Dithering in a Simplex

Zachi Baharav, Doron Shaked
HP Laboratories Israel*
HPL-98-47(R.1)
November, 1999

E-mail:[zachi,dorons]@hp.technion.ac.il

Color
Halftoning,
Dithering.

This report introduces the problem of dithering in a simplex, and suggests a few possible solutions. In general, dithering can be viewed as a method for describing color images using a combination of allowed output colors. Usually the convex hull of the set of allowed output colors is a cube in the color space. However, there are cases when the convex hull is a simplex, which leads to the problem at hand. All the simplex dithering methods suggested in this report are based on a geometrical interpretation of the dithering process, which is described and used throughout this report. Different features of the various dithering methods are discussed, and a few examples are given.

* HP Labs Israel, Technion City, Haifa 32000, Israel.

1 Introduction

Halftoning processes are widely used today for printing images, due to the fact that the number of printable colors in a color-printer is much smaller than the number of colors in the original image. There are various methods for halftoning, like error-diffusion, dithering, and others [1, 2]. In this report we discuss only dithering, which is simple both in terms of computational complexity and memory requirements.

In common color printers, dither processes are used to convert an input color image into an output image consisting of the eight printable-colors, White, Cyan, Magenta, Yellow, Red, Green, Blue and blacK. These colors form a cube in the RGB color space. However, there are cases where for a given input color only *some* of the RGB-cube vertices are printable, due to some constraints [4]. As will be seen, this case can not be treated as a simple extension to the regular case, and a whole different mechanism is called for. This new mechanism, called *dithering in a simplex*, is described in this report. Another case where dithering in a simplex is called for is the introduction of an additional printable color to the original eight.

The organization of the report is as follows. Sec. 2 sets the definitions and notations for the sequel, and the motivation for this report. Sec. 3 describes the conventional process of dithering in a cube and its geometrical interpretation. Sec. 4 describes various properties of dither-processes in general. These properties serve to compare and evaluate the proposed simplex dithering methods. Sec. 5 describes the barycentric-coordinates, which are a basic concept in the case of dithering in a simplex. Sections 6, 7, and 8, describe three methods for dithering in a simplex (respectively: barycentric, Cartesian, and helical dithering in a simplex). In each of these sections, a geometric interpretation is followed by an algebraic description, and a discussion of the method. A few examples are given in Sec. 9, and summary and conclusions are finally given in Sec. 10.

2 Terms, Formulation and Motivation

2.1 Terms

- **Halftoning** is a process which takes an input color-image and a list of 'printable-colors', and produces an output-image where each pixel is one of the printable-colors, so that the input and output images look similar to a human viewer. For example, in the common case of color printing, the input image is a full-color image (24 bits-per-pixel representing RGB values), and the printable-colors are the 8 vertices of the RGB color-cube (White, Cyan, Magenta, Yellow, Red, Green, Blue, and blacK).

- **Dithering** is a point-wise halftoning process. The output color in a certain location, depends *solely* on the value of the input-pixel at that location, and the location coordinates.

- **Cube** is the shape of the unit ball when using the L^∞ norm. We deal with cubes in one, two and three dimensions. For ease of explanation, we use only the positive-coordinates part of the cube, $[0,1]^N$ (where N is the appropriate dimension).

- **Simplex** is the convex-hull of $(N+1)$ non-coplanar points in an N dimensional space. Thus, a 2D simplex is a triangle, and a 3D simplex is a tetrahedron. Fig. 1 describes simplices and cubes in 1, 2, and 3 dimensions.

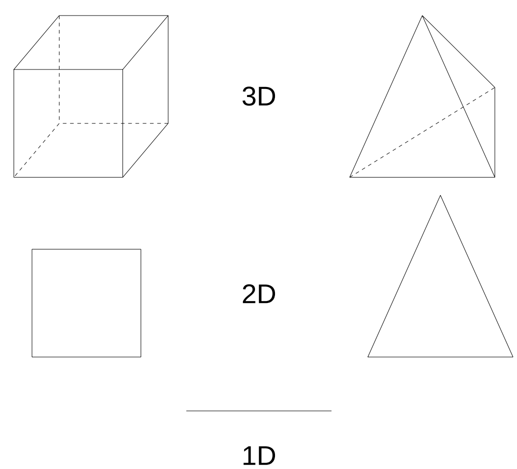

Figure 1: Cubes and simplices in 1, 2, and 3 dimensions.

- **Patch** is an image of constant color.

Most of the following explanations are made for a two-dimensional color-space. This is just in order to keep the description more lucid, and the figures more understandable. However, at the end of each explanation, a reference is made addressing the implementation in a (real-world) 3D color-space. The 2D color-space we use is described in Fig. 2(a), and the real-world 3D color-space is described in Fig. 2(b). In the 2D space each color is described by two-coordinates: Magenta and Cyan. The vertices of the cube are White, Magenta, Cyan, and Blue.

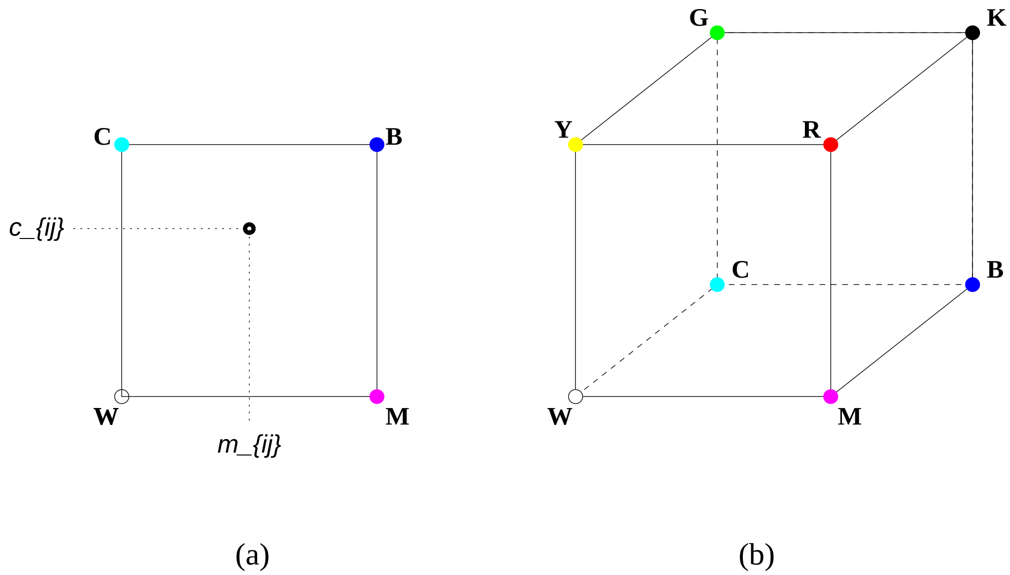

Figure 2: (a) Two dimensional color-space. Each color-point in this space is described by its two coordinates, Magenta and Cyan. (b) Three-dimensional color-space. Each color-point in this space is described by a triplet, Cyan, Magenta and Yellow.

2.2 Formulation

The 2D dithering-in-a-simplex problem is described in Fig. 3. The pixel at location (i, j) in the input-image is denoted by P_{ij}. The output pixel at the (i, j) location is denoted by O_{ij}. The printable colors are the 3 vertices of the simplex, which are denoted by $\{$**A**, **B**, **C**$\}$. For presentation purposes we usually use Red, Green and Blue, to denote **A**, **B**, and **C**, respectively.

The general form of the problem can be phrased as: *Given an input pixel (color and location), which simplex vertex should be printed so that the resulting image approximates the input image for a human viewer.*

Three dimensional color-space

In 3D the 2D triangle extends to a 3D tetrahedron with 4 printable colors. The rest of the formulation is the same.

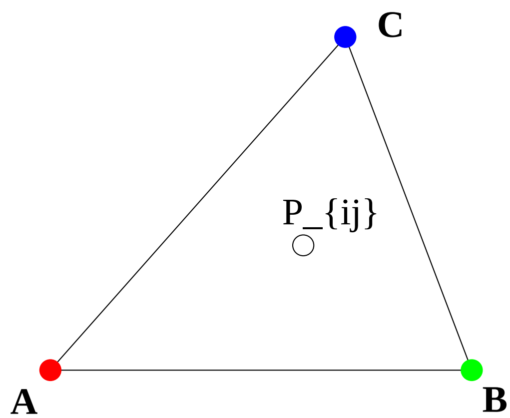

Figure 3: Dithering in a simplex. The input pixel is P_{ij}, and the printable colors are denoted by {**A**,**B**,**C**}.

2.3 Motivation

Two cases were the initial motivation for this research. The first case is the incorporation of various constraints into the dithering process. Traditionally, the printable colors in a printer are combinations of a few basic printable colors. For example, suppose the basic printable colors are Magenta and Cyan over White paper, as seen in Fig. 4(a). Then, each output-pixel can be either Cyan, Magenta, both or none, which brings us to 4 allowed output colors. This leads to classic dithering methods or dithering in a cube, where the input pixel is a point in the cube, and the output should be one of its vertices.

Now, assume that certain colors in the 2D color-space of Fig. 2(a), namely those located in the triangle formed by White-Magenta-Cyan, should be dithered using only White, Magenta or Cyan as output colors (no use of Blue halftones for bright colors). This leads to a problem of dithering in a simplex, where the input color can be halftoned using only vertices of an enclosing simplex, as demonstrated in Fig. 4(a).

The second case which served as a motivation is that of multistate printers, namely printers that can produce general sets of colors, as in Fig. 4(b), where one additional printable color, D, is added. In this case the situation is that of dithering in a simplex. This is because, for example, every color enclosed in the triangle White-Cyan-**D**, can be described using these three colors.

