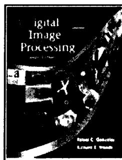
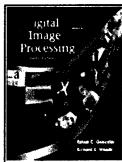


Review

Matrices and Vectors

Objective

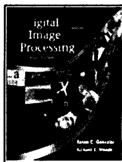
To provide background material in support of topics in ***Digital Image Processing*** that are based on matrices and/or vectors.



Some Definitions

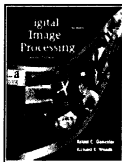
An $m \times n$ (read "m by n") *matrix*, denoted by **A**, is a rectangular array of entries or elements (numbers, or symbols representing numbers) enclosed typically by square brackets, where m is the number of rows and n the number of columns.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$



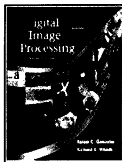
Definitions (Con't)

- **A** is *square* if $m = n$.
- **A** is *diagonal* if all off-diagonal elements are 0, and not all diagonal elements are 0.
- **A** is the *identity matrix* (**I**) if it is diagonal and all diagonal elements are 1.
- **A** is the *zero* or *null matrix* (**0**) if all its elements are 0.
- The *trace* of **A** equals the sum of the elements along its main diagonal.
- Two matrices **A** and **B** are *equal* iff the have the same number of rows and columns, and $a_{ij} = b_{ij}$.



Definitions (Con't)

- The ***transpose*** $\mathbf{A}^T$ of an $m \times n$ matrix $\mathbf{A}$ is an $n \times m$ matrix obtained by interchanging the rows and columns of **A**.
- A square matrix for which $\mathbf{A}^T = \mathbf{A}$ is said to be ***symmetric***.
- Any matrix X for which $\mathbf{XA} = \mathbf{I}$ and $\mathbf{AX} = \mathbf{I}$ is called the ***inverse*** of **A**.
- Let c be a real or complex number (called a ***scalar***). The ***scalar multiple*** of c and matrix **A**, denoted $c\mathbf{A}$, is obtained by multiplying every elements of **A** by c. If $c = -1$, the scalar multiple is called the ***negative*** of **A**.



Definitions (Con't)

A *column vector* is an $m \times 1$ matrix:

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}$$

A *row vector* is a $1 \times n$ matrix:

$$\boxed{\mathbf{b} = [b_1, b_2, \cdots b_n]}$$

A column vector can be expressed as a row vector by using the transpose:

$$\boxed{\mathbf{a}^T = [a_1, a_2, \cdots, a_m]}$$

Some Basic Matrix Operations

- The *sum* of two matrices **A** and **B** (of equal dimension), denoted **A** + **B**, is the matrix with elements $a_{ij} + b_{ij}$.
- The *difference* of two matrices, **A**– **B**, has elements $a_{ij} - b_{ij}$.
- The *product*, **AB**, of $m \times n$ matrix **A** and $p \times q$ matrix **B**, is an $m \times q$ matrix **C** whose (i,j)-th element is formed by multiplying the entries across the ith row of **A** times the entries down the jth column of **B**; that is,

$$\boxed{c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{pj}}$$

Some Basic Matrix Operations (Con't)

The *inner product* (also called *dot product*) of two vectors

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

is defined as

$$\boxed{\begin{aligned} \mathbf{a}^T\mathbf{b} = \mathbf{b}^T\mathbf{a} &= a_1b_1 + a_2b_2 + \cdots + a_mb_m \\ &= \sum_{i=1}^{m} a_ib_i. \end{aligned}}$$

Note that the inner product is a scalar.

Vectors and Vector Spaces

A ***vector space*** is defined as a nonempty set V of entities called *vectors* and associated scalars that satisfy the conditions outlined in A through C below. A vector space is *real* if the scalars are real numbers; it is *complex* if the scalars are complex numbers.

- Condition A: There is in V an operation called *vector addition*, denoted $\mathbf{x} + \mathbf{y}$, that satisfies:
    1. $\mathbf{x} + \mathbf{y} = \mathbf{y} + \mathbf{x}$ for all vectors $\mathbf{x}$ and $\mathbf{y}$ in the space.
    2. $\mathbf{x} + (\mathbf{y} + \mathbf{z}) = (\mathbf{x} + \mathbf{y}) + \mathbf{z}$ for all $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$.
    3. There exists in V a unique vector, called the *zero vector*, and denoted $\mathbf{0}$, such that $\mathbf{x} + \mathbf{0} = \mathbf{x}$ and $\mathbf{0} + \mathbf{x} = \mathbf{x}$ for all vectors $\mathbf{x}$.
    4. For each vector $\mathbf{x}$ in V, there is a unique vector in V, called the *negation* of $\mathbf{x}$, and denoted $-\mathbf{x}$, such that $\mathbf{x} + (-\mathbf{x}) = \mathbf{0}$ and $(-\mathbf{x}) + \mathbf{x} = \mathbf{0}$.

This is where you should start.

Vectors and Vector Spaces (Con't)

- Condition B: There is in V an operation called *multiplication by a scalar* that associates with each scalar c and each vector $\mathbf{x}$ in V a unique vector called the *product* of c and $\mathbf{x}$, denoted by $c\mathbf{x}$ and $\mathbf{x}c$, and which satisfies:
  1. $c(d\mathbf{x}) = (cd)\mathbf{x}$ for all scalars c and d, and all vectors $\mathbf{x}$.
  2. $(c + d)\mathbf{x} = \mathbf{cx} + \mathbf{dx}$ for all scalars c and d, and all vectors $\mathbf{x}$.
  3. $\mathrm{c}(\mathbf{x} + \mathbf{y}) = \mathbf{cx} + \mathbf{cy}$ for all scalars c and all vectors $\mathbf{x}$ and $\mathbf{y}$.
- Condition C: $1\mathbf{x} = \mathbf{x}$ for all vectors $\mathbf{x}$.

Vectors and Vector Spaces (Con't)

We are interested particularly in real vector spaces of real $m \times 1$ column matrices. We denote such spaces by $\Re^m$, with vector addition and multiplication by scalars being as defined earlier for matrices. Vectors (column matrices) in $\Re^m$ are written as

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

Vectors and Vector Spaces (Con't)

Example

The vector space with which we are most familiar is the two-dimensional real vector space $\Re^2$, in which we make frequent use of graphical representations for operations such as vector addition, subtraction, and multiplication by a scalar. For instance, consider the two vectors

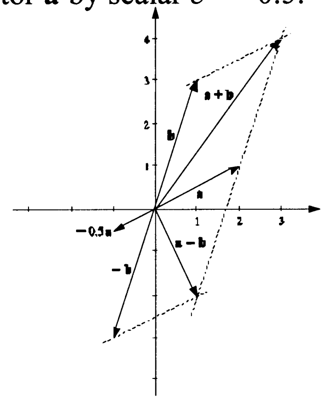
$$\mathbf{a} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

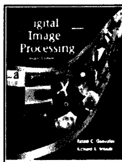
Using the rules of matrix addition and subtraction we have

$$\mathbf{a} + \mathbf{b} = \begin{bmatrix} 3 \\ 4 \end{bmatrix} \qquad \mathbf{a} - \mathbf{b} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

Vectors and Vector Spaces (Con't)

Example (Con't)
The following figure shows the familiar graphical representation of the preceding vector operations, as well as multiplication of vector **a** by scalar $c = -0.5$.





Vectors and Vector Spaces (Con't)

Consider two real vector spaces V_0 and V such that:

- Each element of V_0 is also an element of V (i.e., V_0 is a *subset* of V).
- Operations on elements of V_0 are the same as on elements of V. Under these conditions, V_0 is said to be a *subspace* of V.

A *linear combination* of $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ is an expression of the form

$$\boxed{\alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2 + \cdots + \alpha_n \mathbf{v}_n}$$

where the α's are scalars.

Vectors and Vector Spaces (Con't)

A vector $\mathbf{v}$ is said to be *linearly dependent* on a set, S, of vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ if and only if $\mathbf{v}$ can be written as a linear combination of these vectors. Otherwise, $\mathbf{v}$ is *linearly independent* of the set of vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$.

