Global Optimization Problems in Computer Vision*

P. SUSSNER†, P. M. PARDALOS‡, G. X. RITTER§

University of Florida, Gainesville, FL 32611

Abstract. In the field of computer vision, computer scientists extract knowledge from an image by manipulating it through image transforms. In the mathematical language of image algebra an image transformation often corresponds to an image-template product. When performing this operation on a computer, savings in time and memory as well as a better fit to the specific computer architecture can often be achieved by using the technique of template decomposition. In this paper we use global optimization techniques to solve a general problem of morphological template decomposition.

1. Introduction

Computer vision is the science of automatic image recognition and image understanding [2, 8, 9, 11]. In most cases an image can be thought of as a function from a discrete rectangular array into the set of the real numbers. Hence, there is a natural correspondence between these images and the matrices over the real numbers. A multitude of image processing techniques can be used in order to extract information from the image. Usually the image is transformed through "neighborhood" or "template" operations. In this case each pixel value in the resulting image depends on a certain set of the values of neighboring pixels. Computer vision is an extremely complex task for various reasons such as the following:

1. Due to their large sizes, digital images require huge amounts of memory for storage. For example a typical medical image of size 480×512 contains 245760 pixels whose values range from 0 to 255. Hence, it requires about a quarter of a megabyte of storage space. Even higher resolutions occur in computer tomography, satellite images and military images for automatic target recognition.
2. Neighborhood transformations on an image take a long time to compute, since they not only involve operations on each of the picture elements, but also operations on a number k of it's nearest neighbors. Hence, at least $(k+1) * 245760$ operations have to be performed for the medical image above.

Many disciplines use factorization techniques in order to simplify complex tasks. Computer vision uses two different methods of decomposition:

1. *Matrix decomposition:* Since there is a natural correspondence between matrices and discrete, rectangular images, matrix decomposition can lead to image compression. An example is the Singular Value Decomposition, which is based on the Spectral Theorem from linear algebra [1]. The theory of minimax algebra [4] gives rise to new ways of decomposing matrices. A minimax decomposition analogous to the spectral decomposition has already been discovered [23].
2. *Template decomposition:* In terms of image algebra, a neighborhood operation constitutes a product of a structure called "template" and the mathematical structure of an image. If a template $\mathbf{r}$ *decomposes* as a product of templates $\mathbf{t}$ and $\mathbf{s}$, then an application of $\mathbf{r}$ to the image $\mathbf{a}$ yields the same result as a successive application of $\mathbf{t}$ and $\mathbf{s}$. Hence, the users of image processing software have an alternative, and often faster way of computing a neighborhood transformation at their disposal [6, 5, 14, 15]. The

* This research was partially supported by US Air Force Contract F08635–89–C-0134.

† E-mail: ps0@cis.ufl.edu

‡ E-mail: pardalos@math.ufl.edu

§ E-mail: ritter@cis.ufl.edu

theory of mathematical morphology restricts itself to image–template and template–template products, which involve only "max" or "min" and "+" as basic operations [16, 17, 29, 30, 31, 12].

In this paper we show that decomposition problems in computer vision can be formulated in a very general form as Global Optimization Problems. Since morphological image processing usually involves only the morphological operations "maximum", "minimum" and "+" together with templates of a fairly small size, Global Optimization seems to be well suited for solving morphological decomposition problems.

Before we proceed, let us discuss some frequently used notations. As usual, the symbol $\mathbb{N}$ stands for the set of natural numbers, $\mathbb{Z}$ for the set of integers, $\mathbb{R}$ stands for the set of real numbers and $\mathbb{C}$ for the set of complex numbers. We use the symbols $\vee$ and $\wedge$ to denote the binary operations of maximum and minimum. The positive part of a real number x, denoted by x^+, is defined as $0 \vee x$. Similarly the negative part of x is $x^- = 0 \vee (-x)$. Matrix inequalities are always defined entrywise. For example, given matrices A and B, the inequality $A \leq B$ means $a_{ij} \leq b_{ij} \quad \forall\, i, j$. If γ is an associative and commutative operation on a finite set $X = \{x_1, \ldots, x_n\}$, then we abbreviate the expression $x_1 \gamma \ldots \gamma x_n$ by $\Gamma_{i=1}^{n} x_i$ or by $\Gamma_{i \in \{1,\ldots,n\}} x_i$. If f is a function on X, then $f(X)$ denotes $\{f(x_1), \ldots, f(x_n)\}$. A sequence $(k_1, k_2, \ldots, k_n)$ is called strictly increasing if $k_1 < k_2 < \ldots < k_n$.

2. Image Algebra

In this chapter we give a brief introduction to the theory of image algebra. Image algebra provides all the mathematical definitions and tools for computer vision [25, 24, 26, 21, 22]. The main objects of interest, which require a formal definition, are images and operations on images.

Images and Templates

Point Sets and Value Sets. A *point set* is a subset of a topological space. In digital image processing point sets are often assumed to be rectangular discrete arrays.

A *value set* is a homogeneous algebra such as $\mathbb{Z}$, $\mathbb{R}$, $\mathbb{R}_{-\infty} = \mathbb{R} \cup \{-\infty\}$, $\mathbb{R}_{+\infty}$ or $\mathbb{C}$.

Images. An $\mathbb{F}$–valued *image* $\mathbf{a}$ is the graph of a function $\mathbf{a} : \mathbf{X} \to \mathbb{F}$ where $\mathbf{X}$ denotes some point set and $\mathbb{F}$ denotes some value set. Obviously $\mathbf{a}$ can be represented as a matrix if $\mathbf{X}$ is a rectangular discrete array.

The set of all images $\mathbf{a} : \mathbf{X} \to \mathbb{F}$ is denoted by $\mathbb{F}^{\mathbf{X}}$. The notation $\mathbb{R}_{-\infty}^{\mathbf{X}}$ is frequently used instead of $(\mathbb{R}_{-\infty})^{\mathbf{X}}$.

Templates. Let $\mathbf{X}$, $\mathbf{Y}$ be point sets and $\mathbb{F}$ a value set with neutral element 0. A generalized $\mathbb{F}$–valued *template* $\mathbf{t}$ from $\mathbf{Y}$ to $\mathbf{X}$ is a function $\mathbf{t} : \mathbf{Y} \to \mathbb{F}^{\mathbf{X}}$. The set of all templates $\mathbf{t} : \mathbf{Y} \to \mathbb{F}^{\mathbf{X}}$ is denoted by $\left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{Y}}$. So for each $\mathbf{y} \in \mathbf{Y}$ the value $\mathbf{t}(\mathbf{y})$ is an $\mathbb{F}$–valued image on $\mathbf{X}$. We define $\mathbf{t}_{\mathbf{y}} = \mathbf{t}(\mathbf{y})$ and call the point $\mathbf{y}$ the *target point* of $\mathbf{t}$.

The *support* of $\mathbf{t}_{\mathbf{y}}$ is defined as follows:

$$S(\mathbf{t}_{\mathbf{y}}) = \{\mathbf{x} \in \mathbf{X} \,:\, \mathbf{t}_{\mathbf{y}}(\mathbf{x}) \neq 0\}$$

Note that the neutral element 0 depends on the algebraic structure of $\mathbb{F}$. For example the element 0 represents the neutral element for $\mathbb{R}$ together with $+$, whereas the element $-\infty$ represents the neutral element for $\mathbb{R}_{-\infty}$ together with $\vee$.