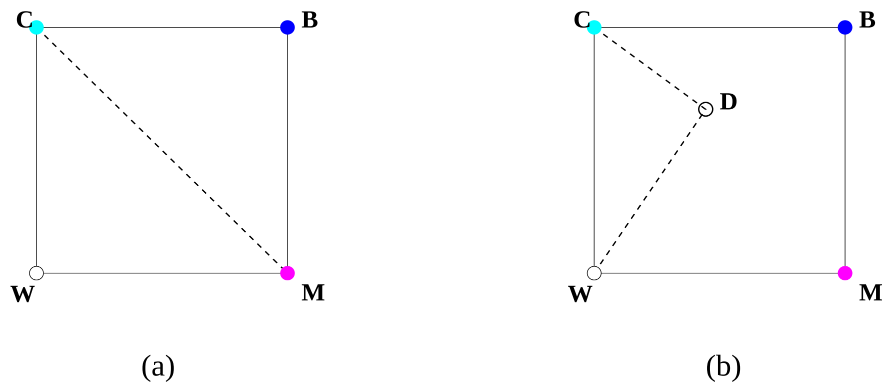

Figure 4: Motivation for investigating dithering in a simplex: (a) Constrained dithering process. (b) Multistate printer, with additional printable color **D**.

3 Cartesian Dithering in a Cube

Before dwelling into the new methods for dithering in a simplex, we give a short review of the conventional case of dithering in a cube.

Let $P_{ij} = (m_{ij}, c_{ij})$, and $O_{ij} \in$ { White, Cyan, Magenta, or Blue }, as in Fig. 2(a). The dithering process is affected by the use of two dither-matrices, Magenta and Cyan. The elements of these matrices are denoted by M_{ij}, C_{ij}, respectively, and are called *thresholds*. Each pixel in the input-image is compared with the corresponding values in the dither-matrices, and the output is determined accordingly. This dithering method is termed *Cartesian Dithering in a Cube*.

Alg. I : Cartesian Dithering in a Cube (Algebraic formulation)

O_{ij}=`White`
  `if` $(m_{ij} >= M_{ij})$ `then :` O_{ij}`+= Magenta`
  `if` $(c_{ij} >= C_{ij})$ `then :` O_{ij}`+= Cyan`

In the above it is assumed that the dither-matrix is of the same size as the input-image. Otherwise, there is a need to duplicate the dither-matrix to cover the whole image.

Note that putting Magenta and Cyan dots, one on top of the other at the same location, amounts to putting a Blue dot. Phrasing it in vector form:

$$\begin{array}{ccccc} \text{Blue} & = & \text{Magenta} & + & \text{Cyan} \\ [\,1{,}1\,] & = & [\,1{,}0\,] & + & [\,0{,}1\,] \end{array}$$

Three dimensional color-space

In the 3D case, there are 8 vertices to the cube, which are the 8 printable-colors { White, Cyan, Magenta, Yellow, Red, Green, Blue, blacK }. There are accordingly 3 dither-matrices, and otherwise the analogy is immediate.

3.1 Geometrical interpretation

Previously we have introduced the two dither-matrices, Magenta and Cyan, whose elements are called thresholds. Now, let us combine these two matrices to produce one vector matrix, called the threshold array **T**, with elements $T_{ij} = (M_{ij}, C_{ij})$. Thus, T_{ij} can be described as a point in the 2D color-space, and **T** is a collection of points in the color-space. It is important to note that just keeping the location of these points in color-space is only part of the story: One has to keep their placement in the array in order to fully describe it. The thresholds location in the color-space determines the colors which are used to halftone any given patch. On the other hand, the thresholds placement in the threshold array determines the halftone pattern, which is important for the human viewer.

Since both thresholds and input-colors can be identified as points in color-space, we describe the whole process of dithering (thresholding) as a geometrical operation in the color-space. This is shown in Fig. 5.

Given the location of an input color $P_{ij} = (m_{ij}, c_{ij})$ in color space as described in Fig. 5, a partition is induced by passing through it lines parallel to the two axes. Assume that the threshold T_{ij} is located in the lower left part of the partition in Fig. 5. In this case, it is true that

$$(m_{ij} >= M_{ij}) \text{ and } (c_{ij} >= C_{ij}) \quad ,$$

and therefore the threshold should be colored Blue. This is true no-matter where specifically it is located in the lower left part. In a similar manner, if the threshold is located in other

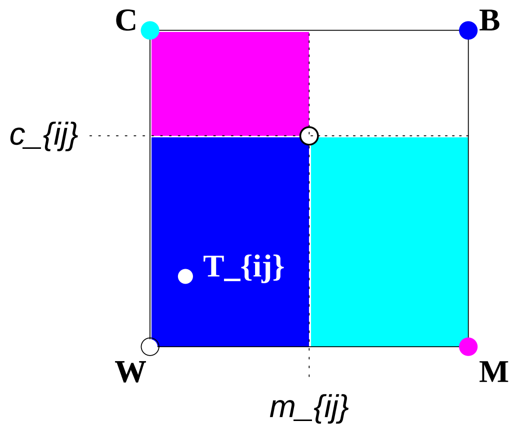

Figure 5: Partition into regions, implied by the input color $P_{ij} = (m_{ij}, c_{ij})$.

parts of the partition, it is always colored with the printable-color opposing it in the cube. In Fig. 5 the different areas are designated with the appropriate colors. Thus, the dithering-process of an input-pixel P_{ij} can be described geometrically:

Alg. II : Cartesian Dithering in a Cube (Geometric formulation)

1. `Locate the input-pixel` P_{ij} `in the color-space.`
2. `Partition the color-space into` 4 `quadrants, by passing through` P_{ij} `hyperplanes parallel to the axes.`
3. `Locate the corresponding threshold value` T_{ij} `in the partition, and color it with the diagonally opposing vertex color.`

Up till now we have described one case, in which an input-color is given, and the output color is determined depending on the threshold location. Now we introduce the dual question:

Assume a given threshold value. Which input-colors render it to a specific halftone-color ?

The answer to this question is shown geometrically in Fig. 6. Each threshold imposes a partition of the color-space. The output color is determined according to the location of the input-color in this partition. Thus, if the input color is in the upper-right square, it renders the threshold Blue. A similar explanation holds for the other parts of the partition. Note the dual relation between thresholds and colors by comparing Fig. 5 to Fig. 6.

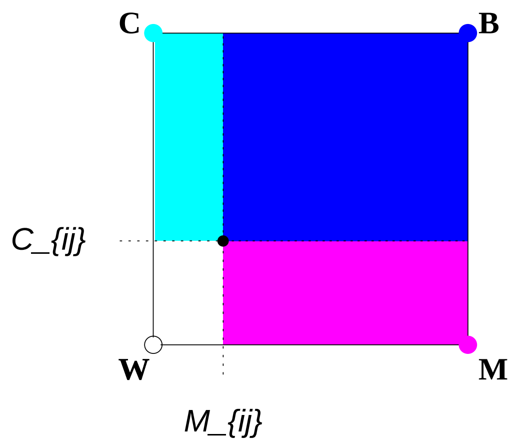

Figure 6: Partition into regions, implied by the threshold $T_{ij} = (M_{ij}, C_{ij})$.

Three dimensional color-space

Extension to 3D is immediate, however its graphical illustration is a little more complicated. Each element of the array $\mathbf{T}$ is a three-component vector, describing a point in a 3D color-space. Each input color imposes a partition of the color-space into 8 octants, by passing through it three planes parallel to the faces of the cube. Again, the output color is determined according to the location of the threshold in this octants-partition (the diagonally opposing color). In the dual problem the partition of the color space is again to octants. This time the colors correspond to the vertex color as in Fig. 6.

4 Various aspects of dithering process

This section reviews a few aspects that are used in the sequel for describing features of the various methods. Traditionally some of these aspects go unnoticed, as they appear implicitly in dithering methods.

4.1 Mean Color Invariance (MCI) rule

The mean color invariance (MCI) rule asserts the following:

Given an input patch, the average color of the (halftoned) output should be equal to that of the input patch.

As noted earlier, threshold matrices have two aspects: The first has to do with locating thresholds in the color-space, and the second deals with their placement (ordering) in the threshold array. Clearly, the MCI rule is concerned only with the first aspect, that of threshold location in color-space.

We demonstrate the idea by means of example: Suppose there are N thresholds uniformly distributed in the 2D cube. An input patch of color (m_{ij}, c_{ij}) imposes a partition of the color-space into four quadrants, as shown in Fig. 5. Now, we compute the average color of the output. From Fig. 5, assuming a uniform distribution of thresholds in the color space, the number of thresholds which are colored Blue in the output image is (recall that the cube is of unit area)

$$t_B = (\ m_{ij} \cdot c_{ij}\ ) \cdot N \quad .$$

In a similar way, the number of thresholds rendered White, Cyan and Magenta, is

$$\begin{aligned} t_W &= (\ (1 - m_{ij}) \cdot (1 - c_{ij})\ ) \cdot N \quad , \\ t_C &= (\ (1 - m_{ij}) \cdot c_{ij}\ ) \cdot N \quad , \\ t_M &= (\ m_{ij} \cdot (1 - c_{ij})\ ) \cdot N \quad . \end{aligned}$$

Taking the average color of the output image amounts to:

$$(\ t_B \cdot (1,1) + t_W \cdot (0,0) + t_C \cdot (0,1) + t_M \cdot (1,0)\ )\ /\ N \quad , \tag{1}$$

which, after doing the algebra, brings us back to $(\ m_{ij}, c_{ij}\ )$, which is indeed the average color of the input.

