A Mathematical Contribution To Dance Notation:

Analysing Labanotation with Euclidean Geometry, Computing Matrices for Dance Notation, and Choreographing with Crystallographic Groups

Claudia Farnesi

A Thesis

in

The Department

of

Mathematics and Statistics

Presented in Partial Fulfilment of the Requirements
for the Degree of Master of Science at
Concordia University
Montreal, Québec, Canada

May 2006

Your file Votre référence
ISBN: 0-494-14233-2
Our file Notre référence
ISBN: 0-494-14233-2

ABSTRACT

**A Mathematical Contribution to Dance Notation:
Analysing Labanotation with Euclidean Geometry,
Computing Matrices for Dance Notation,
and Choreographing with Crystallographic Groups**

Claudia Farnesi

Dances consist of bodies moving through space and time, a concept established by the great choreographer Merce Cunningham. Dance notation is the recording of these movements on paper. This multidisciplinary research aims at bridging the gap between the sciences and the arts. We mathematically investigate an existing system of dance notation, and use mathematical tools to generate new ones.

The arts of dance and dance notation contain numerous mathematical concepts, mostly relating to Euclidean geometry. The first objective of this research is to identify these mathematical structures present in Labanotation. The second is to characterize dances using algebra. In one section, positions of partners in contradancing are defined by matrices and calculated through matrix multiplication using Homogeneous Coordinates. In another section, body movements are encoded into 4×6 matrices; the rows represent the four-dimensional coordinate space, and the columns the different body parts. After raising into 5×7 matrices using the concept of homogeneous coordinates, summing a sequence of matrices provides a choreography matrix representing the final position of a dancer as dictated by the sequence. The third objective is to choreograph using crystallographic groups (or wallpaper groups). Geometric shapes are designed to represent the basic steps of certain ballroom dances, and each group is applied to each symbol using Artlandia's SymmetryWorks in Adobe Illustrator. A brief discussion explains why only five groups are relevant, and the ensuing results illustrate that these groups applied to the dance symbols generate mostly feasible choreographic routines.

ACKNOWLEDGEMENTS

This thesis was a lot of fun but a lot of work, and despite my numerous ideas, it took a long time for everything to be coherent as a whole. I want to express my gratitude to my supervisor, Dr. Fred Szabo, for his guidance, time-investment and ceaseless effort in helping me make sense of all the unlinked portions. I am also grateful for his unwavering commitment to my research topic, and for his inclination towards technology that allowed me to properly conclude most sections of this thesis.

I want to thank the Department of Mathematics and Statistics for having accepted me into the Master's program, confident that my dance-inspired thesis would prove mathematically challenging and acceptable. I also want to thank:

- Dr. Harald Proppe and Dr. Cheryl Dudek, for their interest in accepting a position on my examining committee;
- Dr. Ron Stern, for believing in my potential and for providing some financial support in my participation at the Connecting Women in Mathematics across Canada workshop in Banff, Alberta in July 2005;
- Dr. Chris Cummins and Dr. Galia Dafni, for allowing me to do NSERC summer research projects with them, subsequently developing my strong research and analytical skills;
- Ann-Marie for having always been available to answer my questions.

Furthermore, I extend my deep thanks to the Natural Sciences and Engineering Research Council of Canada for awarding me a Postgraduate Scholarship, without which I would not have had the opportunity to pursue my graduate studies.

I reserve a special thank you to Keith Spriggs, my secondary 4 and 5 mathematics teacher, whose passion for math and teaching I will never forget. It was him who first showed me the "deep" links mathematics has to other disciplines - in his case, gardening and country music.

Lastly, but definitely not least, I want to thank my family and friends for their love, support, encouragement, and enthusiasm throughout my Master's program. To my best friends Nadia and Melissa, I wish you best of luck with your own thesis/studies. To my boyfriend Denis, thank you for your love, support and contributions throughout these challenging years. And to my dearest sister Carla, thank you for being open to all sorts of dance discussions with me. I want to thank my father for instilling in me, at a young age, the "undisputed fact" that $2 + 2 = 4$; I can now present you numerous counter-examples! And to my mom, thank you for registering me into dance classes at the shy, awkward and innocent age of five. As with life in general, my dance experiences were not always easy-going, but my growing love for this art taught me patience and perseverance in attaining my goals.

I dedicate this thesis to you, Mommy, whose hard work in life and with your own thesis motivated me to keep going even when the going got tough. I may not have received the mathematics genes from you, but I definitely received all the other good stuff. Thank you, I love you.

The mathematician's best work is art,
a high perfect art,
as daring as the most secret dreams of imagination,
clear and limpid.
Mathematical genius and artistic genius touch one another.

– Gosta Mittag-Leffler [20]

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$\mathbf{Z}$	set of integers	[22]
$\mathbf{Z} \oplus \mathbf{Z}$	the direct sum of $\mathbf{Z}$ and $\mathbf{Z}$	[11]
$\mathbf{R}$	set of real numbers	[22]
$\mathbf{R}^n$	n –dimensional space	[11]
D_n	dihedral group of order $2n$	p. 8
$\langle \mathbf{R}_{\frac{360}{n}} \rangle$	cyclic rotation group	p. 9
ρ	symmetry of the square: $90°$ rotation	p. 9
ϕ	symmetry of the square: reflection across a horizontal axis	p. 9
I	initial position matrix	p. 42
M_i	movement matrix	p. 42
M	$= \sum_{i=1}^{n} M_i$, the sequence of movement matrix	p. 45
C	$= M + I$, the choreography matrix	p. 45

CHAPTER 1

Introduction

It is widely known and often stated that mathematics is required in almost any field of study. Although it is not so apparent at first, the arts is one domain that involves powerful mathematical concepts. In this thesis, we further investigate the connections between mathematics and the art of dance.

There currently exists very little research connecting dance to mathematics. What could be found relates Group Theory to contradancing and squaredancing formations, and is briefly discussed. However, the focus of this thesis is on notation, specifically of body movements, although a small chapter relates to group formations. On the whole, it consists mostly of an innovative contribution to the dance community using mathematics as dance notation.

1.1 Mathematics in Art

Specialists in the dance community have, at various times, suggested a connection between science and dance. For example, the great choreographer Merce Cunningham states, "The dance is an art in space and time. The object of the dancer is to obliterate that" [17, p. 37]. He explains that a good dancer should, among other things, disregard the apparent fact that weight and gravity exist. Although other interesting points have been raised, most of these connections have been at a philosophical level. But to a certain extent, philosophy and mathematics are intimately linked. Plato argued that mathematics consists of abstract objects that exist separately from the time and space in which we live, and thus move [22]. Since dance is simply a creative combination of everyday body movements, generally performed to some form of musical composition that strings together a given time sequence, dance notation is thus nothing more than movement notation. It is therefore possible to strip a choreography of all its creative elements and be left with the movements in their purest form, which can then be examined at a mathematical level. However, no mathematician ever undertook the task of concretely examining the mathematical aspects evident in dance and movement notation.

The goal of this thesis is precisely to undertake such a challenge. As a motivating example, consider the recording of movement on paper, popularly called dance notation. By definition, "Dance notation is the translation of four-dimensional movements (time being the fourth dimension) into signs written on two-dimensional paper" [14, p. xiv]. From the start, this definition involves mathematical ideas, including affine transformations, vectors and coordinate spaces. By delving further, as this thesis intends to do, one will notice that the art and notation of dance are strongly connected to mathematics.

The purpose of the research is to bring a scientific approach to an art discipline, where a connection to mathematics has been suggested by members of the dance community, yet never fully investigated from a mathematician's perspective. When enlightened by this fact, we hope that artists can learn to be less afraid of the mathematics they perhaps once feared, and more aware of the concepts they use every day. For their part, mathematicians can better appreciate both the natural and man-made art and view it in new light. Furthermore, establishing such an intimate link between mathematics and dance will hopefully attract more women into a predominantly male area of study. Ultimately, this thesis will document the importance of mathematics as a multidisciplinary skill.

1.2 The Art of Notation

Our research concentrates on the role of notation in the art of dance. An existing method is discussed and analysed, and then new methods using only mathematics are presented. For the moment, we compare the influence of good notation in the advance of modern dance with the impact of appropriate notation on the advance of mathematics.

In mathematics, there exist various number systems, each created independently by different civilizations, for different and also similar reasons. As a starting example, consider the ordinal numbers. Today's society generally works in base 10. This is the method taught in schools from a young age, and we continue to view the number system as such throughout our life. However, counting in base $5, 20$ or 60 is also feasible. In fact, the vigesimal system was used by Mayans and Aztecs and its influence is present in the choice of names for numbers in French, such as "quatre-vingts" (four-twenty) for 80. For its part, the sexagesimal system was employed by the Babylonians, as well as Greek and Arab astronomers; but most importantly, it is still beneficial to us today when telling time or calculating angles [4]. This illustrates how although working in base 60 has proven itself more difficult for simple, everyday tasks, the reason astronomers chose this method of counting was its ultimate importance for their discipline.

As another example, we use the Arabic numerals

$$0, 1, 2, 3, \ldots, 10, 11, \ldots, 100, 101, \ldots$$

and the decimal notation to represent numbers. Our ability to calculate with ease strongly depends on this notation, such that the addition

$$\begin{array}{r} 45 \\ +\ 17 \\ \hline 62 \end{array}$$

is easily carried out. On the other hand, calculating the sum of 45 and 17 in Roman numerals,

$$\begin{array}{r} \text{XLV} \\ +\ \text{XVII} \\ \hline \text{LXII} \end{array}$$

involves notational gymnastics and is not built on structural principles. Similarly, the Egyptians also had their own numeral system, where summing 45 and 17 would look like

```
     |||||∩∩∩∩
+       ||||||∩
---------------
    ||∩∩∩∩∩∩
```

One reason for the failure of the Romans to make significant contributions to mathematics is that their numerical notation did not provide a framework for defining computational procedures. The Egyptians, however, went a bit further than the Romans when creating their symbols, and also created one for fractions. In fact, their method of writing fractions as $\frac{1}{x}$ is still employed worldwide today [4], even if the rest of their system has been discarded.

Lastly, consider the evolution of binary numbers, first presented by the Indian mathematician Pangola circa 3 BC. The modern system was fully described by Gottfried Wilhelm Leibniz in 1676, George Boole presented what is now known as Boolean Algebra in 1854 [26], and conversion between binary and decimal numbers became possible thanks to Georg Brander [4]. Nevertheless, this system remained rather unpopular until the twentieth century, following advancements in technology. The use of binary numbers in digital circuit designs and relay-based computer systems were presented by Claude Shannon and George Stibitz, respectively, in 1937 [26]. These inventions led to a cycle of research and development in the field of computer science and engineering which is still ongoing today, illustrating how some discoveries can be beneficial in other disciplines as well. Relating back to developments in mathematics, Euler's use of binary numbers in simplifying calculations led him to a counter-example for one of Fermat's conjectures. We now know that not every number of the form $F_n = 2^k + 1$, with $k = 2^n$, is prime, since for $n = 5$, we have $F_n = 2^{32} + 1$ which is divisible by $1 + 2^7 + 2^9$ [4].

The different developments presented for mathematical notation parallel those involved in dance notation. As early as the mid-15th century, dance performers and teachers took interest in recording their choreography on paper. These included not only classical dances (ballets), but also court dances and English Country-Dances (contredanses) [15]. As a result, various dance notation systems eventually emerged, each providing something of direct use and interest to the inventor. While some of the earlier systems used words or word abbreviations, and others used track drawings, newer systems were visual (or stick figure), music note or abstract symbol systems [14]. Innovators of the three former models are Margaret Morris [3], Raoul Auger Feuillet and Arthur Saint-Léon, respectively [9].

The two systems most widely used today are the Benesh stick figure system and the Laban abstract symbol system. The Benesh system uses a matrix to represent the body. This is actually a part of a horizontal music staff, and the position of the body is represented by stick figures [14]. Laban's system, on the other hand, is more mathematically appealing due to the choice of the abstract symbols. This system can be used to represent any movement performed by the body, which leads to the idea of the fundamental aspect of mathematics in performing everyday actions.

It is important to emphasize the reason behind the reduced use of most dance notation systems. Notation systems are created because their inventors want to keep a record of their choreographies for future dancers to reproduce; therefore, the systems are designed in such a way that benefits their inventors directly. However, only a few of them – those that are beneficial to everyone in the field – become universally accepted and remain as such for years to come. Most methods are either too difficult to write or understand; therefore, the need for their creation is not being properly met. They are slowly discarded until a few remain, mirroring the survival of some mathematical notations over others, as discussed above.

The Laban system itself appears overwhelmingly complex and the lack of a shorthand renders it slower to write; however, it is very visual and thus more legible, and the high amount of detail each symbol contains is what advantages this system over others. Furthermore, the modern version of Laban's system of dance notation has been developed and refined from an analysis of movement conducted worldwide, as opposed to being the product of only one person's needs, adding to its universal appeal [15].

Another possible reason for the success of Labanotation is its indirect use of mathematics, which can be considered as a universal language. An interesting point to raise is that the set up of Labanotation, fully explained in Chapter 3, is analogous to the computational set up of the Abacus, where individual columns hold unique information. While the Abacus was useful at the time it was designed, advancements in technology led to the creation of newer tools, such as the pocket calculator, which eliminated the need for the Abacus. On the other hand, Labanotation remains to this day a powerful method of dance notation, proving that the column set up is advantageous given the proper discipline. Nevertheless, we might ask ourselves whether Labanotation survives because no one has attempted to invent a newer, better method of dance notation, or because anyone who tried has failed.

Currently, Laban's system of dance notation is extremely beneficial in its own field. However, the case may have been different if the mathematical ideas had not existed prior to the system's creation. The premise upon which this thesis is built is that appropriate notation was a major factor in advancing science to its modern state. The research shows that the principles of good notation learned from its role in mathematics have had a major influence in the advance of other disciplines. We illustrate this claim by examining the role of notation in choreography and dance.

1.3 Scope of the Thesis

This thesis is divided into seven chapters, and organized in an evolutionary sequence. Each chapter further develops ideas and concepts discussed in previous chapters. Following this introduction is chapter 2, which introduces the mathematics discussed in all subsequent chapters. Chapter 3 focuses on Labanotation and the Euclidean geometry embedded within it.

In chapter 4, we discuss the art of contradancing and establish a possible system of notation using matrices and the concept of homogeneous coordinates. Chapter 5 examines body movements and their notation by considering a four-dimensional coordinate space, where time represents the fourth dimension. We then present an evolved method of dance notation using matrices.

In chapter 6, we choreograph using groups. Specifically, we create symbols to represent basic steps of a few ballroom dances, and briefly discuss their relation to dihedral groups. Afterwards, we apply the crystallographic groups (also known as wallpaper patterns) to these symbols, in the hopes of generating feasible dance displacements. The resulting floor patterns are presented and analysed in this chapter. The entirety of this thesis is finally concluded in chapter 7.

CHAPTER 2

Mathematical Preliminaries

This chapter presents all the necessary mathematical tools relevant to discussions in subsequent chapters. Although most of the following content can be found in any higher-level undergraduate texts on Linear Algebra, Abstract Algebra, Group Theory and Plane Symmetries, we cite from [6], [11], [23] and [25].

2.1 Dihedral Groups and Permutation Groups

Definition *Consider any regular polygon with n sides, $n \geq 3$. Then the corresponding group, called the **dihedral group of order** $2n$, is denoted D_n. This group is also often called the **group of symmetries of a regular** n**-gon**.*

Definition *A **permutation** of a set A is a function from A to A that is both one-to-one and onto.*

Definition *A **permutation group** of a set A is a set of permutations of A that forms a group under function composition.*

Example *Recall the dihedral group D_4. Associate each motion in D_4 with the permutation of the locations of each of the four corners of a square. These eight permutations, called **symmetries of a square**, are:*

R_0 = *Rotation of* $0°$
R_{90} = *Rotation of* $90°$ *counterclockwise*
R_{180} = *Rotation of* $180°$
R_{270} = *Rotation of* $270°$
H = *Reflection about a horizontal axis*
V = *Reflection about a vertical axis*
D = *Reflection about the main diagonal*
D' = *Reflection about the other diagonal*

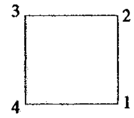
We find that only two elements are required in order to generate the entire *group. These elements are the* $90°$ *rotation* R_{90}, *called* ρ, *and the reflection* H *across a horizontal axis, called* ϕ, *where:*

$$\rho = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{bmatrix} = (1234)$$

and

$$\phi = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{bmatrix} = (12)(34)$$

for the square labeled as such:



2.2 Symmetry Groups and Affine Transformations

Definition *The **symmetry group** of a plane figure* F *is the set of all symmetries of* F.

Definition *A **cyclic rotation group of order** n, denoted* $< \mathbf{R}_{\frac{360}{n}} >$, *is a symmetry group consisting of the rotational symmetries of* $0°, \frac{360°}{n}, \frac{2(360°)}{n}, \ldots, \frac{(n-1)360°}{n}$ *and no other symmetries.*

Definition *An **isometry** of* n*–dimensional space* $\mathbf{R}^n$ *is a function from* $\mathbf{R}^n$ *onto* $\mathbf{R}^n$ *that preserves distance.*

Hence, for any function T from $\mathbf{R}^n$ onto $\mathbf{R}^n$, if the distance from $T(p)$ to $T(q)$ is the same as the distance from p to q $\forall$ pairs of points $p, q \in \mathbf{R}^n$, then the function T is called an isometry. All isometries in $\mathbf{R}^2$ can be classified into one of four types:

1. Rotation
2. Translation
3. Reflection
4. Glide-reflection

Definition *A **translation** is a function that carries all points the same distance in the same direction.*

Definition *A **reflection** across a line L is the transformation that leaves every point of L fixed and takes every point Q not on L to the point Q' such that L is the perpendicular bisector of the line segment joining Q and Q'.*

Definition *The line L is called the **axis of reflection**.*

Definition *A **glide-reflection** is the product of a translation and a reflection across the line containing the translation vector.*

Example *Figure 2.1 illustrates an example of a glide-reflection:*

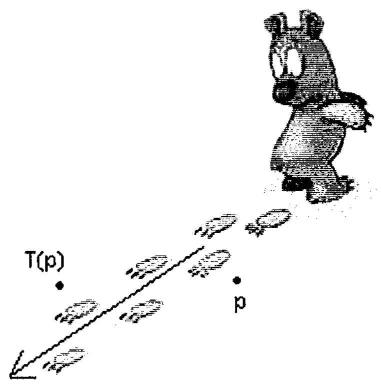


Figure 2.1 Successive footprints in wet sand

2.3 Frieze Groups and Crystallographic Groups

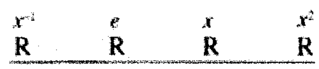
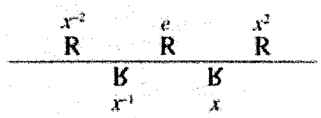
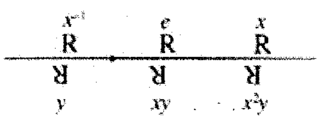
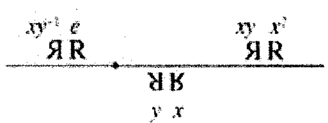
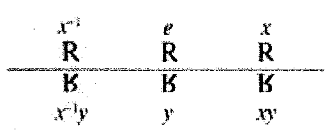
There exist two types of infinite symmetry groups that arise from periodic designs in a plane. They are:

1. the **discrete frieze groups**, the symmetry groups of patterns in $\mathbf{R}^2$ whose subgroup of translations is isomorphic to $\mathbf{Z}$, and
2. the **plane crystallographic groups**, the discrete symmetry groups of patterns in

$\mathbf{R}^2$ whose subgroups of translations are isomorphic to $\mathbf{Z} \oplus \mathbf{Z}$.