Vectors and Vector Spaces (Con't)

A set S of vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ in V is said to ***span*** some subspace V_0 of V if and only if S is a subset of V_0 and every vector $\mathbf{v}_0$ in V_0 is linearly dependent on the vectors in S. The set S is said to be a ***spanning set*** for V_0. A ***basis*** for a vector space V is a linearly independent spanning set for V. The number of vectors in the basis for a vector space is called the ***dimension*** of the vector space. If, for example, the number of vectors in the basis is n, we say that the vector space is n-dimensional.

 15

basis – linearly independent spanning set for V

number of vectors establishes the dimension of the vector space

Vectors and Vector Spaces (Con't)

An important aspect of the concepts just discussed lies in the representation of any vector in $\Re^m$ as a ***linear combination*** of the basis vectors. For example, any vector

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

in $\Re^3$ can be represented as a linear combination of the basis vectors

$$\underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \text{ and} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}$$

 16

These are the usual basis vectors in $\mathbb{R}^3$

Vector Norms

A ***vector norm*** on a vector space V is a function that assigns to each vector $\mathbf{v}$ in V a nonnegative real number, called the ***norm*** of $\mathbf{v}$, denoted by $\|\mathbf{v}\|$. By definition, the norm satisfies the following conditions:

(1) $\|\mathbf{v}\| > 0$ for $\mathbf{v} \neq \mathbf{0}$; $\|\mathbf{0}\| = 0$,

(2) $\|c\mathbf{v}\| = |c|\|\mathbf{v}\|$ for all scalars c and vectors $\mathbf{v}$, and

(3) $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$.

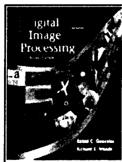
Vector Norms (Con't)

There are numerous norms that are used in practice. In our work, the norm most often used is the so-called 2-***norm***, which, for a vector **x** in real $\Re^m$, space is defined as

$$\boxed{\|\mathbf{x}\| = [x_1^2 + x_2^2 + \cdots + x_m^2]^{1/2}}$$

which is recognized as the *Euclidean distance* from the origin to point **x**; this gives the expression the familiar name Euclidean norm. The expression also is recognized as the length of a vector **x**, with origin at point **0**. From earlier discussions, the norm also can be written as

$$\boxed{\|\mathbf{x}\| = [\mathbf{x}^T\mathbf{x}]^{1/2}}$$



Vector Norms (Con't)

The *Cauchy-Schwartz* inequality states that

$$\boxed{|\mathbf{x}^T\mathbf{y}| \leq \|\mathbf{x}\|\,\|\mathbf{y}\|}$$

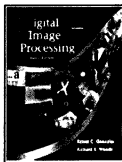
Another well-known result used in the book is the expression

$$\boxed{\cos\theta = \frac{\mathbf{x}^T\mathbf{y}}{\|\mathbf{x}\|\,\|\mathbf{y}\|}}$$

where θ is the angle between vectors **x** and **y**. From these expressions it follows that the inner product of two vectors can be written as

$$\boxed{\mathbf{x}^T\mathbf{y} = \|\mathbf{x}\|\,\|\mathbf{y}\|\cos\theta}$$

Thus, the inner product can be expressed as a function of the norms of the vectors and the angle between the vectors.



Vector Norms (Con't)

From the preceding results, two vectors in $\Re^m$ are ***orthogonal*** if and only if their inner product is zero. Two vectors are ***orthonormal*** if, in addition to being orthogonal, the length of each vector is 1.

From the concepts just discussed, we see that an arbitrary vector $\mathbf{a}$ is turned into a vector $\mathbf{a}_n$ of unit length by performing the operation $\mathbf{a}_n = \mathbf{a}/\|\mathbf{a}\|$. Clearly, then, $\|\mathbf{a}_n\| = 1$.

A ***set of vectors*** is said to be an ***orthogonal*** set if every two vectors in the set are orthogonal. A ***set of vectors*** is ***orthonormal*** if every two vectors in the set are orthonormal.

normalize vectors $\underline{a}_n = \dfrac{\underline{a}}{\|\underline{a}\|}$

Some Important Aspects of Orthogonality

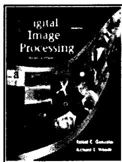
Let $B = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ be an orthogonal or orthonormal basis in the sense defined in the previous section. Then, an important result in vector analysis is that any vector v can be represented with respect to the orthogonal basis B as

$$\boxed{\mathbf{v} = \alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2 + \cdots + \alpha_n \mathbf{v}_n}$$

where the coefficients are given by

$$\boxed{\begin{aligned} \alpha_i &= \frac{\mathbf{v}^T \mathbf{v}_i}{\mathbf{v}_i^T \mathbf{v}_i} \\ &= \frac{\mathbf{v}^T \mathbf{v}_i}{\|\mathbf{v}_i\|^2} \end{aligned}}$$

These are the dot products of $\underline{v}$ with each of the basis vectors



Orthogonality (Con't)

The key importance of this result is that, if we represent a vector as a linear combination of orthogonal or orthonormal basis vectors, we can determine the coefficients directly from simple inner product computations. It is possible to convert a linearly dependent spanning set of vectors into an orthogonal spanning set by using the well-known *Gram-Schmidt* process. There are numerous programs available that implement the Gram-Schmidt and similar processes, so we will not dwell on the details here.

Gram-Schmidt orthogonalization procedure.

Discrete image transforms (Castleman, 2/e, Ch. 13)

$$y_i = \sum_{j=0}^{N-1} t_{ij} x_j \qquad \text{or} \qquad \underline{y} = \underset{\substack{\uparrow \\ \text{kernel matrix}}}{\underline{T}}\,\underline{x} \qquad (1)$$

linear transformation of vector $\underline{x}$

inner product of input vector $\underline{x}$ with i-th row of $\underline{T}$

If $\underline{T}$ is a unitary matrix

$$\underline{T}^{-1} = \underline{T}^{*t}$$

and $\underline{T}\,\underline{T}^{*t} = \underline{T}^{*t}\underline{T} = \underline{I}$

If $\underline{T}$ is unitary and real it is an orthogonal matrix

$$\underline{T}^{-1} = \underline{T}^{t}$$

and $\underline{T}\,\underline{T}^{t} = \underline{T}^{t}\underline{T} = \underline{I}$ (2)

Note that the i,j-th element of $\underline{T}\,\underline{T}^{t}$ is the inner product of rows i and j. By (2) this is zero unless $i=j$ in which case it is zero. $\Rightarrow$ the rows of $\underline{T}$ are a set of orthonormal vectors.

Example: Discrete Fourier Transform

$$\underline{F}_k = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} f_i \, e^{-j 2\pi k \frac{i}{N}} \qquad \text{or} \qquad \underline{F} = \mathcal{W}\underline{f}$$

$\mathcal{W}$ is a unitary (not orthogonal)

$$w_{ik} = \frac{1}{\sqrt{N}} e^{-j 2\pi k \frac{i}{N}}$$

In general, the rows of T (in this case $\mathcal{W}$) form an orthonormal basis set for all $N \times 1$ vectors.

(1) is simply a coordinate transform which rotates $\underline{x}$ but does not change its length.

Two-dimensional discrete <u>linear</u> transformations

$$\underset{N\times N}{G_{mn}} = \sum_{i=0}^{N-1}\sum_{k=0}^{N-1} \underset{N\times N}{F_{ik}}\, \underbrace{J(i,k,m,n)}_{\text{kernel function of the transform}}$$

kernel matrix is a block matrix

$$\begin{array}{cc} & \begin{array}{cccc} n=1 & n=2 & & n=N \end{array} \\ \begin{array}{c} m=1 \\ m=2 \\ \vdots \\ m=N \end{array} & \begin{bmatrix} [\ \ ] & [\ \ ] & \cdots & [\ \ ] \\ [\ \ ] & [\ \ ] & \cdots & [\ \ ] \\ \vdots & \vdots & & \\ [\ \ ] & [\ \ ] & \cdots & [\ \ ] \end{bmatrix} \end{array}$$

If $J(i,k,m,n) = T_r(i,m)\,T_c(k,m)$

then it is called separable and can be carried out as a rowwise operation followed by a column wise operation or vice versa.

$$G_{mn} = \sum_{i=0}^{N-1} T(i,m)\left[\sum_{k=0}^{N-1} F_{ik}\,T(k,n)\right] \quad \text{or} \quad \underline{G} = \underline{T}\,\underline{F}\,\underline{T}$$

Inverse transform is

$$\underline{F} = \underline{T}^{-1}\underline{G}\,\underline{T}^{-1} = \underline{T}^{*t}\underline{G}\,\underline{T}^{*t}$$

Example: 2-D DFT

$$\underline{T} = \underline{\mathcal{W}}$$

$$\underline{T}^{-1} = \underline{\mathcal{W}}^{*t}$$

$$\therefore \quad \underline{G} = \underline{\mathcal{W}}\,\underline{F}\,\underline{\mathcal{W}} \quad \text{and} \quad \underline{F} = \underline{\mathcal{W}}^{*t}\underline{G}\,\underline{\mathcal{W}}^{*t}$$

Orthogonal transforms

Many transforms used in image processing have only real elements in their kernel matrix.