To conclude this example, we see that a uniform distribution of thresholds in the color-space, together with the Cartesian method of thresholding in a cube, leads to an average-preserving halftoning method.

It can be shown that in the case of Cartesian dithering in a cube, an equivalent requirement to MCI is that the marginal distribution of thresholds, along the different cube axes, is uniform.

Three dimensional color-space

The MCI rule extends as-is to 3D color-space.

4.2 Continuity

Continuity has to do with the effect small changes in the input have on the output. In our case, small changes in the input means similar colors, and the output is the halftoned image. To this end, we pose the following question:

Continuity: *Assume an input pixel $P_{ij} = (m_{ij}, c_{ij})$ is halftoned to the output color* $\mathbf{A}$*. What other values of P_{ij} are halftoned to the same output color?*
(note that we made no assumptions on the relevant T_{ij}, so the question relates only to the dithering method).

For example, consider the case of monochrome dithers, namely those with printable-colors only White (denoted by '0') and blacK (denoted by '1'). In this case, assuming that a certain input pixel was dithered to blacK, means that if the input pixel was darker (closer to blacK), it would have been dithered to blacK as well. This property is sometimes referred to as the Stacking property of Cartesian dither. A similar property also holds, separably, for Cartesian color dithering in a cube.

To demonstrate this property, consider Cartesian dithering in a 2D cube, and assume the input color P_{ij} was halftoned to the Blue vertex. This means T_{ij} is located in a specific part of the partition imposed by P_{ij}, as described by the shaded region in Fig. 7(a). We do not, however, know its exact location in that part. This, in turn, means that all the input colors in the blue region in Fig. 7(b) will also be halftoned to Blue by T_{ij}. Surely there might be other colors which are halftoned to Blue by T_{ij}, but we cannot determine them, as we do not have any more information about the exact location of T_{ij}.

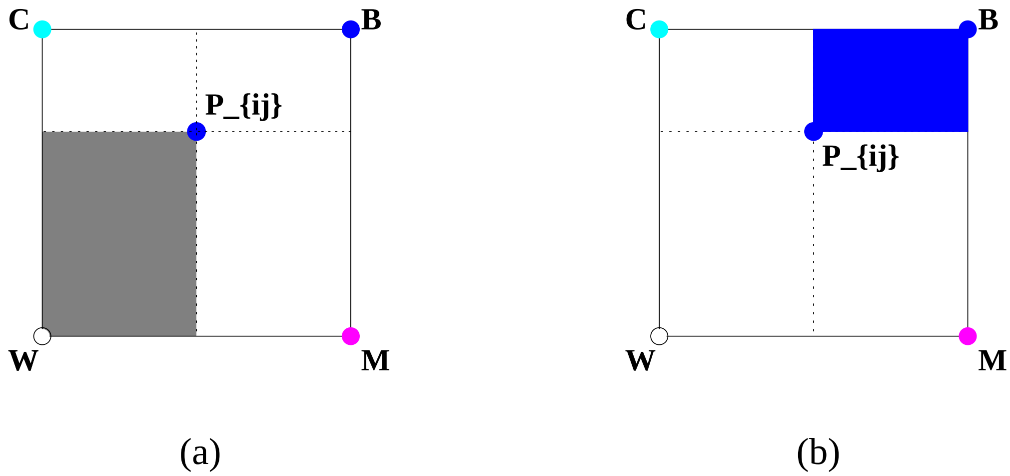

Figure 7: Continuity for the case of Cartesian dithering in a 2D cube. (a) The input pixel P_{ij} is rendered Blue by the unknown threshold T_{ij}, which means it is located somewhere in the shaded region. (b) All colors in the blue region are rendered Blue by this unknown threshold T_{ij}.

4.3 Computational aspects

Dithering is traditionally a very fast method, since it involves only few comparisons. Moreover, its memory requirements are very low, since it only involves the storing of the dither matrices, and there is no need for buffers(like in error-diffusion). However, there are a few inherently different factors when dealing with dithering in a simplex. To start with, a prerequirement for dithering in a simplex is to determine in which simplex the input color is. Secondly, as we will shortly see, some of the dithering methods involve more than simple thresholding: there are methods which involve ratio computations, comparisons, etc. This usually means more computation, however there are methods which are much faster than others.

At any rate, it is worth to keep in mind that dithering is a point operation, and therefore is amenable to parallel processing. This is true regardless of the computational burden per input pixel.

4.4 Design of thresholds placement in dither matrices

As we have seen, threshold location in color space is constrained by the dithering method and the MCI rule. The placement of the thresholds in the matrix is independent of the above, and should be designed so as to yield patterns that are pleasant to the human viewer.

There are a few known placement design methods for the standard Cartesian dithering in a cube, some of which takes explicitly into account the human visual system [3] and the printer model [6].

Threshold placement for simplex dithers is a difficult and important problem. Indeed, without a good method for threshold placement, the resulting dither process produces images of poor visual quality.

We do not refer to this problem in this report, since it involves totally different approaches, which are mostly yet not optimized for the new dither geometries proposed herein. We do use some non-optimal placements in the example section, though we do not present their derivation.

5 Barycentric coordinates

In this section we describe the important notion of barycentric coordinates. Barycentric coordinates of an input color are intimately connected to the simplex geometry and the MCI rule, as they describe the proper amount of the halftone colors to be used in its rendering. A detailed description of barycentric coordinates and some of their properties can be found in [4, 5]. Here we give their definition and mention a few relevant properties.

Assume a triangle with vertices $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$, as in Fig. 8(a). Every point $\mathbf{P}$ in the triangle can be expressed in terms of its barycentric coordinates (a, b, c), which are determined by:

$$\left.\begin{array}{ccccccc} a\mathbf{A} & + & b\mathbf{B} & + & c\mathbf{C} & = & \mathbf{P} \\ a & + & b & + & c & = & 1 \end{array}\right\} \tag{2}$$

Note that the first equation in (2) is a two dimensional vector equation.

Spelling out equation (2) in words, it means that for producing the color $\mathbf{P}$, one needs to use a relative amount of a dots of color $\mathbf{A}$, b dots of color $\mathbf{B}$, and c dots of color $\mathbf{C}$. Remarkably, this is exactly what we look for when dithering an input color. Namely, no-matter what dithering in a simplex method one uses, when halftoning an input patch of color $\mathbf{P}$, the ratio between the amount of dots of the different vertices colors used to render $\mathbf{P}$ is exactly its barycentric coordinates. This is of-course, assuming one wants to follow the MCI rule.

It can be shown [5, 4] that for all points **P** within the triangle, the barycentric coordinates are non-negative. Fig. 8 describes a few features of the barycentric coordinates which are of use to us:

- Lines of constant barycentric coordinate - Lines of constant (e.g.) a coordinate are lines parallel to the edge facing the vertex **A**, namely the edge connecting **B** and **C**, (dashed lines in Fig. 8(a)). Specifically, the vertex **A** is of value $a = 1$, and the edge **B**-**C** is of value $a = 0$. Value of $a = 0$ means that points on this line can be described using linear combination of only **B** and **C**.

- Lines of constant coordinate ratio - Lines of constant coordinate ratio (e.g.) $a : b$ are lines through the other vertex **C** (dotted lines in Fig. 8(a)). The constant ratio is easily determined from their junction with **A**-**B**.

- Area Proportions - Given a point **P** in the triangle, one can induce a partition of the triangle into three sub-triangles whose vertices are **P** and couples of triangle vertices, as shown in Fig. 8(b). The area ratio of those three triangles is equal to the ratio of the barycentric coordinates of **P**, namely:

$$A_{\text{area}} : B_{\text{area}} : C_{\text{area}} = a : b : c \quad . \tag{3}$$

- Affine Invariance - The barycentric coordinates of a point in a triangle are invariant with respect to affine transformations. This enables us to discuss methods with respect to a certain triangle, using barycentric coordinates, and to apply them to any triangle, since all the triangles are affine transformations of each other.

Three dimensional color-space

Full description of 3D extensions can be found in [4]. Here we give only a brief description. The triangle extends to a tetrahedron with 4 vertices. There are 4 barycentric coordinates. Planes with a constant barycentric coordinate are parallel to the faces of the tetrahedron. Planes with constant ratio of two coordinates pass through the other two vertices. A point in a tetrahedron induces partition into 4 sub-tetrahedra whose vertices are that point and

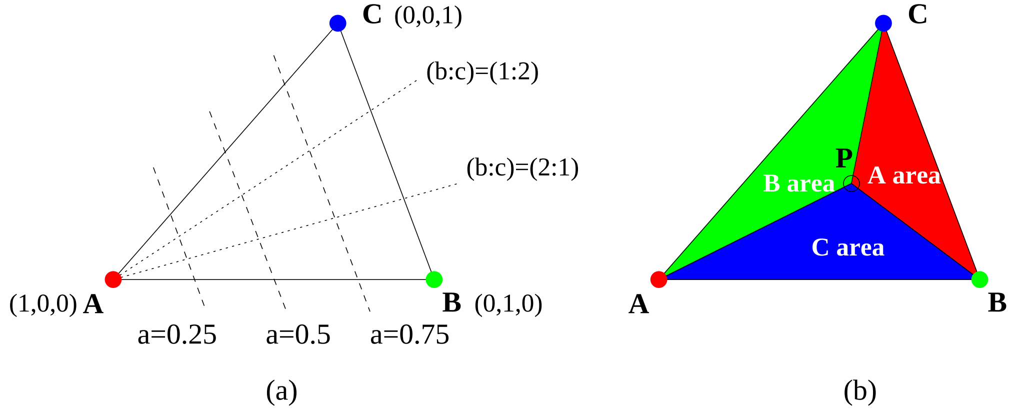

Figure 8: Barycentric coordinates. (a) Triangle with barycentric coordinates, iso-coordinate and iso-ratio lines. (b) Partition induced by a point in a triangle.

every triplet of the tetrahedron vertices. The ratio of the sub-tetrahedron volumes is equal to the ratio of the barycentric coordinates of that point.

