlapping regions. We show results for object recognition with occlusions.

1 Introduction

In the field of signal processing and computer vision an input signal or image is a function f over some subset of $\mathbb{R}$ or $\mathbb{R}^2$. In order to manipulate and analyze f, it is useful to introduce a linear decomposition into basis elements f_j, i.e.,

$$f = \sum_j c_j f_j \,.$$

Such a decomposition can allow not only a compact representation of f but also aid in understanding important aspects of the signal or image. An example of a well known and useful decomposition of this type is the Fourier series expansion. We note that in the Fourier case the elements f_j form a basis (an orthogonal one), thus, there is a unique decomposition, i.e. a unique choice for c_j.

We study the object recognition problem via a robust template decomposition approach. Our main concern is to represent "interest objects" that appear in a given image with a linear combination of image templates from a well established library of templates. The term "interest objects" depends solely on the kind of application. For example, if the application is to recognize faces then, we should have a template library with a lot of faces (and/or features of faces). Let the image to be recognized be I and the template library be $\mathcal{L}$. The task of image recognition is reduced to a function approximation problem of the form

$$I(x) = \sum_j \sum_i c_{ij} A_i(\tau_j)(x) = \sum_{i,j} c_{ij} T_{ij}(x) \tag{1}$$

where $\tau_j \in \mathcal{L}$, $T_{ij} = A_i(\tau_j)$ denotes an affine transformation applied to the template τ_j, and c_{ij} is the choice of coefficients that "best" decompose the image. Typically the library $\mathcal{L}$ is large, in order to accommodate many possible situations and we also have to consider the possible (affine) transformations. Thus, we have an over-redundant basis leading to infinite many solutions, c_{ij}, to this problem. That is not the case for the Fourier decomposition.

The mathematical problem of function decomposition with over-redundant basis can be illustrated as follows. Say the library consists of sinusoids and functions of the form $1/(k+x)$ (where k varies over $\mathbb{N}$). Let us assume that our target function (our image) is accurately represented by the function

$$f(x) = \sin 2x + \frac{4}{(3+x)} \,. \tag{2}$$

It is clear that only two terms from the prototype library are required to represent $f(x)$. However, one could write $f(x)$ using either sinusoids alone or as combinations of $1/(k+x)$ alone, but either representation would require many terms. The problem is to formulate a coefficient selection criterion and a method to compute the coefficients that yields compact representations.

1.1 Coefficient selection criteria

Coefficient selection may be viewed as a constrained optimization problem. One is given the prototype library $\{\tau_j\}$, a set of possible (affine) transformations on the templates $\{A_i\}$, an objective function F (discussed in the following), and the function (image) I that is to be decomposed. The task is to select the coefficients $c = (c_{ij})$, so that the objective function $F(c)$ is minimized subject to the constraint given by (1), i.e. $I(x) = \sum_j \sum_i c_{ij} A_i(\tau_j)(x)$.

The purpose of the objective function F is to select a best representation, c^*, from among all solutions c that satisfy the constraint. One may think of the value $F(c)$ as being the "cost" associated with using the representation (1).

We want the cost associated with adopting a term $A_i(\tau_j)$ in the representation for I ($c_{ij} \neq 0$) to be distinctly high, relative to the cost of not using the template at all ($c_{ij} = 0$) . This favors the selection of economical (minimal) representations. Second, the cost should escalate with the magnitude of c_{ij}, but should not dominate the first condition, i.e., the rate of increase in cost as a function of $|c_{ij}|$ should decrease. This condition allows us to model noise in the function via special "noise templates." We need to allow for noise in our decompositions, but prefer representations that can be explained with minimal noise content. This condition leads us naturally to consider concave objective functions.

The experimental results presented in this paper utilize primarily the objective function

$$F_p(c) = \sum_{j=1}^{M} \sum_{i=1}^{N} |c_{ij}|^p , \tag{3}$$

where N is the number of possible (affine) transformations and M is the size of the template library. We also consider the p-norm error minimization, which is closely related to this cost function. Moreover, we do consider variations of this cost function when applying the greedy matching pursuit strategy. Coifman and Wickerhauser [?], working in another domain, have proposed an entropy like function, $\sum_{i,j} |c_{ij}|^2 log|c_{ij}|^2$. Despite its attractive conceptual origin—minimizing the amount of information—it is not directly applicable to our situation since for us the coefficients c_{ij} are free and are not guaranteed to square-sum to 1 (as is required in their formulation).

1.2 The optimization problem

For the cost (3), if the parameter p is greater than 1, then the objective function F_p is convex and so the minimization problem has a unique local minimum which is also the global minimum. Standard numerical descent techniques can be used to perform this minimization. Since larger values of $|c_{ij}|$ are heavily penalized, this objective function will tend to distribute the weights c_{ij} across as many of

the $A_i(\tau_j)$ as possible. In fact, as $p \to \infty$ the minimizing solution is exactly the one with the largest $|c_{ij}|$ as small as possible. In this case the representation (1) is in some sense maximally stable with respect to arbitrary prototype library element removal.

Alternatively, we consider the situation with $p < 1$. Now the objective function is non-convex, and in fact the optimization problem will generally have multiple local minima, making the optimization more difficult. An objective function of this form will tend to coalesce the weights c_{ij} onto as few $A_i(\tau_j)$ as possible, providing an efficient representation (few non-zero c_{ij}'s). This is the type of representation desired in the example given in (2), and is the chief interest of the present work. Thus, the template matching problems lead naturally to the problem of minimizing

$$F_p(c) = \sum_{j=1}^{M} \sum_{i=1}^{N} |c_{ij}|^p, \qquad 0 < p < 1, \tag{4}$$

subject to the constraint (1), i.e., $I(x) = \sum_j \sum_i c_{ij} A_i(\tau_j)(x)$.

We will show that it is possible to charactherize all local minima and obtain the global one by visiting them. In the case of $p = 1$ a linear programming technique can be applied. Since the number of local minima grows exponential with the size of the template library we consider an alternative greedy algorithm.

1.3 Matching Pursuit

Inspired by Mallat and Zhang's work [8] we consider a matching pursuit strategy where, at each stage, the criteria of best selection is based on minimizing an image residue. In regression statistics, this decomposition method is known as *Projection Pursuit Regression* , a non-parametric method that is concerned with "interesting" projections of high dimensional data (see Friedman and Stuetzle [5], Huber [6]).

The original matching pursuit is based on the standard L^2 (Hilbert space) method. In recognition of image with occlusions, the L^2 norm is not suitable. We propose an L^p matching pursuit with $0 < p < 1$, to improve the robustnesee. With $0 < p < 1$, we lost the sturture of inner product but the notion of projection can be recaptured via the values of cost function, that is, the criterion for a template to be "best matching" or "closest" to the image is to minimize the values of a cost function. We will adopt the term "L^p norm" though it is not really a norm.

Our algorithm is then a multi-stage iterative algorithm that at each stage we apply the L^p matching pursuit to find a "best-matching" template (using the previous results). The image is updated by extracting the object matched by the selected template (see the algorithm section for details).

To test the robustness of the L^p norm, we also consider a cost function built from a robust regression estimator, namely the LTS-method (Least Trimmed Squares, Rousseuw 1983, 1984, [10]).

It is a projection method with the ability to distinguish corrupted data from correct ones. The basic idea is to project a testing image to each templates in the library then pick up a template with the minimal LTS cost to be the best matching one. Simulation results for both the L^p and LTS matching pursuit are shown in the end of this paper.

2 Template Library

To begin with, we must first establish a well-defined, over-redundant library of templates for some specific application.

By a redundant library, we mean it contains many non-canonical templates as well as one canonical template. A canonical template is a trivial template with zero gray-level value pixels everywhere except one pixel at the extreme left and top corner that its gray-level value is 1. Moreover, we will assume we can apply a set of affine transformations to each template, e.g., translation. Clearly, this single canonical template plus a set of all translations form a basis for the image space. Other templates are said to be non-canonical and consist of images of interesting objects. Also, it is convenient, on most occasions, to require $\mathcal{L}$ to be complete and well-defined so that no template in $\mathcal{L}$ can be matched by other template also from $\mathcal{L}$ over half of its total pixels.

2.1 Template and image coordinates

Suppose we have now created a well-defined complete template library $\mathcal{L} = \{\tau_j : j = 1...M\}$ for some application, where we will use $\epsilon_1 \equiv \tau_1$ to represent the canonical template. We still have to consider all possible affine transformations A_i for each template τ_j. In general, these affine transformations include all possible translations and possibly linear transformations to account for scale and changes of viewing the scene Let the image to be recognized be I of dimension N and each template τ_j be of dimension N_T (the dimension of an image is just the image lattice size, i.e. the total number of pixels, and to simplify, we assume that both N and N_T are perfect square numbers). In this paper, we only study the case that A_i is a translation and thus, the lattice size and the number of possible A_i's are the same, N. Furthermore, let $P = \{p_1, p_2, \cdots, p_N\}$ and $Q = \{q_1, q_2, \cdots, q_{N_T}\}$ be the pixel sets of I and any τ_j, respectively. (We order the pixels from top to bottom and left to right.) Since we choose to represent the pixel sets in one dimensional form, the issue of mappings between P and Q resulted from shifting some τ_j over I needed to be addressed clearly. Let $A_i(\tau_j)$ now represent a translation on template τ_j such that its first pixel is positioned at the i-th pixel $p_i \in P$ (see Figure 1). For the sake of simplicity, let's only consider a translation A_i acting on template τ_j such that no

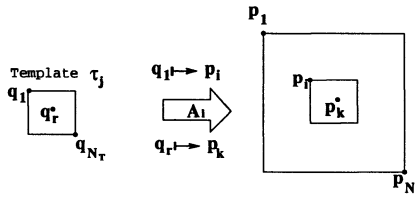


Figure 1: The pixel correspondences between I and $T_\lambda = T_{ij} = A_i(\tau_j)$. We see that pixel q_1 is positioned on p_i and q_r on p_k, respectively.