A template $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ is called *translation invariant* if and only if for each triple $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbf{X}$ with $\mathbf{x} + \mathbf{z} \in \mathbf{X}$ and $\mathbf{y} + \mathbf{z} \in \mathbf{X}$ the following equation holds:

$$\mathbf{t}_{\mathbf{y}}(\mathbf{x}) = \mathbf{t}_{\mathbf{y}+\mathbf{z}}(\mathbf{x} + \mathbf{z}) \; .$$

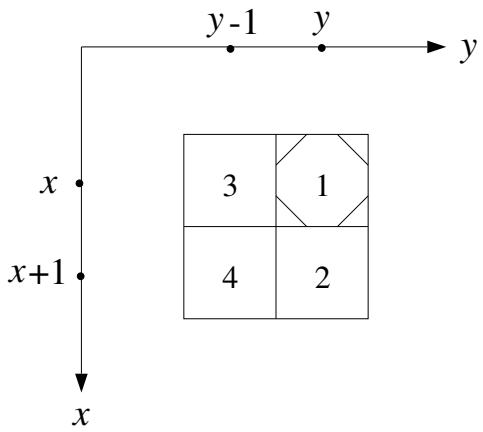
In this case it suffices to know $\mathbf{t}_{\mathbf{y_0}}$ for an arbitrary $\mathbf{y}_0 \in \mathbf{X}$. If $\mathbf{X}$ is discrete and $S(\mathbf{t}_{\mathbf{y_0}})$ is finite then $\mathbf{t}$ can be represented pictorially. If in particular $S(\mathbf{t}_{\mathbf{y_0}})$ is rectangular then $\mathbf{t}$ is called a *rectangular template*.

Example 1. Let $\mathbf{X} = \mathbb{Z}^2 = \mathbb{Z} \times \mathbb{Z}$ and let $\mathbf{t} \in \left(\mathbb{R}^{\mathbf{X}}\right)^{X}$ be an invariant template given by:

$$\begin{aligned} \mathbf{t}_{(0,\,0)}(0,\,0) &= 1 \\ \mathbf{t}_{(0,\,0)}(0,\,1) &= 2 \\ \mathbf{t}_{(0,\,0)}(-1,\,0) &= 3 \\ \mathbf{t}_{(0,\,0)}(-1,\,1) &= 4 \end{aligned}$$

This template can be visualized as follows (the hashed cell indicates the location of the target point):

Figure 1. Pictorial representation of a translation invariant template.



Remark. Invariant templates over a discrete, rectangular support can be characterized by matrices up to the location of the target pixel: Let $\mathbf{X}$ denote the set $\{1, \ldots, n\} \times \{1, \ldots, m\}$. Notice that for each $\mathbf{x} \in \mathbf{X}$ there is a bijection $f^{\mathbf{x}}$ between $n \times m$ matrices over a value set $\mathbb{F}$ and invariant templates from $\mathbf{X}$ to $\mathbb{F}^{\mathbf{X}}$. Under this bijection a matrix $A \in \mathbb{F}^{\mathbf{X}}$ is mapped to a template $\mathbf{r}^{(\mathbf{x},\,A)}$. This template is determined by $\mathbf{r}^{(\mathbf{x},\,A)}_{\mathbf{x}}(i,\,j) = a_{ij}$ for all $(i,\,j) \in \mathbf{X}$.

For the remainder of this chapter, value sets can always be identified by the symbols $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ and point sets by the symbols $\mathbb{F}, \mathbb{G}, \mathbb{H}$. The value set $\mathbf{X}$ is always assumed to be finite.

Operations on Images and Templates

Operations on Images. The basic operations on $\mathbb{F}$–valued images stem from the naturally induced operations on the value set $\mathbb{F}$. In particular, addition, multiplication and maximum are defined as follows:

$$\begin{aligned}
\mathbf{a} + \mathbf{b} &= \{(\mathbf{x}, \mathbf{c}(\mathbf{x})) \,:\, \mathbf{c}(\mathbf{x}) = \mathbf{a}(\mathbf{x}) + \mathbf{b}(\mathbf{x}),\, \forall \mathbf{x} \in \mathbf{X}\} \quad \forall\, \mathbf{a}, \mathbf{b} \in \mathbb{R}_{-\infty}^{\mathbf{X}} \\
\mathbf{a} * \mathbf{b} &= \{(\mathbf{x}, \mathbf{c}(\mathbf{x})) \,:\, \mathbf{c}(\mathbf{x}) = \mathbf{a}(\mathbf{x}) * \mathbf{b}(\mathbf{x}),\, \forall \mathbf{x} \in \mathbf{X}\} \quad \forall\, \mathbf{a}, \mathbf{b} \in \mathbb{R}^{\mathbf{X}} \\
\mathbf{a} \vee \mathbf{b} &= \{(\mathbf{x}, \mathbf{c}(\mathbf{x})) \,:\, \mathbf{c}(\mathbf{x}) = \mathbf{a}(\mathbf{x}) \vee \mathbf{b}(\mathbf{x}),\, \forall \mathbf{x} \in \mathbf{X}\} \quad \forall\, \mathbf{a}, \mathbf{b} \in \mathbf{R}_{-\infty}^{\mathbf{X}} \,.
\end{aligned}$$

We adopt the usual convention $r + (-\infty) = (-\infty) + r = -\infty \quad \forall\, r \in \mathbb{R}_{-\infty}$.

The more complex operations which we define in this section are based on a binary operation between value sets $\circ \,:\, \mathbb{F} \times \mathbb{G} \to \mathbb{H}$ and an associative and commutative operation γ on the value set $\mathbb{H}$.

Operations between Images and Templates. If $\mathbf{a} \in \mathbb{F}^{\mathbf{X}}$, $\mathbf{t} \in \left(\mathbb{G}^{\mathbf{X}}\right)^{\mathbf{Y}}$ then $\mathbf{a} \textcircled{\gamma} \mathbf{t} \in \mathbb{H}^{\mathbf{Y}}$ — the *generalized product* of the image $\mathbf{a}$ with the template $\mathbf{t}$ — can be formed:

$$\mathbf{a} \textcircled{\gamma} \mathbf{t} = \left\{ (\mathbf{y},\, \mathbf{b}(\mathbf{y})) \,:\, \mathbf{b}(\mathbf{y}) = \underset{\mathbf{x} \in \mathbf{X}}{\Gamma} (\mathbf{a}(\mathbf{x}) \circ \mathbf{t}_{\mathbf{y}}(\mathbf{x})),\, \mathbf{y} \in \mathbf{Y} \right\} .$$

If for example $\mathbf{a} \in \mathbb{R}_{-\infty}^{\mathbf{X}}$ and $\mathbf{t} \in \left(\mathbb{R}_{-\infty}^{\mathbf{X}}\right)^{\mathbf{Y}}$, then the *generalized convolution* ("$\oplus$") and the *additive maximum* ("$\boxtimes$") are defined as follows:

$$\begin{aligned}
\mathbf{a} \boxtimes \mathbf{t} &= \left\{ (\mathbf{y},\, \mathbf{b}(\mathbf{y})) \,:\, \mathbf{b}(\mathbf{y}) = \bigvee_{\mathbf{x} \in \mathbf{X}} \mathbf{a}(\mathbf{x}) + \mathbf{t}_{\mathbf{y}}(\mathbf{x}),\, \mathbf{y} \in \mathbf{Y} \right\} \\
\mathbf{a} \oplus \mathbf{t} &= \left\{ (\mathbf{y},\, \mathbf{b}(\mathbf{y})) \,:\, \mathbf{b}(\mathbf{y}) = \sum_{\mathbf{x} \in \mathbf{X}} \mathbf{a}(\mathbf{x}) * \mathbf{t}_{\mathbf{y}}(\mathbf{x}),\, \mathbf{y} \in \mathbf{Y} \right\}
\end{aligned}$$