6 Barycentric Dithering in a Simplex

This section describes the first method for dithering in a simplex, which was first described in [4]. It is described first in geometrical terms, and then using algebraic formulation. The name of the method stems from the fact that it relies heavily on properties of the barycentric coordinate system.

6.1 Geometrical interpretation

This method relies on the fact that the area ratio of the simplex partition induced by an input color, is equal to the ratio of the barycentric coordinates of that input color.

The thresholding method is described as follows (see Fig. 9):

Alg. III : Barycentric Dithering in a Simplex (Geometric formulation)

1. Locate the input-pixel P_{ij} in the color-space.
2. Partition the color-space into 3 sub-simplices - P_{ij} replaces each vertex in turn.

3. Locate the corresponding threshold T_{ij} in the color-space partition.

4. Color it with the color of the vertex not included in the sub-simplex.

It can be shown that the MCI rule is obeyed if and only if the thresholds are uniformly distributed in the simplex [4].

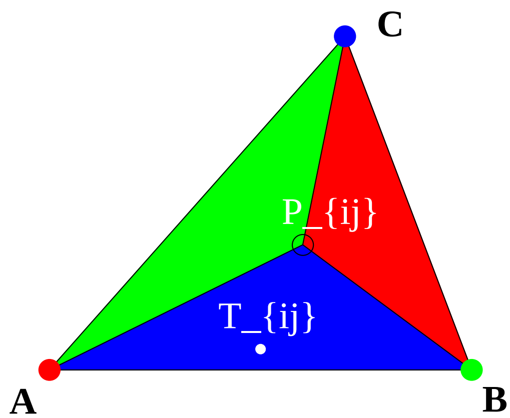

Figure 9: Barycentric Dithering: geometric view.

Three dimensional color-space

In three-dimensions, the partition of the tetrahedral color-space is to 4 sub-tetrahedra. Instead of dealing with areas, one deals with volumes, and the distribution of the thresholds in the color-space is assumed uniform in space rather than in area.

6.1.1 Dual view

As in the case of Cartesian dithering in a cube, one may gain insight by viewing the dual relation between thresholds and colors. Fig. 10(a) depicts an input color partitioning of the threshold space. In the dual problem one assumes a given threshold which partitions the input-colors according to the halftone they render it. The solution is presented graphically in Fig. 10(b). If the input color is in the upper partition, it renders the threshold as **C**. This upper partition is colored **C** in the figure. A similar explanation holds for the other parts of the partition. Thus, we see that each threshold imposes a partition of the color-space. The partition is formed by connecting the threshold to each face with a line segment whose continuation passes through the opposite vertex.

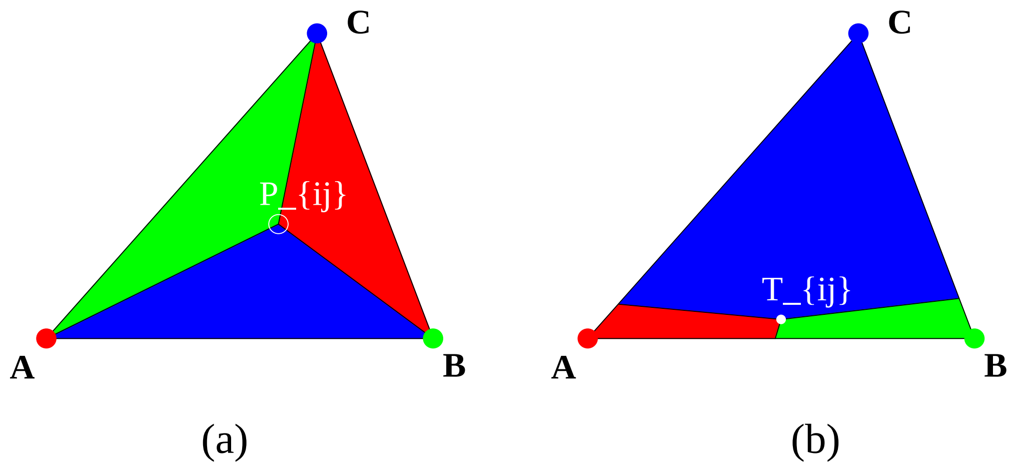

Figure 10: Barycentric dithering in a simplex: Dual relation between thresholds and colors. (a) Partition induced by the input color and the respective coloring of the thresholds. (b) Partition induced by a threshold, and the respective colors it would be halftoned to by an input color.

Three dimensional color-space

The dual partition in the 3D case has a more complex construction. Each of the 4 resulting parts includes the corresponding vertex and all the tetrahedron points that can be reached from that vertex without crossing either of the planes passing through the threshold and two of the other 3 tetrahedron vertices. It can be shown that each part has 6 faces, each of which has 4 edges (the resulting partition on the tetrahedron faces is the same as in Fig. 10(b)).

6.2 Algebraic Interpretation

It can be shown [4] that barycentric thresholding, i.e. the determination of the threshold location, with respect to the partition induced by the input color, can be done using simple ratios, as follows:

Alg. IV : Barycentric Dithering (Algebraic formulation)

1. Compute the barycentric coordinates of the color $\mathbf{P}$ and the threshold $\mathbf{T}$, and denote them as $(\mathbf{P}_a, \mathbf{P}_b, \mathbf{P}_c)$ and $(\mathbf{T}_a, \mathbf{T}_b, \mathbf{T}_c)$, respectively.

2. Compute the ratios

$$r_i = \frac{\mathbf{P}_i}{\mathbf{T}_i} \qquad \texttt{for} \;\; i = \{a, b, c\} \quad . \tag{4}$$

3. The halftone is the argument of

$$\max_i \quad r_i \quad . \tag{5}$$

Three dimensional color-space

3D color-spaces have 4 barycentric coordinates. Thus, 4 ratios have to be computed and compared.

6.3 Comments on Barycentric Dithering

- Continuity (as defined in Sec. 4.2)- Assume an input color P_{ij} was halftoned to vertex $\mathbf{C}$. This means that T_{ij} is located somewhere in the partition imposed by P_{ij}, as described by the shaded region in Fig. 11(a). This in turn means that all colors in the blue region in Fig. 11(b) will be halftoned to vertex $\mathbf{C}$ by T_{ij}. Surely there can be other colors that will be halftoned to vertex $\mathbf{C}$ by T_{ij}, but using only the information we got from the halftoning of P_{ij}, this is the most we can say.

- Computational aspects - In comparison to Cartesian dithering, one has to compute barycentric coordinate ratios, and then find the maximum, rather than simple thresholding. Moreover, in actual applications one also needs to compute the barycentric coordinates, as the input is usually given in (R,G,B) coordinates. Nevertheless, it is clearly a point operation halftoning process, and therefore it is qualified for Dithering.

- Design and usage of dithering matrices - A method for the design of a barycentric dither matrix was proposed in [4]. Sample results are shown in Sec. 9.

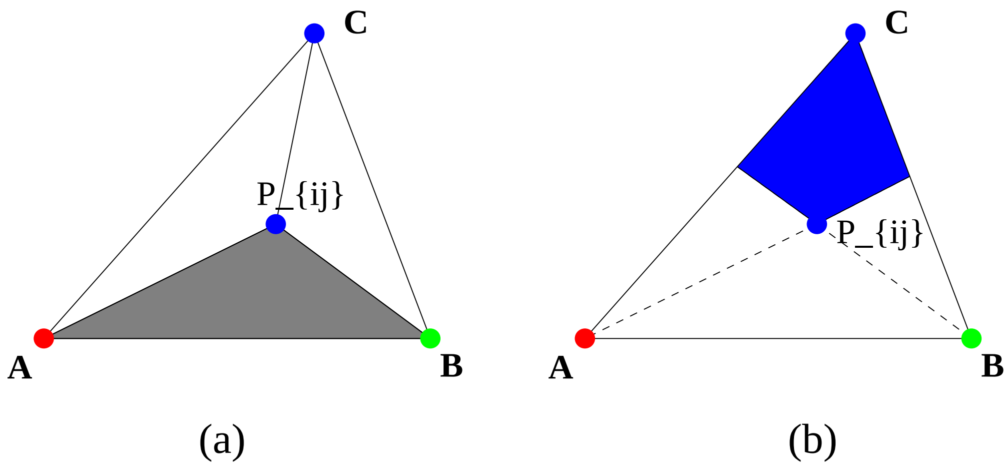

Figure 11: Barycentric dithering in a simplex: Continuity. (a) The input pixel P_{ij} is rendered **C** by the unknown threshold T_{ij}, which means it is located somewhere in the shaded region. (b) All colors in the blue region are rendered **C** by the same unknown threshold T_{ij}.

7 Cartesian Dithering in a Simplex

The method described here is named Cartesian because it partitions the simplex using hyperplanes that are parallel to the simplex faces.

7.1 2D Geometric interpretation

This method relies on the fact that the loci of points with a constant barycentric coordinate are hyperplanes parallel to one of the simplex faces. In this section we present the 2D case. There are two possible 3D geometric interpretations, which are detailed in the sequel.

In this dithering method, all the thresholds are located on one edge of the simplex, (e.g: the **B**-**C** edge). The thresholding method is described as follows (see Fig. 12):

Alg. V : Cartesian Dithering in a Simplex (Geometric formulation)

1. Locate the input-pixel P_{ij} in the color-space.
2. Draw two lines through P_{ij} parallel to the **A**-**B** and **A**-**C** edges of the triangle, respectively. This induces a partition of the third edge of the triangle, **B**-**C**.