portion of Q is shifted outside P. We can explicitly describe such relation as follows:

$$Q \xrightarrow{A_i} \Omega_i = \{p_k : k \in \Gamma_i \,,\, \Gamma_i \text{ is an index set} \subset \{1, 2, ..., N\}\} \subset P. \tag{5}$$

The mapping formula for A_i is such that $q_r \in Q \mapsto p_k = p_{k(r,i)} \in \Omega_i$ where [1]

$$k = i + (\lfloor \frac{r-1}{\sqrt{N_T}} \rfloor \times N) + (r - 1 - \lfloor \frac{r-1}{\sqrt{N_T}} \rfloor \times \sqrt{N_T}) \,.$$

Denote $T_{ij} = A_i(\tau_j)$ and $e_{i1} = T_{i1} = A_i(\epsilon_1)$ [2], then we have $T_{ij}(p_k) = \tau_j(q_r)$. Using these notations, one may prefer to write the decomposition equation (1) as

$$\begin{aligned} I(p_k) &= \sum_{i=1}^{N} c_{i1} e_{i1}(p_k) + \sum_{j=2}^{M} \sum_{i=1}^{N} c_{ij} T_{ij}(p_k) \\ &= \sum_{\lambda=1}^{N} c_\lambda e_\lambda(p_k) + \sum_{\lambda=N+1}^{M.N} c_\lambda T_\lambda(p_k) \end{aligned} \tag{6}$$

where $\lambda = \lambda(i,j) = (j-1) \times N + i$ and p_k is the *k-th* pixel of the image. From now on, we will mostly abide by the more compact form of single index representation such as in Equation (6). Also, we may write $I[k]$ in stead of $I(p_k)$ for simplification.

[1] The expression $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x.

[2] Note that $e_i(p_j) = \delta_{ij}$, where $\delta_{ij} = 1$ for $i = j$ and $\delta_{ij} = 0$ otherwise.

3 Optimization problem and solution

Equation (6) can be written in matrix notation as $\mathrm{T}\hat{c} = \mathrm{I}$ where

$$\mathrm{T} = \begin{pmatrix} e_1(p_1) & \cdots & e_N(p_1) & T_{N+1}(p_1) & \cdots & T_{MN}(p_1) \\ e_1(p_2) & \cdots & e_N(p_2) & T_{N+1}(p_2) & \cdots & T_{MN}(p_2) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ e_1(p_N) & \cdots & e_N(p_N) & T_{N+1}(p_N) & \cdots & T_{MN}(p_N) \end{pmatrix},$$

$$\hat{c} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{MN} \end{pmatrix} \quad \text{and} \quad \mathrm{I} = \begin{pmatrix} I(p_1) \\ I(p_2) \\ \vdots \\ I(p_N) \end{pmatrix}.$$

Note that $e_i(p_j) = \delta_{ij}$, where $\delta_{ij} = 1$ for $i = j$ and $\delta_{ij} = 0$ otherwise. If the prototype library forms a basis (linearly independent), then $M = 1$, and there is no freedom in choosing the coefficients (c_j); the coefficients are uniquely determined by the constraint. If there are linear dependencies in the prototype library, then $M > 1$, the prototype library over-spans, and the set of all solutions (c_j) to the constraint form an $(M-1)\,N$ dimensional affine subspace in the $M.N$-dimensional coefficient space. Let S denote this solution space, i.e., $\dim(S) = (M-1)\,N$.

The task to detect interest objects embedded in I by a linear combination of templates from $\mathcal{L}$ can be formulated as solving the following optimization problem:

$$\underset{\hat{c}}{Min}\, F_p(\hat{c}) \qquad \text{subject to the constraint } \mathrm{T}\hat{c} = \mathrm{I} \tag{7}$$

where $\mathrm{T} \in \mathbb{R}^{N \times M.N}$, $\hat{c} \in \mathbb{R}^{M.N}$, $\mathrm{I} \in \mathbb{R}^N$, $M > 1$. The constraint space, S, is the set of all $\hat{c}$ satisfying $\mathrm{T}\hat{c} = \mathrm{I}$, and is an affine subspace of dimension $(M-1)\,N$. We will first study the L^p-cost function (4). It is natural when analyzing F_p in (4) as a function in the coefficient space $\langle c_i \rangle$ to decompose the domain into orthants, where each coefficient is of constant sign. This allows the removal of the absolute values in (4), so we may treat F_p as a smooth function inside each octhant. For example, if we consider the restriction of F_p to the orthant consisting of all points c such that $c_1 < 0$, $c_2 < 0$, and $c_i > 0$ for $i \geq 3$, then (4) can be written

$$F_p(c) = (-c_1)^p + (-c_2)^p + \sum_{i=3}^{MN} c_i^p.$$

Moreover, it is clear that $F_p(c) \to \infty$ as $\|c\| \to \infty$, so for minimization purposes it suffices to consider bounded c. The bound will depend upon the constraint equation (1), but, for example, if c_0 is any solution to (1), then it suffices to consider only those c satisfying $|c_i| \leq (F_p(c_0))^{1/p}$ for all i.

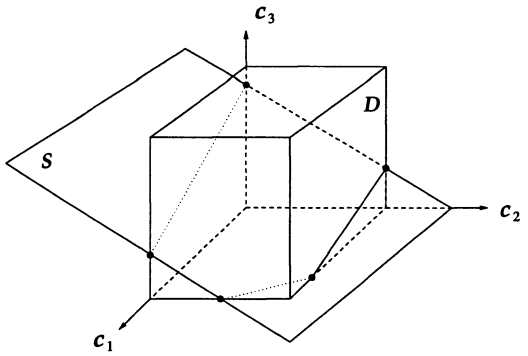


Figure 2: Illustration of a domain restriction polytope obtained from the intersection of a 2 dimensional constraint space S with a rectangular solid bound domain D in a 3 dimensional coefficient space. In this example the intersection is a non-regular pentagon. If the restricted objective function F is concave, then its local minima occur at the vertices of the pentagon.

When combined with the restriction to orthants, we have a decomposition of the pertinent domain of F_p into $M.N$-dimensional cubes of edge length $(F_p(c_0))^{1/p}$.

The intersection of the constraint space S with these domain cubes gives rise to convex polytopes, as illustrated in Figure 2. The system of domain restrictions can be written out explicitly. For the first (positive) orthant they are

$$\begin{aligned} Ac &= b \\ c_i &\leq d_i, \qquad 1 \leq i \leq M.N \qquad (8) \\ -c_i &\leq 0, \qquad 1 \leq i \leq M.N, \qquad (9) \end{aligned}$$

where previously we considered the case with the d_i's a single constant at least as large as $(F_p(c_0))^{1/p}$.

The relation $c_1 = (1, 0, \ldots, 0)^t \cdot c \leq d_1$ describes a half-space in the space $\langle c \rangle$, and the entire collection (8) and (9) together describe the intersection of $2MN$ halfspaces, i.e., a polytope with at most $2MN$ faces. The general inequality defining a half-space is $v \cdot c \leq d_i$, where v is a vector normal to the bounding hyperplane, and d_i determines an offset from the origin. So an arbitrary convex polytope having N' faces can be described in the form $Bc \leq d$, where $B \in \mathbb{R}^{N' \times M.N}$, $d \in \mathbb{R}^{N'}$, and the inequality is interpreted coordinatewise. So the generalized constraint relations can be written:

$$\begin{aligned} Ac &= b \\ Bc &\leq d. \end{aligned} \qquad (10)$$

The relations (10) can be viewed as defining a polytope inside the affine space S. If we were

to perform a basis transformation to obtain coordinates conducive to representations inside S, then F_p under the same transformation would loose its simple form. Even without this consideration, it is useful to study more general objective functions. The specific property of F_p of interest to us is *concavity.* A function F mapping from a convex domain Ω of a vector space X to $\mathbb{R}$ is concave if

$$F(\beta x + (1-\beta)y) \geq \beta F(x) + (1-\beta)F(y)$$

for all x and y in Ω and $\beta \in [0,1]$. The result we desire (Proposition 1) actually requires only a weaker property, which we call *pseudo-concave.* A function $F : \Omega \to \mathbb{R}$ as above is pseudo-concave if

$$F(\beta x + (1-\beta)y) \geq Min\{F(x), F(y)\}$$

for all x and y in Ω and $\beta \in [0,1]$. Clearly any concave function is also pseudo-concave.

Proposition 1 *Let Ω be a closed, bounded, convex polytope in a vector space X and let $F : \Omega \to \mathbb{R}$ be pseudo-concave. Then the global minimum of F on Ω occurs at a vertex of Ω.*

Proof: Let $x \in \Omega$. We will show that there is a vertex v of Ω such that $F(x) \geq F(v)$, from which the proposition follows.