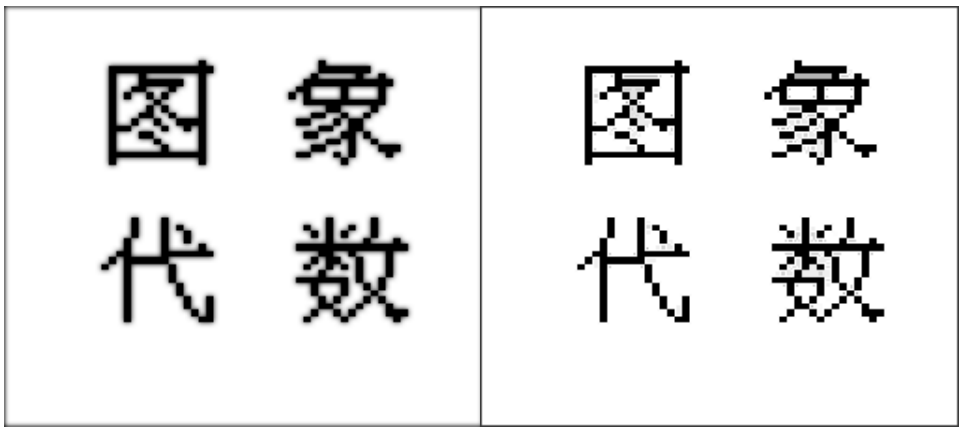
where

$$\bigvee_{\mathbf{x} \in \mathbf{X}} \mathbf{a}(\mathbf{x}) + \mathbf{t}_{\mathbf{y}}(\mathbf{x}) = \max\{\mathbf{a}(\mathbf{x}) + \mathbf{t}_{\mathbf{y}}(\mathbf{x}),\, \mathbf{x} \in \mathbf{X}\} \,.$$

The operation *additive minimum* ("$\boxtimes$") is defined in a similar fashion.

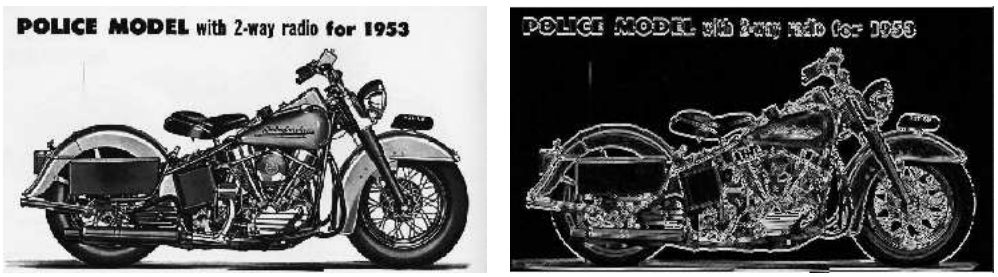
Example 2. Many image processing techniques such as the Rolling Ball Algorithm and algorithms for noise removal employ morphological image–template products of the form $\mathbf{a} \boxtriangledown \mathbf{t}$ or $\mathbf{a} \boxtriangleup \mathbf{t}$. Both forms appear in the Max–Min Sharpening Algorithm which enhances blurred images.

Figure 2. Blurred characters and result of applying max-min sharpening to blurred characters



Example 3. Figure 3 shows the result of applying the Roberts edge detector, based on generalized convolutions, to the image of a motorcycle.

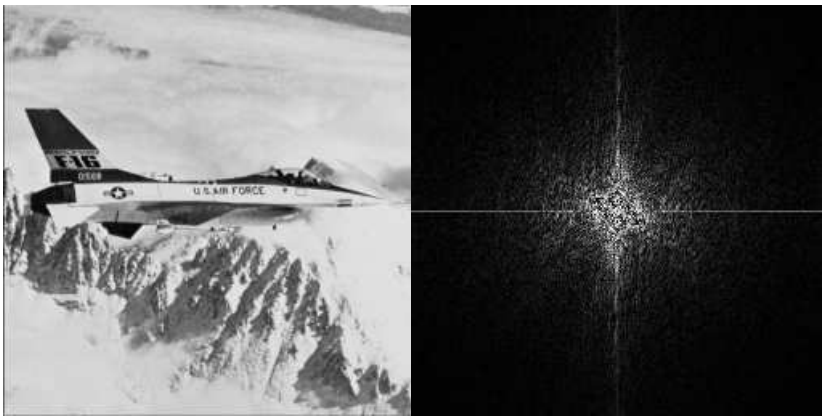
Figure 3. Motorcycle and its Roberts edge enhanced image.



Example 4. One of the most widely used transforms in image processing is the Fourier transform [3, 32]. Like many other image processing transformations the Fourier transform can be expressed in terms of a generalized convolution. This means that there is a template $\mathbf{f}$ such that the Fourier transform of an image $\mathbf{a}$ is given by $\mathbf{a} \oplus \mathbf{f}$.

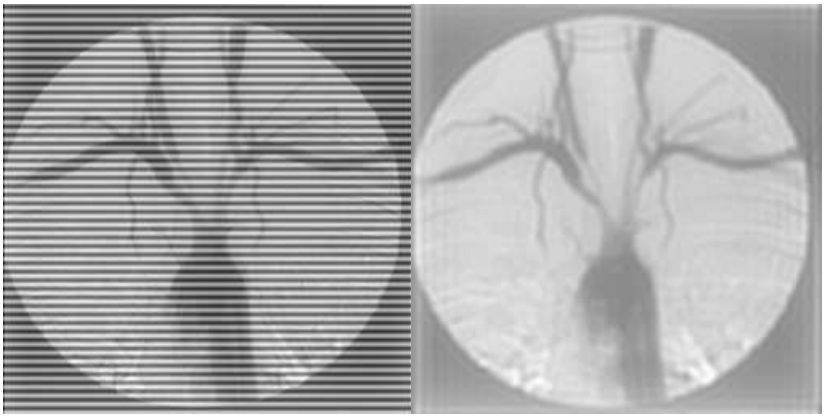
The images in Figure 4 show an image of a jet and its corresponding Fourier transform image after centering. Notice that almost all of the information is contained in the center of the latter image.

Figure 4. The image of a jet and its centered Fourier transform image.



Typical applications of the Fourier transform are low and high pass filtering [8, 20]. The figures below demonstrate how low pass filtering can be applied to medical images.

Figure 5. Noisy angiogram and filtered angiogram using a low pass filter



Operations between Templates. The basic operations between generalized templates are induced by the basic operations on images. Thus, the addition, the multiplication and the maximum of two templates are straightforward:

$$\mathbf{s}+\mathbf{t}=\{(\mathbf{y},\mathbf{r_y}):\ \mathbf{r_y}=\mathbf{s_y}+\mathbf{t_y},\ \mathbf{y}\in\mathbf{Y}\}\quad \forall\,\mathbf{s},\mathbf{t}\in\left(\mathbb{R}^{\mathbf{X}}_{-\infty}\right)^{\mathbf{Y}}$$

$$\mathbf{s}*\mathbf{t}=\{(\mathbf{y},\mathbf{r_y}):\ \mathbf{r_y}=\mathbf{s_y}*\mathbf{t_y},\ \mathbf{y}\in\mathbf{Y}\}\quad \forall\,\mathbf{s},\mathbf{t}\in\left(\mathbb{R}^{\mathbf{X}}\right)^{\mathbf{Y}}$$

$$\mathbf{s}\vee\mathbf{t}=\{(\mathbf{y},\mathbf{r_y}):\ \mathbf{r_y}=\mathbf{s_y}\vee\mathbf{t_y},\ \mathbf{y}\in\mathbf{Y}\}\quad \forall\,\mathbf{s},\mathbf{t}\in\left(\mathbb{R}^{\mathbf{X}}_{-\infty}\right)^{\mathbf{Y}}$$

Templates can not only be convolved with images, but also with other templates. Ritter [22, 26] defines the *generalized product* of a template $\mathbf{t} \in \left(\mathbb{G}^{\mathbf{X}}\right)^{\mathbf{Y}}$ and a template $\mathbf{s} \in \left(\mathbb{F}^{\mathbf{Z}}\right)^{\mathbf{X}}$, also denoted by $\mathbf{t} \textcircled{\gamma} \mathbf{s}$, as the template $\mathbf{r} \in \left(\mathbb{H}^{\mathbf{Z}}\right)^{\mathbf{Y}}$ by setting