3. `Locate the threshold in the above partition, and color it accordingly: If the threshold is in the part adjacent to vertex` $\mathbf{B}$ `it is colored` $\mathbf{C}$`, otherwise if it is in the part adjacent to vertex` $\mathbf{C}$ `it is colored` $\mathbf{B}$`, and finally if the threshold is in the mid-partition, it is colored` $\mathbf{A}$.

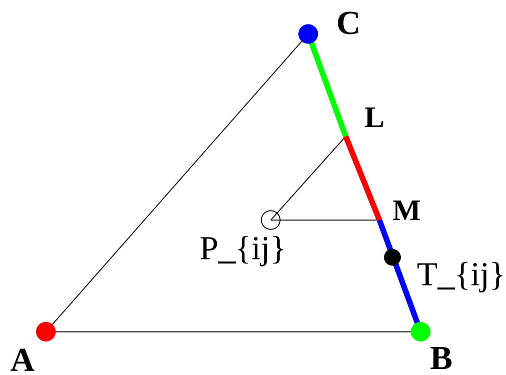

Figure 12: Cartesian Dithering in a simplex: geometric formulation assuming thresholds are located only on the **B**-**C** edge. L and M denote the threshold-space partition points.

Recall that lines parallel to the edges are of constant barycentric coordinate. Therefore, point L in Fig. 12 has the same b coordinate as P_{ij}, and point M has the same c coordinate as P_{ij}. Thus, the length ratio $|\,L - \mathbf{C}\,|/|\,\mathbf{B} - \mathbf{C}\,|$ is b, the length ratio $|\,M - \mathbf{B}\,|/|\,\mathbf{B} - \mathbf{C}\,|$ is c, and since the coordinates are summed to one, the length ratio corresponding to $L - M$ is a. If the thresholds are uniformly distributed on the $\mathbf{B} - \mathbf{C}$ edge, the right proportion of thresholds are colored to each of the halftone colors, and thus the method complies with the MCI rule.

The dual interpretation of Cartesian-Dithering is described in Fig. 13. The threshold-space partition in Fig. 13(a), and the dual input color partition in Fig. 13(b). Note that unlike the previous cases, there is no complete duality between colors and thresholds. Namely, thresholds are limited to lie on the line segment, whereas input-colors can lie anywhere within the simplex.

Also note that there is no preference to a certain edge of the triangle (e.g. $\mathbf{B} - \mathbf{C}$), and the whole procedure can be adjusted to the thresholds located on any other edge.

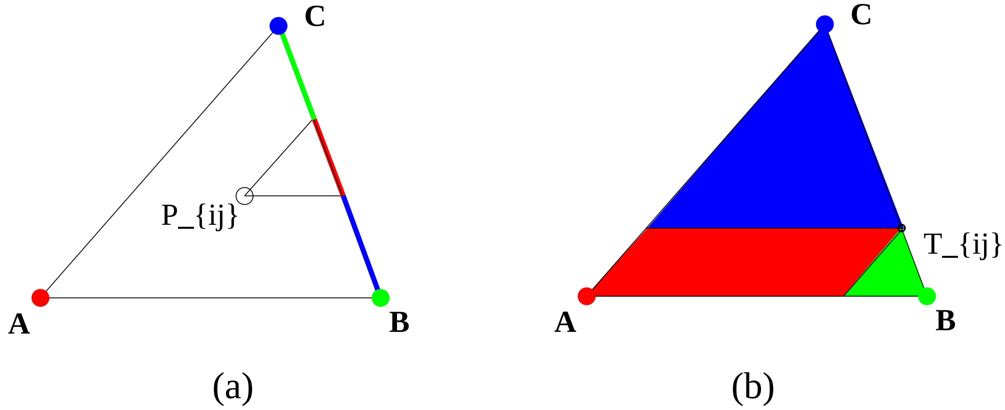

Figure 13: Cartesian dithering in a simplex: Dual relation between thresholds and colors. (a) Partition implied by an input color. (b) Partition implied by a threshold.

Three dimensional color-space

3D extension is somewhat more complicated this time than before. There are two possible extensions: In the first the thresholds are located on an edge of the tetrahedron, whereas in the second extension the thresholds are distributed on its face. Both methods preserve the 'Cartesian' nature of the 2D method, namely the usage of hyperplanes parallel to the axes. The first extension is discussed in Sec. 7.2, and the second is described briefly in Sec. 7.3.

7.2 3D Geometric Interpretation: Thresholds on an Edge

The 3D geometric interpretation is depicted in Fig. 14. We assume that the thresholds are uniformly distributed on the **A**-**B** edge. Three planes pass through point **P** describing the input-pixel color. Each plane is parallel to two out of the following three edges: **A**-**C**, **C**-**D**, **D**-**B**. The first plane (colored magenta in Fig. 14), is parallel to edges **C**-**D** and **D**-**B**, and intersect **A**-**B** at L. The second plane (colored yellow) is parallel to **A**-**C** and **C**-**D**, and intersects **A**-**B** at N. The third plane (colored green) is parallel to **A**-**C** and **D**-**B**, and intersects **A**-**B** at M. Thus, the three planes induce a partition of **A**-**B** into 4 segments. The halftoned color is determined according to the threshold location:

- The threshold is in **A**-$N \Longrightarrow$ output color is **B**.
- The threshold is in N-$M \Longrightarrow$ output color is **D**.
- The threshold is in M-$L \Longrightarrow$ output color is **C**.

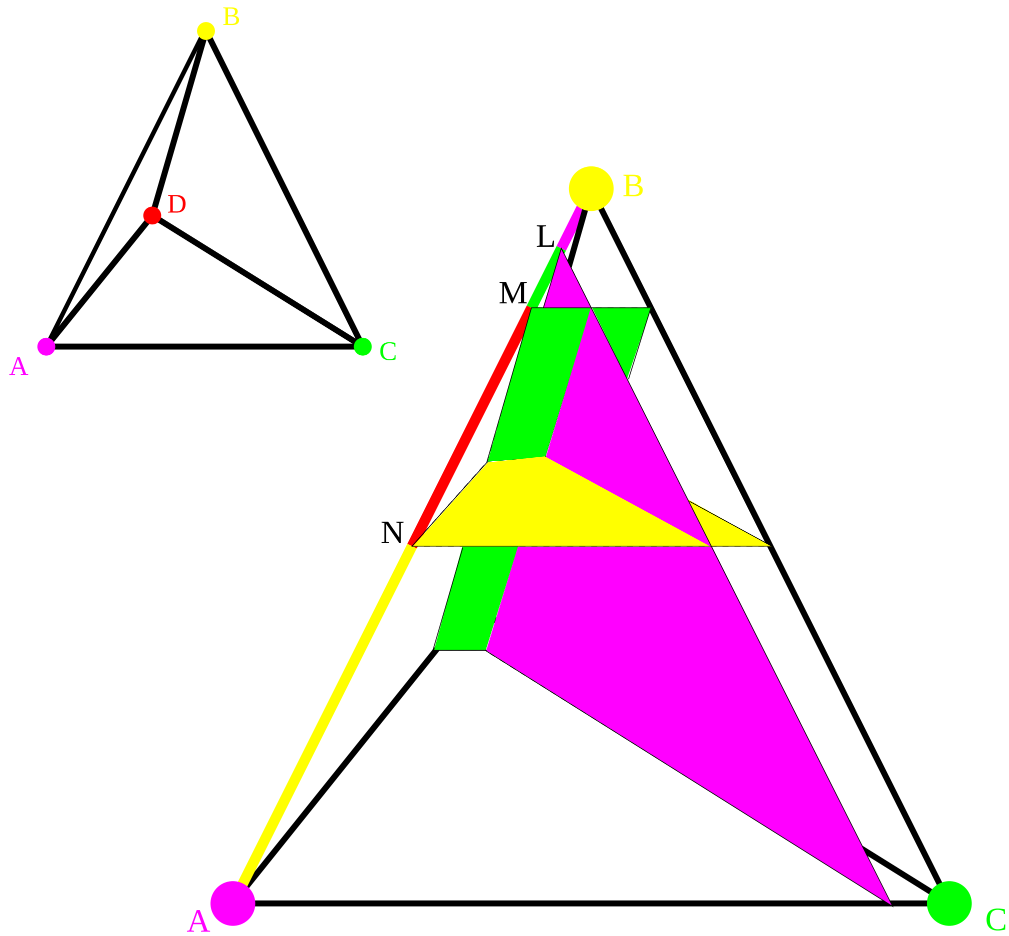

Figure 14: Cartesian thresholding in a 3D simplex, where thresholds are distributed along a line.

- The threshold is in L-$\mathbf{B}\Longrightarrow$ output color is $\mathbf{A}$.

This method, and the fact that the MCI rule holds when the thresholds are distributed uniformly on **A**-**B**, can be explained in a similar way to the 2D case, relying on the property of planes of constant barycentric coordinates.

Note that planes that are parallel to edges not sharing a face in the tetrahedron (e.g, the green plane in Fig. 14) have constant coordinate sums (in the above case $a + c$).

7.2.1 One-dimensional interpretation

We now give a different geometrical interpretation of the dithering method in 3D . This simple interpretation is shown schematically in Fig. 15. Assume the number of thresholds is N. We can order the thresholds and refer to them by an index $i \in \{1, 2, \ldots N\}$. Given an input patch of color $\mathbf{P}$ with coordinates (a, b, c, d), the above dithering method can be described as assigning the first $b \cdot N$ thresholds to color $\mathbf{B}$, the next $d \cdot N$ thresholds to color $\mathbf{D}$, the next $c \cdot N$ thresholds to color $\mathbf{C}$, and the last $a \cdot N$ thresholds to color $\mathbf{A}$.