It is well known that x can be represented as a convex combination of vertices of Ω, i.e., there exists vertices $v_1, v_2, \ldots, v_n$ of Ω and corresponding strictly positive coefficients $a_1, a_2, \ldots, a_n$, with $\sum_{i=1}^{n} a_i = 1$, such that

$$x = \sum_{i=1}^{n} a_i v_i.$$

If $n = 1$, then $a_1 = 1$, and so $x = v_1$ is a vertex and there is nothing to prove. Otherwise we have by pseudo-concavity that

$$\begin{aligned} F(x) &= F\left(a_1 v_1 + (1-a_1)\sum_{i=2}^{n} \frac{a_i}{1-a_1} v_i\right) \\ &\geq Min\left\{F(v_1), F\left(\textstyle\sum_{i=2}^{n} a_i v_i/(1-a_1)\right)\right\}. \end{aligned} \tag{11}$$

If $n = 2$, then $a_2 = 1 - a_1$, so $F(x) \geq Min\{F(v_1), F(v_2)\}$, and we are finished. Otherwise simply iterate the decomposition on the last term in (11) until

$$F(x) \geq Min\{F(v_1), F(v_2), \ldots, F(v_n)\}$$

is obtained. □

4 One template matching and simulations

In many cases we do not want the representation (6) to use more than one template. For example, if we want to find a specific face in an image, then it suffices to use only one face-template. In these cases the non-canonical template represents a key feature and the canonical templates e_λ represents non-interest elements , e.g., noise. Let us consider a template τ_2 of size N_T ($\tau_1 \equiv \epsilon_1$) and a particular translation, A_i, with i fixed. In this case the equation (1) can be restrcited to

$$\begin{pmatrix} e_i(p_i) & \cdots & e_{i+N_T-1}(p_i) & T_{N+1}(p_i) \\ e_i(p_{i+1}) & \cdots & e_{i+N_T-1}(p_{i+1}) & T_{N+1}(p_{i+1}) \\ \vdots & \ddots & \vdots & \vdots \\ e_i(p_{i+N_T-1}) & \cdots & e_{i+N_T-1}(p_{i+N_T-1}) & T_{N+1}(p_{i+N_T-1}) \end{pmatrix} \begin{pmatrix} c_i \\ \vdots \\ c_{i+N_T-1} \\ c_{N+1} \end{pmatrix} = \begin{pmatrix} I(p_i) \\ \vdots \\ I(p_{i+N_T-1}) \end{pmatrix}.$$

where again, $e_i(p_j) = \delta_{ij}$. Thus, we can rewrite the equation above as

$$\begin{pmatrix} 1 & 0 & \cdots & 0 & \tau[1] \\ 0 & 1 & & & \tau[2] \\ \vdots & \vdots & & \vdots & \\ 0 & & & 1 & \tau[N_T] \end{pmatrix} \begin{pmatrix} c_i \\ \vdots \\ \\ c_{i+N_T-1} \\ c_{N+1} \end{pmatrix} = \begin{pmatrix} I[i] \\ \vdots \\ \\ I[i+N_T-1] \end{pmatrix},$$

where we reuse $\tau[j]$ for the value of $T_{N+1}(p_i) = L_i(\tau_j(q_1))$. We can also assume that $\tau[j] \neq 0$ for $j = 1, \ldots, N_T$, since otherwise we can redefine either T or the pixel ordering to get a smaller value for N_T.

It follows from Proposition 1 that the local minima of $F_p(c)$ can be found by setting c_{N+1}, c_i, ..., c_{i+N_T-1} to zero one at a time. If we set $c_{N+1} = 0$ then we get $c_k = I[k]$ for all k. This is the "pure noise" solution. The first nontrivial (template using) solution sets $c_i = 0$. This forces the template coefficient $c_{N+1} = I[i]/\tau[1]$, from which it follows that $c_{i+1} = I[i+1] - c_{N+1}\tau[2]$, and in general $c_{i+k} = I[i+k] - c_{N+1}\tau[k]$ for $k = 1, \ldots, N_T$. For $c_i = 0$ solution we can then explicitly calculate

$$\begin{aligned} F_p(c) &= |I[i]/\tau[1]|^p + \sum_{k=i}^{i+N_T-1} |I[k] - I[i]\tau[k]/\tau[1]|^p + \sum_{k=i+N_T}^{N} |I[k]|^p + \sum_{k=1}^{i-1} |I[k]|^p \\ &= |I[i]/\tau[1]|^p + \sum_{k=i}^{i+N_T-1} (|I[k] - I[i]\tau[k]/\tau[1]|^p - |I[k]|^p) + \sum_{k=1}^{N} |I[k]|^p. \end{aligned}$$

The solution determined by setting $c_{i+j} = 0$ ($1 \leq j \leq N_T$) can be calculated in an analogous fashion. The corresponding value for $F_p(c)$ is then

$$F_p(c) = |I[i]/\tau[j+1]|^p + \sum_{k=i}^{i+N_T} (|I[k] - I[i]\tau[k]/\tau[j+1]|^p - |I[k]|^p) + \sum_{k=1}^{N} |I[k]|^p .$$

The optimal cost of the match of the template in the (translation) position i is the smallest of the values of $F_p(c)$ across all $N_T + 1$ solutions (c). One can perform a similar analysis for all template translations, and define the matching position of the template to be the position which generated the smallest match cost.

It should be noted that this discussion pertains only to single template matching. If one wants to simultaneously match multiple templates (for example two eyes, a nose, and a mouth template on an image of a human face), then to be efficient one would like design some special structure into the optimization problem (??) that would allow the global minimum to be found by optimizing one template at a time. This will be the spirit of the greedy matching pursuit strategy to be discussed in section ??.

4.1 Simulations

We have designed a sequence of experiments focused on the effects of noise and occlusions to compare the L^p template matching method with the conventional correlation techniques.

The experiments consist of numerous trials on random images with fixed occlusion size and fixed noise variance. The latter determines the signal-to-noise ratio (SNR) for the experiment, defined here as the ratio of the standard deviation of the image to the standard deviation of the noise.

Each trial has four components: an image, a template, an occlusion, and noise. The image is 64 pixels wide by 64 pixels high, randomly generated using an uncorrelated uniform distribution across the range $(-256, 256)$. The template is a 4 pixel by 4 pixel subimage of the image. After selecting the template, a portion of the image from which the template is drawn is "occluded" by redrawing from the same distribution that formed the image, i.e., from an uncorrelated uniform distribution with range $(-256, 256)$. (Occlusion sizes range from 0–14 pixels, from a total subimage size of 16 pixels.) Finally, noise is added to the (occluded) image, drawn from an uncorrelated gaussian mean-zero random variable.

Translates of the template are compared against the noisy, occluded image, using both p-norm error minimization and our proposed decomposition method. (Because both the template and the image are drawn from zero-mean random variables, there is little difference between 2-norm error minimization and standard correlation.) For each method the translation position yielding the best score is compared with the position of the original subimage from which the template was formed. If the two agree then the match is considered successful, otherwise the match fails for the trial in

question. The first experiment, displayed in Figure 3-(a), displays the percentage of successful match trials at various occlusion sizes and no noise. Dashed curves there show the results from our proposed technique for $p = 0.125, 0.25, 0.5$. Solid curves correspond to results obtained by minimizing the p-norm for the same p-values and in addition for $p = 1.0, 2.0$, and 4.0. (We do not currently have an appropriate formulation of our method for $p > 1$.) Note that smaller values of p outperform larger values, and that for a given p value our method performs only slightly worse than minimizing with respect to the corresponding p-norm, providing nearly 100% correct results with $p = 0.125$ for occlusions as large as 11 (out of 16) pixels.

This result is somewhat artificial, however, since noise is generally present in real images. Figure 3-(b) presents results for when noise is present at a SNR of 37. Here we note that $p = 0.125$ still performs very well, although good results can not be obtained if the occlusion is larger than half the template size. Notice that the results using larger values of p are less affected by noise, especially those with $p > 1$.

Figure 3-(c) displays the results for varying noise levels with a constant occlusion size of 5 pixels. Note again that larger values of p produce results which are less sensitive to noise. For example, the results for $p = 0.125$, which are best for large SNR, are poorest for SNR of less than about 3.

The final graph, Figure 3-(d), shows the results obtained by varying the p-value for fixed occlusion and noise levels. We see there that for high noise levels (SNR=2) with no occlusions the best p value is 2. For an occlusion size of 4/16 and a SNR of 9.2, the best p value for the p-norm minimization method is somewhere between 0.25 and 0.5, whereas the optimal p value for the decomposition method is somewhat smaller. The remaining curve corresponds to no noise and an occlusion size of 8/16. Also note that for small p (generally best for large occlusions), the performance difference between p-norm minimization and the proposed decomposition method is smallest.

To conclude, these experiments show that both our proposed decomposition method and p-norm minimization with $p < 1$ are superior to standard correlation for template matching in the presence of occlusions and low levels of (gaussian) noise. Smaller values of p tend to be more robust against occlusions at the cost of greater sensitivity to noise. Although p-norm minimization outperforms our proposed decomposition method, the difference is slight and may be outweighed by other factors when applied to natural images.

5 Multiple templates and matching pursuit

In this section, we proceed to elucidate the matching pursuit method for the case of multiple templates. The basic idea is to devise a greedy iterative method where at each stage only one template is selected and thus, we can rely on the previous section result. In this section we will consider the L^p norm