There exist seven Frieze patterns, listed below along with their groups of symmetries. These patterns leave designs invariant under all multiples of just one translation.

Table 2.1 The Seven Frieze Patterns

Pattern	Generated by:	Isomorphic to:
	x = translation	**Z**
	x = glide-reflection	**Z**
	x = translation y = vertical reflection	$\mathbf{D}_\infty$
	x = translation y = rotation of 180°	$\mathbf{D}_\infty$
	x = glide-reflection y = rotation of 180°	$\mathbf{D}_\infty$
	x = translation y = horizontal reflection	$\mathbf{Z} \oplus \mathbf{Z}_2$
	x = translation y = horizontal reflection z = vertical reflection	$\mathbf{D}_\infty \oplus \mathbf{Z}_2$

The type of infinite symmetry group relevant to this thesis, however, are the crystallographic groups, also known as the wallpaper groups. These patterns arise from infinitely repeating designs in a plane, and they are invariant under linear combinations of two linearly independent translations [11]. There exist a total of seventeen 2-dimensional wallpaper patterns, named and classified according to the geometrical transformations used to generate them. Five contain only translations and rotations. The remaining twelve patterns contain opposite isometries. Their names include an m for mirror reflections and a g for glide reflections, as listed below [6]:

Table 2.2 The Seventeen Crystallographic Patterns

Pattern	Generated by:
$p1$	2 translations
$p2$	2 translations, 180° rotation
$p3$	120° rotations about 2 different points
$p4$	180° and 90° rotations
$p6$	180° and 120° rotations
$g1$	glide reflection
pm	2 reflections in parallel, 1 translation
pg	3 parallel glide reflections
pmm	reflections in 4 sides of a rectangle
pgg	2 perpendicular glide reflections
pmg	reflection and glide in perpendicular axes, translation not parallel to glide
cm	reflection and glide in parallel
cmm	2 perpendicular reflections, 180° rotation
$p4g$	reflection, 90° rotation
$p6m$	3 reflections in sides of $30-60-90$ triangle
$p31m$	3 reflections in sides of equilateral triangle
$p3m1$	reflection, 120° rotation

2.4 Matrices and Homogeneous Coordinates

Matrix multiplication is a necessary tool in creating affine transformations of two-dimensional images. The idea is to embed these images in $\mathbf{R}^3$ in order to use matrix multiplication in $\mathbf{R}^3$, before converting the images back into $\mathbf{R}^2$.

Definition *The corresponding coordinates* $(x,y,1) \in \mathbf{R}^3$ *of* $(x,y) \in \mathbf{R}^2$ *are called* ***homogeneous coordinates***.

Suppose we want to translate a point (x,y) to $(x+h, y+k)$. We would apply the 3×3

matrix

$$A = \begin{bmatrix} 1 & 0 & h \\ 0 & 1 & k \\ 0 & 0 & 1 \end{bmatrix}$$

to the homogeneous coordinates $(x, y, 1)$. The resulting position $(x+h, y+k, 1)$ would also be in homogeneous coordinates, but it is easily observed that the new position can be obtained by removing the 1 and converting back to $\mathbf{R}^2$.

Now suppose that, after translation, we want to reflect the point $(x+h, y+k)$ about the x –axis. We would apply the augmented matrix

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

to the homogeneous coordinates $(x+h, y+k, 1)$. After conversion, we would have the new position vector $(x+h, -(y+k)) \in \mathbf{R}^2$.

CHAPTER 3

Labanotation and Affine Geometry

In this chapter, we present Rudolf von Laban's system of dance notation, also called Labanotation. After defining the majority of the symbols used to record movements with this system, we discuss the elements of Euclidean geometry embedded within it. We also explain Laban's associated system for recording the displacement of dancers on a stage, and analyse the possible Affine transformations present in these floor plans.

3.1 The Laban System

As with math and dance notations, the Laban system has undergone many changes. During the Second World War, isolation between countries created differences between the European Kinetography established at the Kinetographisches Institute in Essen and the American Labanotation developed at the Dance Notation Bureau in New York, USA. After the war, the International Council of Kinetography Laban tried to unify the systems to eliminate most of the differences, which it succeeded in doing [18]. Nonetheless, some minor variations exist, and this paper focuses on the American Labanotation.

Rudolf von Laban (1879 – 1958) was, among other things, a ballet master and movement theorist [1]. He first presented his notation in *Schrifttanz* (*Written Dance*), published in Vienna in 1928. The English and French editions of the book appeared two years later. From the start, Laban credited the inventors of systems from which he had been inspired, notably Raoul Feuillet's track drawing system itself inspired from Pierre Beauchamp and possibly even André Lorin. The Beauchamp-Feuillet system first published in 1700 in *Chorégraphie ou l'art de décrire la danse* [15] included numerous principles from which Laban built his notation. The two principles of most interest are the central line which divides the movements of the right and left sides of the body, and the use of specific directional signs to indicate to the dancer how to move. The other two principles are the partitions by bar-strokes along the central line to indicate a metrical time division, and the use of special stress signs to indicate basic body actions [18].

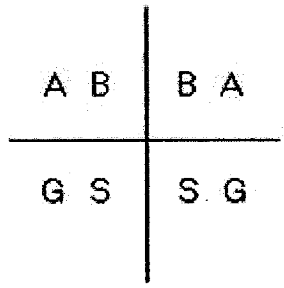


Figure 3.1 The cross staff

The key difference between the systems is that Beauchamp and Feuillet wrote their notation along a floor pattern line, and restricted the notation to steps and leg gestures. Instead, the Laban notation is recorded along a vertical staff, and it includes the upper body movements as well [18]. Before deciding on the vertical staff, however, Laban designed a cross staff, written and read in a horizontal progression (Figure 3.1). The letters A, B, G and S indicate where the arm and body movements and leg and support gestures were written, respectively. The right side represents the right side of the body and the left represents the left side [15]. Eventually, one of his pupils, a German dancer by the name of Kurt Jooss, suggested a vertical division for both the upper and lower body parts, thus inspiring the vertical staff known today [1]. This staff is composed of eleven columns. The first ten are symmetric columns representing the left and right sides of the body, and the right-most column indicates the head movements, as shown in Figure 3.2. In actuality, only the three solid lines are present on the staff [15,16].

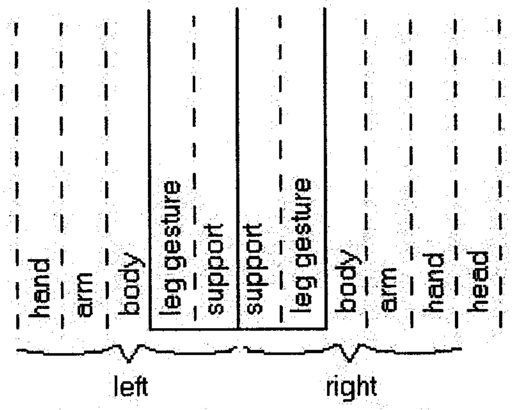


Figure 3.2 Laban's modern vertical staff

The main advantage of this vertical staff is that the lateral symmetry of the body is visually apparent. Furthermore, the continuous flow of the movements is more evident due to the lack of breaks between symbols, especially when movements overlap. This same visual advantage exists with the direction symbols [15]. These symbols are symmetric and represent eight basic directions illustrated in Figure 3.3. Starting from the top left and moving clockwise, they are: forward (on the left or right side), right forward diagonal, right side, right back diagonal, back (on the right or left side), left back diagonal, left side, and left forward diagonal. Finally, a complete rectangle represents the "in place" direction [16].

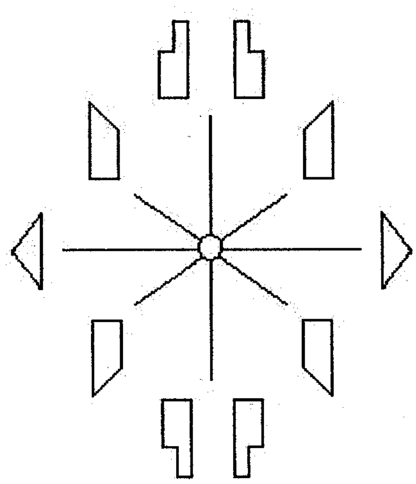


Figure 3.3 Direction symbols

These basic directions are used for all body parts, and are shaded accordingly to indicate at what level the movement occurs. For example, the symbols in Figure 3.4 represent a low, middle and high in place movement, respectively [14]. It is important to note, however, that these directions and levels are relative to the body parts being moved. For example, with respect to steps in the support columns, "low" refers to a bend in the knee and "high" implies being on half toes, whereas "middle" is the normal placement of the foot on the ground. For arm and leg gestures, "middle" refers to the level of the point of attachment of the body to either the arm or the leg – that is, the shoulder and hip joints, respectively. Furthermore, "forward" implies that the extremity of the arm (the hand) or the leg (the foot) is brought in front of the body [15].

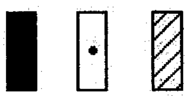


Figure 3.4 Levels

Another advantage of the Laban system is therefore its compactness. Just one symbol can tell the reader what body part is moving (depending on the column in which it is placed), to where it is moving (direction and level), as well as when to start the movement and how long it should last [15]. The duration and rhythm part is achieved through the ticking of the central line of the staff, where each tick indicates one count and a bar line marks the end of the measure [16]. The amount of space covered by one symbol illustrates for how long the movements should last. The breaks between, or overlaps of different symbols, specify whether they should be performed in a discontinuous or continuous manner [14]. Furthermore, if a series of movements should not overlap but must appear as one flowing movement, phrasing bows such as ( ) for the left and right sides are used around the movements in question [16].

In addition to the direction and level of a move, a contraction or extension at the elbow or the knee can also define arm and leg gestures and the degree of knee bend while the legs are used as support. Furthermore, lengths of steps and positions of the feet can also be modified. The two basic symbols used to express any such modifications are an X and a reversed N [16]. Using the hand as an example, the former indicates grasping and the latter stretching [18]. The length of a step and the position of the feet are more specifically determined with four symbols. These are shown in Table 3.1, with a brief description of what they mean. Similarly, Table 3.2 describes the six degrees of contraction and two degrees of extension of the arms and the legs, as well as the six degrees of knee bends while the legs support [16].

Table 3.1 Symbols Modifying Steps and Feet Positions

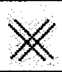
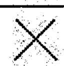
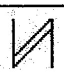
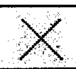
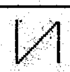
Symbol	Length of Step Description	Position of Feet Description
	very short	very narrow
	short	narrow
	long	wide
	very long	very wide

Table 3.2 Degrees of Contraction and Extension

Symbol	Degree of Contraction	Degree of Extension	Arm and Leg Description / Position	Knee Bend Description
	1		slightly bent	slightly bent
	2		more rounded	more bent
	3		right angle at the elbow/knee	typical demi-plié
	4		slightly past 90°	bent with heels off the floor
	5		more contracted	typical grand-plié
	6		hand/foot touches shoulder/hip	squat
		1	more extended than normal	N/A
		2	hyper-extension (rarely used)	N/A

Specification symbols such as parallelograms and pins also exist to express pivot turns, revolutions on a straight path, as well as curved paths. Pivot turns, or rotations around one's own vertical axis, are indicated with the use of parallelograms. These are drawn vertically in the appropriate support columns, with the slanted edges at the top and at the bottom. The tip of the slant indicates the direction of the turn, as shown in Figure 3.5.

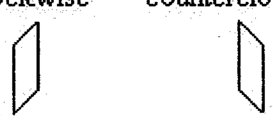


Figure 3.5 Parallelograms of rotation

Pins are then added inside the parallelograms to tell the performer how much of a rotation must be completed. There exist eight different degrees of rotation, with the position of the pins being relative to the direction of the turn. Figure 3.6 indicates the pins and their degree of rotation within a wheel.

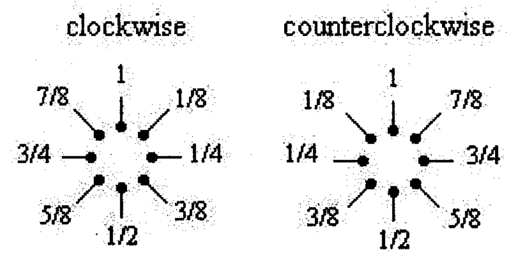


Figure 3.6 Pins indicating degrees of rotation

If more than one rotation is to be completed, the smallest whole number of rotations is indicated with an actual number, and a pin is added to specify the remaining degree of rotation. Figure 3.7 illustrates a 2.5 counterclockwise rotation [16].

Figure 3.7 A 2.5 counterclockwise rotation

These same parallelograms and pins are used to represent curved paths and revolutions on straight paths (Figure 3.8). For example, if a sequence of movements is being performed along a curved path, a vertical line is placed to the right of the staff. As with the parallelogram, edges at a slant at the top and at the bottom indicate the direction of the turn. A pin is then added in the break of the vertical line to show how much of a rotation to complete. The center of rotation of these curved paths is either to the right or to the left of the dancer, depending if the curve is clockwise or counterclockwise, and whether the dancer is walking forwards or backwards. If a revolution occurs on a straight path, the same vertical line is placed to the right of the staff. However, perpendicular edges instead of slanted ones are added at the top and bottom. This combination is called a way sign. The parallelogram and the pin are then added inside the break of the vertical line to indicate the direction and degree of the revolution completed along the straight path [16].

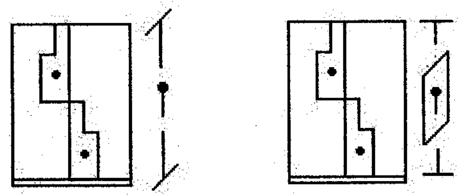


Figure 3.8 Different rotations possible

3.1.1 Other Important Symbols

Symbols that should be mentioned in this context, in addition to those used to identify different body parts, include touching bows, and *return to normal*, *space hold*, *body hold*, *hold weight* and *repeat* signs. These signs will not be discussed in a mathematical concept, but are included at this point because of their use in the final section of this chapter. Nonetheless, some of the mathematics involved with each sign is relatively obvious.

The touching bows are vertical brackets that link two body parts together, indicating touching between them. An X can be placed either centered on the bow or closer to one end to specify whether both body parts are active in the touching, or if only one is acting and the other is passive [18]. Examples include two feet touching during an aerial movement [16], or one hand gripping the other [18]. These bows are allowed to cross the staff if necessary.

The *return to normal* sign is a circle with a dot inside ⊙. Depending in which column and with which body part symbol it is placed, it indicates that the respective body part should re-assume its normal position – normal implying the state that the body part naturally takes. The *return to normal* sign is thus often used to cancel *space hold*, *body hold* and *hold weight* signs. The *space hold* sign is a small diamond ◇ placed after a movement. It indicates that until a new movement is performed within that column or a *return to normal* sign is encountered, then the position should be held with respect to the space it occupies. The *body hold* sign is simply an empty circle ○ used in the same manner as the *space hold* sign, but it indicates that the position is being held with respect to the body. An example is when a dancer has his arm stretched frontward and then pivots 180° counterclockwise. If a *space hold* sign appeared after the raising of the arm, then the arm would finish in back of the dancer. If the *body hold* sign were used instead, then the arm would remain in front of the body as the dancer pivots. The *hold weight* sign is also a small, empty circle. It is placed in the support columns to indicate that the weight must be held on the body part doing the supporting. Only movement performed by the body part supporting the weight can cancel the effect of the *hold weight* sign [16].

Repeat signs in their basic form are a slanted line with two dots, one above the line to the left, and the other below the line to the right, resembling the percentage sign %. They indicate that a movement is to be repeated exactly. If the sign has two parallel slanted lines instead of one, it is called laterally symmetric, and implies that the exact movement is to be repeated to the opposite side. These signs can be centered within the staff crossing the central line to indicate that a count, measure or several measures are to be repeated, in which case the sign is drawn continuously bigger. To specify the measure to be repeated, the number of that measure replaces the bottom dot [16].

The *repeat* signs can also be used for short or long sectional repeats. In such cases, the sign is placed outside of the staff. To indicate the parts to be repeated, one sign is placed to the left of where it begins, and another is placed at the right of the staff to show where the repetition ends. When repeating measures that have already been repeated, the repeat signs are placed further left and right. If some parts are to be repeated more than once, then the top dot of the initial sign and the bottom dot of the end sign are replaced by the number of repetitions to be made. To indicate that the steps of another performer are being repeated, then the letter of the said performer replaces the top dot [16].

Lastly, for precise positioning of every part of the body, symbols have been created to identify each such part. The diagram in Figure 3.9 illustrates these symbols [15].