Figure 15: Cartesian dithering in a simplex: One dimensional geometric interpretation.

7.2.2 Algebraic Interpretation

An algebraic formulation follows immediately from the above one dimensional interpretation.

Alg. VI : <u>Cartesian Dithering in a Simplex (Algebraic formulation)</u>

1. `Compute the barycentric coordinates` $(\mathbf{P}_a, \mathbf{P}_b, \mathbf{P}_c, \mathbf{P}_d)$ `of` $\mathbf{P}$ `and find the order-index` $\mathbf{T}_i$ `of threshold` $\mathbf{T}$.

2. `If` $(\mathbf{P}_b + \mathbf{P}_d) \cdot N \geq i$ `If` $\mathbf{P}_b \cdot N \geq i$ `Output` $\mathbf{B}$.
   `else` `Output` $\mathbf{D}$.
   `else` `If` $\mathbf{P}_a \cdot N \geq (N - i)$ `Output` $\mathbf{A}$.
   `else` `Output` $\mathbf{C}$.

7.3 3D Geometric Interpretation: Thresholds on a Plane

So far we delt with a 3D extension of 2D Cartesian dithering in a simplex placing the thresholds on an edge of the simplex. Here we briefly describe a different 3D extension. In this extension, as described in Fig. 16, the thresholds are uniformly distributed on a face of a simplex.

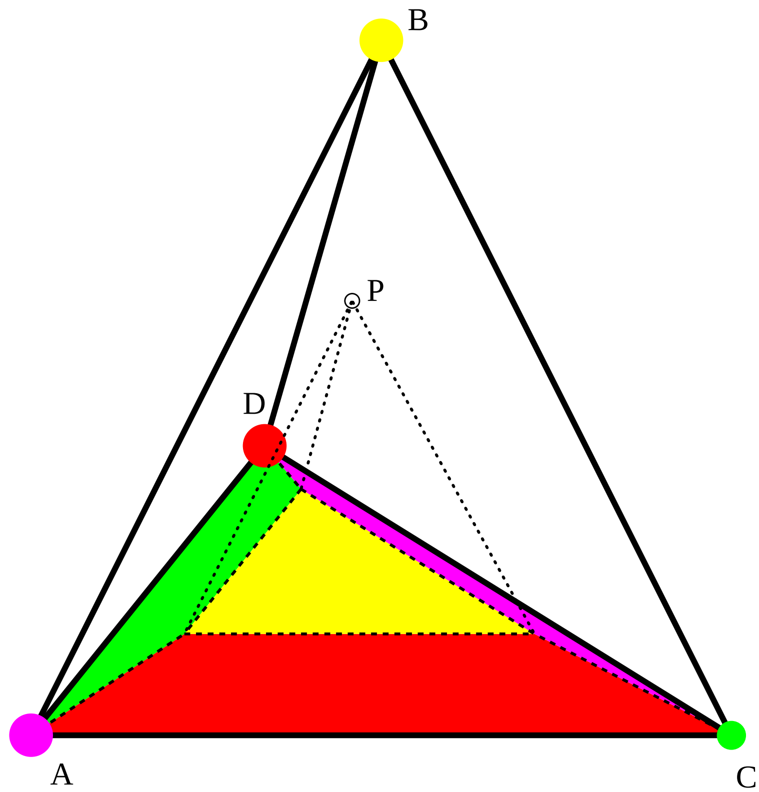

Figure 16: Cartesian thresholding in a 3D simplex, where thresholds are on a face.

Without loss of generality, we assume that the thresholds are uniformly distributed on the **A**-**C**-**D** face, which is denoted as the basis of the tetrahedron. Three planes parallel to the other three simplex faces are passed through the input-color **P**. These planes intersects the basis, and form the partition described in Fig. 16. The thresholds are then colored as shown. We state, without proof, that the MCI rule holds when the thresholds are distributed uniformly on the basis.

In some sense this method is an hybridization of Cartesian dithering in a simplex and Barycentric dithering.

7.4 Comments on Cartesian dithering in a simplex

- Continuity- Following a reasoning similar to the previous cases (Sec. 4.2 and Sec. 6.3), we draw Fig. 17.

- Computational aspects - In terms of comparisons, Cartesian dithering in a simplex is faster then dithering in a cube (2 comparisons in Alg. VI (section 7.2.2) as opposed to 3). However, it should be remembered that for all simplex dithering, there is an extra cost of determining the simplex the input-color is in, and its barycentric coordinates in that simplex. All these overshadow the differences due to comparisons.

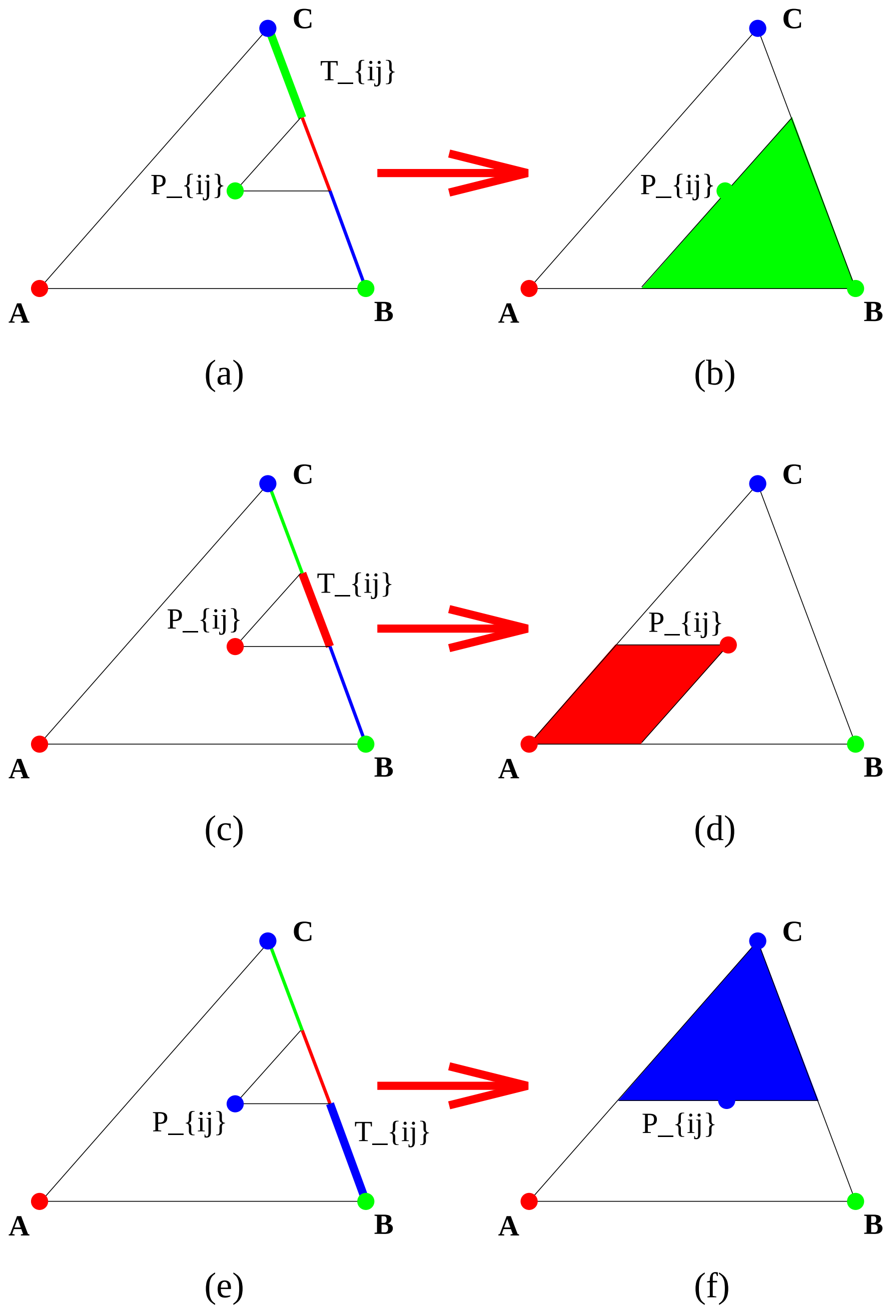

Figure 17: Cartesian dithering in a simplex, continuity property. (a)-(b) : Assuming input pixel P_{ij} is dithered to color **B** (Green), then threshold T_{ij} is located somewhere on the Green segment in (a). Therefore, all input colors in the area denoted by Green in (b) are also halftoned to **B** (Green). (c)-(d) : In this case assume P_{ij} is dithered to **A** (Red). (e)-(f) : In this case assume P_{ij} is dithered to **C** (Blue).

8 Helical Dithering in a Simplex

This method is unique in that it is inherently 3D , and does not have an immediate 2D counterpart. The explanations are therefore somewhat more difficult, but nevertheless there are some interesting features which are worth pointing.

8.1 Algebraic Motivation

In 3D simplices, there are 4 vertices which are denoted here by **A**,**B**,**C**, and **D**, as in Fig. 18(a). Now, let us associate them with the vertices of a 2D square, as shown in Fig. 18(b), and assume the thresholds are uniformly distributed in this square. Each threshold has two coordinates, $\mathbf{T} = (\mathbf{T}_x, \mathbf{T}_y)$. Note that in this formulation, there is no immediate equivalence between colors and thresholds (we establish a relation shortly, however, only for the sake of completeness).