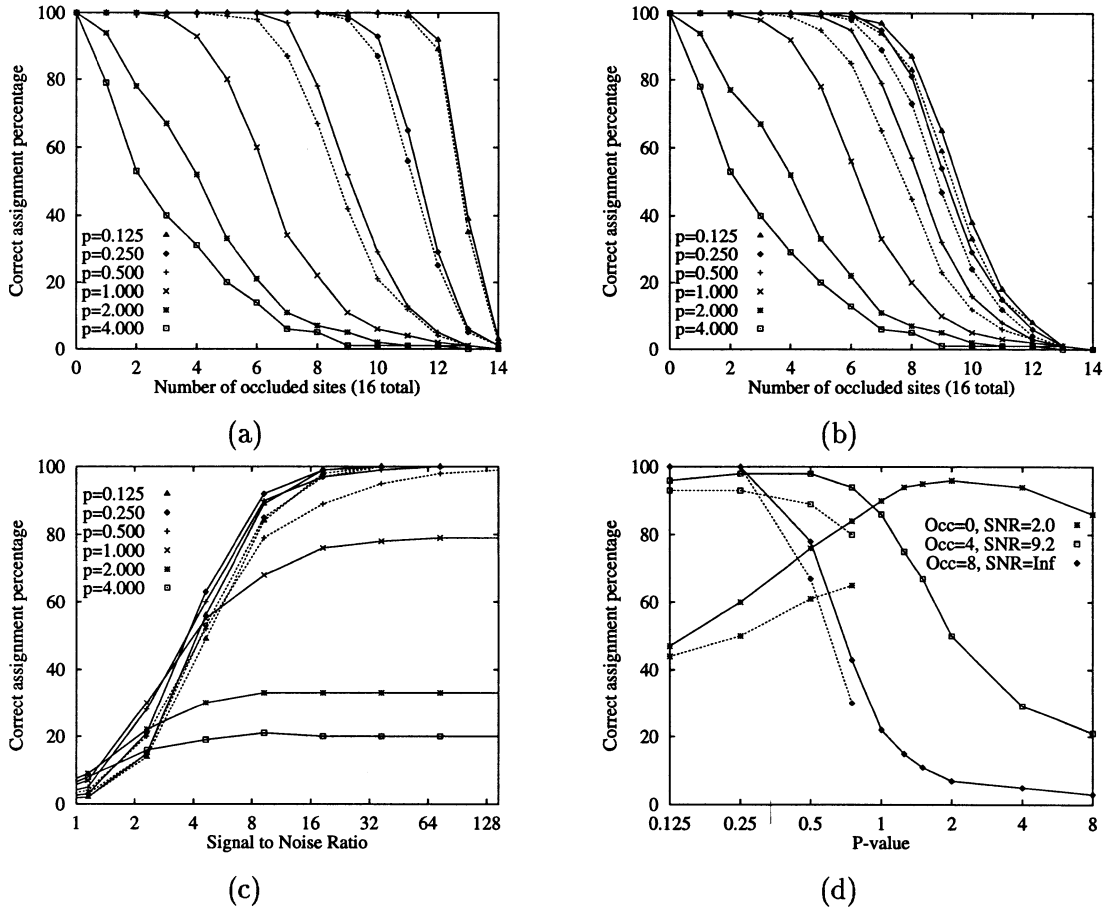


Figure 3: (a)Experimental matching accuracy as a function of occlusion size with no noise. The solid curves correspond to p-norm minimization, dashed to our proposed method. (b)Experimental matching accuracy as a function of occlusion size at a SNR of 37. The solid curves correspond to p-norm minimization, dashed to our proposed method. (c)Experimental matching accuracy as a function of noise level at a fixed occlusion size of 5 (out of 16) pixels. The solid curves correspond to p-norm minimization, dashed to our proposed method. (d)Experimental matching accuracy as a function of p, at various noise levels and occlusion size. The solid curves correspond to p-norm minimization, dashed to our proposed method.

and also, for comparison, we will consider the LTS (Least Trimmed Squares). The LTS is a high breakdown regression method (a high breakdown regression estimator means it can tolerate a large amount of outliers contamination and the LTS is known ([10]) to be capable of sustaining data contamination up to 50%).

The cost function F_p in (7) is total cost. The matching pursuit method is an iterative greedy algorithm so the whole optimization problem is indeed solved in many steps. Let's denote the stage-wise cost function as $\mathcal{F}_p$ to be distinguished from the global F_p. In case of restricting the coefficient vector $\hat{c}$ to only one non-zero element, the global F_p is reduced to $\mathcal{F}_p$. This implies that the cost function $\mathcal{F}_p$ at each stage is actually a function of some scalar c.

We briefly review the (L^2) matching pursuit below. Suppose it is given a signal f, and a library of functions $D = \{g_\gamma\}_{\gamma\in\Gamma}$ where Γ is a set of index tuples and D represents a large, over-redundant family of functions. The *Matching Pursuit* introduced for signal processing in [8] is a greedy stage-wise algorithm and relies on the inner product methods on Hilbert spaces. A "best" matching library element to the residual signal structures at each stage is decided by successive approximations of the residual signal with orthogonal projections on elements in the library. That is, say at stage n, for any element $g_\gamma \in D$, we consider

$$R^{n-1}f =< R^{n-1}f, g_\gamma > g_\gamma + R^n f \tag{12}$$

where $R^n f$ is the *n-th* residue after approximating $R^{n-1}f$ in the direction of g_γ (assume that the initial residue is the function f, i.e. $R^0 f = f$). The matching pursuit strategy is to find g_{γ^*} that minimizes $||R^n f||$ (or the g_{γ^*} closest to $R^{n-1}f$), i.e.

$$||R^{n-1}f - < R^{n-1}f, g_{\gamma^*} > g_{\gamma^*}||_{L^2} = \underset{\gamma\in\Gamma}{Min}\, ||R^{n-1}f - < R^{n-1}f, g_\gamma > g_\gamma||_{L^2}\,.$$

Next, We describe the L^p matching pursuit. At stage n, if a transfromed template $T_\lambda (= T_{ij} = A_i(\tau_j))$ is chosen, the *n-th* residual image can be updated as follows:

$$R^n I(p_k) = R^{n-1} I(p_k) - c_\lambda T_\lambda(p_k) \qquad \text{for } k = 1...N \quad \text{and} \quad \lambda = (j-1)\times N + i.$$

We can derive an analogous form as (12) by rewriting the above equations as

$$R^{n-1} I(p_k) = c_\lambda T_\lambda(p_k) + R^n I(p_k). \tag{13}$$

From (13), we see $R^n I$ can be derived by "projecting" $R^{n-1} I$ in the direction of T_λ and the best matching T_λ is the one that minimizes the cost at stage n.

Cost function (L^p-method) The cost function $\mathcal{F}_p^n$, at stage n, using the L^p norm is defined as

$$\mathcal{F}_p^n(c_\lambda) = ||R^n I||_{L^p} = \frac{1}{|c_\lambda|^p \cdot |Variance(\tau_j)|^{\frac{p}{2}}} \sum_{k \in \Gamma_i} |r_k|^p = \sum_{k \in \Gamma_i} |\tilde{r}_k|^p \tag{14}$$

where $r_k = |R^{n-1}I(p_k) - c_\lambda T_\lambda(p_k)|$ is the residue at pixel p_k, $k \in \Gamma_i$, the index set as in (5), when $R^{n-1}I$ is matched by the template T_λ. $Variance(\tau_j)$ is the variance of the gray-level-value distribution for τ_j. The normalized residues that can be computed off line

$$\tilde{r}_k = \frac{r_k}{|c_\lambda| \cdot |Variance(\tau_j)|^{\frac{1}{2}}}$$

are used to avoid "over-utilization" of templates on darker regions of the image (0 is black, 255 is white). To see this, if we did not normalize r_k with $|c_\lambda|$, then it is possible for a template to match very well in a darker region due to a small residue sum caused by a small value of $|c_\lambda|$. We now briefly address one extreme case: $\mathbf{p} = 1$.

Case $\mathbf{p} = 1$: When $\mathbf{p} = 1$, it is the L^1-method under consideration. Unlike the L^2-method (least squares), theoretically, L^1-method works if the template library $\mathcal{L}$ is robust. This property is due to the fact that L^1 regression is sensitive to leverage-point outliers but not to outliers in the y-direction. (Our experiments show that L^p with $0 < p < 1$ is more robust than L^1.)

Cost function II (LTS-method) The cost function $\mathcal{F}_{LTS}^n$ using the LTS-method is defined as

$$\mathcal{F}_{LTS}^n(c_\lambda) = ||R^n I||_{LTS} = \sum_{k \in \Gamma_i'} |\tilde{r}_k|^2 \tag{15}$$

where Γ_i' is a subset of Γ_i and $|\Gamma_i'| = h$ that it contains those indices k's for the first h smallest $|\tilde{r}_k|^2$'s among all the N_T squares of normalized residuals. Note that the $Variance(\tau_j)$ used to normalize residues r_k's in (15) is computed from only the h pixels actually used. (So, it cannot be calculated off line.) The scalar $h = \lceil \alpha N_T \rceil$ where the notation $\lceil x \rceil$ denotes the smallest integer greater than or equal to x and α is some constant between 1/2 and 1. The value of α is crucial. For any interesting object O embedded in the image I, $h = \lceil \alpha N_T \rceil$ is the expected number of robust pixels in O. We call α the *robust constant* of our template matching algorithm and the range of it ensures the capability of recognizing objects with occlusions up to 50% of their total pixels.

Case $\alpha = 1/2$: The best robustness properties are achieved when the robust constant α is approximately 1/2, i.e. h is close to $N_T/2$. On the other hand, this may increase the probability of false recognitions (see simulation results).

Case $\alpha = 1$: For $\alpha = 1$, the optimization problem is back to the original L^2 matching pursuit. To recognize images with noises and occlusions becomes difficult.

5.1 One template matching

As discussed in the previous section the local minima to $\mathcal{F}^n(c)$ are at the vertices of the polytope that is the intersection between the subspace expanded by the reduced linear system and the N_T hyperplanes, each defined by $r_k = 0$, $k \in \Gamma_i$. For each hyperplane $r_k = 0$ the solution of the constraint equation (1) is $c_k = I(p_k)/T_\lambda(p_k)$. Based on this, we can formulate a definition for the solution space associated with each template T_λ. An "interesting" filter can then be developed to speed up the template matching algorithm.

Definition 1 *Given an image I with pixel set P, we define the coarse solution space S_λ generated by T_λ with respect to I as:*

$$S_\lambda = \{c_k : c_k = [I(p_k)/T_\lambda(p_k)]^*, \quad p_k \in \Omega_i, \quad I(p_k) \neq 0 \quad \textit{and} \quad T_\lambda(p_k) \neq 0\}$$

where Ω_i is defined as in (5). [3]

The coarse solution space S_λ can be further classified according to the next definition.

Definition 2 *Let S_λ be the solution space of some non-canonical template T_λ. Assume that $S_\lambda = S_{\lambda,1} \dot{\cup} S_{\lambda,2} \dot{\cup} \cdots \dot{\cup} S_{\lambda,l}$ ($\dot{\cup}$ means disjoint union) of which $m_\lambda = |S_{\lambda,1}| \geq |S_{\lambda,2}| \geq \cdots \geq |S_{\lambda,l}|$ and l, m_λ are some positive integers. Then a sub-solution set $S_{\lambda,k}$, for $k \in \{1, 2, \cdots, l\}$, is said to be a maximal sub-solution set if $|S_{\lambda,k}| = m_\lambda$.*

As an example of illustration, let $S_\lambda = \{2, 2, 2\} \dot{\cup} \{4, 4\}$ then $|S_{\lambda,1}| = |\{2, 2, 2\}| = 3$. We will call the value, 2, the principal contrast scalar of S_λ and denote it as c_λ.