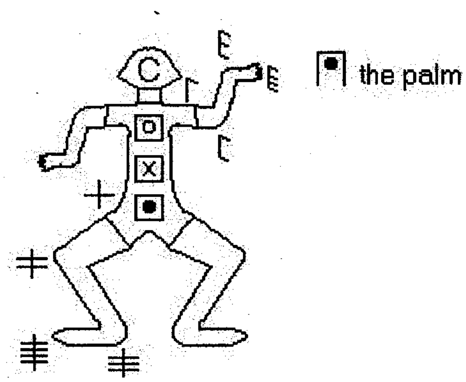


Figure 3.9 Symbols for body parts

3.2 The Mathematics of Labanotation

When reading through the description of the various symbols used in Labanotation, there are many instances where the mathematics involved is very clear. Indeed, possibly every element contains at least one mathematical idea. More interesting is the fact that often, this inadvertently leads us to connect mathematics to other non-Labanotation concepts.

3.2.1 Symmetry and Geometric Shapes

Let us begin with the vertical staff. Of its eleven columns, ten are set up in a very symmetrical manner. The main purpose, as described earlier, is to imitate the human body's own set up and clearly distinguish the movements of its right and left sides. Hence, the mathematical concept of symmetry visible in the staff leads us to consider the symmetry of the human body. Furthermore, Labanotation is a method used to record movement on paper. Any movement that the human body performs can also be recorded in this manner, without restrictions to dance steps. This forces the idea that perhaps some common, everyday movements are also performed symmetrically; hence, mathematically.

Next, consider the choice of the direction symbols. These include a triangle, a tetragon and a hexagon – all of which are geometric shapes. However, it is not only the symbols themselves that are directly related to mathematics. The idea of symmetry is also present here. By observing the symbols, we can assume that there are actually five directions: forwards, diagonal forward, side, diagonal backward and backwards. These become ten directions since each can be performed either to the right or to the left.

3.2.2 Coordinate Systems

Now consider a person standing upright. Let the sideways direction represent the x-axis and the forward/backward direction be the y-axis. It is thus evident that the direction symbols represent movement on an xy-plane, where the axes are as defined above. One can delve further into the mathematics and consider instead an xyz-coordinate system, where the z-axis is the body's own vertical axis. This enables the idea of levels in the movements. Recall that high, middle and low movements are relative to the body part moving. For example, the middle level for the arm would be shoulder-height, whereas middle level for the leg would be at the height of the hip. Hence, the point of origin for the xyz-plane is not fixed, as is the case with the xy-plane. Rather, the point of origin shifts upwards or downwards, depending on the body part being moved. For example, the origin of the xyz-coordinate system for a hand movement would be the wrist.

3.2.3 Angles

Another important mathematical concept involved in Labanotation is that of angles. The most prominent elements of Labanotation with this feature are, in particular, the degrees of contraction of the arms and the legs. As described in Table 3.2, there exist six degrees of contraction, each equivalent to a certain angle formed at the elbow or the knee. For example, a contraction of degree 3 is geometrically equivalent to a right angle. Hence, degrees 1 and 2 represent obtuse angles and degrees 4 and 5 acute angles. Degree 6, with the hand/foot touching the shoulder/hip, theoretically represents a $0°$ angle. The $180°$ angle is the normal extension of the arm/leg.

Just as an example, consider a person standing upright. Let his right arm be in a degree 3 contraction at position right, middle level. Since the right arm is moving, the point of origin for the xyz –coordinate system is the right shoulder. The upper part of the arm is thus along the x –axis, and the forearm is parallel to the y –axis. Now imagine that there is a line connecting the shoulder to the tip of the fingers, creating a right triangle. If measurements of the arm were known, a mathematician would be intrigued in using trigonometry to determine the angle of inclination between the shoulder and the fingers.

The concept of angles is also exploited with the choice of pins to indicate the fraction of a rotation completed in any type of turn. The diagram used to present the pins and the fact that they are used to represent degrees of turns lead us to consider angles within a circle. Let us first consider the pins established for the counterclockwise direction. Their directions are directly related to the angles of a unit circle in the xy –plane. If we let one rotation be equivalent to 2π radians (R) on the unit circle, then we obtain the correspondence listed in Table 3.3 below.

Table 3.3 Angles of Rotation

Rotation	Angle (R)
$\frac{1}{8}$	$\frac{\pi}{4}$
$\frac{1}{4}$	$\frac{\pi}{2}$
$\frac{3}{8}$	$\frac{3\pi}{4}$
$\frac{1}{2}$	π
$\frac{5}{8}$	$\frac{5\pi}{4}$
$\frac{3}{4}$	$\frac{3\pi}{2}$
$\frac{7}{8}$	$\frac{7\pi}{4}$
1	2π

Let us now consider the pins established for the clockwise direction. Recall that for any counterclockwise angle θ, where $0 \leq \theta \leq 2\pi$, its clockwise-equivalent angle is $\theta' = \theta - 2\pi$, where the resulting negative sign indicates that the angle is clockwise. Yet, if we temporarily neglect this negative sign, then the above still holds. After all, in Labanotation, the pins represent a fraction of a turn completed in the indicated direction, and are thus always positive.

3.2.4 Rotations

To continue the discussion of rotations, recall that pivot turns are illustrated with the use of parallelograms. As was the case for the direction symbols, the mere use of a geometric shape implies a mathematical connection. Now let us consider the similarities and differences existing between pivot turns, curved paths, and revolutions on a straight path. We already discussed the difference in their notations, although they all make use of the pins and/or parallelograms. So let us shift our focus to the actual movements being performed. In the case of pivot turns, the dancer stays in place, pivoting about its own vertical axis. This is the z –axis of our imaginary xyz –coordinate system.

In the case of revolutions on a straight path, the dancer again pivots about its own vertical axis; however, because the dancer is also moving in a certain direction, the origin of the xyz –coordinate system is being translated in a continuous fashion as the dancer moves. For example, if the dancer is pivoting clockwise while walking "ahead" on what represents the y –axis, then the origin of the xyz –coordinate system is being shifted along the y –axis to maintain it at the body of the dancer. With curved paths, however, the axis of rotation is dependent on the direction of the turn and on the position of the body. Let us suppose the dancer is simply walking along on a curved path – the axis of rotation is therefore at the center of the circle formed by this walk. Yet, where is the axis relative to the dancer? If the dancer is walking forward clockwise or backward counterclockwise, the axis of rotation is to his right. If the dancer is walking forward counterclockwise or backward clockwise, the axis of rotation is to his left.

3.3 Floor Plans and Affine Transformations

One final aspect to mention regarding any form of dance notation is floor plans. At the time of the Renaissance, movement of the dancers on the ballroom floor was rather simple and generally known, and it was therefore not necessary to record these movements on paper. Eventually, however, more intricate patterns were created and it became a necessity to have these steps recorded. Floor plans are thus simply a record of the area occupied by dancers on a ballroom floor or on a stage. One of the earliest, most widely known floor patterns is Fabritio Caroso's 1600 rose design from *Nobilità de Dame* (see Figure 3.10). It depicts the paths of Dames and Cavaliers as they cross one another. Interestingly, this pattern is titled "The Contrepasso according to the true mathematics after the verses of Ovid" [14].

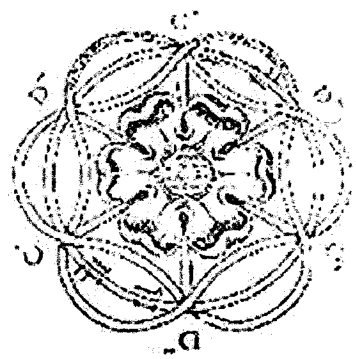


Figure 3.10 Fabritio Caroso's rose design

As with the dance notation systems, different people created different symbols to represent the men and women in the floor plans. Stepanov's method, for example, represented the men with an X and the women with a circle ○. Published in Russia in 1892, this method was much clearer to recognize and easier to draw. In all instances, however, the floor patterns created were generally repeating. Hence, the symbols were just necessary to indicate where the man and woman started, and perhaps also to include their positions at various moments in the dance [14].

The emergence of ballets then led to floor plans being recorded relative to a stage. With Labanotation, the patterns are recorded onto a stage as illustrated and labeled in Figure 3.11 below.

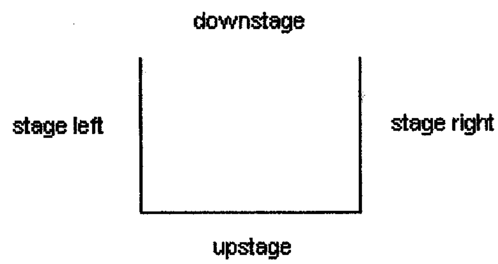


Figure 3.11 The stage

A tack ⊥, which is the symbol used to indicate perpendicular lines in mathematics, represents the starting position of the dancer, regardless if it is a male or female. The horizontal line represents the back of the dancer, and the tip of the vertical line indicates where the dancer is facing. When more than one dancer is present on the stage, letters are placed next to the tack to distinguish one from another. These same letters are placed underneath the central columns of the staff when different dancers perform different steps [16].

To show the movement of the dancers, arrows are drawn from the tack to the desired end location, curving the arrow as necessary to illustrate the path taken. If paths are retraced, two arrows are drawn, with the second line drawn shorter. When more than one dancer is present on the stage and paths cross, the arrow of the dancer passing second is cut at the point of intersection [16]. Figure 3.12 depicts two dancers – one facing the audience, the other facing stage right – whose paths cross, after which the dancer B retraces his steps.

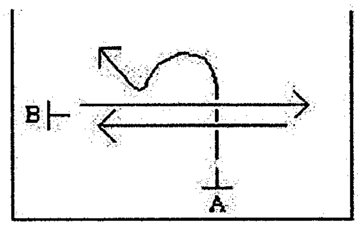


Figure 3.12 Example of a floor plan

Floor plans are drawn for any given set of measures. The range of these measures is then indicated under the plans. For example, if the above floor plan depicts the pattern followed for the first three measures of a given notated dance, then the range 1 – 3 is added underneath it. The floor plans are included next to or underneath the notated steps, near the measures to which they relate [16].

The mathematical concepts involved in floor plans are mostly geometrical. In fact, every pattern illustrated corresponds to a transformation. In all cases, however, it is preferable to disregard the movements performed by the dancer between these transformations, as well as the final position of the dancer. Since we are dealing solely with the floor plans, we should instead assume that the position of the dancer's body is held throughout the transformations.

The most obvious transformation illustrated in any floor plan is a translation, where an object retains its orientation and is simply shifted along any straight line. In general, every floor plan clearly shows a translation. The latter is in fact the basic component of any choreography. Another type of transformation that might be visible in the floor plans is a reflection. Reflections involve shifting an object, currently established in a given set of coordinates, to its relative position on the other side of a given line, called the axis of reflection. Sometimes, choreographers might find it necessary and aesthetically pleasing to have dancers change place along an imaginary line. This line is in fact the axis of reflection. Finally, choreographers also make use of rotations, where an object is shifted in a circular fashion, maintaining its distance from an imaginary point called the axis of rotation. As was discussed in the previous section, these rotations are created when dancers perform steps along a curved path.

3.4 An Example: The Matrix Choreography

In September 2002, I was asked to join the team at Les Ballets Jazz du Québec (Brossard, Québec) and was offered to teach an adolescent-adult class of Jazz 2 level. For the dance school's annual show presented June 28^{th} and 29^{th}, 2003, I choreographed a dance to music from the soundtrack of the motion picture *The Matrix*.

In order to put into practice the concepts of Labanotation and floor plans, I notated a small excerpt of my choreography using Laban's system (Figure 3.13). Also included are floor plans, the first of which corresponds to the notated steps (Figure 3.14). The affine transformations involved in these floor plans are clearly visible, and by performing the steps that are notated, some of the other mathematical concepts discussed earlier also become apparent. Although this is but a small example, it serves its purpose in summing up the various mathematical ideas that have been brought forth in this chapter, and their connection with one of the most known dance notation systems in the world, Labanotation.

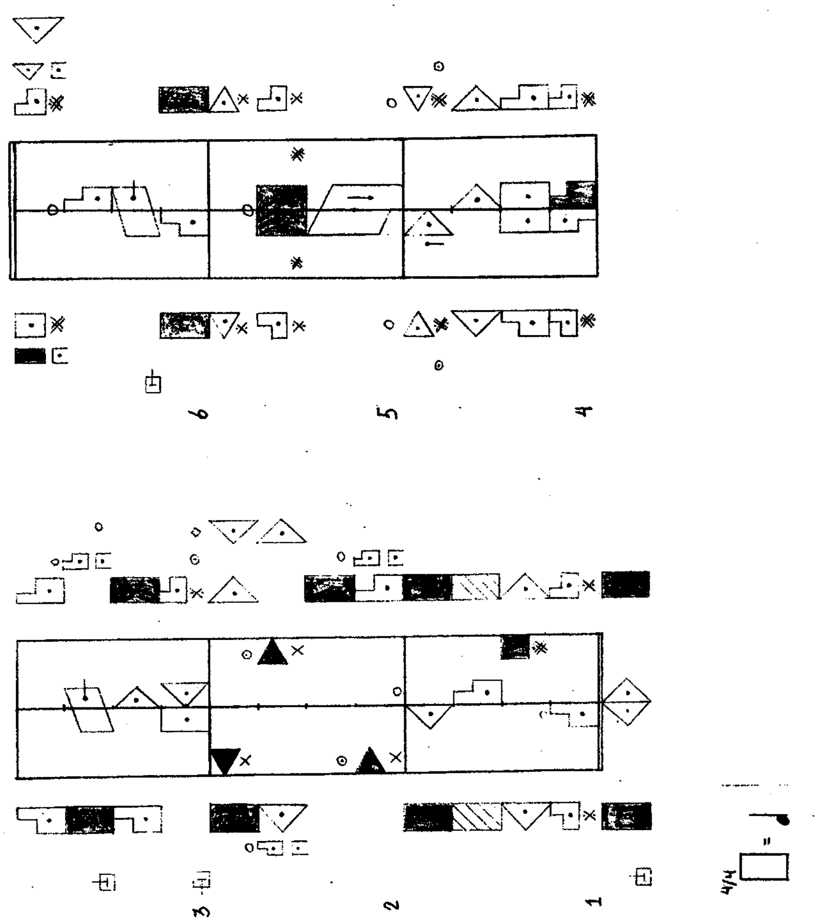


Figure 3.13 Labanotation of measures 1 – 6 for *The Matrix*

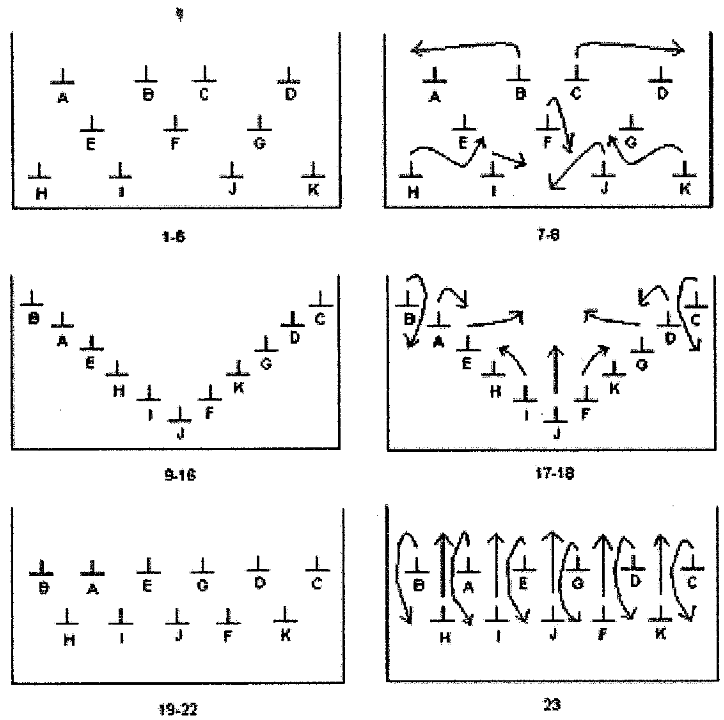


Figure 3.14 Floor plans for measures 1 – 23 of *The Matrix*

The various aspects of Labanotation that have a direct relationship with mathematics include the choice of geometric shapes for various symbols and the symmetry in the set up of the staff, which reflects the symmetry of the human body. Also, there exists the theoretical concept of both an xy –plan and an xyz –coordinate system to understand the movement of different parts of the body in terms of direction and level. Finally, angles are indirectly used to indicate to what extent a body part must be bent, or how much of a rotation must be completed.

These ideas are easily identified in the above notation, particularly the concept of symmetric steps and arm movements. The idea of the xyz –coordinate system also is exploited in measure 1 for the symmetric arm movements. Furthermore, measures 3, 5 and 6 illustrate the use of angles in body rotations, and the degree of contraction for arms and legs is visible throughout.

Also related to Labanotation are floor plans, which in their depiction of the movement of dancers across a stage make use of different affine transformations. In the examples provided, translations and reflections are present, clearly illustrating the intimate link between mathematics and Labanotation.

CHAPTER 4

Contradancing and Homogeneous Coordinates

In this chapter, we begin working on the idea of using matrices for notation by applying the already established concept of matrix multiplication for translations. We encode displacements of partners in an improper formation of contradancing, generalized to a total of c couples, and a working example is provided for a sample of $c = 4$ couples.

4.1 A Brief Description of Contradancing

The term contradancing encompasses various types of folk dances stemming from both the French court dances and the English country-dances. Unarguably, the setup consists of two straight lines of an even number of couples, compared to the square formation of two couples in square-dancing [8]. Two of three possible arrangements for these lines are a proper or improper formation (the Beckett formation will not be discussed). The latter implies an alternating male-female arrangement with partners facing each other, whereas the former consists of all men in one line facing their female partner in the other line [27]. Regardless, each couple will perform a sequence of steps; at the end of the 64 –beat sequence, two neighboring couples will have exchanged places. The sequence is repeated until the first couple reaches the end of the line [8]. Note that in every other set of steps, the two couples at each end dance by themselves and are called "out". The end couples then proceed to dance their way back to their starting position while performing the same sequence of steps [5].

This is all the information necessary for understanding the following section of our chapter; however, a slightly more detailed description of contradancing is available at [24] for those interested. Before proceeding with our trial method of contradance notation, note that formations in contradancing is one area that has been largely studied by mathematicians. Research relating to group theory, specifically symmetries of the square and the dihedral group D_4, can be found in [5], [7] and [19].

4.2 Notation using Homogeneous Coordinates

Let ↑ represent a couple, where the tip of the arrow represents the man. Assume the arrow is 2 units in length.

Now visualize graphically the arrangement of four couples. Arrange each couple at positions $1, 2, 3$ and 4, with the x-axis centered between each man and woman.