Assume the input color has barycentric coordinates $(\mathbf{P}_a, \mathbf{P}_b, \mathbf{P}_c, \mathbf{P}_d)$. These 4 coordinates imply a partition of the square, as described in Fig. 18(b). Namely, the 4 barycentric coordinates are transformed into a pair of two points in the square:

$$P_1 = (\frac{\mathbf{P}_b}{\mathbf{P}_b + \mathbf{P}_d},\ \mathbf{P}_b + \mathbf{P}_d) \qquad , \quad P_2 = (\frac{\mathbf{P}_c}{\mathbf{P}_a + \mathbf{P}_c},\ \mathbf{P}_b + \mathbf{P}_d) \quad . \tag{6}$$

The halftoned color of a threshold depends on its location in this partition. In Fig. 18(b) thresholds are colored according to the vertex opposite to their location. An extreme case may help to see that the method really works: Assume the input color is very close to vertex **B** in the simplex. This means that $\mathbf{P}_b$ is close to one, and the other coordinates are close to null. In this case, we see from (6) that indeed almost all the thresholds are colored **B**, which is the desired result for this case.

The algorithm follows immediately:

Alg. VII : <u>Helical Dithering in a Simplex (Algebraic formulation)</u>

1. `Compute the barycentric coordinates` $(\mathbf{P}_a, \mathbf{P}_b, \mathbf{P}_c, \mathbf{P}_d)$ `of the input color` $\mathbf{P}$`, and denote the threshold coordinates as` $(\mathbf{T}_x, \mathbf{T}_y)$`.`

2. If $\mathbf{P}_b + \mathbf{P}_d \geq \mathbf{T}_y$,

if $\frac{\mathbf{P}_b}{\mathbf{P}_b + \mathbf{P}_d} \geq \mathbf{T}_x$ threshold is colored $\mathbf{B}$.

else threshold is colored $\mathbf{D}$.

else,

if $\frac{\mathbf{P}_c}{\mathbf{P}_a + \mathbf{P}_c} \geq \mathbf{T}_x$ threshold is colored $\mathbf{C}$.

else threshold is colored $\mathbf{A}$.

Assuming the thresholds are uniformly distributed in the square, the MCI rule is obeyed by this method, since the average in this case is equal to the areas enclosed by the various partitions.

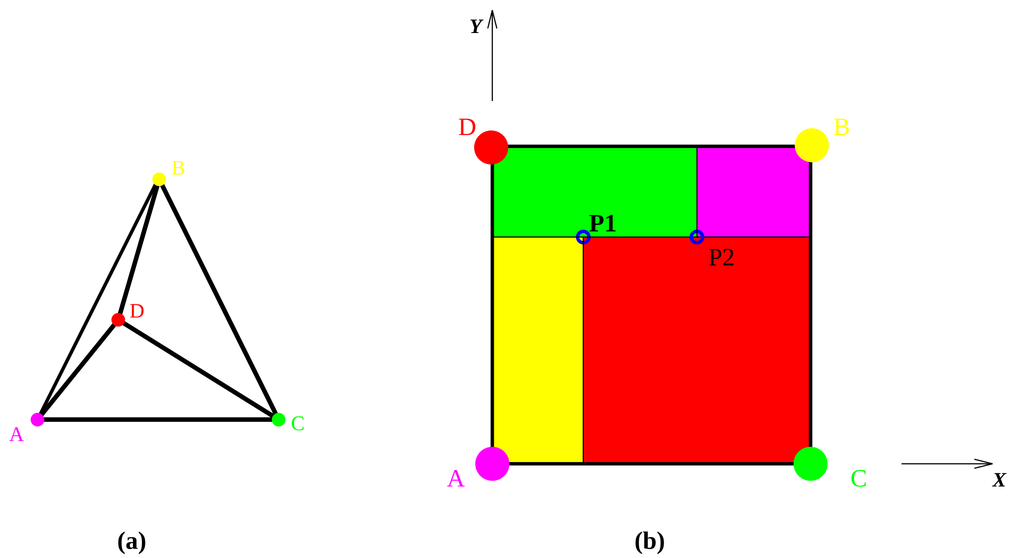

Figure 18: Helical dithering in a simplex: (a) Original simplex. (b) Colored partition induced by an input color according to (6). Also shown are (x, y), the axes of the threshold space.

8.2 Geometrical Interpretation

Although the following geometrical interpretation can hardly motivate Helical Dithering, it is still a 'nice-to-have' property, and after all, the geometric interpretation was the reason for the name coined for the method.

Recall that each threshold has two coordinates, $(\mathbf{T}_x, \mathbf{T}_y)$. The location of a threshold in the color space can be determined according to the input-color variations which make it change its output color. Thus, we see that each threshold can be characterized by the following:

1. Colors for which $\mathbf{P}_b + \mathbf{P}_d \geq \mathbf{T}_y$ are halftoned to either **B** or **D**, whereas the other colors are halftoned to either **A** or **C**.
2. Colors for which either $\frac{\mathbf{P}_b}{\mathbf{P}_b+\mathbf{P}_d} \geq \mathbf{T}_x$ or $\frac{\mathbf{P}_c}{\mathbf{P}_a+\mathbf{P}_c} \geq \mathbf{T}_x$ are colored to either one of **B** or **C**, or to one of **A** or **D**, respectively. Which of the inequalities is effective is determined by the previous item.

Since we would like to have a consistent description of the threshold location in the color-space, the following should hold for the threshold coordinates in color-space $(\mathbf{T}_a, \mathbf{T}_b, \mathbf{T}_c, \mathbf{T}_d)$:

$$\mathbf{T}_b + \mathbf{T}_d = \mathbf{T}_y \tag{7}$$
$$\frac{\mathbf{T}_b}{\mathbf{T}_b + \mathbf{T}_d} = \mathbf{T}_x \tag{8}$$
$$\frac{\mathbf{T}_c}{\mathbf{T}_a + \mathbf{T}_c} = \mathbf{T}_x \tag{9}$$

These three equations, together with the requirement of the barycentric coordinates that the coordinates sums up to one, determine the values of the threshold coordinates in color-space. It is nice to note that the thresholds are located on points for which

$$\frac{\mathbf{T}_b}{\mathbf{T}_d} = \frac{\mathbf{T}_c}{\mathbf{T}_a} \quad , \tag{10}$$

which in turn describes a Helical surface, extending from the **A**-**C** edge to the **B**-**D** edge, as shown in Fig. 19(c).

We now turn to describe the dithering process geometrically.

Alg. VIII : Helical Dithering in a Simplex (Geometric formulation)

1. Compute the barycentric coordinates $(\mathbf{P}_a, \mathbf{P}_b, \mathbf{P}_c, \mathbf{P}_d)$ of the input color $\mathbf{P}$.

2. Draw three planes through $\mathbf{P}$, as shown in Fig. 19(a):

   (a) A plane through $\mathbf{P}$ with constant $\mathbf{P}_b + \mathbf{P}_d$ coordinates, (colored Red in Fig. 19(a)).

   (b) A plane through $\mathbf{P}$, $\mathbf{B}$, and $\mathbf{D}$, (colored Green in Fig. 19(a)).

   (c) A plane through $\mathbf{P}$, $\mathbf{A}$, and $\mathbf{C}$, (colored Yellow in Fig. 19(a)).

3. Locate the threshold in the above partition, and color the threshold according to its location, as described in Fig. 19(b).

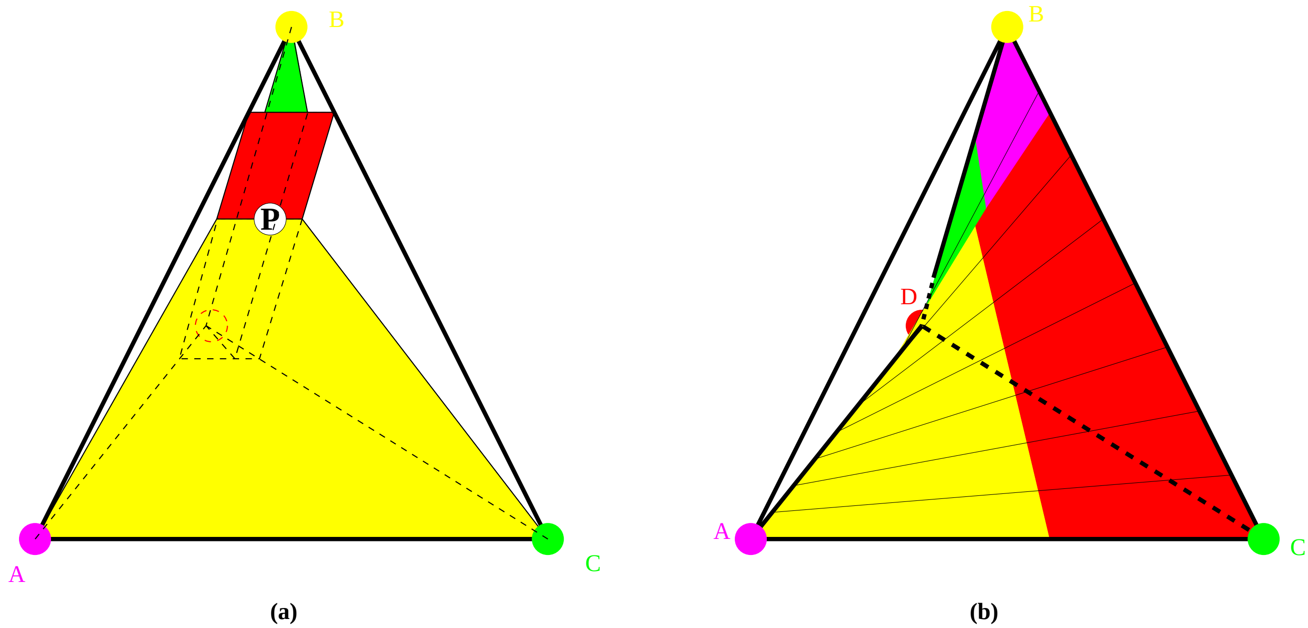

Figure 19: Helical dithering in a simplex: Geometric interpretation.