Observation : *If T_{λ^*} is indeed embedded in image I and its principal contrast scalar is c^*, then*

$$\mathcal{F}_p(c^*) = \underset{c \in S_{\lambda^*}}{Min}\, \mathcal{F}_p(c) \simeq \underset{c \in S_{\lambda^*,1}}{Min}\, \mathcal{F}_p(c) = \mathcal{F}_p(c_{\lambda^*})$$

where $\lambda^* = (j^* - 1) \times N + i^*$. That is, the cost $\mathcal{F}_p(c^*)$ (or $\mathcal{F}_{LTS}(c^*)$) can be approximated by only considering for those maximal sub-solution sets. For $p = 0$, the above approximation becomes exact,

[3] The notation $[x]^*$ is defined as

$$[x]^* = \begin{cases} [x], & \text{when } x \geq 1 \\ \frac{1}{[1/x]}, & \text{otherwise} \end{cases}$$

where $[x]$ denotes the closest integer to x.

since in this case, to minimize the cost $\mathcal{F}_p$ is equivalent to minimize the number of nonzero residues. For L^p with $0 < \mathbf{p} < 1$ and LTS-methods the above assertion is then no longer clear and we adopt it as a further approximation.

5.2 Stopping criteria and further simplifications

We now define an "interesting" operator to improve our algorithm. This operator provides an explicit measurement for degree of interests for any possible matchings and more importantly, it functions as a preliminary filter to avoid unnecessary computation.

Definition 3 *Given a transformed non-canonical template T_λ, using the same notations as in the last definition, let's define*

$$Siml_{T_\lambda} = \frac{|S_{\lambda,1}|}{|S_\lambda|}.$$

Clearly, $0 < Siml \leq 1$. We call Siml the similarity ratio of T_λ to its corresponding part in the image I.

Applied $Siml_{T_\lambda}$ to, e.g. the tasks of face recognition, we may set $Siml_{Threshold} = 0.85$ then this suggests any face template T_λ can match to some interesting object in I only when it resembles the object more than 85%. It is constructive to see how much algorithm complexity can be reduced by applying this operator. To give an overall correct answer for question of this kind in image processing is difficult since it involves the uncertainties of the properties of images under consideration. However, we have made the following assumption that it seems to be fair enough in most cases. *Similarity assumption: for an N-dimensional image, the $Siml_{Threshold}$ filter will reduce the total N searches of possible matching positions into $O(N^{1-Siml_{Threshold}})$.*

The last and most important threshold is the threshold for the costs and we denote it as $COST_{Threshold}$. The values of $COST_{Threshold,L^p}$ and $COST_{Threshold,LTS}$ vary with $\mathbf{p}$ and α, respectively. The cost threshold serves as a stopping criterion for our algorithm. Further study in this aspect is shown in the simulation section.

5.3 Algorithm

Before describing the template matching algorithm, let's first define the initial conditions. Notice that we have made the assumption each object (human face, as in our simulation) can appear only once in the image I. Hence a selected template is no longer under consideration later. In the beginning of program simulation, we have

$$\begin{cases} P = \{p_1, p_2, \cdots, p_N\} = \text{pixel set} \\ R^0 I = I = \text{input image} \\ \mathcal{L}^0 = \mathcal{L} = \text{template library} \\ Matched = \emptyset \\ Overlapped = \emptyset \end{cases}$$

The set *Matched* contains those pixels in P that have been matched by some template from the template library and the set *Overlapped* contains pixels matched by more than one templates at two or more matching pursuit stages. The updating step in the algorithm description will give a detailed rules for updating pixels' gray-level values as well as these two sets, *Matched* and *Overlapped*. Initially, both *Matched* and *Overlapped* are empty. A correspondence map *MatchBy* is used to memorize the template that each pixel in the set *Matched* is matched by.

The multistage template matching algorithm is illustrated by the execution of one iterative stage and the stopping criterion. Suppose we are at stage n then the completion of stage n is done by executing the following four steps.

A. Matching step : For each non-canonical template τ_j in the template library $\mathcal{L}^{n-1}$ and for each possible translation A_i, let $T_\lambda = A_i(\tau_j)$:

(A–1) compute the solution space S_λ for T_λ where $\lambda = (j-1) \times N + i$ and

$$S_\lambda = \left\{ c \mid c = \left[\frac{R^{n-1} I(p_k)}{T_\lambda(p_k)} \right]^*, p_k \in \Omega_i - Matched, R^{n-1} I(p_k) \neq 0 \quad \text{and} \quad T_\lambda(p_k) \neq 0 \right\};$$

(A–2) compute the similarity ratio $Siml_{T_\lambda}$. If $Siml_{T_\lambda}$ is less than $Siml_{Threshold}$ this indicates that T_λ is not a candidate for possible matching. Otherwise, proceed to (A-3).

(A–3) compute the robust cost $COST_\lambda$ as if image $R^{n-1} I$ is projected onto the direction of T_λ and its associated principal contrast scalar c_λ. Record them as an ordered quadruple entry $(T_\lambda, c_\lambda, Siml_{T_\lambda}, COST_\lambda)$ into the *n-th* cost catalogue $\mathcal{C}^n$. In fact, for $n > 2$, the robust cost $COST_\lambda$ for T_λ can be looked up directly from the previous cost catalogue $\mathcal{C}^{n-1}$ if, at stage $n-1$, the position of T_λ is not overlapped with the best matching template selected in the previous stage. So, in most cases, step (**A**) can be reduced to a single look-up operation.

If all robust costs in catalogue $\mathcal{C}^n$ are greater than $COST_{Threshold}$, this suggests we have recovered all interested objects in the image I then the algorithm jumps to the stopping stage. Otherwise, continue to the next selection step.

B. Selection step : Choose a template which "best" matches the residual image $R^{n-1}I$ by looking up the robust cost catalogue $\mathcal{C}^n$. A best matching template T_λ to image $R^{n-1}I$ is a template that has the minimal $COST_\lambda$.

D. Updating step : Remove τ_{j^*} from the stage-wise template library $\mathcal{L}^{n-1}$, that is, $\mathcal{L}^n \leftarrow \mathcal{L}^{n-1} - \{\tau_{j^*}\}$. To update the gray-level values is more complicated, we have to consider the possible overlapping between interesting objects or occlusions. For each pixel $p_k \in P$ and $T_{\lambda^*}(p_k) \neq 0$,

(Case 1) **if** $p_k \notin Matched$ **then**

$$R^nI(p_k) \leftarrow R^{n-1}I(p_k) - c_{\lambda^*} \times T_{\lambda^*}(p_k)$$
$$Matched \leftarrow Matched \cup \{p_k\}$$
$$MatchBy(p_k) \leftarrow T_{\lambda^*}$$

(Case 2) **if** $p_k \in Matched$ and $p_k \notin Overlapped$ **then**

$$Overlapped \leftarrow Overlapped \cup \{p_k\}$$
$$NewOverlapped \leftarrow NewOverlapped \cup \{p_k\}$$

(Case 3) **if** $p_k \in Overlapped$ **then**

$$NewOverlapped \leftarrow NewOverlapped \cup \{p_k\}$$

At each stage, if the local set $NewOverlapped$ (initialized to $\emptyset$ at the beginning of each stage) is not empty then this implies there are overlappings occurred in this stage. We then decompose the set $NewOverlapped$ into one or several regions based on the rule that p_{k_1} and p_{k_2} are in the same region if $MatchBy(p_{k_1}) = MatchBy(p_{k_2})$. Compute the robust costs restricted in each overlapped region for both the old "top" template, say $T_{\lambda'}$ and the newly recognized template T_{λ^*}. The one with smaller robust cost on this region will be the new "top" template. Then updating the $MatchBy$ and gray-level values become clear after all overlapping ambiguities are resolved.

Stopping stage : When the algorithm jumps to this stopping stage, say at stage n, it implies that all robust costs for the non-canonical templates in $\mathcal{L}^{n-1}$ are greater than the robust cost threshold $COST_{Threshold}$. This indicates we have recovered the main decomposition for I and the remaining task is to check if there are overlapping regions happened during the whole matching pursuit process and if this is, indeed, the case, we then find a residual representation for those regions by only using the canonical template e_0. So, if the set $Overlapped$ is not empty, the steps listed below are executed.

while there exists a pixel $p_k \in Overlapped$ **do**

$$\text{match } R^{n-1}I(p_k) \text{ with } T_{k0} = L_k e_0$$
$$Overlapped \leftarrow Overlapped - \{p_k\}$$

As pointed out that, after stage 1, computation for robust cost of any template T_λ is, in most cases, reduced to a single look up operation. So, the first stage is the most expensive stage from the complexity point of view. At any stage n ($n > 1$), there are $O(MN_T)$ templates that require re-calculation for their robust costs since among all the $O(MN)$ template T_λ's, there are $O(MN_T)$ of them will overlap the best matching template chosen in stage $n-1$. Hence we can approximately derive the following relation that for $n > 1$,

$$\text{Complexity of stage } n \simeq \frac{N_T}{N} \cdot \text{Complexity of stage 1.}$$

We conclude this section with a complexity analysis study for matching pursuit stage 1.

Complexity Analysis : The complexity for carrying out stage 1 in our proposed algorithm is $O(MN \cdot N_T) + O(MN^{1-Siml} \cdot Time_{L^p})$ for the L^p-matching-pursuit method and $O(MN \cdot N_T) + O(MN^{1-Siml} \cdot Time_{LTS})$ for the LTS-matching-pursuit method. Here $O(Time_{L^p})$ and $O(Time_{LTS})$ are the time to compute (14) and (15), respectively.