Consider an improper formation, such that there are never two men or two women next to each other; arrange each arrow with the tips alternating up/down/up/down. This gives roughly:

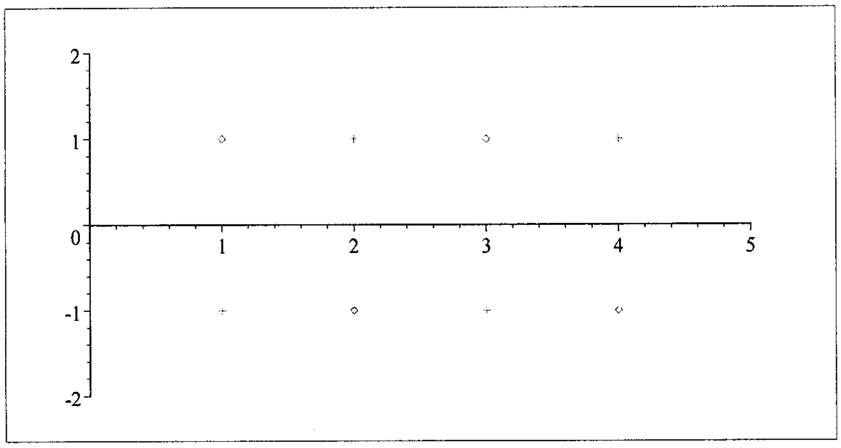


Figure 4.1 Improper formation of 4 couples

(Note that here, the diamond ◇ represents the tip of the arrow (the man) and the + represents the woman.)

Let us now generalize to c couples. We shall use a structure similar to the idea of shape grammar to deduce the matrix representatives of contradancing. The goal is to shift dancers down an imaginary line, although the men and women of a pair also need to be interchanged every other turn. Hence, only two possible steps exist, repeated alternately:

Step #1 : To move all couples at positions with x odd by $+1$, and all couples at positions with x even by -1.

Step #2 : To move all couples at positions with x even, $x < c$, by $+1$, and all couples at positions with x odd, $x > 1$, by -1. Then, reflect along the x-axis all couples at positions $1 < x < c$.

These simple steps are translations, easily expressed in matrix form using a homogeneous coordinate system. In general, recall from section 2.4 that the matrix

$$A = \begin{bmatrix} 1 & 0 & h \\ 0 & 1 & k \\ 0 & 0 & 1 \end{bmatrix}$$

represents the displacement made by the dancer's feet on an xy-plane, and the matrix

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

represents a reflection of the dancer about the x-axis. So for the steps described above, we obtain:

Step #1 :

- $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ for position (x,y) with x odd (more specifically, with $x = 2k+1, k = 0, 1, \ldots, \frac{c}{2} - 1)$

- $\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ for position (x,y) with x even (more specifically, with $x = 2k, k = 1, 2, \ldots, \frac{c}{2})$

Step #2 :

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

for position (x,y) with $x > 1$ odd $(x = 2k+1, k = 1,2,\ldots,\frac{c}{2}-1)$

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

for position (x,y) with $x < c$ even $(x = 2k, k = 1,2,\ldots,\frac{c}{2}-1)$

Multiplying the augmented position vector $(x,y,1)$ of one partner by the appropriate matrix (one of the four above) would result in the partner's new position, as such:

$$\begin{bmatrix} 1 & 0 & \pm 1 \\ 0 & \pm 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x \pm 1 \\ \pm y \\ 1 \end{bmatrix}$$

Is it possible to deduce how many steps it would take for c couples to return to their starting positions? Consider first two examples, where $c = 4$ and $c = 6$:

Example *For $c = 4$ couples:*

ABCD $\xrightarrow{1}$ *BADC* $\xrightarrow{2}$ *BDAC* $\xrightarrow{1}$ *DBCA* $\xrightarrow{2}$ *DCBA* $\xrightarrow{1}$ *CDAB* $\xrightarrow{2}$ *CADB* $\xrightarrow{1}$ *ACBD* $\xrightarrow{2}$ *ABCD*

$\therefore$ *It takes* 8 *steps.*

Example *For $c = 6$ couples:*

ABCDEF $\xrightarrow{1}$ *BADCFE* $\xrightarrow{2}$ *BDAFCE* $\xrightarrow{1}$ *DBFAEC* $\xrightarrow{2}$ *DFBEAC* $\xrightarrow{1}$ *FDEBCA* $\xrightarrow{2}$ *FEDCBA* $\xrightarrow{1}$ *EFCDAB* $\xrightarrow{2}$ *ECFADB* $\xrightarrow{1}$ *CEAFBD* $\xrightarrow{2}$ *CAEBFD* $\xrightarrow{1}$ *ACBEDF* $\xrightarrow{2}$ *ABCDEF*

$\therefore$ *It takes* 12 *steps.*

We can generalize:

Claim *It will take c couples a total of $2c$ steps to return to their starting positions.*

Proof Suppose the c couples are arranged in order $1,2,\ldots,c-1,c$. Note that it takes couple c, $c-1$ steps to get from its starting position to position 1. The couple then stays in this position for 1 step while all couples not at the ends switch. This sums to $(c-1)+1=c$ steps in one direction. The entire process is then repeated in the opposite direction in order to return all the couples to their starting positions; this will take another c steps. Hence, it takes $c+c=2c$ steps in all. □

4.3 A Working Example

Suppose we have $c = 4$ couples. Let us consider couple #1. The man and woman begin at positions $(1,1)$ and $(1,-1)$, respectively. Applying step #1 (for x odd):

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

and

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

In other words, the man is now at position $(2,1)$ and the woman at position $(2,-1)$. Applying step #2 (for $x < 4$ even):

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ -1 \\ 1 \end{bmatrix}$$

and

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix}$$

such that the man is now at position $(3,-1)$ and the woman is at position $(3,1)$. Applying step #1 (again for x odd):

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 3 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \\ 1 \end{bmatrix}$$

and

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \\ 1 \end{bmatrix}$$

with the man ending at position $(4,-1)$ and the woman at position $(4,1)$. Note that we cannot apply step #2 here because $x = 4$. This means that the couple has reached one end of the line and stays in position for one step while the middle couples change place. We now apply step $\#1$, but this time for x even:

$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 4 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ -1 \\ 1 \end{bmatrix}$$

and

$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 4 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix}$$

This brings the man back to $(3,-1)$ and the woman back to $(3,1)$. Applying step $\#2$ (here for $x > 1$ odd), such that

$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 3 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}$$

and

$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$$

leaves the man at $(2,1)$ and the woman at $(2,-1)$.

Applying step $\#1$ for x even, we get:

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

and

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

such that the man is back at its starting position $(1,1)$ and the same for the woman, who is back at $(1,-1)$. Since $x = 1$, we cannot apply step $\#2$ for x odd, such that the couple stays in place while the others move, as required by contradancing rules.

We observe that our mathematics works well, since our couple completed each of the required steps in contradancing (moving along the line up to one end, staying in position, and then coming back all the way with a final stay in position). Furthermore, it took $2(4) = 8$ steps for the couple to complete the routine and finish in their starting position.

CHAPTER 5

Matrices and Dance Notation

One main goal of this thesis pertains to devising an encoding process for general body movements using mathematical tools directly. In this chapter, we define such a process using matrices. Note that for simplicity, we are considering only basic body parts, and restricting movements.

5.1 Body Matrix

First picture the human body, and recall the analysis of Labanotation discussed in Chapter 3. Suppose that each basic body part (head, right arm, left arm, torso, right leg, left leg) has its own coordinate system, with the origin centered at the joint where the body part attaches to the body. In all cases, the x –axis goes from left to right, the y –axis goes from back to front, and the z –axis goes upwards. Assume that the initial position is standing legs straight, arms down on either side, looking straight ahead. (This is the anatomical position of the body, but with the palms of the hands facing each other instead of forwards.)

Now consider a 4×6 matrix, where each column represents the six basic body parts listed above. The first three rows represent movement along the x,y and z axis, with the coordinate system as defined previously. Similarly to Labanotation, a negative in the z direction implies bending in the knees for the legs, a bending of the torso, and so forth.

For simplicity, we are restricting all possible entries in the first three rows to the set $\{-2,-1,0,1,2\}$. The negatives mean the movement is made "backwards" along the axis, the 0 represents no movement, and the positives are movements made "forwards". If a movement is made from the origin, then it can only move 1 in any direction. But if a certain body part has already moved, say down 1, then it can either return to its starting position by moving 1 unit, or go to its opposing position, in this case up, by moving 2 units. The fourth row represents time t, where the entries are positive integers indicating how long the movement for its respective column should take to complete. If a body part is motionless during this time, the entry in that column is 0.

In the same manner as with homogeneous coordinates, we now raise this matrix into a 5×7 matrix. All extra entries are 0, except for the corner, which instead of being 1 is the maximum of all the numbers entered in the time row. (This is explained in detail in section 5.4.)

Hence, we have the movement matrix:

$$M_i = \begin{bmatrix} h_x & ra_x & la_x & to_x & rl_x & ll_x & 0 \\ h_y & ra_y & la_y & to_y & rl_y & ll_y & 0 \\ h_z & ra_z & la_z & to_z & rl_z & ll_z & 0 \\ h_t & ra_t & la_t & to_t & rl_t & ll_t & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & t_{\max} \end{bmatrix}$$

with

$$t_{\max} = \max\{h_t, ra_t, la_t, to_t, rl_t, ll_t\}.$$

For the initial position described above, we define the initial position matrix

$$I = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

5.2 Examples

1. Move your head down to look at the ground in 1 beat while lifting both your arms up towards the sky in 2 beats. This is represented by the matrix

$$M_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & 2 & 0 & 0 & 0 & 0 \\ 1 & 2 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

2. Afterwards, move your arms straight out on both sides while jumping with both legs opening also on both sides, everything in 1 beat. This is:

$$M_2 = \begin{bmatrix} 0 & 1 & -1 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

3. In 1 beat, lift your head back to look in front of you, such that:

$$M_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

4. In 2 beats, twist your whole torso to the left with the head following, the left arm staying left relative to the torso, and the right arm coming in front of the torso (so left of its original position). We get:

$$M_4 = \begin{bmatrix} -1 & -2 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

5. Coming back in 1 beat would simply be the negative of above:

$$M_5 = \begin{bmatrix} 1 & 2 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

6. To complete an 8 –beat sequence, close your left leg to your right leg, while lowering your left arm to its starting position, and lifting your right arm towards the sky. Then:

$$M_6 = \begin{bmatrix} 0 & -1 & 1 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

5.3 Addition for Choreography

So what shall we do with all these matrices? The idea is to sum them. The one resulting matrix would then indicate the final position of the dancer. For example, summing the above matrices gives:

$$M = M_1 + M_2 + M_3 + M_4 + M_5 + M_6$$

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

$$+ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} -1 & -2 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

$$+ \begin{bmatrix} 1 & 2 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 1 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 5 & 7 & 7 & 3 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 8 \end{bmatrix}$$

Based on our definitions, this means that the dancer completes the sequence of movements having moved right 1 unit, with the right arm stretched up to the sky and all other body parts in their initial position.

But what about the 2 at the right-arm / z-axis position? We know that this number is only significant when the body part is coming from an opposite direction. We recall that the initial position of the dancer was with both arms down on either side, as indicated in matrix I. So for our results to be more telling, we must always sum the sequence of movement matrix $M = \sum_{i=1}^{n} M_i$ with the initial position matrix I. The resulting matrix is the choreography matrix C, which properly shows the final position of the dancer. In fact, the matrix C is analogous to the idea of a terminal matrix.

In our example, we thus get:

$$C = M + I$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 5 & 7 & 7 & 3 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 8 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 5 & 7 & 7 & 3 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 8 \end{bmatrix}$$

which is in fact the appropriate final position of the dancer relative to the coordinate axes.

It is important to mention that we defined one possible initial matrix I; however, there exist an infinite number of different starting positions for a choreography. It is thus important to always specify this initial position and to sum the sequence of movements with the appropriate matrix I to obtain the correct choreography matrix.

Now note that although the head and torso are in their initial positions, their nonzero entries in the time row indicates that these body parts were nevertheless moving during this sequence. These entries thus indicate how many beats were required for the given body part to move and return to its initial position. So in our example, the head moved during 5 beats, ending where it began.

5.4 Time Sequence

Now what about the time sequence? According to each step of our example above, we should have an 8 –beat sequence. Notice that this number appears in the final entry of the choreography matrix C, but nowhere in the time row. So what do these numbers actually signify?

First, recall that each entry in the time row indicates the total number of beats the respective body part moved. So for example, the arms moved for a total of 7 beats during the entire sequence, whereas the right leg only moved for 1 beat. Therefore, the reason that the total 8 beats does not appear in this row is precisely because not one body part was moving for the entire sequence, which is an appropriate representation of most choreographies. However, we cannot just sum the entries of the time row either, because there were beats during which different body parts moved at the same time, which is also customary.

This is why we defined the last entry of our matrix M_i as

$$t_{\max} = \max\{h_t, ra_t, la_t, to_t, rl_t, ll_t\}.$$

By using the addition operation, this entry provides us with an effective counter. In fact, the final number indicates how long the entire sequence M requires for completion, and within M we still have the individual times for each body part. Hence in our example above, we had

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 5 & 7 & 7 & 3 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \boxed{8} \end{bmatrix}$$

where the 8 appropriately indicates the total 8 –beat sequence. Note that 8 is not the $t_{\max}$ value of this matrix M; instead, it is the sum of all the $t_{\max}$ values contained in the matrices M_i.

Recall that to obtain the choreography matrix C, we needed to sum the matrix M with the initial position matrix I. Given

$$I = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

we observe that the last entry is 0. Although this is done on purpose, it is not false in any way, since our initial position does not take up any time in the sequence.

So we obtain

$$C = M + I = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 5 & 7 & 7 & 3 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 8 \end{bmatrix}$$

and everything works out as required.

5.5 Special Cases

5.5.1 Holding Positions

Sometimes, a choreographer may decide that the dancer should hold a position for a specific amount of time, be it at the beginning, during or at the end of a performance. In these instances, we must make an exception to the definition provided for our entry $t_{\max} = \max\{h_t, ra_t, la_t, to_t, rl_t, ll_t\}$. Supposing the position is to be held for s beats, then we calculate

$$t_{\max} = s + \max\{h_t, ra_t, la_t, to_t, rl_t, ll_t\}$$

and input instead this value in the matrix.

By observation, one can then see that the entry is greater than the maximum of the time row, deduce that the position is being held, and by simple subtraction, one can calculate for how long this occurs.

As examples, consider:

1. The initial position defined above, to be held for $s = 4$ beats. Then we define:

$$I = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 \end{bmatrix}$$

2. The position in example 4 above, to be held for $s = 8$ beats. Then:

$$M_4 = \begin{bmatrix} -1 & -2 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2+8 \end{bmatrix} = \begin{bmatrix} -1 & -2 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10 \end{bmatrix}$$

5.5.2 Possible Time Overlaps

In choreographies, there are often moments when a body part is moving for numerous beats, while other body parts take turns moving for less beats. For example, we could have both arms taking 6 beats to extend upwards and 2 to come back down; meanwhile, the head could be moving to look down in 2 beats, up in 2 beats, and back to its starting position in 2 beats, followed by the right leg moving right in 1 beat and the left leg moving left in a subsequent beat.

In this case, the arising problem is how to encode this information in a matrix so that:

1. we still have all necessary information (ie: which body part moves, where, and for how long);
2. we know which movements are being performed at the same time;
3. we do not end up with an excess number of total beats for the choreography.

The solution we propose is simple. For any movement performed during the "leftover" beats of the previous matrix, we set $t_{max} = 0$. Only simple observation is then required to notice this difference. For the example provided above, we encode the first set of movements as before, such that:

$$M_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & 2 & 0 & 0 & 0 & 0 \\ 2 & 6 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 6 \end{bmatrix}$$

To encode all subsequent movements that are performed at the same time as the remaining $6 - 2 = 4$ beats, we write:

$$M_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad M_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

We proceed to encode the remaining movements:

$$M_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \quad M_5 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$M_6 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Summing these matrices, we obtain:

$$
\begin{aligned}
M &= M_1 + M_2 + M_3 + M_4 + M_5 + M_6 \\
&= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & 2 & 0 & 0 & 0 & 0 \\ 2 & 6 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 6 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
&+ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \\
&+ \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 8 \end{bmatrix}
\end{aligned}
$$

and assuming that we hold our initial position for 4 beats before beginning our sequence of movements, we calculate the choreography matrix:

$$
\begin{aligned}
C &= M + I \\
&= \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 8 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 2 & 8 & 8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 12 \end{bmatrix}
\end{aligned}
$$

As can be seen, we lose nothing in our final result, which appropriately illustrates the final position of the dancer, as well as the total 12 beats required to complete this sequence of movements.

5.6 Computer Animation

With the advancements in technology, computer animation has become a great interest of today's society. In particular, choreographers and dance notators welcome this form of science in their artistic domain. Various computer animation programs currently exist or are being developed as a means to choreograph and simulate what a tentative dance routine might look like. Most of these programs use Labanotation as their source code, either to facilitate or complement its use [10, 12, 21].

In our case, we decided that it would be interesting to test what the above notation might look like once animated. Included in the Appendix is the beginnings of such a process, provided courtesy of Denis Wong Wong Keet, B. Eng. Computer Engineering. The dance notation is automated using the language OpenGL and the interface of C++. First, the basic OpenGL functions required for viewing (the camera angles) and for the environment (the dance floor and dance box) were set up. Second, the term *dancerBody* was defined to refer to the dancer's body. Finally, by isolating the different body parts of the dancer into separate classes, it is possible to apply different animations to the relevant body parts.

For example, given the matrix

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

we require that just the right arm be raised, in two counts, from its initial position (down on the side) to above the head. Hence, only the rightArm() class would be given new values. Note that the variables utilized in the code signify the angle of rotation for each body part. It is necessary to have three variables for both rotation and translation purposes – one for each axis (x,y,z). These necessary angle and translation variables have been created.

Anyone interested in pursuing the programming can refer to the Appendix for the initial code with comments, frame images, as well as a detailed explanation of what steps must be undertaken next. Overall, what we are trying to show is that the notation presented in this chapter is simple and useful for basic movement notation. Furthermore, it is also possible to generate a computer program in which we input the values of a sequence of matrices, and obtain as output an animation of the dance sequence dictated by this sequence.