This is then an orthogonal matrix

The inverse transform is

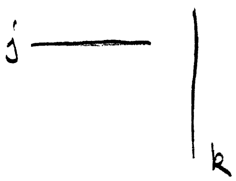
$$\underline{F} = \underline{T}^t G \underline{T}^t$$

If T is symmetric

$$\underline{G} = \underline{T}\underline{F}\underline{T} \quad \text{and} \quad \underline{F} = \underline{T}\underline{G}\underline{T}$$

The rows of the kernel matrix form a set of basis vectors in an N-dimensional vector space.

The rows are orthonormal



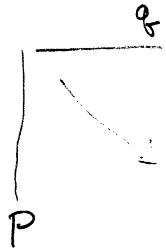
$$\underline{T}\,\underline{T}^{*t} = I \quad \text{or} \quad \sum_{i=0}^{N-1} T_{ji} T^*_{ki} = \delta_{jk}$$

A basis image is generated by inverse transforming a coefficient matrix containing only one non-zero element which is one,

i.e. $\underline{G}^{p,q} = \left\{ \underbrace{\delta_{i-p,\,j-q}}_{\text{only non-zero for } p=q \text{ which is } 1} \right\}$ This is the coefficient matrix

this is the p,q basis image

$$F_{mn} = \sum_{i=0}^{N-1} T(i,m) \sum_{k=0}^{N-1} \delta_{i-p,\,k-q}\, T(k,n) = \underbrace{T(p,m)\, T(q,n)}_{\text{this is the outer product of two rows of } T}$$



fills this in

Sinusoidal transforms

matrix kernel for the DFT

$$\mathcal{W} = \begin{bmatrix} \omega_{00} & \cdots & \omega_{0,N-1} \\ \vdots & \ddots & \vdots \\ \omega_{N-1,0} & \cdots & \omega_{N-1,N-1} \end{bmatrix}$$

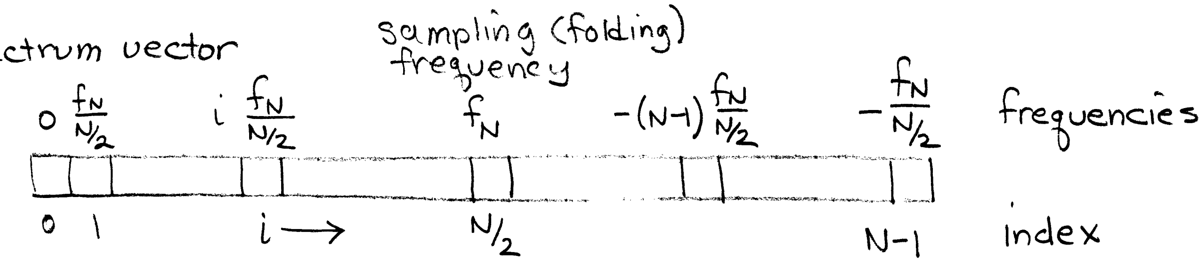
$$\text{where} \quad w_{ik} = \frac{1}{\sqrt{N}} e^{-j2\pi \frac{ik}{N}}$$

$\mathcal{W}$ is unitary

In one dimension

$$\underline{F} = \underline{\mathcal{W}}\underline{f} \quad \text{and} \quad \underline{f} = \underline{\mathcal{W}}^{*t}\underline{F}$$

spectrum vector



This is usually shifted

$$F(u) \Leftrightarrow f(x) \Rightarrow F(u-u_0) \Leftrightarrow e^{j2\pi x \frac{u_0}{N}} f(x)$$

$$= e^{j\pi x} f(x)$$

$$= (-1)^x f(x)$$

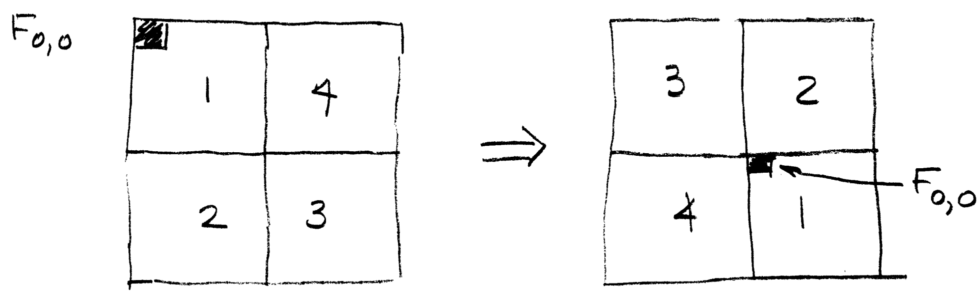
where $u_0 = \frac{N}{2}$ is the shift required to center

In two dimensions

$$\underline{G} = \underline{\mathcal{W}}\,\underline{F}\,\underline{\mathcal{W}} \quad \text{and} \quad \underline{F} = \underline{\mathcal{W}}^{*t}\,\underline{G}\,\underline{\mathcal{W}}^{*t}$$

spectrum matrix ($\underline{G}$)   image matrix ($\underline{F}$)

$$F(u,v) \Leftrightarrow f(x,y) \Rightarrow F\left(u-\tfrac{N}{2},\, v-\tfrac{N}{2}\right) \Leftrightarrow (-1)^{x+y} f(x,y)$$



transformation rearranges spectrum to be symmetric for plotting

discrete cosine transform

$$G_c(m,n) = \alpha(m)\,\alpha(n) \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} g(i,k) \cos\left[\frac{\pi(2i+1)m}{2N}\right] \cos\left[\frac{\pi(2k+1)n}{2N}\right]$$

$$g(i,k) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} \alpha(m)\alpha(n)\, G_c(m,n) \cos\left[\frac{\pi(2i+1)m}{2N}\right] \cos\left[\frac{\pi(2k+1)n}{2N}\right]$$

where $\alpha(0) = \sqrt{\frac{1}{N}}$ and $\alpha(m) = \sqrt{\frac{2}{N}}$ for $1 \le m \le N$

In matrix form, the DCT can be written as a unitary matrix operation

$$\underline{G_c} = \underline{C}\,\underline{g}\,\underline{C}$$

where

$$C_{im} = \alpha(m) \cos\left[\frac{\pi(2i+1)m}{2N}\right]$$

Notes

- can be computed by a fast algorithm
- real valued
- widely used in image compression

discrete sine transform

- similar to DCT but $N = 2^P$
- used in certain image compression applications

$$G_s(m,n) = \frac{2}{N+1} \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} g(i,k) \sin\left[\frac{\pi(i+1)(m+1)}{N+1}\right] \sin\left[\frac{\pi(k+1)(n+1)}{N+1}\right]$$

$$g(i,k) = \frac{2}{N+1} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} G_s(m,n) \sin\left[\frac{\pi(i+1)(m+1)}{N+1}\right] \sin\left[\frac{\pi(k+1)(n+1)}{N+1}\right]$$

$$T_{i,k} = \sqrt{\frac{2}{N+1}}\, \sin\left[\frac{\pi(i+1)(k+1)}{N+1}\right]$$

Discrete Hartley Transform (DHT)

use the basis function $cas(\theta) = \cos(\theta) + \sin(\theta) = \sqrt{2}\cos\left(\theta - \frac{\pi}{4}\right)$