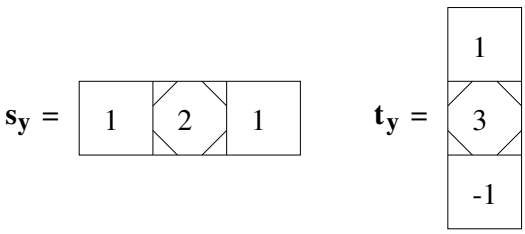
$$\mathbf{r}_{\mathbf{y}}(\mathbf{z}) = \prod_{\mathbf{x} \in \mathbf{X}} \left(\mathbf{s}_{\mathbf{x}}(\mathbf{z}) \circ \mathbf{t}_{\mathbf{y}}(\mathbf{x})\right) \quad \forall \mathbf{y} \in \mathbf{Y},\ \forall \mathbf{z} \in \mathbf{Z}\,.$$

For example, real–valued templates $\mathbf{t}$ and $\mathbf{s}$ allow us to form a *generalized convolution* ("$\bigoplus$") and an *additive maximum* ("$\boxtimes$"):

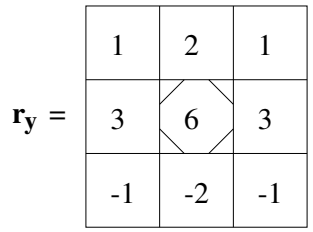
$$\begin{aligned}
\left(\mathbf{t} \bigoplus \mathbf{s}\right)_{\mathbf{y}}(\mathbf{z}) &= \sum_{\mathbf{x} \in \mathbf{X}} \left(\mathbf{s}_{\mathbf{x}}(\mathbf{z}) * \mathbf{t}_{\mathbf{y}}(\mathbf{x})\right) \quad \forall \mathbf{y} \in \mathbf{Y},\ \forall \mathbf{z} \in \mathbf{Z} \\
\left(\mathbf{t} \boxtimes \mathbf{s}\right)_{\mathbf{y}}(\mathbf{z}) &= \bigvee_{\mathbf{x} \in \mathbf{X}} \left(\mathbf{s}_{\mathbf{x}}(\mathbf{z}) + \mathbf{t}_{\mathbf{y}}(\mathbf{x})\right) \quad \forall \mathbf{y} \in \mathbf{Y},\ \forall \mathbf{z} \in \mathbf{Z}\,.
\end{aligned}$$

Note that the generalized products $\mathbf{s} \textcircled{\gamma} \mathbf{t}$ and $\mathbf{t} \textcircled{\gamma} \mathbf{s}$ of an $n \times 1$ rectangular template $\mathbf{s}$ and a $1 \times m$ rectangular template $\mathbf{t}$ always lead to $n \times m$ rectangular templates.

Example 5. Suppose $\mathbf{s}, \mathbf{t} \in \left(\mathbb{R}^{\mathbb{Z}^2}\right)^{\mathbb{Z}^2}$ are the following translation invariant templates:



The template product $\mathbf{r} = \mathbf{s} \bigoplus \mathbf{t}$ — also called the generalized convolution of $\mathbf{s}$ with $\mathbf{t}$ — has the following representation:



The Generalized Matrix Product. As before let $\mathbb{F}$, $\mathbb{G}$, $\mathbb{H}$ be value sets and $\circ : \mathbb{F} \times \mathbb{G} \to \mathbb{H}$ be a binary operation. Given integers n, m, k , an operation $\textcircled{\gamma} : \mathbb{F}_{n \times k} \times \mathbb{G}_{k \times m} \to \mathbb{H}_{n \times k}$ between matrices is defined as a special case of the *generalized matrix product* [27]:

$$\left(A \textcircled{\gamma} B\right)_{ij} = \Gamma_{l=1}^{k} \left(a_{il} \circ b_{lj}\right) \quad \forall\ i = 1, \ldots, n,\ j = 1, \ldots, m\ .$$

For example, given $A \in \mathbb{R}_{n \times k}$, $B \in \mathbb{R}_{k \times m}$ the product $A \bigoplus B \in \mathbb{R}_{n \times m}$ corresponds to the regular matrix product.

The additive maximum $A \boxtriangledown B \in \mathbb{R}_{n\times m}$ on the other hand is given by

$$(A \boxtriangledown B)_{ij} = \bigvee_{l=1}^{k} a_{il} + b_{lj} \;\; \forall\, i = 1, \ldots, n,\; j = 1, \ldots, m\ .$$

As mentioned earlier, there is a direct correspondence between matrices over a value set $\mathbb{F}$ and $\mathbb{F}$–valued images over a rectangular discrete array. Thus the higher level operations on matrices give rise to possible ways of decomposing the corresponding images. However, the higher level operations "$\textcircled{\gamma}$" are not defined as operations between images, i.e. the product $\mathbf{a}\,\textcircled{\gamma}\mathbf{b}$ between images $\mathbf{a}, \mathbf{b}$ does not exist.

Example 6.

$$\begin{pmatrix} 4 & 3 \\ 1 & 9 \\ 0 & 5 \end{pmatrix} \boxtriangledown \begin{pmatrix} 3 & 8 & 7 \\ 0 & 9 & 0 \end{pmatrix} =$$

$$\left[\begin{pmatrix} 4 \\ 1 \\ 0 \end{pmatrix} \boxtriangledown (3 \quad 8 \quad 7) \right] \vee \left[\begin{pmatrix} 3 \\ 9 \\ 5 \end{pmatrix} \boxtriangledown (0 \quad 9 \quad 0) \right] =$$

$$\begin{pmatrix} 7 & 12 & 11 \\ 9 & 18 & 9 \\ 5 & 14 & 7 \end{pmatrix}\ .$$

Decomposability of Matrices and Templates

The definitions in this section require a binary operation $\circ : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$ and an associative and commutative operation $\gamma : \mathbb{F} \to \mathbb{F}$.

Decomposability of Templates. A template $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ is called *decomposable* (with respect to the operation "$\textcircled{\gamma}$") if there exist templates $\mathbf{t}^1,\ \mathbf{t}^2 \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ such that

$$\mathbf{t} = \mathbf{t}^1\, \textcircled{\gamma}\, \mathbf{t}^2\ .$$

The template $\mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ is called *weakly decomposable* (with respect to "$\textcircled{\gamma}$") if there exist an strictly increasing sequence of natural numbers $(k_1, k_2, \ldots, k_n)$ and templates $\mathbf{t}^1, \mathbf{t}^2, \ldots, \mathbf{t}^{k_n} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ such that

$$\mathbf{t} = \left(\mathbf{t}^1\, \textcircled{\gamma} \ldots \textcircled{\gamma} \mathbf{t}^{k_1}\right) \gamma \ldots \gamma \left(\mathbf{t}^{k_{n-1}+1}\, \textcircled{\gamma} \ldots \textcircled{\gamma} \mathbf{t}^{k_n}\right)\ .$$

Such a representation of $\mathbf{t}$ is called a weak decomposition of $\mathbf{t}$.

Example 7. The image algebra expression for the one-dimensional Fourier transform of $\mathbf{a} \in \mathbb{C}^{\mathbb{Z}_n}$ is given by $\mathbf{a} \oplus \mathbf{f}$ where $\mathbf{f}_j(k) = e^{-2\pi kji/n}$. In this original form $O(n^2)$ complex additions and multiplications are needed for the computation. If n is a power of 2, this number can be reduced to $O(n\ log_2\ n)$ by incorporating the Cooley-Tukey radix–2 fast Fourier algorithm which is based on template decomposition.

The fundamental results which establish the importance of template decomposition are the following properties of image algebra products:

Some Properties of Image and Template Operations. The following associative laws hold for arbitrary $\mathbf{a} \in \mathbb{F}^{\mathbf{X}}$ and $\mathbf{s}, \mathbf{t} \in \left(\mathbb{F}^{\mathbf{X}}\right)^{\mathbf{X}}$ [22]:

$$\mathbf{a} \textcircled{\gamma} (\mathbf{s} \textcircled{\gamma} \mathbf{t}) = (\mathbf{a} \textcircled{\gamma} \mathbf{s}) \textcircled{\gamma} \mathbf{t}$$
$$\mathbf{a} \textcircled{\gamma} (\mathbf{s} \gamma \mathbf{t}) = (\mathbf{a} \textcircled{\gamma} \mathbf{s}) \gamma (\mathbf{a} \textcircled{\gamma} \mathbf{t}) \ .$$

In view of these results, users of image processing software have alternative ways of computing image–template products at their disposal.