CHAPTER 6

Choreographing with Wallpaper Patterns

The main question in this chapter is whether it is possible to take basic ballroom dance steps and generate feasible routines with the same process used to create 2 –dimensional wallpaper patterns. These seventeen crystallographic patterns and the geometric transformations employed to generate them were discussed in section 2.3.

6.1 Choreographing Procedure

We begin by choosing a given number of ballroom dances with simple, yet interesting basic steps, and assign a symbol to each. Each symbol should clearly indicate the starting position, pause position and ending position of the dancer, with an extra identification to distinguish the right foot from the left foot. The pause position refers to the half-way point of the basic step, not always involving an actual pause in time. Usually, this pause implies that the steps will then be performed in "reverse" order, to come back to the starting position. As such, the starting and ending positions are usually the same. (Note that in the cases below, this is true for all the dances except the Tango.) All steps of the right foot are connected and all steps of the left foot are connected. An extra line connects the starting positions of the right and left feet. If a foot steps in place, a perpendicular line is added as indication.

The symbols presented in section 6.3 are kept to scale, to adequately indicate the proportional differences in the steps for each dance. However, each symbol was reduced to a 1 inch $\times 1$ inch square in order to generate the patterns using the Artlandia *SymmetryWorks* add-in for *Adobe Illustrator* 10 [2]. The resulting patterns for $p1, p2, p3, p4$ and $p6$ are shown below and discussed in section 6.5.

6.2 Ballroom Dancing Specifications

It is important to note that the designed symbols are specific to the female. Due to the face-to-face positioning of the couple for performing basic steps in ballroom dancing, the leader (generally the male) steps forward into a sequence, whereas the follower (usually the female) mirrors the footwork and steps back [29]. Furthermore, the female always begins her steps with the right foot. This will be necessary when trying to determine which routines generated can be said feasible under these conditions. In fact, given this information, we can immediately determine that all routines generated by either a mirror reflection m or glide reflection g will not be feasible. These reflections imply that the female dancer will alternate the performance of her basic step between starting with the right foot, and starting with the left foot; however, in ballroom dancing, only the male begins the steps with his left foot, so this switch is impossible.

This implies that already twelve of the seventeen wallpaper patterns – precisely all those containing opposite isometries – cannot generate feasible routines. Given the restrictions of ballroom dancing mentioned above, we now investigate the remaining five patterns – those containing only translations and rotations – for the following dances: Waltz, Tango, Chacha, Samba, and Salsa/Mambo. Note that we are investigating only the steps and neglecting all aspects of time. In this case, the Salsa basic step is exactly like the Mambo basic step, and we henceforth refer to either of them only as the Mambo step.

Nevertheless, these timing differences affect the labeling of the pause. For example, the Chacha has a fake pause (identified as *pause), which is used solely to indicate the "reverse" repetition. In fact, there does not exist any actual pause when performing the Chacha. With the set up and the Samba, there exists a very short pause (identified as ~pause). The Waltz is counted $1-2-3$ on 4 beats/measure, and the Samba is counted $1-\&-2$ on 2 beats/measure, creating a small pause of $\frac{1}{4}$ measure. The Mambo, however, has 3 steps performed to 4 beats/measure, with each step taking one full beat, such that an actual pause occurs before repetition of steps in the opposite direction. The difference between the Salsa and the Mambo is precisely with the positioning of the pause, called a *hold* in ballroom dancing terminology. While the Mambo hold occurs at count 1, the Salsa hold occurs at count 4. Therefore, since the Mambo usually begins on count 2, whereas the Salsa begins on count 1, the latter is called “Salsa On One” or “On One”, and the former, “On Two” [30].

As with dance notation, ballroom dances have also evolved differently. There currently exists two styles of ballroom dancing categories: American and International. Both of these are subdivided to distinguish the soft, slow, flowing dances from the faster, rhythmic Latin dances. The American Style is divided into American Smooth, which includes the Waltz and the Tango, and American Rhythm, which includes the Chacha, Mambo and sometimes the Samba. The loosely-equivalent categories in the International Style are, respectively, Standard and Latin [28]. The symbols designed below and the subsequent discussions pertain specifically to the American Style, taught at the Arthur Murray Ballroom Dance Studios.

6.3 Symbols are to Steps as Patterns are to Dances

6.3.1 Waltz

The Waltz consists of six steps in total. The female steps with the following feet: right, left, right, (~pause), left, right, left. These steps are taken in the following directions: back, left, left, (~pause), front, right, right. The symbol designed to represent the Waltz is:

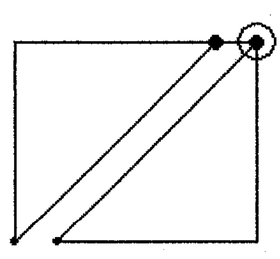


Figure 6.1 The Waltz

The five routines generated with SymmetryWorks are illustrated in Figures 6.2 – 6.6.

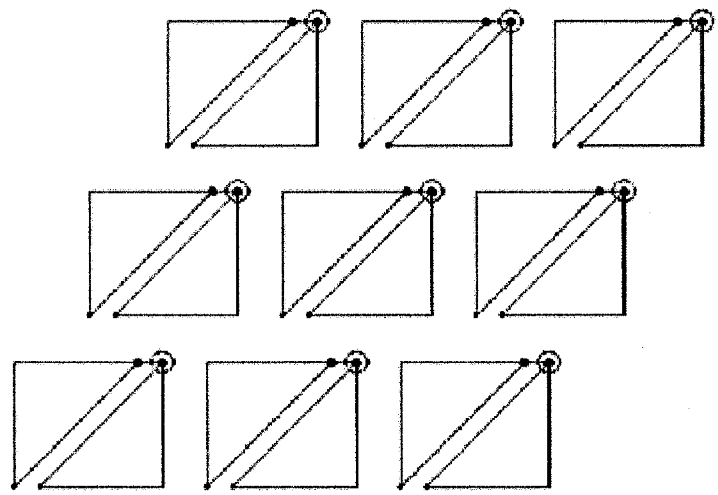


Figure 6.2 Pattern $p1$ generated with the Waltz

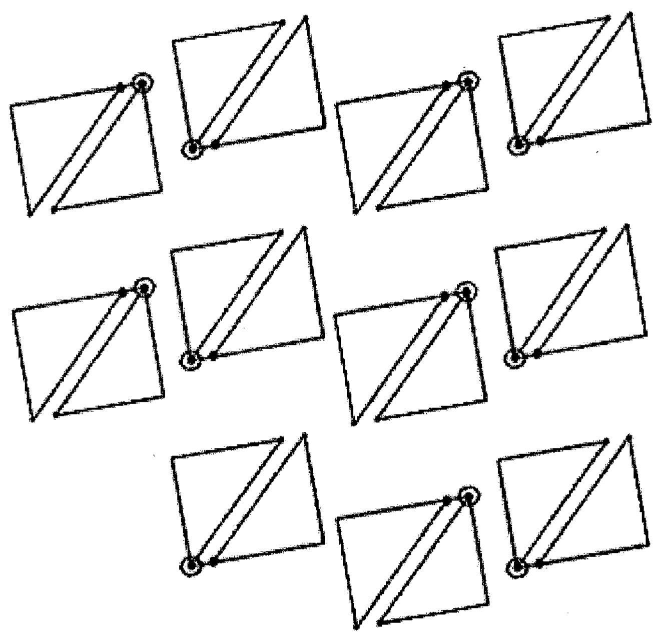


Figure 6.3 Pattern $p2$ generated with the Waltz

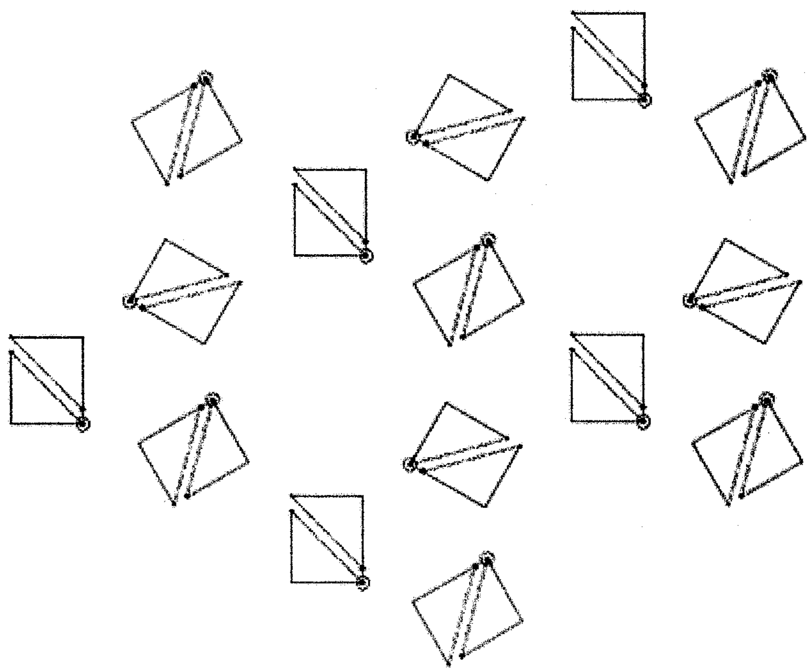


Figure 6.4 Pattern $p3$ generated with the Waltz

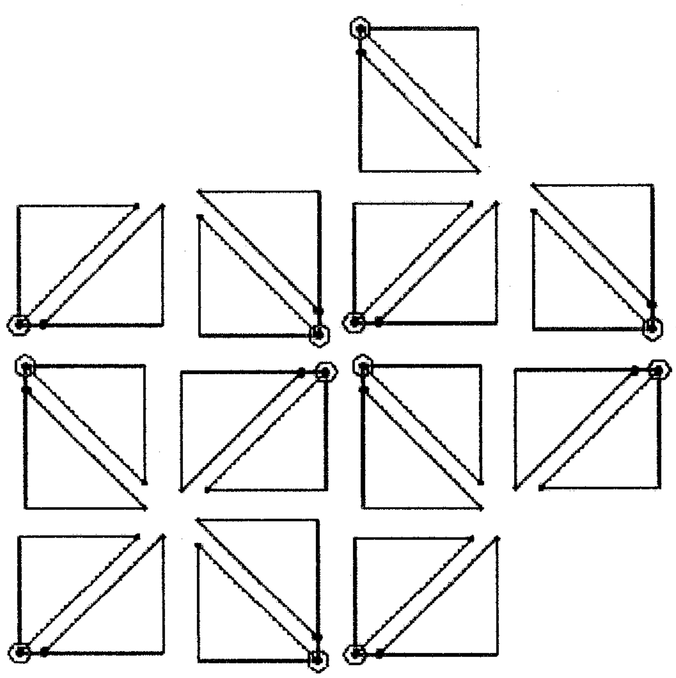


Figure 6.5 Pattern $p4$ generated with the Waltz

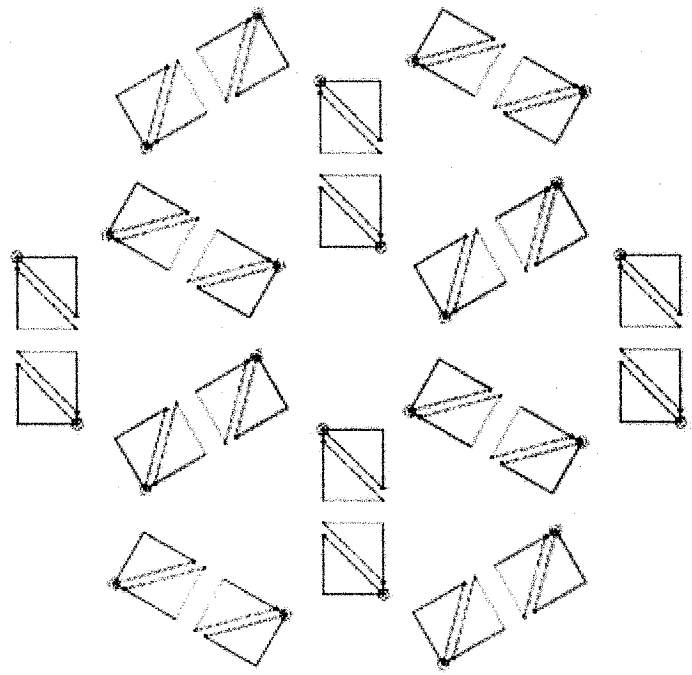


Figure 6.6 Pattern $p6$ generated with the Waltz

6.3.2 Tango

The Tango consists of five steps in total. The female steps with the following feet: right, left, right, left, right. These steps are taken in the following directions: back, back, back, left, left. As mentioned before, the Tango is the only one of the five dances discussed in this chapter that does not have repetition of steps in the opposite direction; hence, the ending position is not the same as the starting position, which is clearly visible on the chosen symbol:

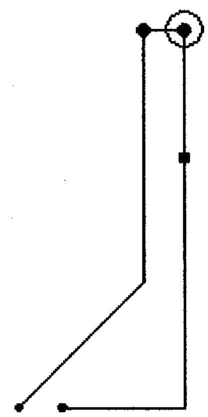


Figure 6.7 The Tango

The five routines generated using SymmetryWorks are illustrated in Figures 6.8 – 6.12.

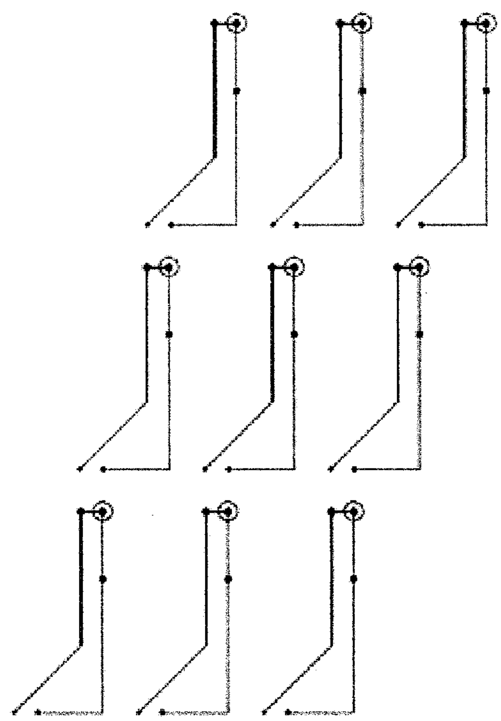


Figure 6.8 Pattern $p1$ generated with the Tango

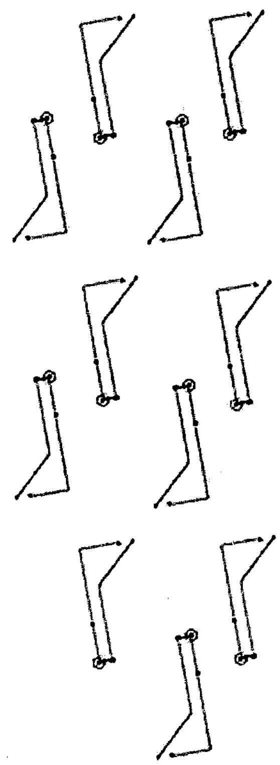


Figure 6.9 Pattern $p2$ generated with the Tango

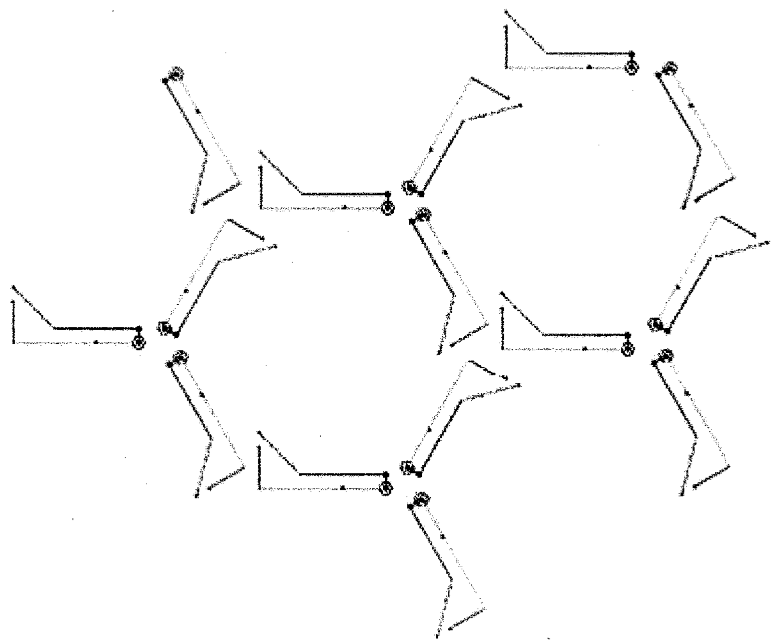


Figure 6.10 Pattern $p3$ generated with the Tango

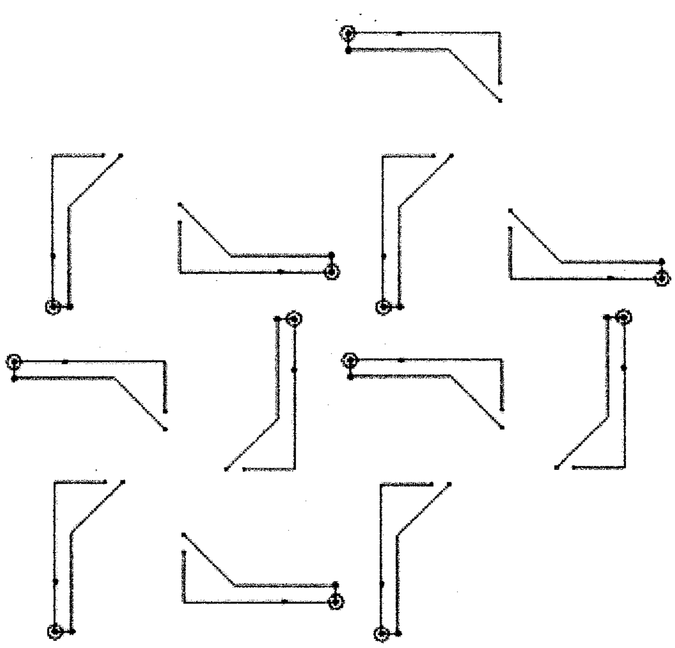


Figure 6.11 Pattern $p4$ generated with the Tango

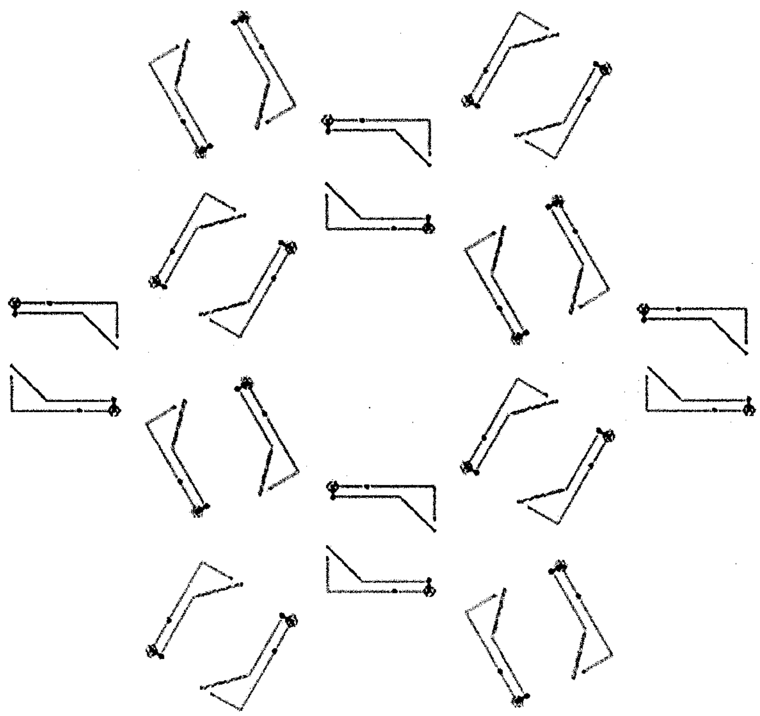


Figure 6.12 Pattern $p6$ generated with the Tango

6.3.3 Chacha

The Chacha basic step has ten steps in total. The female steps with the feet: right, left, right, left, right, (*pause), left, right, left, right, left. These feet step in the following directions: right, front, in place, left, left, (*pause), left, back, in place, right, right. The symbol designed to represent the Chacha is:

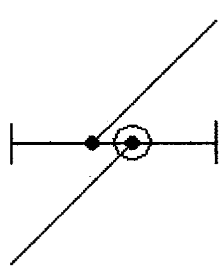


Figure 6.13 The Chacha

The five routines generated with SymmetryWorks are illustrated in Figures 6.14 – 6.18.