Forward

$$G_{mn} = \frac{1}{N} \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} g_{ik} \, cas\left[\frac{2\pi(im+kn)}{N}\right]$$

$$g_{ik} = \frac{1}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} G_{mn} \, cas\left[\frac{2\pi(im+kn)}{N}\right]$$

identical transforms

$$T_{ik} = \frac{1}{\sqrt{N}} \, cas\left[\frac{2\pi ik}{N}\right]$$

DFT N numbers $\longrightarrow$ N complex symmetric numbers

DHT N numbers $\longrightarrow$ N real numbers

computational alternative to Fourier transform avoiding complex arithmetic

Convolution theorem

$$g(x) = f(x) * h(x) \iff F(u)H_e(u) + F(-u)H_o(u) = G(u)$$

$F(u)$ – Hartley transform of f

$G(u)$ – Hartley transform of g

$H_e(u)$ – even components of Hartley transform of h

$H_o(u)$ – odd components of Hartley transform of h

Eigenvalues & Eigenvectors (Con't)

Example

Suppose that we have a random population of vectors, denoted by $\{\mathbf{x}\}$, with covariance matrix (see the review of probability):

$$\boxed{\mathbf{C}_\mathbf{x} = E\{(\mathbf{x}-\mathbf{m}_\mathbf{x})(\mathbf{x}-\mathbf{m}_\mathbf{x})^T\}}$$

Suppose that we perform a transformation of the form $\mathbf{y} = \mathbf{Ax}$ on each vector $\mathbf{x}$, where the rows of $\mathbf{A}$ are the orthonormal eigenvectors of $\mathbf{C}_\mathbf{x}$. The covariance matrix of the population $\{\mathbf{y}\}$ is

$$\begin{aligned}\mathbf{C}_\mathbf{y} &= E\{(\mathbf{y}-\mathbf{m}_\mathbf{y})(\mathbf{y}-\mathbf{m}_\mathbf{y})^T\}\\ &= E\{(\mathbf{Ax}-\mathbf{Am}_\mathbf{x})(\mathbf{Ax}-\mathbf{Am}_\mathbf{x})^T\}\\ &= E\{\mathbf{A}(\mathbf{x}-\mathbf{m}_\mathbf{x})(\mathbf{x}-\mathbf{m}_\mathbf{x})^T\mathbf{A}^T\}\\ &= \mathbf{A}E\{(\mathbf{x}-\mathbf{m}_\mathbf{x})(\mathbf{x}-\mathbf{m}_\mathbf{x})^T\}\mathbf{A}^T\\ &= \mathbf{AC}_\mathbf{x}\mathbf{A}^T\end{aligned}$$

Suppose we have a random set of vectors

$$\underline{x}_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{im} \end{bmatrix}$$

Compute the covariance matrix

$$\underline{C}_x = E\{(\underline{x}-\underline{m}_x)(\underline{x}-\underline{m}_x)^T\}$$

Consider the transformation $\underline{y} = \underline{A}\,\underline{x}$ where rows of $\underline{A}$ are eigenvectors of $\underline{C}_x$

Then

$$\begin{aligned}\underline{C}_y &= E\{(\underline{y}-\underline{m}_y)(\underline{y}-\underline{m}_y)^T\}\\ &= E\{(\underline{A}\underline{x}-\underline{A}\underline{m}_x)(\underline{A}\underline{x}-\underline{A}\underline{m}_x)^T\}\\ &= E\{\underline{A}(\underline{x}-\underline{m}_x)(\underline{x}-\underline{m}_x)^T\underline{A}^T\}\\ &= \underline{A}\,E\{(\underline{x}-\underline{m}_x)(\underline{x}-\underline{m}_x)^T\}\,\underline{A}^T\end{aligned}$$

diagonal matrix of eigenvalues (variances) $\longrightarrow \underline{C}_y = \underline{A}\,\underline{C}_x\,\underline{A}^T$ so $\underline{y} = \underline{A}\,\underline{x}$ decorrelates the data

Eigenvalues & Eigenvectors

Definition: The *eigenvalues* of a real matrix **M** are the real numbers λ for which there is a nonzero vector **e** such that

$$\boxed{\mathbf{Me} = \lambda\, \mathbf{e}.}$$

The *eigenvectors* of **M** are the nonzero vectors **e** for which there is a real number λ such that **Me** = λ **e**.

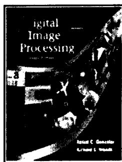
If **Me** = λ **e** for **e** ≠ 0, then **e** is an *eigenvector* of **M** associated with *eigenvalue* λ, and vice versa. The eigenvectors and corresponding eigenvalues of **M** constitute the *eigensystem* of **M**.

Numerous theoretical and truly practical results in the application of matrices and vectors stem from this beautifully simple definition.

eigenvalues.

$$\underline{M}\,\underline{e} = \lambda\, \underline{e}$$

eigenvectors.



Eigenvalues & Eigenvectors (Con't)

Example: Consider the matrix

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

It is easy to verify that $\mathbf{M}\mathbf{e}_1 = \lambda_1\mathbf{e}_1$ and $\mathbf{M}\mathbf{e}_2 = \lambda_2\mathbf{e}_2$ for $\lambda_1 = 1, \lambda_2 = 2$ and

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

In other words, $\mathbf{e}_1$ is an eigenvector of $\mathbf{M}$ with associated
eigenvalue λ_1, and similarly for $\mathbf{e}_2$ and λ_2.

24

$$\underline{M}\,\underline{e} = \lambda\,\underline{e}$$

For example,

$$\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = 1\begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

and

$$\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = 2\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

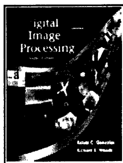
$$\begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

Eigenvalues & Eigenvectors (Con't)

The following properties, which we give without proof, are essential background in the use of vectors and matrices in digital image processing. In each case, we assume a real matrix of order $m \times m$ although, as stated earlier, these results are equally applicable to complex numbers.

1. If $\{\lambda_1, \lambda_2, \ldots, \lambda_q$, $q \leq m$, is set of distinct eigenvalues of $\mathbf{M}$, and $\mathbf{e}_i$ is an eigenvector of $\mathbf{M}$ with corresponding eigenvalue λ_i, $i = 1,2,\ldots,q$, then $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_q\}$ is a linearly independent set of vectors. An important implication of this property: If an $m \times m$ matrix $\mathbf{M}$ has m distinct eigenvalues, its eigenvectors will constitute an orthogonal (orthonormal) set, which means that any m-dimensional vector can be expressed as a linear combination of the eigenvectors of $\mathbf{M}$.

1. The eigenvectors of $\underline{M}$ represent a orthonormal basis set basis set for the m-dimensional vectors



Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Review: Matrices and Vectors

Eigenvalues & Eigenvectors (Con't)

2. The numbers along the main diagonal of a diagonal matrix are equal to its eigenvalues. It is not difficult to show using the definition **Me** = λ **e** that the eigenvectors can be written by inspection when **M** is diagonal.

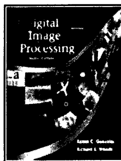
3. A real, symmetric $m \times m$ matrix **M** has a set of m linearly independent eigenvectors that may be chosen to form an orthonormal set. This property is of particular importance when dealing with covariance matrices (e.g., see Section 11.4 and our review of probability) which are real and symmetric.

26

2. $\underline{M}\,\underline{e} = \lambda \underline{e}$

If $\underline{m}$ is diagonal then the diagonal elements of $\underline{m}$ are its eigenvalues.

3. Any real, symmetric $m \times m$ matrix has m lineary, independent eigenvectors that can be used to form an orthonormal basis set.



Digital Image Processing, 2nd ed. www.imageprocessingbook.com

Review: Matrices and Vectors

Eigenvalues & Eigenvectors (Con't)

4. A corollary of Property 3 is that the eigenvalues of an $m \times m$ real symmetric matrix are real, and the associated eigenvectors may be chosen to form an orthonormal set of m vectors.

5. Suppose that **M** is a real, symmetric $m \times m$ matrix, and that we form a matrix **A** whose rows are the m orthonormal eigenvectors of **M**. Then, the product $\mathbf{AA}^T$=I because the rows of **A** are orthonormal vectors. Thus, we see that $\mathbf{A}^{-1} = \mathbf{A}^T$ when matrix **A** is formed in the manner just described.

6. Consider matrices **M** and **A** in 5. The product $\mathbf{D} = \mathbf{AMA}^{-1} = \mathbf{AMA}^T$ is a diagonal matrix whose elements along the main diagonal are the eigenvalues of **M**. The eigenvectors of **D** are the same as the eigenvectors of **M**.