Example 8. The two-dimensional Fourier transform template is separable, i.e. there are two one-dimensional templates $\mathbf{s}$ and $\mathbf{t}$, such that $\mathbf{f} = \mathbf{s} \bigoplus \mathbf{t}$. Hence, a two-dimensional Fourier transform can be computed by a two stage application of one-dimensional Fourier transforms. This technique reduces the number of operations per pixel by an order of magnitude.

Decomposability of Matrices. We say a matrix $A \in \mathbb{F}_{n \times m}$ is *weakly decomposable into k vector pairs*, if there are column vectors $\mathbf{b}^i \in \mathbb{F}_{\mathbf{n} \times \mathbf{1}}$ and row vectors $\mathbf{c}^i \in \mathbb{F}_{\mathbf{1} \times \mathbf{m}}$ for $i = 1, \ldots, k$ such that A has the following representation:

$$A = (\mathbf{b}^1 \textcircled{\gamma} \mathbf{c}^1) \gamma (\mathbf{b}^2 \textcircled{\gamma} \mathbf{c}^2) \gamma \ldots \gamma \left(\mathbf{b}^k \textcircled{\gamma} \mathbf{c}^k\right) \ .$$

In other words, the matrix $A \in \mathbb{F}_{n \times m}$ can be written as $A = B \textcircled{\gamma} C$ where $B \in \mathbb{F}_{n \times k}$ and $C \in \mathbb{F}_{k \times m}$. Together the matrices B and C have $k * (n + m)$ entries compared to the $n * m$ entries of the matrix A. Hence, the representation of A in terms of $B \textcircled{\gamma} C$ leads to savings of information, if k is smaller than $\frac{n*m}{n+m}$.

Remark. Matrix decompositions are useful tools in image processing for the following reasons:

1. The weak decomposition of a matrix into k vector pairs induces a weak decomposition of the corresponding rectangular template into k vertical and k horizontal strip templates. According to Paul Gader a factorization of a rectangular template into strip templates transforms a two–dimensional problem into a one-dimensional problem [5]. Moreover it greatly simplifies an eventual factorization into smaller neighborhood (e.g. 3x3) templates. So far only decomposition techniques for separable templates are known.
2. Due to the natural correspondence between matrices and images, matrix decompositions can be used for image compression. A direct implementation of this technique is hardly ever possible, since a $n \times m$ matrix usually does not factorize into k vector pairs, where k is much smaller than $\frac{n*m}{n+m}$. For most practical applications it suffices anyway if the product of a small number of vector pairs yields a very good approximation of the original matrix in terms of a certain metric. An example for this approach is the Singular Value Decomposition which provides the best approximation in the least squares sense by means of a matrix of rank k.

Metrics on Images and Invariant Templates with Finite Support. It is well known that a metric on a metric space $\mathcal{M}$ induces a metric on the space $\mathcal{M}_{n \times m}$ of all $n \times m$ matrices. Let d be the metric on $\mathbb{R}$ given by:

$$d(a, b) = |a - b| \quad \forall \ a, \ b \in \mathbb{R} \ .$$

Then the following definitions induce metrics on $\mathbb{R}_{n\times m}$:

$$d_1(A,\, B) = \sum_{i=1}^{n}\sum_{j=1}^{m} d(a_{ij},\, b_{ij})$$
$$d_2(A,\, B) = \bigvee_{i,\,j} d(a_{ij},\, b_{ij})\ .$$

Similarly if $\mathrm{s} \in \left(\mathbb{R}^{\mathbf{X}}\right)^{\mathbf{Y}}$ and $\mathrm{t} \in \left(\mathbb{R}^{\mathbf{X}}\right)^{\mathbf{Y}}$ are invariant templates with the same finite support S at some target point $\mathbf{y} \in \mathbf{Y}$, then the following equations provide metrics on the set of all invariant templates with that particular finite support:

$$d_1(\mathrm{s},\, \mathrm{t}) = \sum_{\mathbf{x}\in\mathbf{S}} d(\mathrm{t}_{\mathbf{y}}(\mathbf{x}),\, \mathrm{s}_{\mathbf{y}}(\mathbf{x}))$$
$$d_2(\mathrm{s},\, \mathrm{t}) = \bigvee_{\mathbf{x}\in\mathbf{S}} d(\mathrm{t}_{\mathbf{y}}(\mathbf{x}),\, \mathrm{s}_{\mathbf{y}}(\mathbf{x}))\ .$$

Since the templates s and t are invariant these definitions do not depend on the choice of the target point $\mathbf{y} \in \mathbf{Y}$.

3. Mixed Integer Programming (MIP) and the Decomposition of Matrices and Templates

The decomposition problems we are primarily interested in are matrix and template decompositions with respect to the operation "$\boxtimes$". Furthermore, we assume that the templates which are to be factorized are elements of $\left(R_{-\infty}^{\mathbf{X}}\right)^{\mathbf{X}}$ where $\mathbf{X}$ is a finite subset of $\mathbb{Z}^2$ (without loss of generality $\mathbf{X}$ contains $\mathrm{z} := (0,0)$. Linear mixed integer programming is suited to solve a very general form of these problems.

The General Form of the Decomposition Problem

Let $A \in \mathbb{R}_{n\times m}$. Assume that A is decomposable as follows for some natural number k:

$$A = B \boxtimes C,$$
$$B \in \mathbb{R}_{n\times k},$$
$$C \in \mathbb{R}_{k\times m}\ .$$

Suppose a set Y is defined as follows:

$$Y = \{y_1, \ldots, y_{n*k+k*m}\} = \{b_{11}, \ldots, b_{1k}, \ldots, b_{n1}, \ldots b_{nk}, c_{11}, \ldots, c_{1m}, \ldots, c_{k1}, \ldots, c_{km}\}$$

Then there are linear functions f_{ij}^{l} $(i = 1, \ldots, n;\ j = 1, \ldots, m;\ l = 1, \ldots, k)$ such that for each value of i and j the following inequalities hold:

$$\begin{aligned} f_{ij}^{1}(y_1, \ldots, y_{n*k+k*m}) &\le a_{ij} \\ f_{ij}^{2}(y_1, \ldots, y_{n*k+k*m}) &\le a_{ij} \\ &\vdots \\ f_{ij}^{k}(y_1, \ldots, y_{n*k+k*m}) &\le a_{ij}\ . \end{aligned} \tag{ij}$$

Moreover, for each i, j there exists an index $l^{ij} \in \{1, \ldots, k\}$ such that:

$$f_{ij}^{l^{ij}}(y_1, \ldots, y_{n*k+k*m}) = a_{ij} \ .$$

Obviously the linear functions f_{ij}^{l} are simply derived from the additive maximum $B \boxtimes C$:

$$f_{ij}^{l}(y_1, \ldots, y_{n*k+k*m}) = b_{il} + c_{lj} \ .$$

A similar result holds for any weak $\boxtimes$–decomposition of an invariant template $\mathbf{t} \in \left(\mathbb{R}_{-\infty}^{\mathbf{X}}\right)^{\mathbf{X}}$ with finite support into invariant templates $\mathbf{t}^{1}, \ldots, \mathbf{t}^{k_1}, \ldots, \mathbf{t}^{k_{n-1}+1}, \ldots, \mathbf{t}^{k_n} \in \left(\mathbb{R}_{-\infty}^{\mathbf{X}}\right)^{\mathbf{X}}$. The support of each template must be given in advance. Suppose the following definitions are given as well:

$$\begin{gathered} \mathbf{x} = \mathbf{x}_1 \\ \{\mathbf{x}_1, \ldots, \mathbf{x}_m\} = S(\mathbf{t}) \\ Y = \{y_1, \ldots, y_{m'}\} = \mathbf{t}^{1}(S(\mathbf{t}^{1})) \cup \ldots \cup \mathbf{t}^{k_n}\left(S\left(\mathbf{t}^{k_n}\right)\right) \ . \end{gathered}$$