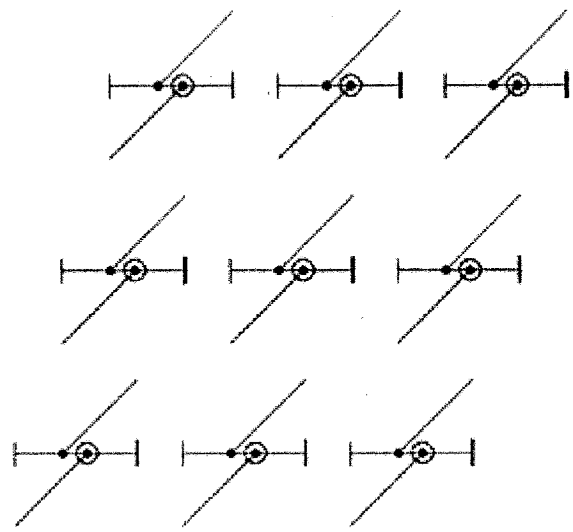


Figure 6.14 Pattern $p1$ generated with the Chacha

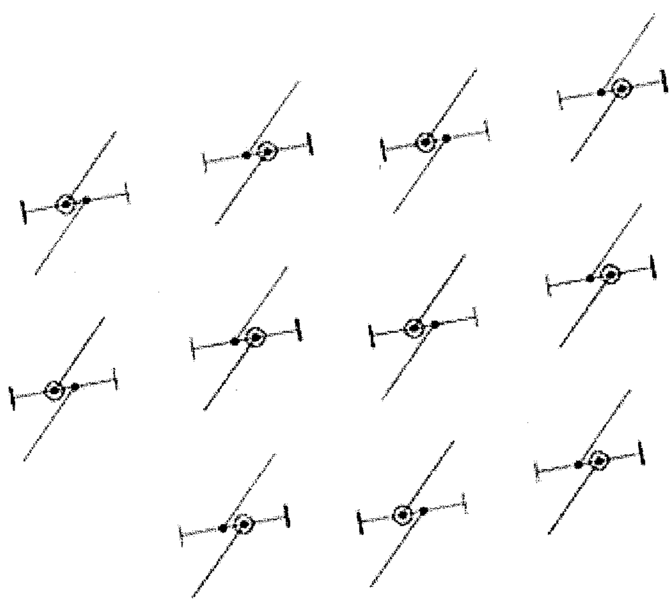


Figure 6.15 Pattern $p2$ generated with the Chacha

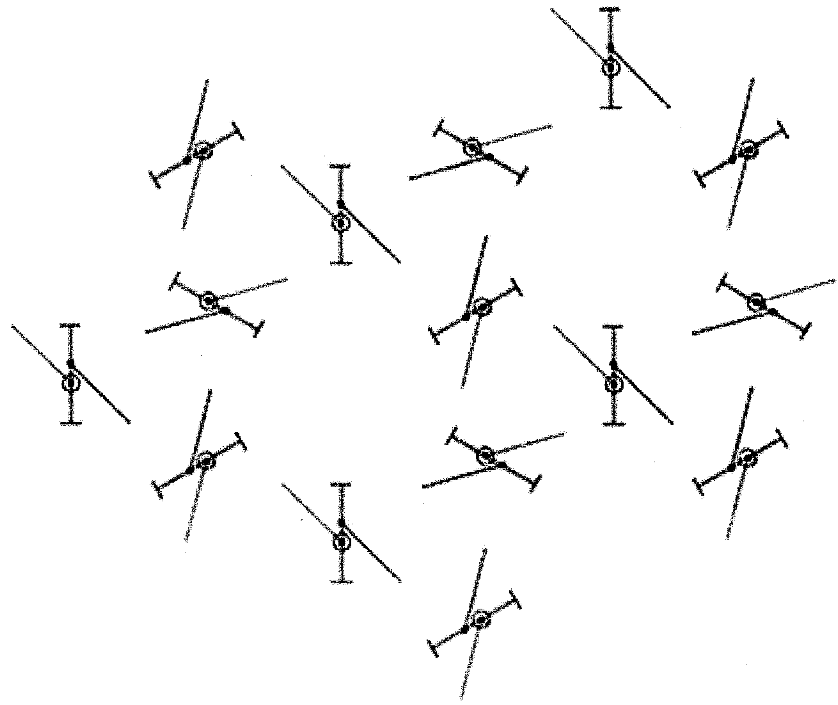


Figure 6.16 Pattern $p3$ generated with the Chacha

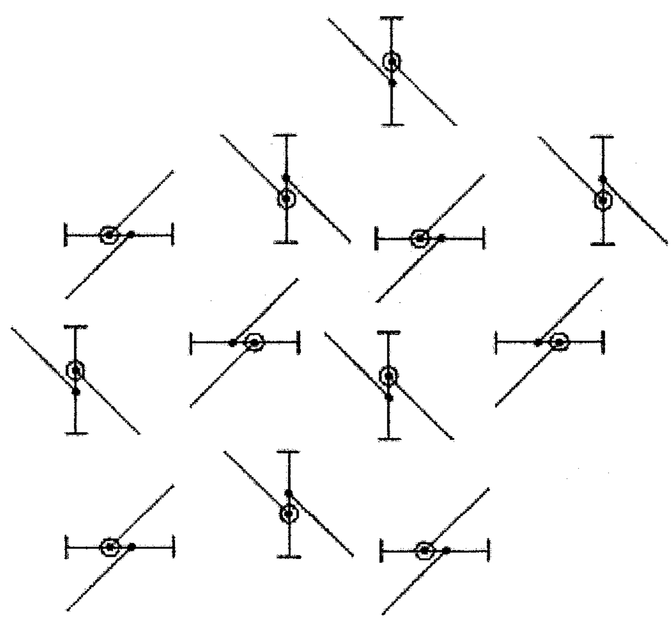


Figure 6.17 Pattern $p4$ generated with the Chacha

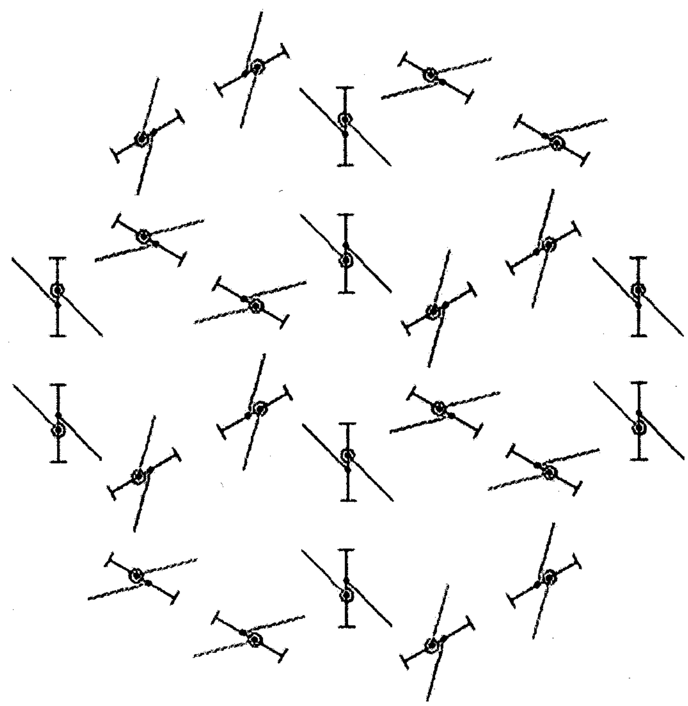


Figure 6.18 Pattern *p6* generated with the Chacha

6.3.4 Samba

There are a total of six steps in the Samba, performed by the following feet: right, left, right, (~pause), left, right, left. The direction of the steps are: back, back, in place, (~pause), front, front, in place. The chosen symbol for the Samba is:

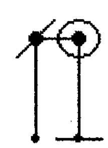


Figure 6.19 The Samba

The five routines generated with SymmetryWorks are illustrated in Figures 6.20 – 6.24.

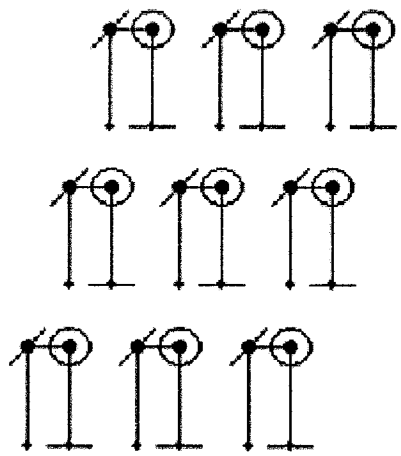


Figure 6.20 Pattern $p1$ generated with the Samba

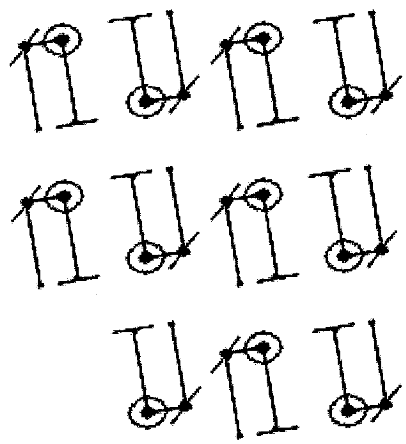


Figure 6.21 Pattern $p2$ generated with the Samba

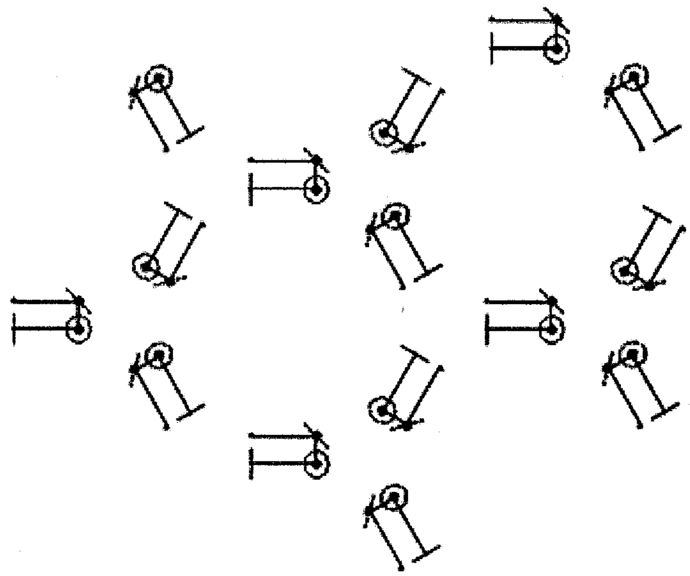


Figure 6.22 Pattern $p3$ generated with the Samba

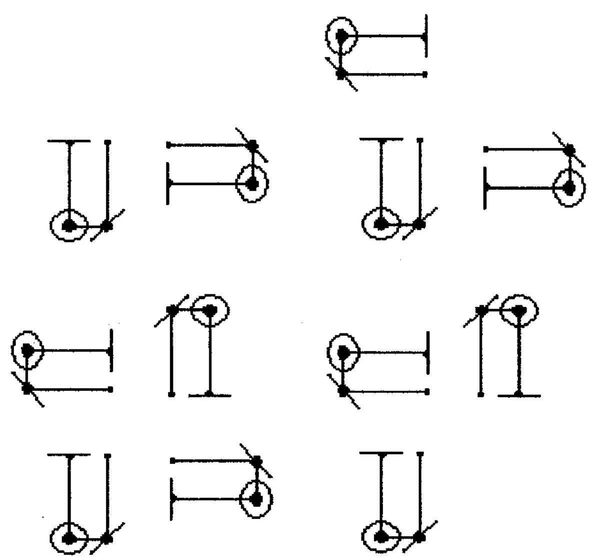


Figure 6.23 Pattern $p4$ generated with the Samba

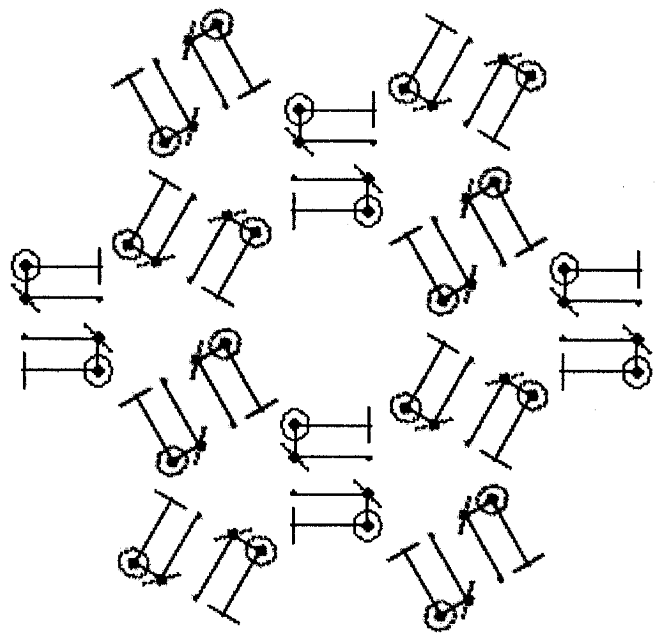


Figure 6.24 Pattern $p6$ generated with the Samba

6.3.5 Mambo

The Mambo consists of a total of six steps. The female steps with her right, left, right, pause, left, right, left feet, in the following directions: back, in place, front, (pause), front, in place, back. Recall that the Mambo steps are exactly like the Salsa steps, except for the timing, which affects the position of the hold (or pause). Regardless, the symbol designed to represent the music-less steps of the Mambo (and Salsa) is:

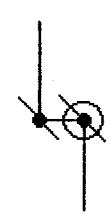


Figure 6.25 The Mambo

The five routines generated with SymmetryWorks are illustrated in Figures 6.26 – 6.30.