 27

5. let $\underline{A} = \begin{bmatrix} \underline{e}_1^T \\ \underline{e}_2^T \\ \vdots \\ \underline{e}_m^T \end{bmatrix}$ $\quad A^T = [\underline{e}_1 \ \underline{e}_2 \ \dots \ \underline{e}_m]$.

$$\underline{A}\,\underline{A}^T = \begin{bmatrix} \underline{e}_1^T \\ \underline{e}_2^T \\ \vdots \\ \underline{e}_m^T \end{bmatrix} [\underline{e}_1 \ \underline{e}_2 \ \dots \ \underline{e}_m] = \underbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & & 1 \end{bmatrix}}_{\text{identity matrix } I}$$

$\Rightarrow \underline{A}^T = \underline{A}^{-1}$

6. If $\underline{A}\,\underline{M}\,\underline{A}^{-1} = \underline{D}$, a diagonal matrix, then the elements of $\underline{D}$ are the eigenvalues of $\underline{M}$

Eigenvalues & Eigenvectors (Con't)

From Property 6, we know that $\mathbf{C}_\mathbf{y}=\mathbf{A}\mathbf{C}_\mathbf{x}\mathbf{A}^T$ is a diagonal matrix with the eigenvalues of $\mathbf{C}_\mathbf{x}$ along its main diagonal. The elements along the main diagonal of a covariance matrix are the variances of the components of the vectors in the population. The off diagonal elements are the covariances of the components of these vectors.

The fact that $\mathbf{C}_\mathbf{y}$ is diagonal means that the elements of the vectors in the population $\{\mathbf{y}\}$ are ***uncorrelated*** (their covariances are 0). Thus, we see that application of the linear transformation $\mathbf{y} = \mathbf{A}\mathbf{x}$ involving the eigenvectors of $\mathbf{C}_\mathbf{x}$ decorrelates the data, and the elements of $\mathbf{C}_\mathbf{y}$ along its main diagonal give the variances of the components of the **y**'s along the eigenvectors. Basically, what has

Eigenvalues & Eigenvectors (Con't)

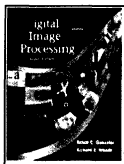
been accomplished here is a coordinate transformation that aligns the data along the eigenvectors of the covariance matrix of the population.

The preceding concepts are illustrated in the following figure. Part (a) shows a data population $\{\mathbf{x}\}$ in two dimensions, along with the eigenvectors of $\mathbf{C}_\mathbf{x}$ (the black dot is the mean). The result of performing the transformation $\mathbf{y}=\mathbf{A}(\mathbf{x} - \mathbf{m}_\mathbf{x})$ on the $\mathbf{x}$'s is shown in Part (b) of the figure.

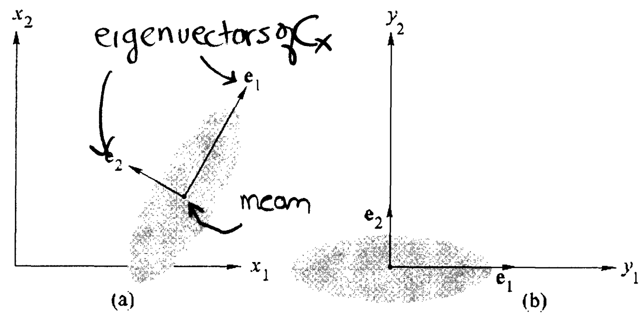
The fact that we subtracted the mean from the $\mathbf{x}$'s caused the $\mathbf{y}$'s to have zero mean, so the population is centered on the coordinate system of the transformed data. It is important to note that all we have done here is make the eigenvectors the

Eigenvalues & Eigenvectors (Con't)

new coordinate system (y_1,y_2). Because the covariance matrix of the **y**'s is diagonal, this in fact also decorrelated the data. The fact that the main data spread is along $\mathbf{e}_1$ is due to the fact that the rows of the transformation matrix **A** were chosen according the order of the eigenvalues, with the first row being the eigenvector corresponding to the largest eigenvalue.



Eigenvalues & Eigenvectors (Con't)



$\underline{y} = \underline{A}(\underline{x} - \underline{m}_x)$

eigenvectors form

the coordinate system
in which the data is decorrelated

Note λ_1 corresponding to $y_1 > \lambda_2$
which corresponds to y_2

Eigenvector based transforms

For an $N \times N$ matrix $\underline{A}$ there are N scalars λ_k, $k=0,\ldots,N-1$

$$|\underline{A} - \lambda_k \underline{I}| = 0$$

These are called the eigenvalues. There are N corresponding eigenvectors $\underline{v}_k$

$$\underline{A}\,\underline{v}_k = \lambda_k \underline{v}_k$$

The eigenvectors form an orthonormal basis set.

Let $\underline{x}$ be a $N \times 1$ random vector, i.e., each x_i is a random variable

estimate mean $\quad \underline{m}_x \approx \frac{1}{L} \sum_{l=1}^{L} \underline{x}_l$

set of L random vectors

estimate covariance matrix $\quad \underline{C}_x = E\left\{(\underline{x} - \underline{m}_x)(\underline{x} - \underline{m}_x)^t\right\} \approx \frac{1}{L} \sum_{l=1}^{L} (\underline{x}_l \underline{x}_l^t - \underline{m}_x \underline{m}_x^t)$

$N \times N$, real, symmetric $\qquad$ for collected data

define $\quad \underline{y} = \underline{A}(\underline{x} - \underline{m}_x)$

rows are eigenvectors of $\underline{C}_x$

Compute $\quad \underline{C}_y = \underline{A}\,C_x \underline{A}^t = [\lambda_k \underline{v}_k]\,\underline{A}^t = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_N \end{bmatrix}$

but $\underline{v}_k$ are the rows (eigenvalues) of A

IMPORTANT RESULTS

(1) A removes correlation among the variables

(2) λ_k is the variance of the k-th transformed variable

Dimensional Reduction

Assume $\underline{m} = \underline{0}$

Drop the lower N−M rows of $\underline{A}$. to give $\underline{B}$

$$\hat{y} = Bx$$

estimate of variable y.

Invert to get

$$\hat{x} = B^t \hat{y}$$

where $\underline{\hat{x}}$ is an approximation of $\underline{x}$ using only M components.

The error is only $\sum_{k=M+1}^{N} \lambda_k = MSE$

Karhunen-Loéve Transform.

$$y = \underset{\substack{\uparrow \\ \text{eigenvectors of } C_x}}{\underline{A}} (\underline{x} - \underline{m}_x)$$

By dropping m−N components we can dramatically reduce the dimensionality of a set of vectors (images)

Some comments

- As correlation between adjacent pixels approach one, the K-L basis functions approach those of the DCT

Example: 24 band multispectral image, 1000 x 1000 pixels/band

Compute covariance matrix for 1,000,000 24-element random variables.

Transform, keeping only m largest eigenvalues.

If M is small this is a terrific savings in memory.

SVD (singular value decomposition)

Write $N \times N$ matrix $\underline{A}$ as

$$\underline{A} = \underline{U}\,\underline{\Lambda}\,\underline{V}^t \qquad (1)$$

columns $\underline{U}$ are eigenvectors of $\underline{A}\,\underline{A}^t$ } orthogonal.

columns $\underline{V}$ are eigenvectors of $\underline{A}^t\underline{A}$ }

$\underline{\Lambda}$ is a diagonal matrix with the singular values of $\underline{A}$ along the diagonal

$$\underline{\Lambda} = \underline{U}^t A\, \underline{V} \qquad (2)$$

Equations (1) and (2) define a transform pair.

- eigenvectors must be computed for each image
- has at most N non zero elements and, if some are zero, this gives data compression
- usually several of the singular values are small enough to ignore for even more data compression.