For each $i = 1, \ldots, m$ there exists a natural number l_i and linear functions f_i^l for all $l = 1, \ldots, l_i$ satisfying inequalities of the following form:

$$\begin{gathered} f_i^1(y_1, \ldots, y_{m'}) \leq \mathbf{t}_{\mathbf{x}}(\mathbf{x_i}) \\ f_i^2(y_1, \ldots, y_{m'}) \leq \mathbf{t}_{\mathbf{x}}(\mathbf{x_i}) \\ \vdots \\ f_i^{l_i}(y_1, \ldots, y_{m'}) \leq \mathbf{t}_{\mathbf{x}}(\mathbf{x_i}) \ . \end{gathered} \qquad \text{[ij]}$$

Furthermore there must be at least one $k_i \in \{1, \ldots, l_i\}$ which allows us to write one of these inequalities as an equality:

$$f_i^{k_i}(\mathbf{y_1}, \ldots, \mathbf{y_{m'}}) = \mathbf{t}_{\mathbf{x}}(\mathbf{x_i}) \ .$$

The problem of finding such a decomposition could be solved as in Takriti and Gader who are particularly interested in weak decompositions of invariant templates into 3×3 rectangular templates [19]. However, we have taken a more general approach. Moreover, we are also able to find the best approximation of a template with respect to a certain norm.

Decomposition and Approximation

In the previous section we introduced the reader to the concepts of matrix and template decompositions with respect to "$\boxtimes$" (we also call them morphological decompositions). Now it is time to give exact definitions of the problems we are concerned with:

i. *Morphological matrix decomposition problem:* Given a matrix $A \in \mathbb{R}_{n\times m}$ and $k \in \mathbb{N}$, determine if A is weakly decomposable into k vector pairs. If possible, find $B \in \mathbb{R}_{n\times k}$ and $C \in \mathbb{R}_{k\times m}$ such that

$$B \boxtimes C = A \ .$$

ii. *Morphological template decomposition problem:* Let $\mathbf{t} \in \left(\mathbb{R}_{-\infty}^{\mathbf{X}}\right)^{\mathbf{X}}$ be an invariant template, $(k_1, k_2, \ldots, k_n)$ an strictly increasing sequence of natural numbers and $\left(\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^{k_n}\right)$ a sequence of subsets of $\mathbf{X}$. Find out if there exist invariant templates $\mathbf{t}^1, \mathbf{t}^2, \ldots, \mathbf{t}^{k_n} \in \left(\mathbb{R}_{-\infty}^{\mathbf{X}}\right)^{\mathbf{X}}$ such that:

a. For all $i = 1, \ldots, k_n$ the set $\mathbf{X}^i$ contains the support of $\mathbf{t}_{\mathbf{z}}^i$.
b. The template $\mathbf{t}$ factorizes as follows into a product of the templates $\mathbf{t}^1, \mathbf{t}^2, \ldots, \mathbf{t}^{k_n}$:

$$\mathbf{t} = \left(\mathbf{t}^1 \boxtimes \ldots \boxtimes \mathbf{t}^{k_1}\right) \vee \ldots \vee \left(\mathbf{t}^{k_{n-1}+1} \boxtimes \ldots \boxtimes \mathbf{t}^{k_n}\right) .$$

Determine the templates $\mathbf{t}^1, \mathbf{t}^2, \ldots, \mathbf{t}^{k_n}$ in this case.

These decomposition problems are closely related to the following approximation problems in terms of the metric d_1 as it is defined in chapter 2.

I. *Morphological matrix approximation problem:* Given a matrix $A \in \mathbb{R}_{n \times m}$ and $k \in \mathbb{N}$, find $B \in \mathbb{R}_{n \times k}$ and $C \in \mathbb{R}_{k \times m}$ such that

$$d_1(B \boxtimes C, A) \leq d_1(B' \boxtimes C', A) \quad \forall\, B' \in \mathbb{R}_{n \times k},\, C' \in \mathbb{R}_{k \times m} .$$

II. *Morphological template approximation problem:* Suppose $\mathbf{t} \in \left(\mathbb{R}_{-\infty}^{\mathbf{X}}\right)^{\mathbf{X}}$ is an invariant template, $(k_1, k_2, \ldots, k_n)$ an strictly increasing sequence of natural numbers and $\left(\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^{k_n}\right)$ a sequence of subsets of $\mathbf{X}$. Define $\mathbf{T}$ to be the set of all sequences $\left(\mathbf{s}^1, \mathbf{s}^2, \ldots, \mathbf{s}^{k_n}\right)$ which have the following properties:

$$\begin{gathered} \mathbf{s}^i \in \left(\mathbb{R}_{-\infty}^{\mathbf{X}}\right)^{\mathbf{X}} \\ S\left(\mathbf{s}_{\mathbf{z}}^i\right) \subseteq \mathbf{X}^i \\ \forall\, i = 1, \ldots, k_n . \end{gathered}$$

Find a sequence of templates $\left(\mathbf{t}^1, \mathbf{t}^2, \ldots, \mathbf{t}^{k_n}\right) \in \mathbf{T}$ which satisfies the inequality

$$\begin{gathered} d_1\Big(\left[\mathbf{t}^1 \boxtimes \ldots \boxtimes \mathbf{t}^{k_1}\right] \vee \ldots \vee \left[\mathbf{t}^{k_{n-1}+1} \boxtimes \ldots \boxtimes \mathbf{t}^{k_n}\right], \mathbf{t}\Big) \\ \leq d_1\Big(\left[\mathbf{s}^1 \boxtimes \ldots \boxtimes \mathbf{s}^{k_1}\right] \vee \ldots \vee \left[\mathbf{s}^{k_{n-1}+1} \boxtimes \ldots \boxtimes \mathbf{s}^{k_n}\right], \mathbf{t}\Big) \\ \forall \left(\mathbf{s}^1, \ldots, \mathbf{s}^{k_n}\right) \in \mathbf{T} \end{gathered}$$

Since we have the equivalences $d_1(A, D) = 0 \Leftrightarrow A = D$ and $d_1(\mathbf{t}, \mathbf{r}) = 0 \Leftrightarrow \mathbf{t} = \mathbf{r}$ for all matrices $D \in \mathbb{R}_{n \times m}$ and all invariant templates $\mathbf{r} \in \left(\mathbb{R}_{-\infty}^{\mathbf{X}}\right)^{\mathbf{X}}$, the solutions of the approximation problems lead to solutions of the decomposition problems as well. The existence part of problem i and ii can be answered depending on the distance of the best approximation to the original matrix, and the original template, respectively. If this distance in terms of the metric d_1 equals 0, then the approximation problems and the decomposition problems have the same solutions.

The Mixed Integer Programming Solution of the Approximation Problems

Consider the following slight variation of the matrix approximation problem I described in the last section.

I'. Suppose that $A \in \mathbb{R}_{n\times m}$ and $k \in \mathbb{N}$ are given. Define $\mathcal{D}'$ as the set of all matrices $D' = B' \boxtimes C'$ such that $B' \in \mathbb{R}_{n\times k}$, $C' \in \mathbb{R}_{k\times m}$ and $D' \leq A$. Find matrices $B \in \mathbb{R}_{n\times k}$ and $C \in \mathbb{R}_{k\times m}$ which satisfy:

$$B \boxtimes C \leq A$$
$$d_1(A, B \boxtimes C) \leq d_1(A, D') \ \ \forall\ D' \in \mathcal{D}' .$$

Since we are restricting ourselves to the set $\mathcal{D}'$, we will not find the best approximation overall of the matrix $A \in \mathbb{R}_{n\times m}$ in terms of an additive maximum $B \boxtimes C$. However, we are still able to find out if A can be decomposed exactly as $B \boxtimes C$ for some $B \in \mathbb{R}_{n\times k}$ and $C \in \mathbb{R}_{k\times m}$. In this case, problem I' yields the same solution as the problems I and i.