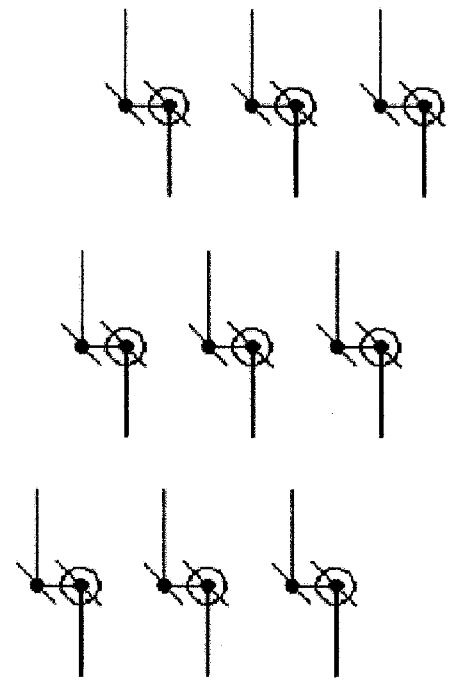


Figure 6.26 Pattern $p1$ generated with the Mambo

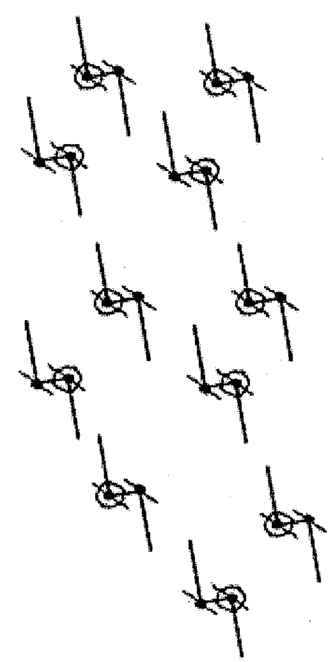


Figure 6.27 Pattern $p2$ generated with the Mambo

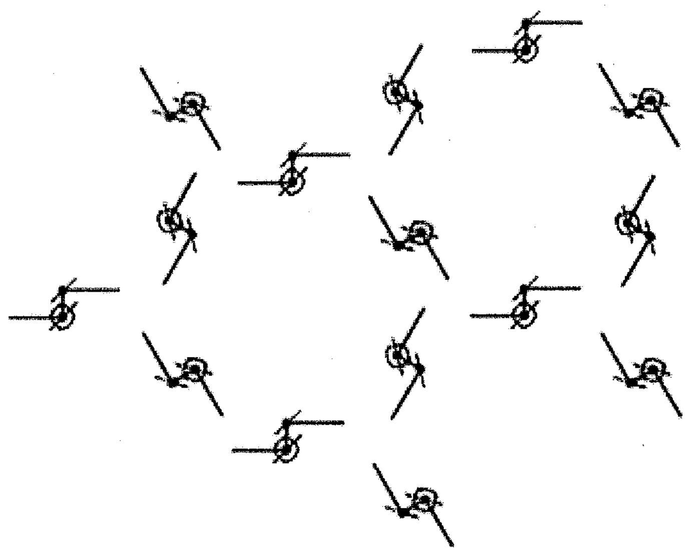


Figure 6.28 Pattern $p3$ generated with the Mambo

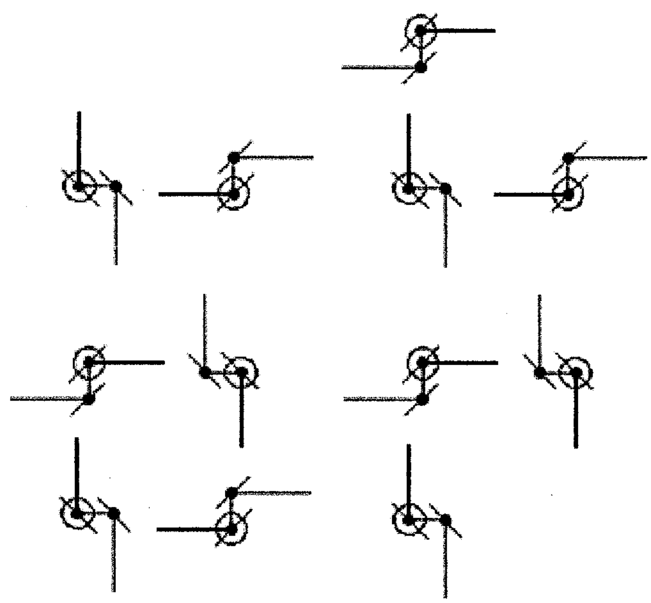


Figure 6.29 Pattern $p4$ generated with the Mambo

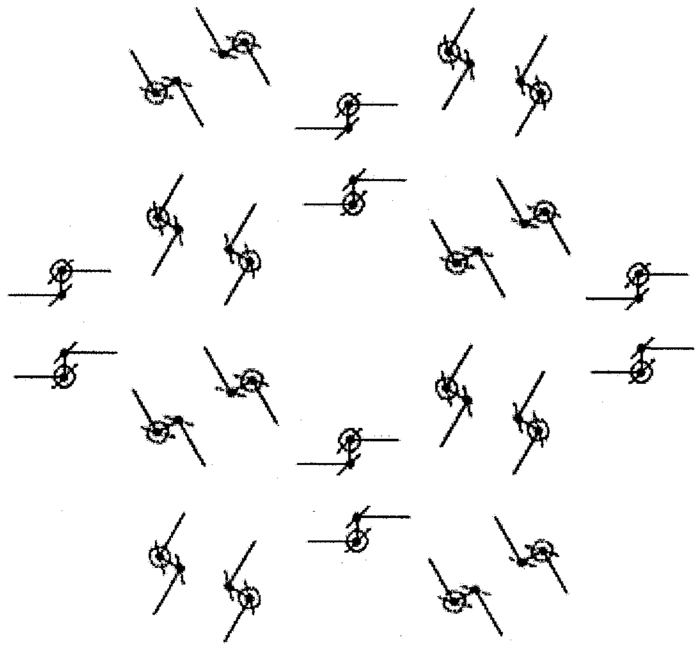


Figure 6.30 Pattern $p6$ generated with the Mambo

6.4 Analysis of Symbols

Symbols were created to define the female basic step of the Waltz, Tango, Chacha, Samba and Mambo. Applying to these symbols the five wallpaper patterns that do not contain opposite isometries subsequently generated the above floor patterns. Of importance now is to analyse each pattern and determine which are feasible given the constraints of the respective dances. In doing so, we discuss the geometric properties present, and whether these can be executed by the dancer as she retains the style of the dance.

Before, however, we shall discuss the separate symbols to determine some of their associations to groups other than the crystallographic patterns. To do so, we omit the circle identifying the initial position of the right foot, and all other extra symbols; instead, we focus our attention on the lines. As a first example, consider rotational symmetries, discussed previously in section 2.2. Unfortunately in our cases, none of the symbols contain more than one rotational symmetry, so none of the dances can be called a cyclic rotation group, regardless of which order. Nevertheless, there are two symbols to which we can apply a $180°$ rotation and they still remain in position. These are the symbols for the Chacha and the Mambo (see Figures $6.31 - 6.32$).

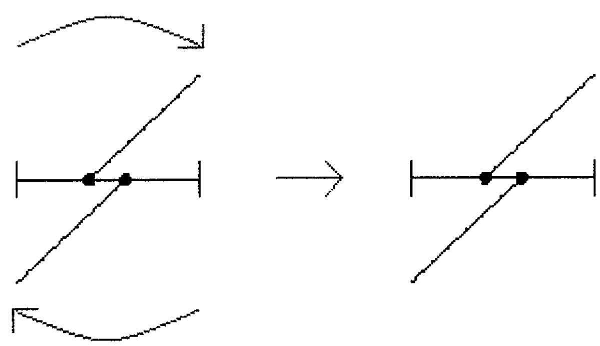


Figure 6.31 R_{180} applied to the Chacha

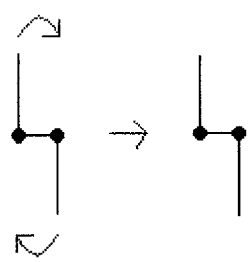


Figure 6.32 R_{180} applied to the Mambo

Relating this back to the eight symmetries of the square discussed in section 2.1, this rotation is called R_{180}, rotation of $180°$. Furthermore, given the specific symbols, this rotation is also equivalent to D', a reflection about the non-main diagonal. Conversely, in the case of the Waltz symbol, the reflection D' is not equivalent to the rotation R_{180}. While applying D' generates the Waltz symbol as is:

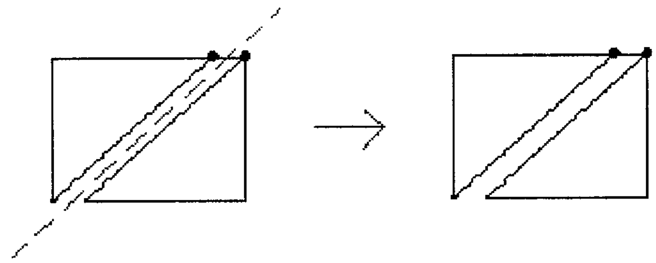


Figure 6.33 D' applied to the Waltz

applying R_{180} creates a small difference, notably in the connection of the initial position (purple circle):

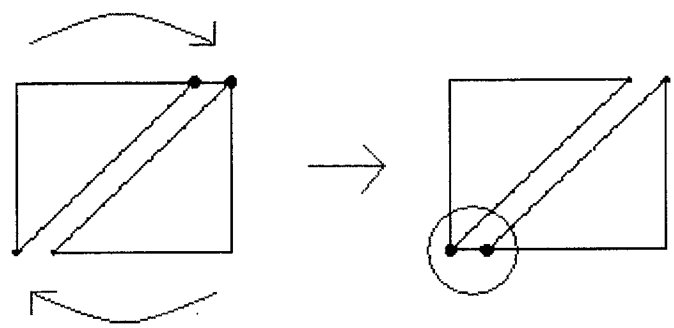


Figure 6.34 R_{180} applied to the Waltz

In the case of the Tango, there does not exist any transformation that can regenerate the symbol as is when applied. We already established that the Tango basic step is the only one of the five discussed to not include repetition in reverse direction; we can thus assume that this directly explains the lack of symmetry in the symbol.

Lastly, consider the symbol for the Samba and imagine it as an open square. We can easily perform a reflection along a vertical axis and still obtain the same symbol:

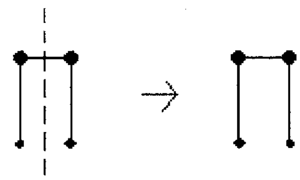


Figure 6.35 V applied to the Samba

This transformation is equivalent to one of the eight symmetries of the square, specifically V, the reflection about a vertical axis. Now consider the Samba symbol rotated 90°, say counterclockwise. The previous symmetry V no longer exists; instead, we have the symmetry H, a reflection about a horizontal axis:

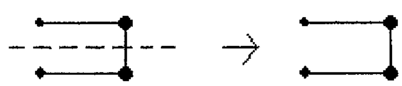


Figure 6.36 H applied to the Samba

Recall that we defined this transformation as ϕ in section 2.1; hence, notice that the Samba, rotated 90°, creates an element of the dihedral group D_4. (In fact, the transformation ϕ generates two of the eight symmetries of the square, R_{180} and H : rotation of 180° and reflection about a horizontal axis, respectively.) Unfortunately, since the Samba symbol is an open square, we cannot apply a 90° rotation ρ and still obtain the same symbol (the difference is visible above), so the Samba does not generate all elements of D_4.

Hence, we conclude that although the dance symbols possess symmetries equivalent to those of the square, the Samba is the only one with potential to create a generator of the dihedral group D_4. We now proceed to analyse the floor patterns generated by applying the crystallographic groups to our symbols.

6.5 Analysis of Patterns

We begin this discussion with the Chacha, since of the five patterns generated, only $p4$ is a feasible floor pattern. Given the basic Chacha step, the female dancer cannot achieve displacements equivalent to a backwards or side translation, a half-turn or a 120° rotation. Hence, the five patterns generated as such are discarded. In $p4$, only the quarter-turn (see the blue arrows in Figure 6.37) is feasible due to the alignment of the right-most steps (pink horizontal line); however, note that this quarter-turn occurs clockwise, opposite the theoretical counterclockwise direction, precisely due to the feet restrictions.

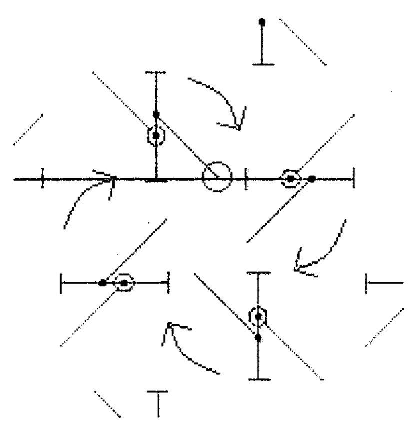


Figure 6.37 Analysis of Chacha $p4$

After the dancer steps forward with her left foot (green circle), she rotates slightly to her right as she completes the in-place step with her right foot, thereby creating the clockwise quarter-turn. The dancer then continues the Chacha step facing a new direction. Note that the quarter-turn established is not an actual Chacha step, although it mimics the half-turn used in the *Sweetheart Chase*. In addition, given the Chacha basic step, this half-turn can also only be performed clockwise.

We now move on to the Tango, since $p2$ and $p3$ are both entirely rejected as feasible floor patterns. Recall that the Tango is the only one of the five dances where the starting and ending positions are not the same. In order for the patterns to be feasible, it is thus a necessity that the start and end positions meet. Unfortunately in $p3$, all the starts are grouped together and all the ends together. Also in the Tango, the amount of space covered by the basic step is the largest among all five dances. The distance covered in trying to displace the movement in $p2$, either backwards or with the half-turn, is hence too big to be performed comfortably. However, the backwards translation in $p1$ is perfectly executable (Figure 6.38). In fact, this displacement is the exact performance of the Tango basic step in real life.

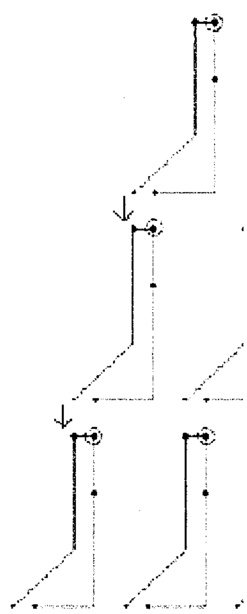


Figure 6.38 Analysis of Tango $p1$

The floor patterns generated by $p4$ and $p6$ are also feasible. In $p4$, neither the quarter-turn p nor the half-turn a can be achieved from the initial position 1 while maintaining the graceful composure of the Tango. Yet it is possible to rotate directly to position pa from 1 (see blue arrow in Figure 6.39) given the position of the feet and the fact that the female dancer must always step back into position.

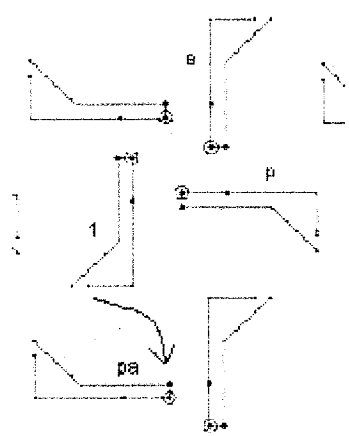


Figure 6.39 Analysis of Tango $p4$

In $p6$, the half-turn is feasible due to the closeness between the starting and ending positions of the feet (see blue arrows in Figure 6.40). Also, this half-turn can only be executed clockwise. A counterclockwise rotation would force the dancer to step forward into the second execution of the basic step, contradicting the fact that she should be stepping backwards. This is the key point in determining the rotation direction required to make a floor pattern feasible. Lastly, there does not exist any feasible 120° rotation in $p6$. As shown in red below, the distance between the start and end positions is too wide for a graceful execution. However, note that it would be possible to dance our way through the inner-circle pattern counterclockwise due to the closeness of the steps (green arrow).

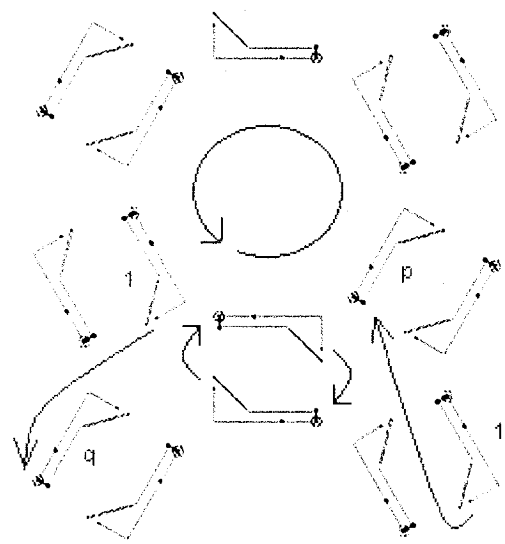


Figure 6.40 Analysis of Tango $p6$

In the previous two dances, there existed unfeasible floor patterns. The constraints of either the Chacha or the Tango rendered the execution of some patterns either uncomfortable or simply not possible. However, for the Waltz, Samba and Mambo, all the patterns generated contain feasible parts. We now discuss each of these patterns.

Due to the diagonal movement already present in the basic step of the Waltz, it is quickly seen how a backwards translation s^{-1} is possible (blue arrows in Figure 6.41). Unfortunately, this left-backwards diagonal movement is also what prevents any rightward translation. In addition, the first step being backwards also prevents any leftward translation.

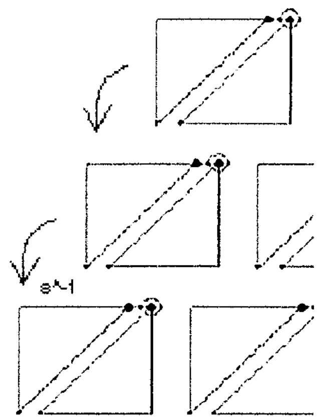


Figure 6.41 Analysis of Waltz $p1$

For $p2$, we already established above that a backwards translation is feasible. The half-turn is also feasible, but only counterclockwise due to the positioning of the feet. Hence, as the blue arrows illustrate in Figure 6.42, a dancer could begin at one end, move backwards, rotate, and then move back "up". At this point, however, she would have no choice but to rotate back into her initial starting position. Moving "right" as shown by the red arrow is not feasible because the displacement step would be too wide, thus removing the ease and grace required of the Waltz performer. It is important to note that the half-turn a discussed is generated by the dancer from position s^{-1}, whereas in theory, it is generated directly from the initial position 1.

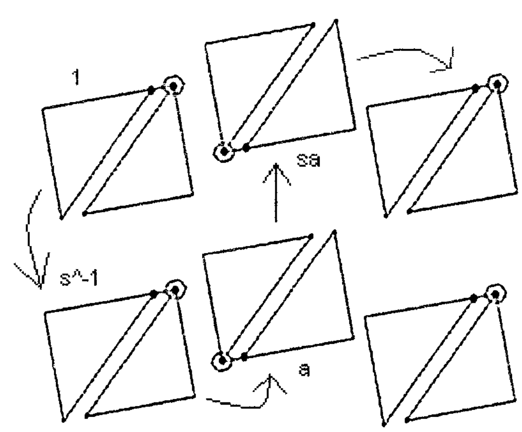


Figure 6.42 Analysis of Waltz $p2$

The pattern generated for $p3$ using the Waltz basic step is also feasible given the positioning of the feet and the required steps. The $120°$ rotation about point p (counterclockwise, facing inwards) (see blue arrow in Figure 6.43) is feasible because as she steps forward with her left foot, the dancer rotates in preparation of the second basic step, with the right foot moving in a right, forwards fashion relative to herself (pink arrow):

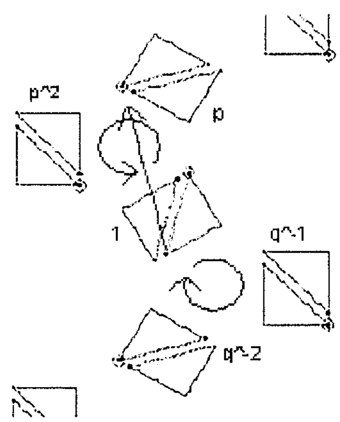


Figure 6.43 Analysis of Waltz $p3$

Similarly, the second $120°$ rotation is feasible about point q^{-1} (green arrow), in the sense that instead of rotating counterclockwise (facing outwards) about point q, the rotation occurs clockwise (still facing outwards). Observing the above diagram, it would not be in context of the Waltz, nor comfortable for the dancer or aesthetically pleasing to the spectator, to move opposite the direction of the green arrow.