SVD Example.

$$\underline{A} = \begin{bmatrix} 0 & 1 & 2 & 1 & 0 \\ 1 & 3 & 4 & 3 & 1 \\ 2 & 4 & 5 & 4 & 2 \\ 1 & 3 & 4 & 3 & 1 \\ 0 & 1 & 2 & 1 & 0 \end{bmatrix}$$

$$\underline{A}\,\underline{A}^t = \begin{bmatrix} 6 & 14 & 18 & 14 & 6 \\ 14 & 36 & 48 & 36 & 14 \\ 18 & 48 & 65 & 48 & 18 \\ 14 & 36 & 48 & 36 & 14 \\ 6 & 14 & 18 & 14 & 6 \end{bmatrix}$$

Now compute eigenvectors and eigenvalues of this matrix.

$$\underline{U} = \begin{bmatrix} 0.186 & 0.638 & 0.241 & -0.695 & -0.695 \\ 0.476 & 0.058 & -0.52 & -0.133 & -0.128 \\ 0.691 & -0.422 & 0.587 & 0 & 0 \\ 0.476 & 0.058 & -0.52 & 0.133 & 0.128 \\ 0.186 & 0.638 & 0.241 & 0.695 & 0.695 \end{bmatrix} \qquad \underline{\lambda} = \begin{bmatrix} 147.07 \\ 1.872 \\ 0.058 \\ 0 \\ 0 \end{bmatrix}$$

Forward transform

$$\underline{\Lambda} = \underline{U}^t \underline{A}\,\underline{U} = \begin{bmatrix} 12.585 & 0 & 0 & 0 & 0 \\ 0 & -1.142 & 0 & 0 & 0 \\ 0 & 0 & 0.557 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Inverse transform

$$\underline{A} = \underline{U}\,\underline{\Lambda}\,\underline{U}^t = \begin{bmatrix} 0 & 1 & 2 & 1 & 0 \\ 1 & 3 & 4 & 3 & 1 \\ 2 & 4 & 6 & 4 & 2 \\ 1 & 3 & 4 & 3 & 1 \\ 0 & 1 & 2 & 1 & 0 \end{bmatrix}$$

as expected.

Hadamard transform (also called the Walsh transform)

- symmetric
- seperable
- unitary
- only +1 and −1 as elements.
- exists for $N = 2^n$

$$\frac{1}{\sqrt{2}}\underline{H}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Can be successively generated as

$$\frac{1}{\sqrt{N}}\underline{H}_N = \frac{1}{\sqrt{N}}\begin{bmatrix} \underline{H}_{N/2} & \underline{H}_{N/2} \\ \underline{H}_{N/2} & -\underline{H}_{N/2} \end{bmatrix}$$

For example for $N = 8$

$$H_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$H_4 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

sequency (sign changes)

$$H_8 = \frac{1}{\sqrt{8}}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}\begin{matrix} 0 \\ 7 \\ 3 \\ 4 \\ 1 \\ 6 \\ 2 \\ 5 \end{matrix}$$

ordered Hadamard transform

$$H_8 = \frac{1}{\sqrt{8}}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}\begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix}$$

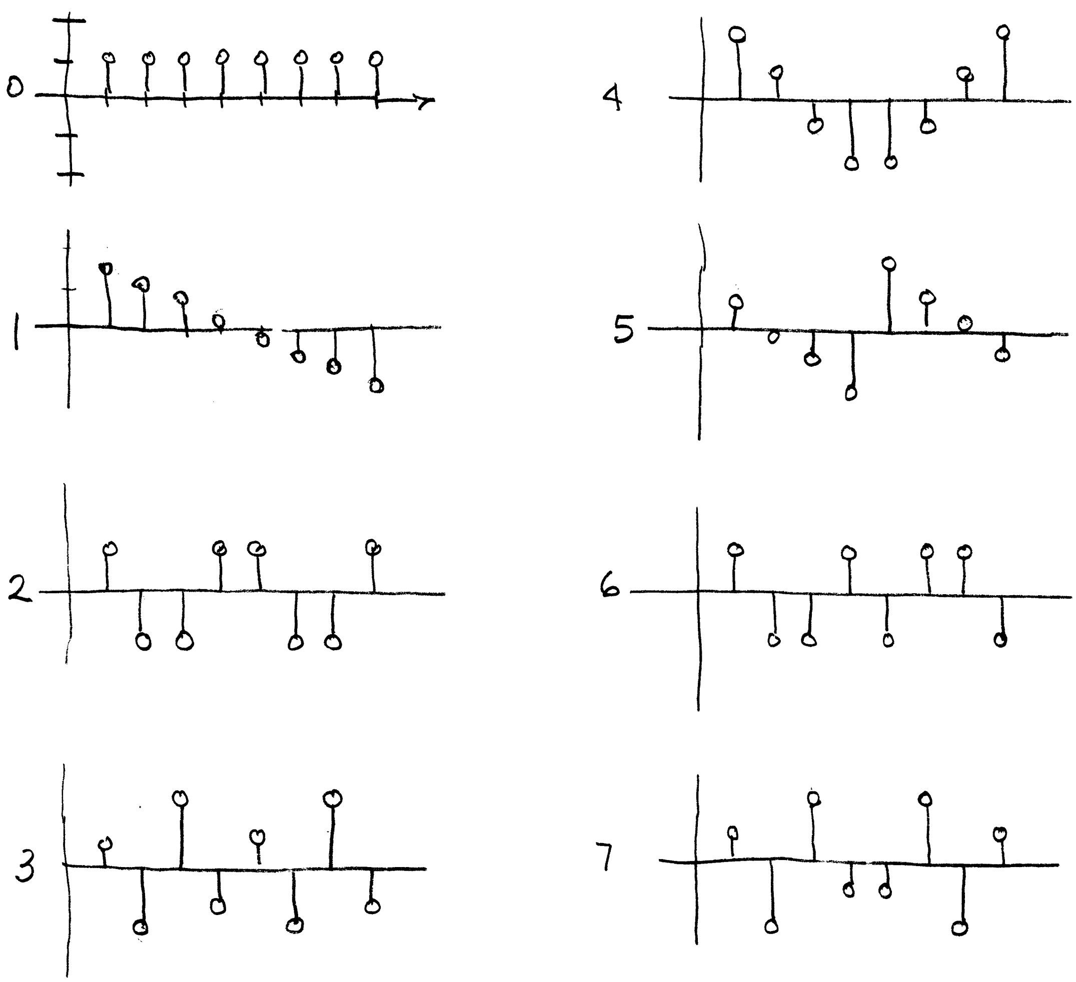
The slant transform – has a fast transform, used for image compression

$$S_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$S_N = \frac{1}{\sqrt{2}} \left[\begin{array}{cc|c|cc|c} 1 & 0 & & 1 & 0 & \\ a_N & b_N & \underline{0} & -a_N & b_N & \underline{0} \\ \hline \multicolumn{2}{c|}{\underline{0}} & \underline{I} & \multicolumn{2}{c|}{\underline{0}} & \underline{I} \\ \hline 0 & 1 & & 0 & -1 & \\ -b_N & a_N & \underline{0} & b_N & a_N & \underline{0} \\ \hline \multicolumn{2}{c|}{\underline{0}} & \underline{I} & \multicolumn{2}{c|}{\underline{0}} & -\underline{I} \end{array} \right] \left[\begin{array}{c|c} \underline{S}_{N/2} & \underline{0} \\ \underline{0} & \underline{S}_{N/2} \end{array} \right]$$