For all $\forall\ D' \in \mathcal{D}'$ the distance $d_1(A, D')$ amounts to nothing else than $\sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij} - d'_{ij}$. because $D' \leq A$. Therefore, it suffices to find matrices $B \in \mathbb{R}_{n\times k}$ and $C \in \mathbb{R}_{k\times m}$ such that $D = B \boxtimes C \leq A$ and $\sum_{i=1}^{n}\sum_{j=1}^{m} d_{ij}$ is maximal.

Now let us return to the system of inequalities (ij) given in the first section of this chapter. Notice that for each group of inequalities depending on i, j only one equality must hold (Takriti and Gader call this fact "the group equality restriction" [19]). We introduce variables r_{ij}^{l} indicating which equality constraints in (ij) are fulfilled and deduce the following formulation:

$$Max \sum_{l=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij} * (b_{il} + c_{lj})$$
$$subject\ to$$
$$r_{ij}^{l} = 0 \ \ or\ 1$$
$$\sum_{l=1}^{k} r_{ij}^{l} = 1$$
$$b_{il} + c_{lj} \leq a_{ij}$$
$$\forall\ \ i = 1, \ldots, n,\ j = 1, \ldots, m,\ l = 1, \ldots, k\ .$$

Equivalently, this maximization problem (I') can be formulated as the linear mixed 0-1 integer programming problem given below.

$$
\begin{gathered}
Max \sum_{l=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij}^{l} \\
subject\ to \\
-|a_{ij}| * r_{ij}^{l} \le p_{ij}^{l} \le |a_{ij}| * r_{ij}^{l} \\
r_{ij}^{l} = 0 \ \ or\ 1 \\
\sum_{l=1}^{k} r_{ij}^{l} = 1 \\
p_{ij}^{l} \le b_{il} + c_{lj} \\
b_{il} + c_{lj} \le a_{ij} \\
\forall\ i = 1,\ldots,n,\ j = 1,\ldots,m,\ l = 1,\ldots,k\ .
\end{gathered}
$$

Let us return to problem I. The best approximation with respect to the metric d of a given matrix $A \in \mathbb{R}_{n\times m}$ in terms of a matrix product $B \boxtimes C$, where $B \in \mathbb{R}_{n\times k},\ C \in \mathbb{R}_{k\times m}$, is requested. Matrices B^* and C^*, which solve this problem, can be found by solving the following minimization problem. The optimal objective value will equal $d(A, B^* \boxtimes C^*) = \sum_{i=1}^{n}\sum_{j=1}^{m}\left|a_{ij} - \bigvee_{l=1}^{k}(b_{il} + c_{lj})\right|$.

$$
\begin{gathered}
Min \sum_{i=1}^{n}\sum_{j=1}^{m}\left(q_{ij} + \sum_{l=1}^{k} r_{ij}^{l} * p_{ij}^{l}\right) \\
subject\ to \\
p_{ij}^{l} \ge a_{ij} - b_{il} - c_{lj} \\
q_{ij} \ge b_{il} + c_{lj} - a_{ij} \\
p_{ij}^{l} \ge 0 \\
q_{ij} \ge 0 \\
r_{ij}^{l} = 0 \ \ or\ 1 \\
\sum_{l=1}^{k} r_{ij}^{l} = 1 \\
\forall\ i = 1,\ldots,n,\ j = 1,\ldots,m,\ l = 1,\ldots,k\ .
\end{gathered}
$$

We would like to point out some helpful observations to the reader:

- Notice that each variable p_{ij}^{l} is greater than or equal to $(a_{ij} - b_{il} - c_{lj})^{+}$, the positive part of $a_{ij} - b_{il} + c_{lj}$.
- The variables r_{ij}^{l} select an index $l \in \{1,\ldots,k\}$ such that $(a_{ij} - b_{il} - c_{lj})^{+}$ is minimal, or equivalently $b_{il} + c_{lj}$ is maximal whenever $b_{il} + c_{lj} < a_{ij}$. Consequently, for each i, j the optimal solution satisfies
$$\sum_{l=1}^{k} r_{ij}^{l} * p_{ij}^{l} = \left(a_{ij} - \bigvee_{l=1}^{k}(b_{il} + c_{lj})\right)^{+}\ .$$
- The variables q_{ij} exceed the values 0 and $b_{il} + c_{lj} - a_{ij}$ for all $l = 1,\ldots,k$. Hence, at optimality
$$q_{ij} = \bigvee_{l=1}^{k}(b_{il} + c_{lj} - a_{ij})^{+} = \left(a_{ij} - \bigvee_{l=1}^{k}(b_{il} + c_{lj})\right)^{-}\ .$$

A linear 0–1 MIP formulation can be given as follows:

$$Min \sum_{i=1}^{n}\sum_{j=1}^{m}\left(q_{ij}+\sum_{l=1}^{k}p_{ij}^{l}\right)$$
$$subject\ to$$
$$p_{ij}^{l} \geq a_{ij}-b_{il}-c_{lj}-M*s_{ij}^{l}$$
$$q_{ij} \geq b_{il}+c_{lj}-a_{ij}$$
$$p_{ij}^{l} \geq 0$$
$$q_{ij} \geq 0$$
$$s_{ij}^{l} = 0 \ \ or\ 1$$
$$\sum_{l=1}^{k} s_{ij}^{l} = k-1$$
$$\forall\ i=1,\ldots,n,\ j=1,\ldots,m,\ l=1,\ldots,k\ .$$

The variables s_{ij}^{l} take on the role of the variables r_{ij}^{l} from the quadratic mixed 0-1 approach. The number M should be at least as large as $\bigvee\limits_{i,j,l}|a_{ij}-(b_{il}+c_{lj})|$.

Similar 0–1 MIP formulations exist for the corresponding template approximation problem II. These formulations are based on systems of inequalities of the form [ij]. Mixed 0–1 Integer programming can also be applied when the maximum metric d_2 instead of d_1 is used.

Example 9. Let $k=2$ and $A \in \mathbb{R}_{3\times 3}$ be the following matrix:

$$A=\begin{pmatrix}7 & 12 & 11\\ 9 & 18 & 9\\ 5 & 14 & 7\end{pmatrix}$$

By Example 3 the matrix A can be written as $A = B \boxtimes C$, where $B \in \mathbb{R}_{3\times 2}$ and $C \in \mathbb{R}_{2\times 3}$. Therefore, the 0–1 MIP program below will yield the optimal objective value $Z^{*}=\sum\limits_{i=1}^{3}\sum\limits_{j=1}^{3}a_{ij}=92$:

$$Max\ Z=\sum_{l=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m}p_{ij}^{l}$$
$$subject\ to$$
$$-18*r_{ij}^{l} \leq p_{ij}^{l} \leq 18*r_{ij}^{l}$$
$$r_{ij}^{l} = 0 \ \ or\ 1$$
$$\sum_{l=1}^{k} r_{ij}^{l} = 1$$
$$p_{ij}^{l} \leq b_{il}+c_{lj}$$
$$b_{il}+c_{lj} \leq a_{ij}$$
$$\forall\ i=1,2,3;\ j=1,2,3;\ l=1,2\ .$$

In most cases we only have to deal with images which only assume non-negative values, i.e. non-negative matrices. Therefore, we can replace $\sum\limits_{l=1}^{k} r_{ij}^{l}=1$ by $\sum\limits_{l=1}^{k} r_{ij}^{l}\leq 1$ which will replace negative entries in$B \boxtimes C$ by zeros and will thus improve the approximation of A.

Computational Aspects

The sizes of our mixed 0–1 linear programming problems increase with the size of the matrices and templates which are to be decomposed. For example, the morphological matrix approximation I' for an $n \times m$ matrix and k vector pairs has $k(2nm + n + m)$ variables and $(3k + 1)nm$ constraints (cf. example 9).