Proof: We'll prove for the L^p case since the reasoning is the same for both methods. The complexity can be derived by observing the total times that step (**A**) is executed since it is the main step required extensive computation. Let's now begin with matching pursuit stage 1. For each execution of step (**A**), we see that (A-1) and (A-2) are definitely carried out but (A-3) may or may not be computed depending on the result of (A-2). By taking advantage of the integral approximation for the definition of coarse solution spaces, we can achieve a linear-time complexity, $O(N_T)$, for (A-1) and (A-2). Following the previous similarity assumption that each prototype template τ_j is expected to generate $O(N^{1-Siml})$ possibly interesting matching positions, total complexity required for stage 1 is then $O(MN \cdot N_T) + O(MN^{1-Siml} \cdot Time_{L^p})$ where $O(Time_{L^p})$ is the complexity for (A-3), i.e., the time to compute (14).

6 Simulation Results

Synthetically Randomized Images : Let's begin with a simple but instructive experiment to test our template matching algorithm for a synthetic example. In this experiment, the template library $\mathcal{L}$ consists of three different types (or shapes) of templates ((a), (b), (c) in Figure **??**). There are 40 templates for each type so that $\mathcal{L}$ includes 120 non-canonical templates and one canonical template e_0. Each of the non-canonical template is a synthetically randomized image with gray-level values between $(0, 200)$ generating from a random number generator. For each value of p and α, there are three testing input images, I_1, I_2 and I_3 (see Figure **??** and **??**). The exact image I_1 to be recognized is constructed by selecting one non-canonical template randomly from each type in $\mathcal{L}$ then

p	$Siml_{Threshold}$	$\frac{1}{N_T}COST_{Threshold}$
0.25	85%	0.79
0.50	85%	0.67
0.75	85%	0.60

α	$Siml_{Threshold}$	$\frac{1}{h}COST_{Threshold}$
0.51	85%	0.072
0.60	85%	0.137
0.75	85%	0.200

Table 1: Threshold values used in the simulation with synthetic images for the L^p and LTS matching pursuit.

put together to form it such as all images (a), (d) and (g) in Figure ?? and ??. Image I_2 is I_1 plus noises that have a uniform distribution in $(0, 10)$ and I_3 is I_2 covered by some unknown occluded square generated uniformly from $(245, 255)$. Notice that each I_3 used for the LTS simulation is occluded by a larger square compared to the case for L^p. We summarize all threshold values in Table 1 and show simulation results in Figure ?? and Figure ??.

Face Recognition : We then test our template matching algorithm for real image for a more interesting application: face recognition. A small library of face templates has been established (see Figure ??). The dimension of all four face template (d) - (g) and two book-like templates (h) - (i) is 64×64. All images, I_1 - I_6, to be recognized in Figure ?? are created by using mosaic (with a robot camera) and are all of dimension 240×240. For each p and α, we do the following experiment: simulations (a) - (g) with input images I_1 - I_6, respectively, using only the four face templates and simulations (h) - (i) with input images I_4 - I_6, respectively but using all six templates. We also investigate the case that $p = 2$, it becomes the L^2 matching pursuit. By comparing the simulation results, we can see both the L^p and LTS are more robust than the L^2. Again, all threshold values are listed in Table 2 and Table 3 and simulation results are shown from Figure ?? through Figure ??.

7 Conclusion

To this end, we have proposed an iterative and robust template matching method using matching pursuit. The two objective functions, F_{L^p} and F_{LTS}, adopted for the optimization formulation in our iterative matching pursuit algorithm displayed very robust results in recognition for synthetic images. As for the case of face recognition, fair results are shown in our experiments. In all, for an image that contains noises and all interested objects embedded in it are not severely occluded, matching pursuit with the L^p approach is very effective and robust. On the other hand, we showed

p	$Siml_{Threshold}$									$\frac{1}{N_T}COST_{Threshold}$
	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	
0.25	95%	95%	90%	70%	70%	70%	70%	70%	70%	0.79
0.50	95%	95%	90%	70%	70%	70%	70%	70%	70%	0.67
0.75	95%	95%	90%	70%	70%	70%	70%	70%	70%	0.60
1.00	85%	85%	85%	70%	70%	70%	70%	70%	70%	0.56
2.00	85%	85%	85%	70%	70%	70%	70%	70%	70%	0.40

Table 2: Threshold values used in the simulation with real images for the L^p matching pursuit.

α	$Siml_{Threshold}$									$\frac{1}{h}COST_{Threshold}$
	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	
0.51	90%	90%	90%	75%	75%	75%	75%	75%	75%	0.072
0.60	90%	90%	90%	75%	75%	75%	75%	75%	75%	0.137
0.75	90%	90%	90%	75%	75%	75%	75%	75%	75%	0.200

Table 3: Threshold values used in the simulation with real images for the LTS matching pursuit.

that the LTS-methods are capable of handling occlusions up to 50% with the penalty of possible false recognitions.

There are much left to be accomplished for this work. We are now working on improving the robustness and effectiveness of our matching pursuit algorithm, especially for a complex task such as face recognition. In addition, we plan to establish a more complete template library of faces. The other direction is to extend our results to be able to identify objects in an image that are affine transformation of some non-canonical templates in template library $\mathcal{L}$.

A Special cases of optimization criteria

This section presents results to the optimization problem (??) for some special values of p. Typically the finite dimensionality of the spaces $\mathbb{R}^N$ and $\mathbb{R}^M$ plays a prominent role.

Case $p = \infty$

Let us define

$$\begin{aligned} F_\infty(c) &= \lim_{p\to\infty} (F_p(c))^{1/p} && (16)\\ &= \lim_{p\to\infty} \left(\sum_{j=1}^{M} |c_j|^p \right)^{1/p} && (17)\\ &= \max_{1\le j\le M} |c_j|. && (18) \end{aligned}$$

(Note that we take the p^{th} root of F_p to normalize the limit.) So for $p = \infty$ the minimization criterion is

$$\begin{aligned} \mathop{Min}_{c\in S^*} F_\infty(c) &= \mathop{Min}_{c\in S^*} \lim_{p\to\infty} (F_p(c))^{1/p}. && (19)\\ &= \mathop{Min}_{c\in S^*} \max_{1\le j\le M} |c_j| && (20) \end{aligned}$$

The minimizing solution c^* is the element of S^* which has smallest maximal component $|c_j^*|$. Furthermore, for all $p > 0$,

$$\mathop{Min}_{c\in S^*} F_\infty(c) \le \mathop{Min}_{c\in S^*} (F_p(c))^{1/p} \le M^{1/p} \mathop{Min}_{c\in S^*} F_\infty(c),$$

so

$$\mathop{Min}_{c\in S^*} F_\infty(c) = \lim_{p\to\infty} \mathop{Min}_{c\in S^*} (F_p(c))^{1/p}.$$

Therefore, for p large enough, the minimizing solution c^* for F_p will have maximal component $|c_j^*|$ close to the smallest possible subject to the constraint in (??).

Case $p = 0$

Let us define

$$\begin{aligned} F_0(c) &= \lim_{p\downarrow 0} F_p(c) && (21)\\ &= |\{c_j \neq 0 \mid j = 1, 2, \ldots, M\}|, && (22) \end{aligned}$$

and the minimization criterion for $p = 0$ is

$$\mathop{Min}_{c\in S^*} F_0(c) = \mathop{Min}_{c\in S^*} |\{j \mid c_j \neq 0,\ j = 1, 2, \ldots, M\}|.$$