9 Examples

In order to demonstrate a dithering method, one has to actually produce a dither-matrix, which means assigning a placement in the dither-matrix for each threshold. This operation requires an extensive effort to modify existing dither matrix design methodologies to the new geometric constraints accompanied by a lot of tuning to the specific printer and print mode in question.

The following images were produced using HP1200C DeskJet, and were printed on premium paper. The dither-matrices used were not optimized, but rather patched on an existing design-procedure for classic dithering-in-a-cube [3, 4]. The examples merely show the feasibility of the above methods, and that reasonable results are attained even without a fine-tuned optimization.

We now describe in detail the production of the examples. First, a dither matrix is designed off-line. The same matrix applies for all simplices [4]. Note that for the case of cartesian dithering in a simplex, we can use a matrix whose thresholds lie on a straight line (which happens to be an edge of a simplex). Each element of the matrix describes the appropriate threshold, stored as a quadruple of barycentric coordinates. Then, the input image is dithered pointwise: For each input pixel, one determines the simplex to be used. Following [7], the RGB cube is partitioned into the following 6 tetrahedra:

$$\{\text{WCMY}\} \quad \{\text{GCMY}\} \quad \{\text{RGMY}\} \quad \{\text{CGMB}\} \quad \{\text{RGMB}\} \quad \{\text{RGKB}\} \quad , \tag{11}$$

in which colors are dithered (each color in the simplex it is contained in). Next, the barycentric coordinates of the input color are computed, with respect to the selected simplex, see [4]. Finally, the coordinate values are compared with the appropriate threshold, and a decision as to the output color is made.

Three examples are shown in Fig. 20. The top image, produced using color-smooth dither screen [3], should serve as a reference. It was produced using the conventional dithering in a cube method, using a state of the art dither-matrix. The image in the middle was produced using barycentric dithering in a simplex, and the one at the bottom was produced using Cartesian dithering in a simplex (thresholds distributed on an edge). One can see that the color-content in these two images is different from that in the one based on the Cartesian dithering in a cube. This can be contributed to the fact that *before* the input color is halftoned, it is associated with a specific simplex, and therefore only part of the printable-colors can participate in rendering that specific input-color. According to the Minimum Brightness Variation Criterion (MBVC [7]), this choice of color content produces better results. The partition into tetrahedra may produce an undesirable effects in smoothly varying backgrounds, as can be seen in the background gray color. This will be hopefully taken care-off by a careful design of the dither-matrix. Another effect which can be contributed to the quality of the dither matrix is the clustering of points in the barycentric dithering. This is mainly because of the poor quality of the dither matrix used. On the other hand, in the case of Cartesian dithering in a simplex, we used a reasonable dither matrix, which was obtained based on a color-smooth design, and therefore there is no such clustering.

To conclude, beside the feasibility that these images demonstrate, we note the different color-content of the images, the banding-effect due to the partition into tetrahedra, and the impact of the quality of the dither matrix.

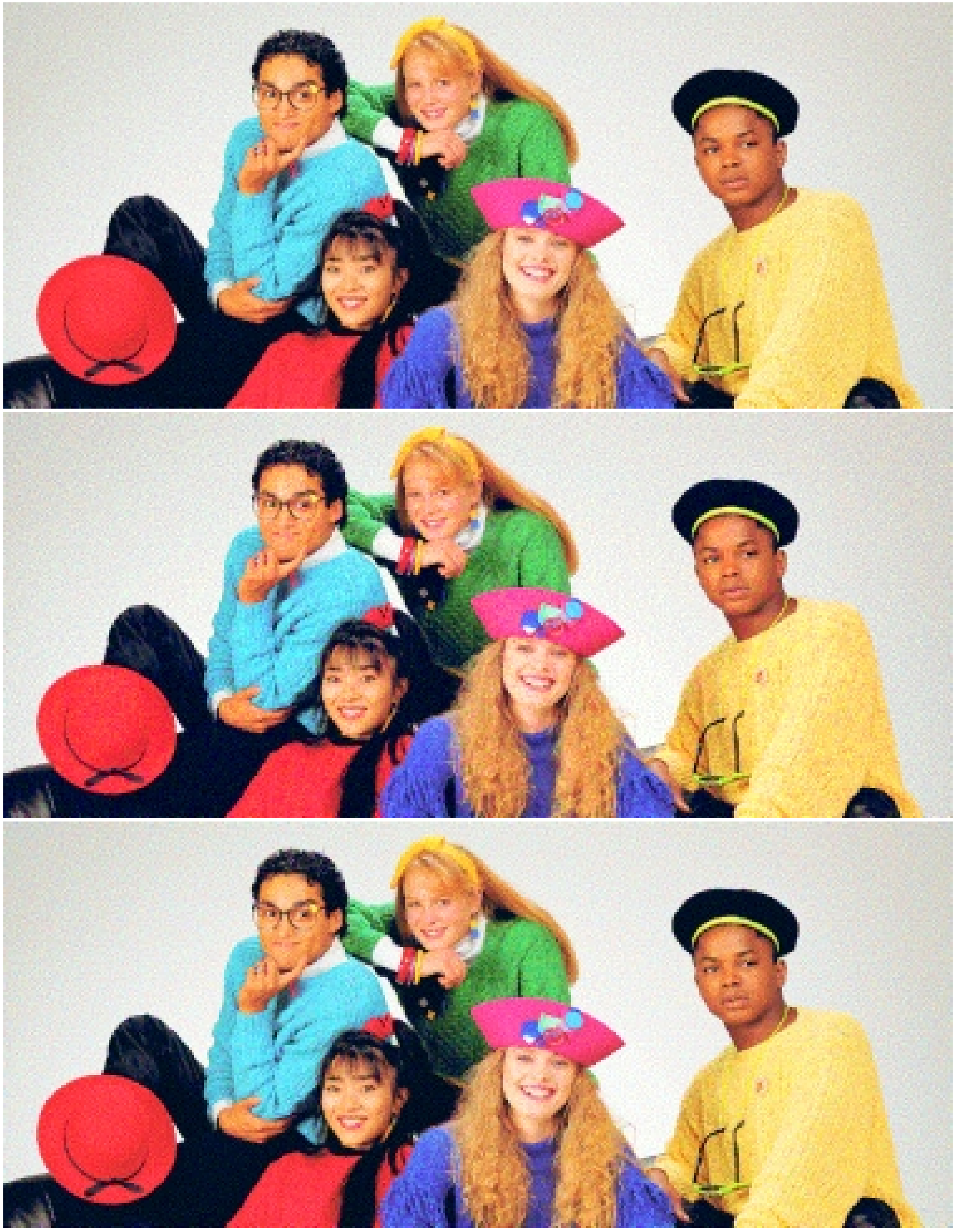

Figure 20: Dithering using three different methods: Top - Dithering in a cube, using Color-Smooth method [3]. Middle - Barycentric dithering in a simplex. Bottom - Cartesian dithering in a simplex (thresholds on an edge).

10 Summary and Conclusions

This report introduces the problem of dithering in a simplex, and suggests a few possible solutions. These were the Barycentric Dithering, Cartesian Dithering in a Simplex, and the Helical Dithering. It is important to note that there are many other possible methods, as pointed out in the text.

No attempt was made to describe which method is better, because a quality criterion was not defined. However, a few remarks concerning continuity and computational costs were given.

We would also like to note that any dithering method, and in particular dithering in a simplex, can be formulated as a one dimensional dithering if the order of the thresholds is allowed to change depending on the input color. By one dimensional dithering we mean that the thresholds are ordered by one index. Thus, a specific 'quantizer' is associated with each input color. So far, in all the methods discussed, we used the same 'quantizer-family' for all the colors, merely with different parameters. However, this is certianlly not a must in a general setting.

Future directions may include the definition of a quality criterion which will enable grading the various methods. Also, a very important problem to solve is the design of good dither matrices, or in other words, the placement of thresholds in the matrix.

11 References

[1] R. Ulichney, *Digital Halftoning*, Cambridge, Ma: MIT Press,1987.

[2] Y. Pnueli, A. M. Bruckstein, *Gridless Halftoning: A reincarnation of the old Method*, Graphical Models and Image Processing, Vol. 58, No. 1, January, pp. 38-64,1996.

[3] Q. Lin, and J. Allebach, *Color Smooth Dither*, HPL Technical Report HPL-97-76, June 1997.

[4] N. Arad, D. Shaked, Z. Baharav, and Q. Lin, *Barycentric Screening: Dithering in the World of Color*, HPL Technical Report HPL-97-103, August 1997.

[5] J. Hoschek and D. Lasser, *Fundamentals of Computer Aided Geometric Design*, A. K. Peters, Wellesley, Massachusetts, 1993. Translated by L. L. Schumaker.

[6] T. N. Pappas, *Model Based Halftoning of Color Images*, IEEE Transactions on image processing, Vol. 6, No. 7, July 1991.

[7] D. Shaked, N. Arad, A. Fitzhugh, and I. Sobel, "Color-diffusion: Tailoring error-diffusion to the world of color", Technical Report HPL 96-129, Hewlett-Packard Laboratories, August 1996. Proceedings version in *Proc. of SPIE Electronic Imaging*, Vol. 3648, January 1999.