We tested our linear 0–1 MIP formulations on some matrices of sizes 8×8 and 16×16 by using the commercially available optimization package LINDO [28]. In spite of very limited CPU–times, we were always able to obtain good approximations of the given matrix (i.e. an objective function value within $10\,\%$ of the exact solution). A detailed description of two specific examples is included at the end of our report.

In the course of the last few decades, many efficient 0–1 mixed linear algorithms have been developed and implemented to solve larger problems [10, 7, 18]. More recently, interior point approaches have been successfully used to solve very large 0–1 mixed integer problems optimally or approximately. The success of these approaches is due to very efficient implementations of interior point methods for solving very large scale linear programs [13]. We concentrate our current efforts on solving computer vision related mixed 0–1 integer problems with interior point methods. Extensive computational results with interior point methods will be presented in a forthcoming article.

Some Preliminary Results: Decomposition of a Matrix into 3 Vector Pairs with LINDO.

Let the following matrix $A \in \mathbb{R}_{8\times 8}$ be given. This matrix is known to be weakly $\boxtimes$ –decomposable into 3 vector pairs.

$$A = \begin{pmatrix} 19 & 15 & 17 & 9 & 20 & 15 & 14 & 15 \\ 19 & 21 & 24 & 13 & 23 & 18 & 21 & 22 \\ 18 & 14 & 18 & 7 & 18 & 13 & 15 & 16 \\ 21 & 17 & 21 & 13 & 24 & 19 & 18 & 19 \\ 22 & 18 & 22 & 20 & 31 & 26 & 22 & 21 \\ 25 & 21 & 21 & 9 & 15 & 16 & 17 & 10 \\ 24 & 20 & 20 & 8 & 17 & 15 & 16 & 12 \\ 28 & 24 & 24 & 16 & 27 & 22 & 20 & 17 \end{pmatrix}$$

The corresponding 0-1 MIP formulation for problem I' is characterized by the following data (cf. Example 9):

- *number of 0–1 variables:* $3 * 64 = 192$
- *total number of variables:* $3 * (2 * 64 + 2 * 8) = 432$
- *number of constraints:* $(3 * 3 + 1) * 64 = 640$
- *exact solution*: 1182
- *approximate integer solution (5 min CPU time on VAX8600)*: 1073
- *number of pivots*: 2451

Finally we present some data and results of an application of the MIP algorithm to a matrix $B \in \mathbb{R}_{16\times 16}$. The matrix B also factors into a $\boxtimes$ –product of 3 vector pairs.

- *number of 0–1 variables:* $3 * 16^2 = 768$
- *total number of variables:* $3 * \left(2 * 16^2 + 2 * 16\right) = 1632$
- *number of constraints:* $(3 * 3 + 1) * 16^2 = 2560$
- *exact solution*: 4256
- *relaxed linear programming solution (15 min CPU time on VAX8600)*: 4630
- *number of pivots*: 2116

Acknowledgements. The authors thank Dr. P. Coffield, Eglin Air Force Base, for his continued support of this research.

Bibliography

[1] H.C. Andrews and B.R. Hunt. *Digital Image Restoration*. Prentice Hall, Reading, MA, 1977.

[2] D.H. Ballard and C.M. Brown. *Computer Vision*. Prentice-Hall, Englewood Cliffs, NJ, 1982.

[3] J.W. Cooley, P.A. Lewis, and P.D. Welch. The fast Fourier transform and its applications. *IEEE Trans. Educ.*, E-12(1):27–34, 1969.

[4] R. Cuninghame-Green. *Minimax Algebra: Lecture Notes in Economics and Mathematical Systems 166*. Springer-Verlag, New York, 1979.

[5] P.D. Gader. Separable decompositions and approximations for gray-scale morphological templates. *CGVIP*, 53:288–296, July 1991.

[6] P.D. Gader and E.G. Dunn. Image algebra and morphological template decomposition. In *Aerospace Pattern Recognition*, volume 1098 of *Proceedings of SPIE*, pages 134–145. Orlando, FL, March 1989.

[7] A.M. Geoffrion and R.E. Marsten. Integer programming algorithms: A framework and a state of the art survey. *Management Science*, 18:465–491, 1972.

[8] R.C. Gonzalez and P. Wintz. *Digital Image Processing*. Addison-Wesley, Reading, MA, second edition, 1987.

[9] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Addison-Wesley, Reading, MA, 1992.

[10] E.L. Johnson H. Crowder and M.W. Padberg. Solving large-scale zero-one linear programming problems. *Operations Research*, 31:803–834, 1982.

[11] R.M. Haralick and L.G. Shapiro. *Computer and Robot Vision*, volume I. Addison-Wesley, New York, 1992.

[12] H. J. A. M. Heijmans. Mathematical morphology: A modern approach in image processing based on algebra and geometry. *SIAM Review*, 37(1):1–36, 1995.

[13] N. Karmarkar and K. Ramakrishnan. Computational results of an interior point algorithm for large scale linear programming. *Mathematical Programming*, 52:555–585, 1991.

[14] D. Li and G. X. Ritter. Decomposition of separable and symmetric convex templates. In *Image Algebra and Morphological Image Processing*, volume 1350 of *Proceedings of SPIE*, pages 408–418, San Diego, CA, July 1990.

[15] Z.Z. Manseur and D.C. Wilson. Decomposition methods for convolution operators. *Computer Vision, Graphics, and Image Processing*, 53(5):428–434, 1991.

[16] G. Matheron. *Elements pour une Theorie des Milieux Poreux*. Masson, Paris, 1967.

[17] G. Matheron. *Random Sets and Integral Geometry*. Wiley, New York, 1975.

[18] P.M. Pardalos and Y. Li. Integer programming. In C.R. Rao, editor, *Handbook of Statistics*, volume 9, pages 279–302. Elsevier, 1993.

[19] S. Takriti P.D. Gader. Local decomposition of gray-scale morphological templates. *Journal of Mathematical Imaging and Vision*, 2:39–50, 1992.

[20] G.X. Ritter. Image algebra with applications. Unpublished manuscript.

[21] G.X. Ritter. Image algebra. Technical Report CCVR-92–1, University of Florida Center for Computer Vision and Visualization, Gainesville, FL, 1992.

[22] G.X. Ritter. *Image Algebra*. unpublished manuscript, available via anonymous ftp from `ftp.cis.ufl.edu` in `/pub/ia/documents`, 1994.

[23] G.X. Ritter and P. Sussner. The minimax eigenvector transform. In *Image Algebra and Morphological Image Processing III*, Proceedings of SPIE, San Diego, CA, July 1992.

[24] G.X. Ritter and J.N. Wilson. Image algebra: A unified approach to image processing. In *Medical Imaging*, volume 767 of *Proceedings of SPIE*, Newport Beach, CA, February 1987.

[25] G.X. Ritter, J.N. Wilson, and J.L. Davidson. AFATL standard image algebra. Technical Report TR 87–04, University of Florida CIS Department, October 1987.

[26] G.X. Ritter, J.N. Wilson, and J.L. Davidson. Image algebra: An overview. *Computer Vision, Graphics, and Image Processing*, 49(3):297–331, March 1990.

[27] G.X. Ritter and H. Zhu. The generalized matrix product and its applications. *Journal of Mathematical Imaging and Vision*, 1(3):201–213, 1992.

[28] L. Schrage. *Linear, Integer and Quadratic Programming with LINDO*. The Scientific Press, Palo Alto, CA, 1983.

[29] J. Serra. Introduction a la morphologie mathematique. Booklet no. 3, cahiers du centre de morphologie mathematique, Fontainebleau, France, 1969.

[30] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, London, 1982.

[31] J. Serra. *Image Analysis and Mathematical Morphology, Volume 2: Theoretical Advances*. Academic Press, New York, 1988.

[32] University of Florida Center for Computer Vision and Visualization, Gainesville. *Handbook of Image Algebra*, 1993.