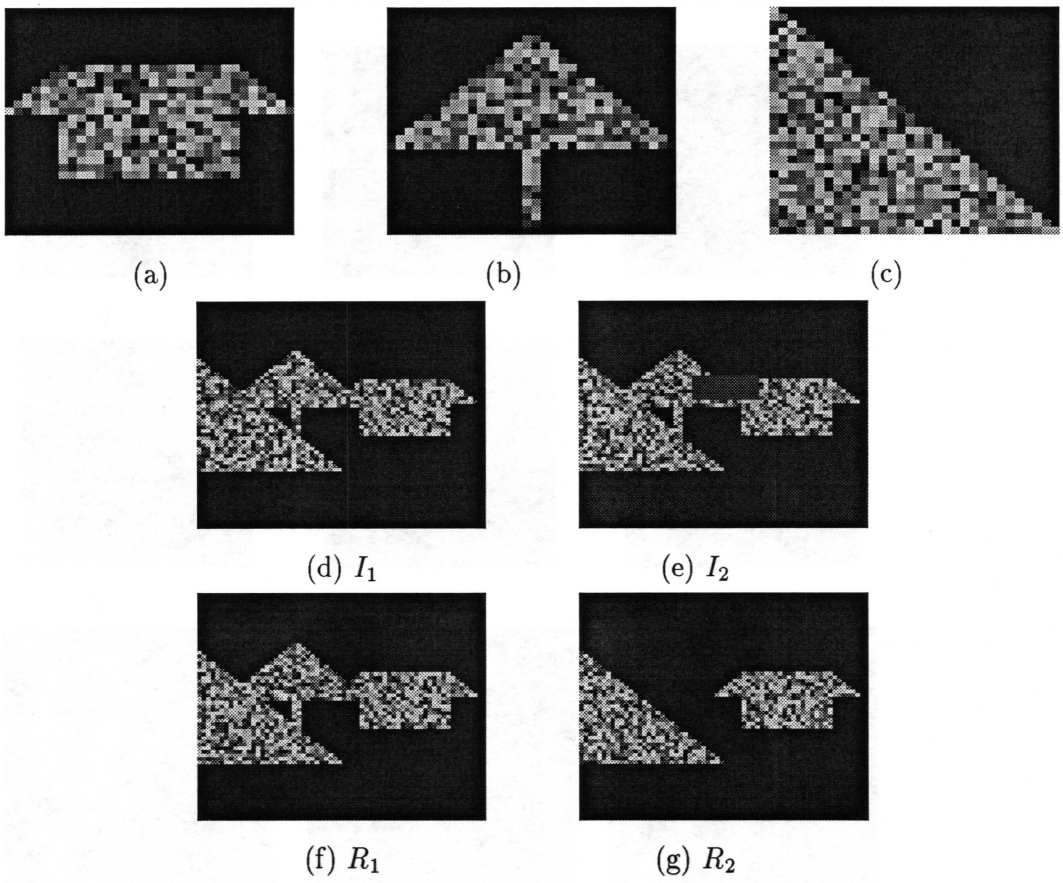
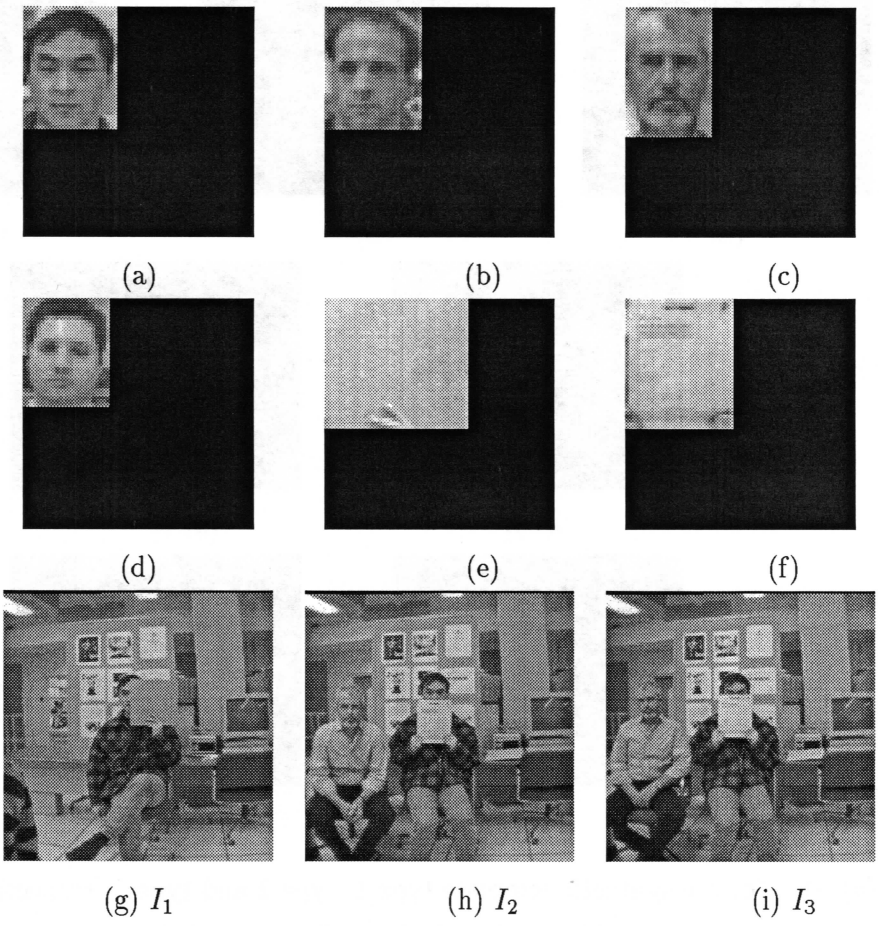


Figure 4: (a), (b), (c) are synthetic template type 1, type 2 and type 3, respectively. (d) test image and (e) test image with noise added and occlusions (f) Result of the decomposition for the L_p with $p = 0.25$, and also for the LTS with $\alpha = 0.51$. (g) Results once the breakdown limits are reached, and occluded templates are not recognized. For L_p, with $p = 0.75$ and for LTS with $\alpha = 0.75$.



(a) (b) (c)

(d) (e) (f)

(g) I_1 (h) I_2 (i) I_3

Figure 5: (a)- (f) are the template library, with faces and books. (g)-(i) The test images, where some templates are present with **small** distortions (scale and viewing angle), noise and occlusions.

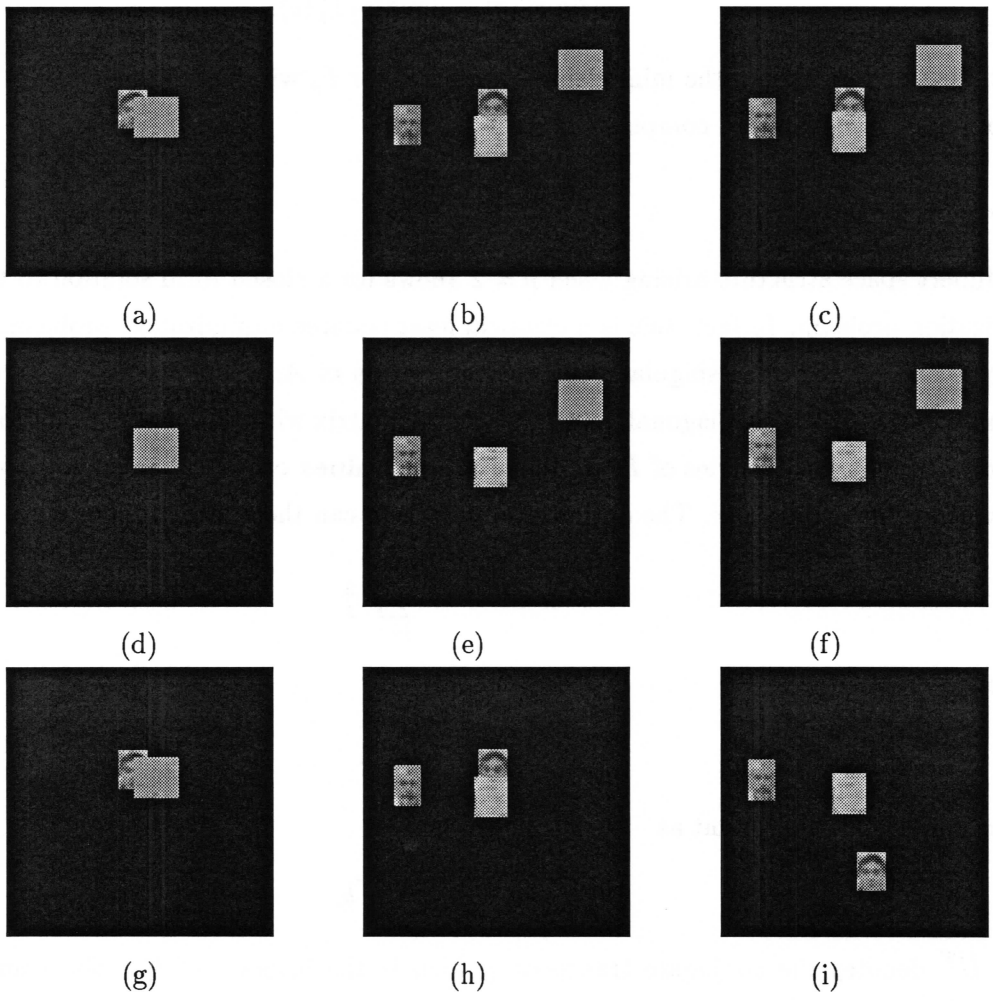


Figure 6: Image decomposition for L^p matching pursuit with $p = 0.25$ (similar results are obtained for p up to 0.75). Results (d) - (f) Image decomposition for $p = 2.0$ and recognition is destroyed (this is equivalent to use correlations methods, like in the L_2 regression pursuit). (g)-(i) Image decomposition with LTS. False recognitions occur in (f) and (i) but we are still able to recover the occluded face in (d).

In other words, the minimizing solution c^* minimizes the number of nonzero components c_j^*. It will be shown later in this document that there exists a finite set $S^{**} \subset S^*$ such that for each $0 < p < 1$, the minimizing point $c^* \in S^*$ of $F_p(c)$ satisfies $c^* \in S^{**}$. It follows from this that

$$\underset{c \in S^*}{Min}\, F_0(c) = \lim_{p \downarrow 0} \underset{c \in S^*}{Min}\, F_p(c),$$

and so for small enough p, the minimizing solution c^* for F_p will be an element of S^* which has the minimal number of nonzero components.

Case $p = 2$

The Hilbert space structure arising when $p = 2$ allows for a closed form solution to the constrained optimization problem. In fact, this is a classical least squares minimization problem.

Let $UDI_{N\times M}V$ be the singular value decomposition of A, where $U \in \mathbb{C}^{N\times N}$ and $V \in \mathbb{C}^{M\times M}$ are unitary, $D \in \mathbb{R}^{N\times N}$ is diagonal, and $I_{N\times M}$ is the matrix with ij-th entry equal to the Kronecker delta δ_{ij}. The diagonal entries of D are the **singular values** of A, and are strictly positive since A is assumed to have full rank. The optimization problem can then be written as

$$\text{Minimize} \qquad \sum_{j=1}^{M} c_j^2 \tag{23}$$

subject to

$$UDI_{N\times M}Vc = b. \tag{24}$$

We can rewrite the constraint as

$$I_{N\times M}Vc = D^{-1}\bar{U}^T b, \tag{25}$$

where $\bar{U}^T$ denotes the conjugate transpose (which is the inverse) of U. Since unitary matrices preserve the L_2-norm, minimizing the norm of c is equivalent to minimizing the norm of Vc. But clearly the last $M - N$ components of Vc, $(Vc)_{N+1}$, $(Vc)_{N+2}$, ..., $(Vc)_M$ are unconstrained, so the solution with minimal norm has these components set to 0. This means that the minimal solution satisfies

$$Vc = I_{M\times N}D^{-1}\bar{U}^T b, \tag{26}$$

or

$$c = \bar{V}^T I_{M\times N}D^{-1}\bar{U}^T b. \tag{27}$$

The matrix $A^I = \bar{V}^T I_{M\times N}D^{-1}\bar{U}^T$ is called the **pseudo-inverse** of A. (See [?].)