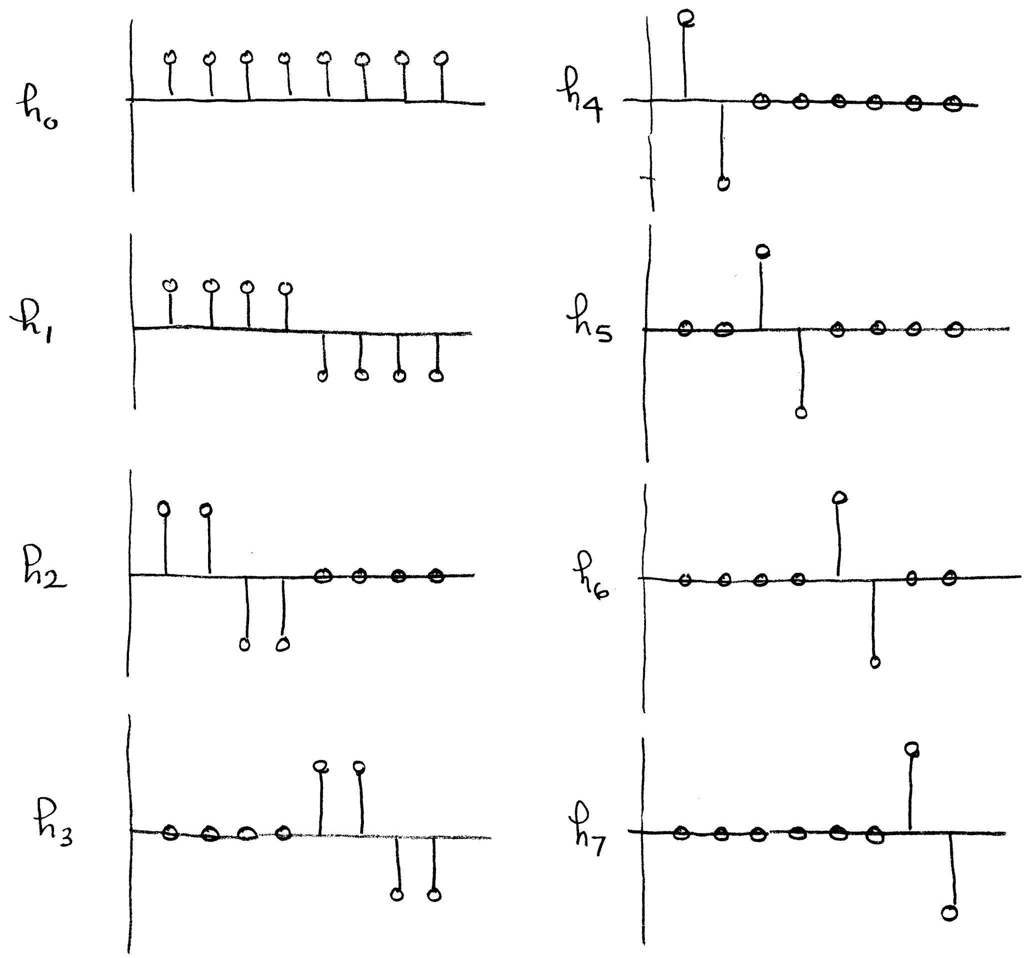
$$a_{2N} = \sqrt{\frac{3N^2}{4N^2-1}} \quad \text{and} \quad b_{2N} = \sqrt{\frac{N^2-1}{4N^2-1}}$$

called slant because it has linear basis functions



Haar

- symmetric
- separable
- unitary
- $N = 2^p$
- vary in both scale (width) and position; used for wavelets



$N=8$ Haar transform basis functions
p specifies scale, q specifies width

Mathematically,

$$h_0(x) = \frac{1}{\sqrt{N}}$$

$$h_k(x) = \frac{1}{\sqrt{N}} \begin{cases} 2^{p/2} & \frac{q-1}{2^p} \le x \le \frac{q-\frac{1}{2}}{2^p} \\ -2^{p/2} & \frac{q-\frac{1}{2}}{2^p} \le x \le \frac{q}{2^p} \\ 0 & \text{otherwise} \end{cases}$$

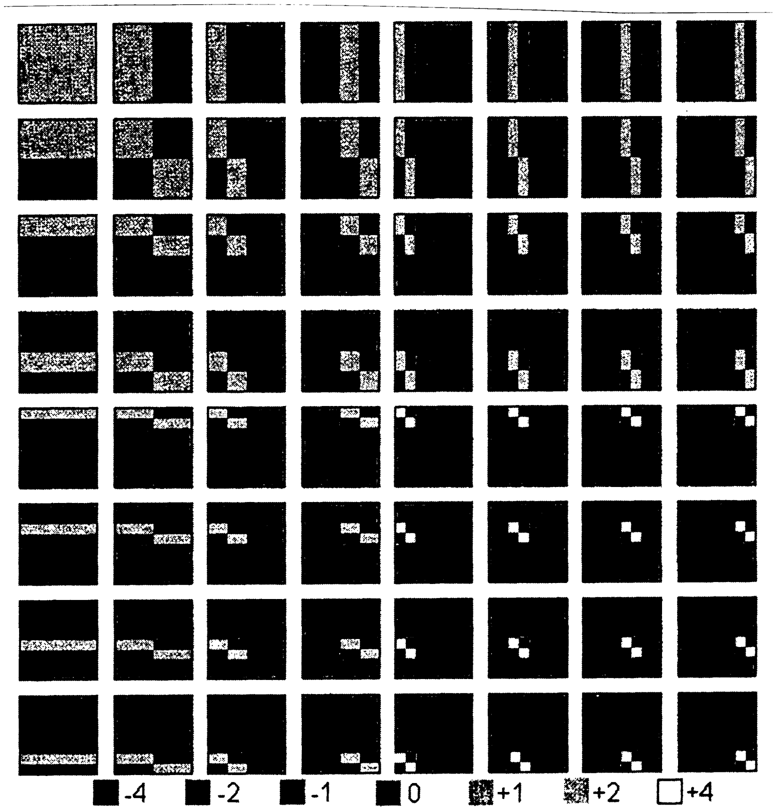
where $k = 2^p + q - 1$ — 2^p is the largest power of 2 such that $2^p \le k$ and $q-1$ is the remainder.

$$\underline{Hr} = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$

Haar functions are scaled, shifted versions of an odd rectangular pulse.

Notes

- not as useful to plot in terms of k since p and q refer to scale and shift
- addresses lines and edges directly so it can be used to call attention to line and edge features



lines

lines

looks for small features at different locations in image

Figure 13–6 The Haar transform basis images for $N = 8$

Transform domain filtering

- modify weighting coefficients prior to reconstructing image via inverse transform

- if either (desirable) signal or (undesirable) noise components of the image resemble one or a few of the basis images of a particular transform, then that transform will be useful in separating the two.

- The Haar transform is a good candidate for detecting vertical and horizontal lines and edges since several of its basis images match those features.

$$
\underbrace{\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}}_{\text{Image}}
\qquad
\underbrace{\begin{bmatrix}
2 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
2 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
2.83 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}}_{\text{Harr transform}}
$$

← lines in this row

Haar features match this edge

edges in this column

Figure 13–8 Edge detection in an eight-by-eight image

Examples of various transforms of a impulse at (0, 0)

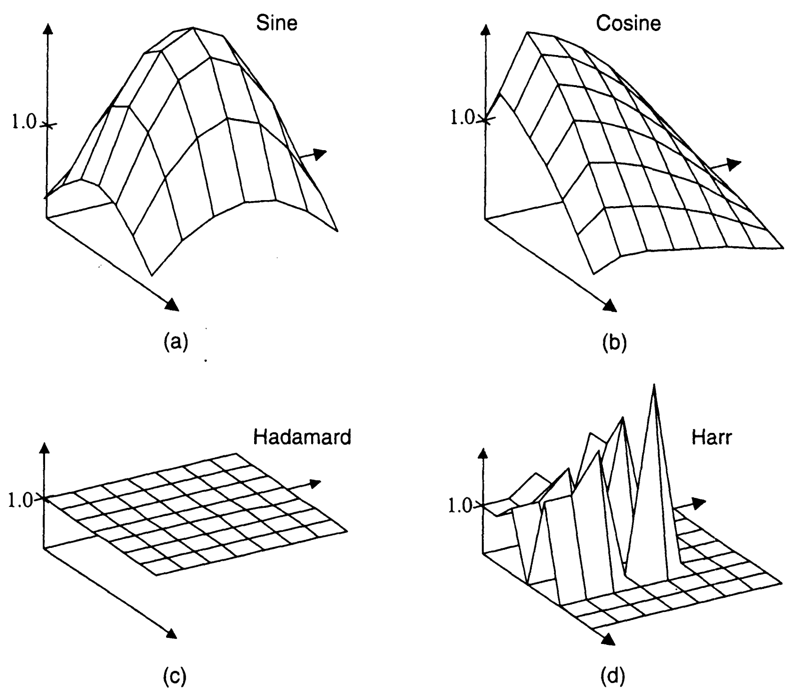


Figure 13–9 Transforms of an image containing an impulse: (a) DST; (b) DCT; (c) Hadamard: (d) Haar. The input is an eight-by-eight matrix, zero everywhere except the upper left element, which has value eight

11.4 Use of principal components for analysis.

For an RGB image we can write each pixel as

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

For n registered images the corresponding pixel vector will be

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

The mean vector for a population of pixels, i.e., random vectors $\underline{x}$, is

$$\underline{m}_x = E\{\underline{x}\}$$

where we compute the expected value of each element.

The covariance matrix of this vector population is given by

$$\underline{C}_x = E\{(\underline{x}-\underline{m}_x)(\underline{x}-\underline{m}_x)^T\}$$

For k samples from a random population

$$\underline{m}_x = \frac{1}{K}\sum_{k=1}^{K} \underline{x}_k$$

$$\underline{C}_x = \frac{1}{K}\sum_{k=1}^{K} \underline{x}_k\underline{x}_k^T - \underline{m}_x\underline{m}_x^T$$

Example: $\underline{x}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ $\underline{x}_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ $\underline{x}_3 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$ $\underline{x}_4 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$

Using above formulae

$$\underline{m}_x = \frac{1}{4}\begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} \qquad \underline{C}_x = \frac{1}{16}\begin{bmatrix} 3 & 1 & 1 \\ 1 & 3 & -1 \\ 1 & -1 & 3 \end{bmatrix}$$

Transform the data $\underline{x}$ by

$$\underline{y} = \underline{A}(\underline{x} - \underline{m}_x)$$

where $\underline{A} = \begin{bmatrix} \underline{e}_1^T \\ \underline{e}_2^T \\ \vdots \\ \underline{e}_n^T \end{bmatrix}$ where $\underline{e}_1, \ldots \underline{e}_n$ are the eigenvectors of $\underline{C}_x$

and $\lambda_1 > \lambda_2 > \ldots > \lambda_n$

This is called a Hotelling transformation. It is optimum in the sense that it minimizes error between $\underline{x}$ and an eigen vector approximation $\hat{\underline{x}}_k$.

$$\underline{m}_y = E\{\underline{y}\} = 0$$

and $\underline{C}_y = \underline{A}\,\underline{C}_x\,\underline{A}^T$

where $\underline{C}_y = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{bmatrix}$

This incidentally indicates that the $\underline{y}$ vectors are uncorrelated.

This can be inverted to give

$$\underline{x} = \underline{A}^T \underline{y} + m_x$$

note: $\underline{A}^{-1} = \underline{A}^T$ because $\underline{A}$ is orthonormal

Construct an estimate $\hat{\underline{x}}$ of $\underline{x}$ using only the k largest eigenvalues

$$\hat{\underline{x}} = \underline{A}_k^T \underline{y} + \underline{m}_x$$

$$A = \begin{bmatrix} \underline{e}_1^T \\ \underline{e}_2^T \\ \\ \underline{e}_k^T \end{bmatrix}$$

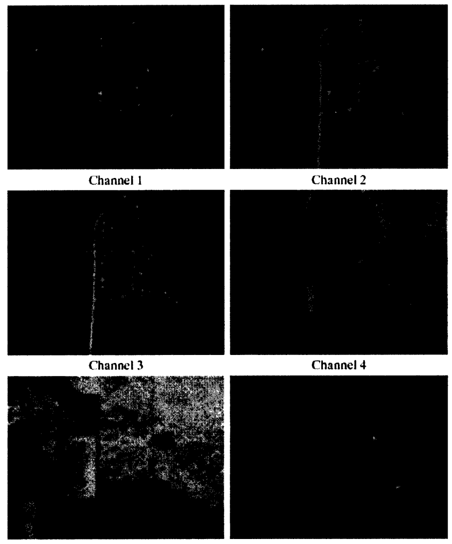
The rms error between $\hat{x}$ and x is

$$e_{rms} = \sum_{j=k+1}^{n} \lambda_j$$

Chapter 11
Representation & Description

FIGURE 11.26 Six spectral images from an airborne scanner. (Courtesy of the Laboratory for Applications of Remote Sensing, Purdue University.)

384x239 pixel images

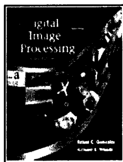


 33

Images from a six-band multispectral scanner.

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}$$

This is an example of region analysis using eigenvectors.



Chapter 11
Representation & Description

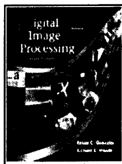
TABLE 11.4
Channel numbers and wavelengths.

Channel	Wavelength band (microns)
1	0.40–0.44
2	0.62–0.66
3	0.66–0.72
4	0.80–1.00
5	1.00–1.40
6	2.00–2.60

visible

infrared

Six wave bands corresponding to the images in Figure 11.26



Digital Image Processing, 2nd ed.

www.imageprocessingbook.com

Chapter 11
Representation & Description

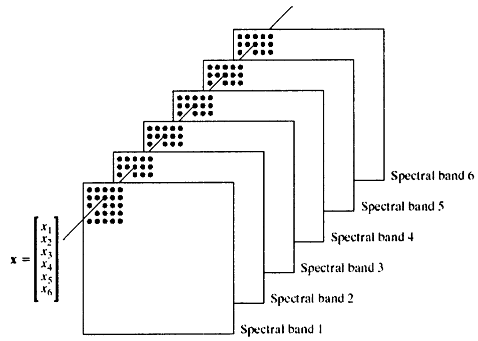


FIGURE 11.27 Formation of a vector from corresponding pixels in six images.

λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
3210	931.4	118.5	83.88	64.00	13.40

TABLE 11.5 Eigenvalues of the covariance matrix obtained from the images in Fig. 11.26.

35

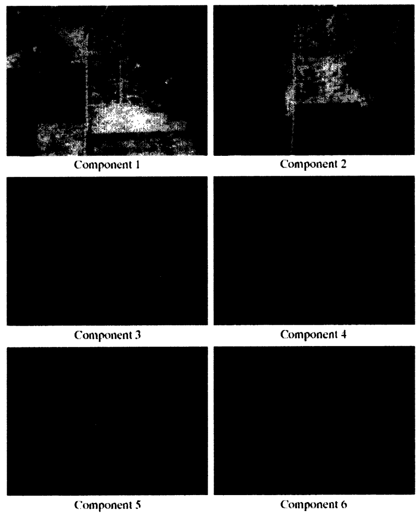
Computed eigen values

Chapter 11
Representation & Description

$\underline{e}_1$ corresponding to λ_1

$\underline{y} = \underline{e}_1^T(\underline{x} - \underline{m}_x)$

$\underline{e}_2$ corresponding to λ_2



$\underline{e}_6$ corresponding to λ_6

FIGURE 11.28 Six principal-component images computed from the data in Fig. 11.26. (Courtesy of the Laboratory for Applications of Remote Sensing, Purdue University.)

 36

Principal component images.

What the Purdue researchers did was transform the images according to

$$\underline{y} = \underline{A}(\underline{x} - \underline{m}_x)$$

They then approximated the images by $\hat{\underline{y}} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ dropping $y_3 - y_6$.

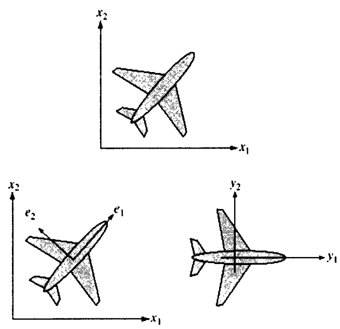
Then $$\hat{x} = \begin{bmatrix} \underline{e}_1^T \\ e_2^T \end{bmatrix}^T \hat{\underline{y}} + \underline{m_x}$$

$$\hat{\underline{x}} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} = [\underline{e}_1 \ \underline{e}_2] \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \underline{m}_x$$

images in components $\hat{\underline{x}}_1$ $\hat{\underline{x}}_2$ correspond to about 94% of the total variance.

Data compression: store only $\underbrace{\hat{x}_1, \hat{x}_2}_{\text{images}}$, $\underline{m}_x$ and the $\underbrace{\text{first two rows of } \underline{A}}_{\text{the eigenvectors}}$.

Chapter 11
Representation & Description



a
b c

FIGURE 11.29 (a) An object. (b) Eigenvectors. (c) Object rotated by using Eq. (11.4-6). The net effect is to align the object along its eigen axes.

Form 2-D vectors from the coordinates of the boundary or region.

Analyze this set of random vectors $\underline{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

Compute $\underline{C}_x$ and $\underline{m}_x$

Compute eigenvectors $\underline{e}_1$ and $\underline{e}_2$ of $\underline{C}_x$

Allows us to account for rotation of objects when we do pattern recognition.
Translation handled by $\underline{m}_x$

This is an example of boundary analysis using eigenvectors.