Theorem 1 *The minimizing point $c^* \in S^*$ of $F_2(c)$ is given by*

$$c^* = A^I b.$$

A special case occurs when the singular values D_{ii} are identical:

Theorem 2 *If the singular values of A are identically equal, say $D_{ii} = \sigma$ for all $i = 1, 2, \ldots, N$, then the minimizing point $c^* \in S^*$ of $F_2(c)$ is given by*

$$c^* = \sigma^{-2} A^T b.$$

Proof: Using the preceding theorem and $D = \sigma I_{N \times N}$ we get

$$\begin{aligned} c^* &= \bar{V}^T I_{M \times N} D^{-1} \bar{U}^T b \\ &= \sigma^{-1} \bar{V}^T I_{M \times N} \bar{U}^T b \\ &= \sigma^{-2} \bar{V}^T I_{M \times N} D \bar{U}^T b \\ &= \sigma^{-2} A^T b. \end{aligned}$$

□

As an example, if the matrix A is composed of columns forming separate sets of orthonormal bases (for $\mathbb{R}^N$), then the solution is given after proper renormalization by projecting b onto each column.

Corollary 1 *Let $p = 2$ and assume N divides evenly into M. Suppose, moreover, that the columns A_{kN+1}, A_{kN+2}, $\ldots$, $A_{(k+1)N}$ of A form an orthonormal basis for $\mathbb{R}^N$ for each $k = 0, 1, \ldots, (M/N) - 1$. Then the minimizing point $c^* \in S^*$ of $F_2(c)$ is given by*

$$c_j^* = \frac{N}{M} \langle A_j, b \rangle, \tag{28}$$

and

$$F_2(c^*) = \frac{N}{M} \|b\|_2^2. \tag{29}$$

Proof: The rows of A are pairwise orthogonal, and have magnitude (with respect to the 2-norm in $\mathbb{R}^M$) of $\sqrt{M/N}$. Therefore the N singular values of A are identically equal to $\sqrt{M/N}$, and the result follows immediately from the preceding theorem. □

Unconstrained optimization

The singular value decomposition presented above for $p = 2$ can be used in general to recast the constrained optimization problem as an unconstrained optimization problem. Recall that we want to minimize $F_p(c)$ subject to the constraint $Ac = b$, which can be rewritten

$$I_{N\times M}Vc = D^{-1}\bar{U}^T b.$$

Letting $\tilde{c}$ denote the vector Vc, note that the first N components of $\tilde{c}$ are fixed by the constraint, but the last $M - N$ are unconstrained. So the original optimization problem is equivalent to

$$\underset{\tilde{c}_{N+1},\tilde{c}_{N+2},\ldots,\tilde{c}_M}{Min} F_p(\bar{V}^T\tilde{c}) \tag{30}$$

where

$$\begin{pmatrix} \tilde{c}_1 \\ \tilde{c}_2 \\ \vdots \\ \tilde{c}_N \end{pmatrix} = D^{-1}\bar{U}^T b. \tag{31}$$

This result cannot be used to provide a closed form solution for $p \neq 2$ because the matrix V does not in general preserve the p-norm.

B "Weight spreading"

It was mentioned earlier that if $p > 1$ then the minimizing solution c^* to the optimization problem (??) will have a tendency to "spread the weight," while for $0 < p < 1$ the opposite occurs. In Section A it was shown that for $p = \infty$ the minimizing solution c^* has the smallest possible largest component $|c_j^*|$, and that this is also approximately true for p large enough. Also, for $p = 0$ (and for p small enough) it was shown that c^* has as few non-zero components as possible. In this section we shall show that in some sense these results hold as well for $0 < p < \infty$.

The following result shows that for $N = 1$, $A = (1, 1, \ldots, 1)$, $c_j \geq 0$, the above statements concerning "spreading the weight" hold in a very natural sense. It would be nice to be able to generalize this result to $N > 1$ and arbitrary c.

Theorem 3 *Suppose*

$$\sum_{j=1}^{M} c_j = \sum_{j=1}^{M} \tilde{c}_j, \tag{32}$$

$$0 \leq c_1 \leq c_2 \leq \ldots \leq c_M, \tag{33}$$

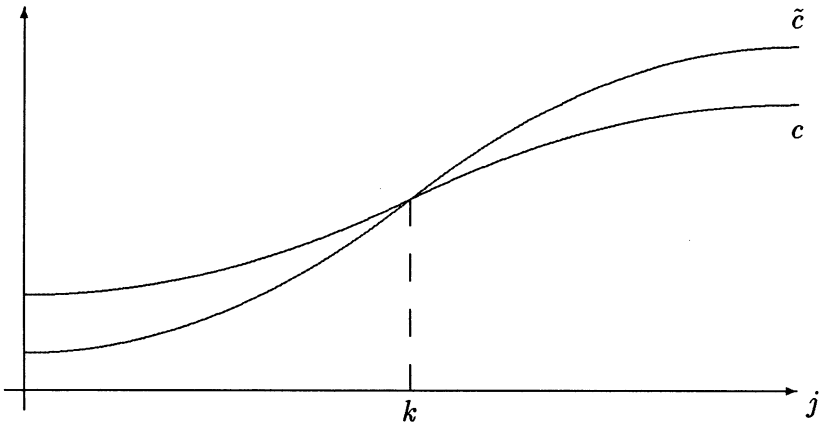


Figure 7: Illustration of two finite sequences c_j and $\tilde{c}_j$ meeting the hypothesis of Theorem 3. The curve c produces a smaller value for F_p if $p > 1$, while the curve $\tilde{c}$ produces a smaller value for F_p if $0 < p < 1$.

$$0 \leq \tilde{c}_1 \leq \tilde{c}_2 \leq \ldots \leq \tilde{c}_M. \tag{34}$$

Suppose further that there exist k, $1 \leq k \leq N$, *such that*

$$c_j \geq \tilde{c}_j \qquad \textit{for } j \leq k, \tag{35}$$

and

$$c_j \leq \tilde{c}_j \qquad \textit{for } j > k. \tag{36}$$

Then

$$F_p(c) \geq F_p(\tilde{c}) \qquad \textit{if } 0 < p < 1, \tag{37}$$

and

$$F_p(c) \leq F_p(\tilde{c}) \qquad \textit{if } p > 1. \tag{38}$$

(Clearly $F_1(c) = F_1(\tilde{c})$*.) Moreover, for* $0 < p < 1$ *or* $p > 1$ *we have*

$$F_p(c) = F_p(\tilde{c}) \qquad \textit{iff } c_j = \tilde{c}_j \ \forall j. \tag{39}$$

Proof: Without loss of generality let us assume $c_1 > 0$, since otherwise $c_1 = \tilde{c}_1 = 0$, and so the first terms do not contribute and we may start with the first $c_j > 0$. Figure 7 illustrates the requirements for the two finite sequences c_j and $\tilde{c}_j$.

Consider first $0 < p < 1$. Let $g(x) = x^p$. Since $c_j \geq \tilde{c}_j$ for $j \leq k$, the concavity of g implies

$$\sum_{j=1}^{k} c_j^p - \tilde{c}_j^p \geq \sum_{j=1}^{k} g'(c_j)(c_j - \tilde{c}_j) \tag{40}$$

$$\geq g'(c_k) \sum_{j=1}^{k} c_j - \tilde{c}_j. \tag{41}$$

Similarly, $c_j \leq \tilde{c}_j$ for $j > k$ implies

$$\sum_{j=1}^{k} c_j^p - \tilde{c}_j^p \leq g'(c_{k+1}) \sum_{j=1}^{k} \tilde{c}_j - c_j. \tag{42}$$

But it follows from (32) that

$$\sum_{j=1}^{k} c_j - \tilde{c}_j = \sum_{j=k+1}^{M} \tilde{c}_j - c_j. \tag{43}$$

This combines with

$$g'(c_k) \geq g'(c_{k+1}) > 0 \tag{44}$$

to show

$$\sum_{j=1}^{k} c_j^p - \tilde{c}_j^p \geq \sum_{j=k+1}^{M} \tilde{c}_j^p - c_j^p, \tag{45}$$

which can be rewritten as

$$\sum_{j=1}^{M} c_j^p \geq \sum_{j=1}^{M} \tilde{c}_j^p. \tag{46}$$

Moreover, the inequalities in (40) and (42) are strict unless $c_j = \tilde{c}_j$ for all j.

For $p > 1$ the same argument holds with all inequalities reversed. □

References

[1] J. Ben-Arie and K. R. Rao, *"On The Recognition of Occluded Shapes and Generic Faces Using Multiple-Template Expansion Matching"*, Proceedings IEEE International Conference on Pattern Recognition, New York City, 1993.

[2] T. H. Cormen, C. E. Leiserson and R. L. Rivest, *Introduction to Algorithms*, McGraw-Hill, 1990.

[3] M. J. Donahue and D. Geiger, *"Template Matching and Function Decomposition Using Non-Minimal Spanning Sets"*, Manuscript, 1994.

[4] H. Ekblom, *"L_p-methods For Robust Regression"*, BIT 14, p.22-32, 1973.

[5] J. H. Friedman and W. Stuetzle, *"Projection Pursuit Regression"*, Journal of the American Statistical Association, vol. 76, p.817-823, 1981.

[6] P. J. Huber, *"Projection Pursuit"*, The Annals of Statistics, vol. 13, No.2, p.435-475, 1985.

[7] P. J. Huber, *Robust Statistics*, John Wiley & Sons, New York, 1981.

[8] S. Mallat and Z. Zhang, *"Matching Pursuit with Time-Frequency Dictionaries"*, IEEE Trans. on Signal Processing, Dec. 1993.

[9] J. L. Mundy and A. Zisserman, *"Towards a New Framework for Vision"* , Geometric Invariance in Computer Vision, editors J. L. Mundy and A. Zisserman, MIT Press, p.1-39, 1992.

[10] P. J. Rousseeuw and A. Leroy, *Robust Regression and Outlier Detection*, John Wiley, New York, 1987.

[11] B. Uhrin, *"An Elementary Constructive Approach to Discrete Linear l_p-approximation, $0 < p \leq +\infty$"*, Colloquia Mathematica Societatis János Bolyai, 58. Approximation Theory, Kecskemét, 1990.