Following this reasoning, half-turns in $p4$ are not feasible because the right foot would have to step too far, inhibiting the graceful poise of the dancer. Nevertheless, the quarter-turn p (counterclockwise, blue arrows in Figure 6.44) is definitely feasible due to the closeness of the position of the right foot in each sequence. This is illustrated by the pink square.

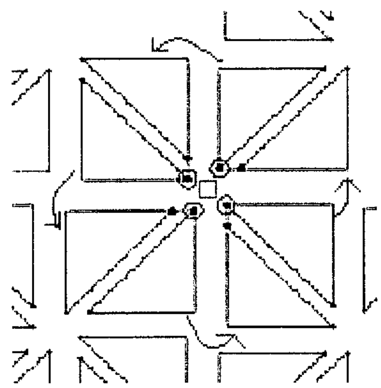


Figure 6.44 Analysis of Waltz $p4$

Using the same previous discussions, it is concluded that a side-by-side half-turn is not feasible given the Waltz basic step. Furthermore, it is already established that 120° rotations are feasible about both p and q^{-1}, as shown in Figure 6.45. However, note that the dancer might face more difficulty in gracefully executing the rotations in $p6$ compared to $p3$ due to the bigger displacement required.

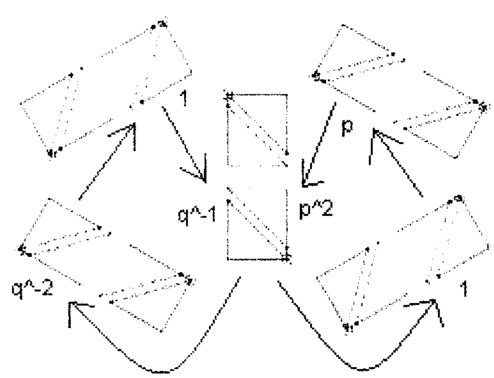


Figure 6.45 Analysis of Waltz $p6$

There now remain two dances to investigate: the Samba and the Mambo, both rhythmic Latin dances. As in all previous dances, only a backward translation s^{-1} is feasible in $p1$ (and $p2$) for both cases:

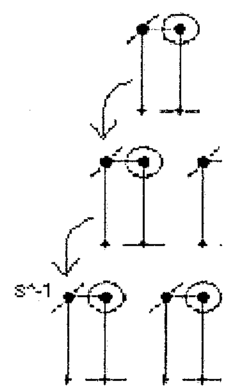


Figure 6.46 Analysis of Samba $p1$

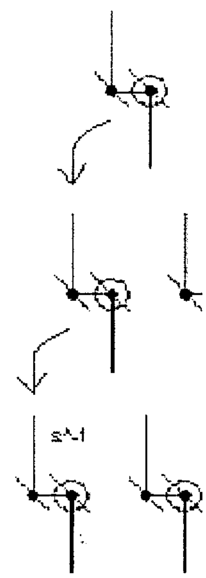


Figure 6.47 Analysis of Mambo $p1$

We now consider the half-turn a present in $p2$. Given the closeness of the steps of the right foot, such a counterclockwise turn is definitely feasible for the Samba, thus generating the following possible sequence:

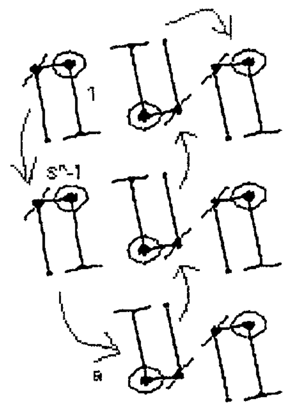


Figure 6.48 Analysis of Samba $p2$

However, the sequence can only loop back because too much distance would need to be covered if trying to move right, as illustrated by the red arrow. Furthermore, as with the Waltz, the half-turn a is theoretically generated from the initial position, whereas the dancer steps into it following the s^{-1} translation. This occurs due to the large distance between position 1 and a, which would be difficult to execute even with the little bounce present in the Samba.

We already established that a backwards translation s^{-1} is feasible in $p2$ for the Mambo. As before, there also exists a feasible half-turn, although this is not the usual half-turn a identified in red (see Figure 6.49). Rather, the only half-turn feasible given the close positioning of the feet is the half-turn sa, completed clockwise, as shown by the green arrows. The execution of this rotation is possible by the dancer starting a slight rotation while stepping forward with her left foot. She continues the rotation with her right foot, cheating slightly to move into her new position. She completes the half-turn with her last left step, now in its starting position in sa (green circle).

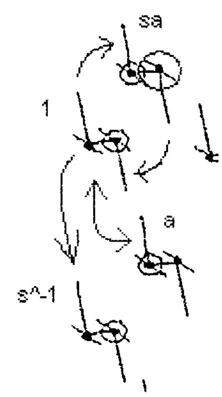


Figure 6.49 Analysis of Mambo $p2$

The situation in $p3$ for the Samba is exactly the same as all the previous feasible cases: Two feasible $120°$ rotations are about p, counterclockwise facing inwards (blue arrow in Figure 6.50), and about q^{-1}, clockwise facing outwards (green arrow).

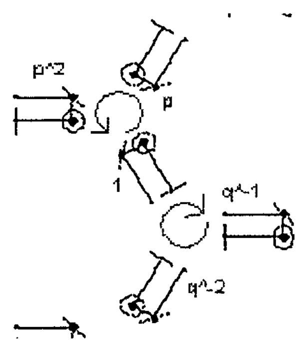


Figure 6.50 Analysis of Samba $p3$

The reasoning is also the same: the steps are relatively small, and so the feet are always close to their next position. Also, the bounce present in the Samba provides ease for the dancer to execute the rotations. In the Mambo, however, the dancer is relatively more constrained by her posture. Hence, the feasible patterns change slightly. Both $120°$ rotations can be executed, but in both cases, this occurs only in opposite direction from the theory. In other words, one rotation occurs as before about point q^{-1}, clockwise facing outwards (blue arrow in Figure 6.51). The second rotation occurs about point p^{-1}, also clockwise but facing inwards (green arrow).

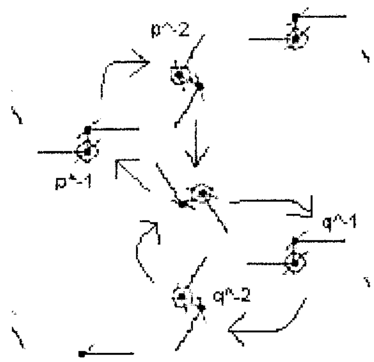


Figure 6.51 Analysis of Mambo $p3$

This difference is allocated to the restriction of having to step back with the right foot first; if rotating counterclockwise about point p, the dancer would seem to be stepping forwards, conflicting with the style of the Mambo. In addition, a variation of the rotation about p^{-1} actually occurs in Mambo and Salsa. In the execution of a turn (also called *open*), the dancer is required to step back with her right foot as such – the difference is in the steps that follow.

In $p4$, both the half-turn a and quarter-turn p are feasible for the Samba. Once again, the tight positioning of the feet and the presence of the bounce render these transformations feasible for the dancer. The quarter-turn must be executed counterclockwise due to the "right step back" restriction (green arrow in Figure 6.52). On the other hand, the half-turn can be completed either clockwise or counterclockwise (blue arrows). The latter permits the dancer to spot into position before "bouncing" back with her right foot, and is more easily performed. If choosing to rotate clockwise instead, the dancer must start rotating her feet slightly when stepping in place at the end of the sequence, in order to be able to execute the second sequence without removing style from the dance.

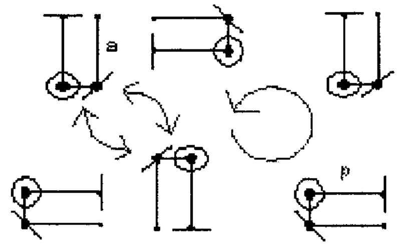


Figure 6.52 Analysis of Samba $p4$

The counterclockwise quarter-turn p is also feasible in the case of the Mambo, still due to the positioning of the feet and the "right step back" restriction (green arrow in Figure 6.53). The half-turn a, however, cannot be directly executed because of the feet alignment. Based on the starting position, the only feasible half-turn is sa as in pattern $p2$ (see Figure 6.49). Nevertheless, it is possible for the dancer to get into position a by completing clockwise quarter-turns as indicated by the blue arrows below. Like the other quarter-turns, these are feasible due to the tight position of the feet/steps.

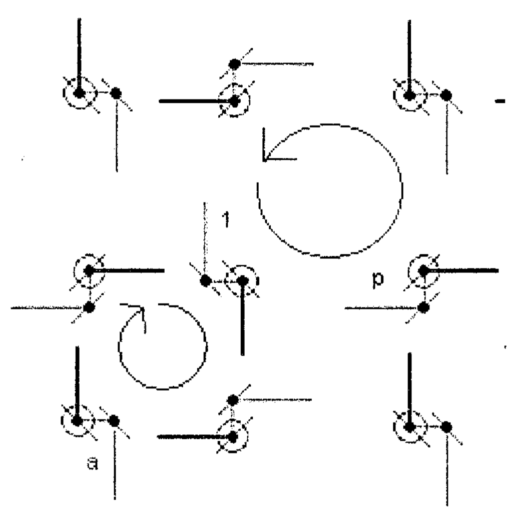


Figure 6.53 Analysis of Mambo $p4$

We conclude this discussion with pattern $p6$ for both the Samba and the Mambo. The side-by-side half-turn in $p6$ is actually advantaged by the steps of both these dances. Their alignment makes the half-turn easily executed both clockwise and counterclockwise;

however, due to the female dancer's restriction of having to step back first, we accept only the clockwise turn (green arrow in Figure 6.54). In the case of the Mambo, no other transformations are feasible. The tight movements of this dance inhibit the execution of any 120° rotation due to the wide spacing between each position.

Figure 6.54 Analysis of Mambo $p6$

For the Samba, in addition to the half-turn (green arrow in Figure 6.55), a 120° rotation is also feasible. Always as a consequence of the bounce which distinguishes the Samba from any of the other dances discussed, this rotation can be executed both clockwise or counterclockwise, illustrated below by the blue arrows.

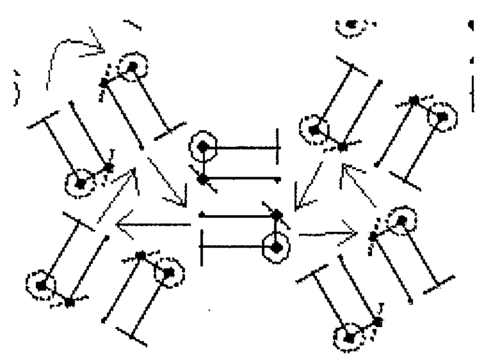


Figure 6.55 Analysis of Samba $p6$

Given the analysis of this section, it is safe to conclude that applying crystallographic groups to symbols representing specific ballroom dancing steps generates mostly feasible routines. In fact, almost every pattern generated contained at least one feasible displacement. The exceptions are $p1$,$p2$,$p3$ and $p6$ for the Chacha, and $p2$ and $p3$ for the Tango. Nevertheless, it is important to note that all the patterns presented were obtained by shifting the control path in the SymmetryWorks add-on; perhaps a different control path might have generated feasible patterns for these exceptions. In addition, changing the control path was itself creating displacements of symbols that could produce visually-entertaining floor plans. Hence, using crystallographic groups for choreographing purposes is not only an interesting suggestion, but also a feasible one.

CHAPTER 7

Conclusion

The original purpose of this research was to present an innovative contribution to the dance community using mathematics as dance notation, in the hopes of bridging the gap between science and art. In Chapter 3, we discussed an existing method of dance notation called Labanotation, which is widely-used today due to its compactness, preciseness, and ability to encode any body movement without restriction to dance steps. We presented the different aspects of mathematics embedded within it, such as symmetry, geometric shapes, 2 –dimensional and 3 –dimensional coordinate systems, angles and rotations. We also described the different affine transformations involved in the displacement of dancers relative to a stage. Translations, reflections and rotations are all necessary choreographic elements of floor plans. These mathematical concepts were made evident through an example, which included a very small excerpt of a choreography performed to music from the motion picture *The Matrix.*

The ideas presented in Chapter 3 were put on hold in Chapter 4. Instead, we presented the art of contradancing, which has been often examined by mathematicians from a group theory point of view. However, our focus was on encoding the displacements of couples in an improper contradancing formation. Using homogeneous coordinates, we generated translation matrices and reflection matrices to indicate the displacements of each partner in a couple. Depending on the values of the position vector (x,y), the augmented vector $(x,y,1)$ was multiplied by one of the four different matrices

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & -1 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 1 \\ 0 & -1 & k \\ 0 & 0 & 1 \end{bmatrix}$$

to obtain the new position vector. An example illustrating the effective use of this system was provided for 4 couples.

In Chapter 5, we combined together the ideas of the two previous chapters. We further developed the concept of matrix notation by creating a 4×6 matrix. Each column represents a different body part, as per the modern vertical staff used in Labanotation. The first three rows represent the x,y and z –axis of a three-dimensional coordinate space, with the origin centered as in Labanotation (at the point of attachment of the body part to the body). Possible values for these entries are restricted to the set $\{-2,-1,0,1,2\}$, depending on the previous position of the dancer. The fourth row represents the non-negative time variable $t \geq 0$, indicating how many beats a movement should take for completion. Augmenting the matrix with the concept of homogeneous coordinates resulted in the 5×7 movement matrix

$$M_i = \begin{bmatrix} h_x & ra_x & la_x & to_x & rl_x & ll_x & 0 \\ h_y & ra_y & la_y & to_y & rl_y & ll_y & 0 \\ h_z & ra_z & la_z & to_z & rl_z & ll_z & 0 \\ h_t & ra_t & la_t & to_t & rl_t & ll_t & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & t_{\max} \end{bmatrix}$$

where $t_{\max} = \max\{h_t, ra_t, la_t, to_t, rl_t, ll_t\}$. Recall that if positions are held for a length of s beats, this value is added to the amount of time the movement takes for completion (such as $s + t_{\max}$).

Summing a sequence of matrices M_i results in the sequence of movement matrix M; however, the entries of this matrix only make sense relative to the initial position matrix I. This matrix usually contains the values $h_t = ra_t = la_t = to_t = rl_t = ll_t = 0$ and

$$t_{\max} = \begin{cases} 0 \\ s \end{cases}.$$

Hence, summing these matrices together results in the choreography matrix $C = M + I$, which was shown to appropriately indicate the final position of the dancer given the sequence described by the matrices M_i.

To further emphasize the effectiveness of this dance notation, despite its restrictions to very basic body movements, we discussed the possible implementation of a computer program for animation. The beginnings of such a program is presented in the Appendix, including the commented code and some screen shots. Presently, the animation occurs without input of values; instead, one presses a specific letter on the keyboard for rotation of the head about the x –axis, say, and the head rotates as long as the letter is pressed. Nevertheless, it is possible to continue the implementation in order to have matrix values as input and the relevant animation as output. The interested developer can find in the Appendix a description of the steps required for completion of the program.

Finally, the focus of Chapter 6 was on choreographing with crystallographic groups. Also known as wallpaper patterns, there exist seventeen such patterns. Given the restriction that the female generally steps backwards with her right foot in most lead-and-follow dances, twelve of the seventeen patterns were dismissed because they contain opposite isometries, including reflections and glide-reflections. These opposite isometries would result in the female stepping forwards with her left foot, contradicting the structure of the lead-and-follow dances.

Five ballroom dances were chosen for analysis: two American Smooth dances (the Waltz and the Tango) and three American Rhythm dances (the Chacha, Samba and Mambo). A symbol was created to represent the basic step of each dance as performed by the female. An analysis of these symbols illustrated that, with the exception of the Tango, they all possess exactly one symmetry equivalent to that of the square. The Chacha and Mambo both contain the symmetry R_{180} (rotation of $180°$) and the Waltz is symmetric about the non-main diagonal (D'). The Samba is symmetric about the vertical axis (V), but if rotated by $90°$, it is also symmetric about the horizontal axis (H), thus creating an element of the dihedral group D_4. In the case of the Tango, the lack of repetition of steps in the reverse direction directly explains the lack of symmetry in the symbol.

The five crystallographic groups not containing opposite isometries ($p1$,$p2$,$p3$,$p4$ and $p6$) were applied to each symbol using Artlandia's *SymmetryWorks* add-in for *Adobe Illustrator*.

We observed all the patterns to determine whether any of them contained feasible displacements by the female dancer. We found that with few exceptions (namely $p1, p2, p3$ and $p6$ for the Chacha and $p2$ and $p3$ for the Tango), each pattern contained at least one feasible displacement. Furthermore, these patterns were obtained by testing different control paths in the *SymmetryWorks* program. Although the basic restrictions of each pattern remains unchanged, it is nonetheless possible that a different control path could either generate feasible patterns when none currently exist (such as in the exceptions above), or create more feasible displacements within the already feasible patterns. Thus, we can safely conclude that applying crystallographic groups to symbols representing specific dance steps is an effective method for choreographing feasible dance routines.

As intended, this research employed various mathematical tools to analyse an existing method of dance notation and to create new methods. Hence, we have shown that mathematics is indeed a universal language required in any field of study, including the arts. Connections between dance and physics (particularly time and space) were already established by the great choreographer Merce Cunningham. Also, research had already been done on the presence of group theory in contradancing formations, and the use of technology to create animation programs for the purpose of choreographing had already been started. However, the possibility of using mathematics directly for dance notation was still vague, and this thesis has rendered it more realistic. Furthermore, no one had actually researched the connections between mathematics and Labanotation, and it is intriguing that the survival of this method of dance notation might be directly linked to the mathematics. Overall, the research showed the direct impact of good notation on the development of certain disciplines, particularly the use of mathematical notation in dance.

BIBLIOGRAPHY

[1] *About Dance Notation*, [online] available: http://dance.edionysus.com/choreography/notation/.

[2] Artlandia, *SymmetryWorks* in *Adobe Illustrator* 10, Adobe Systems Incorporated, Copywrite 1987-2001.

[3] S. Allen, *Morris Dance Notation using abc*, [online] available: http://www.ucolick.org/~sla/morris/music/wwwdance.html.

[4] M. Béchard, *The Power of Numerical Notation*, MATH 496 Honour's Project, Dr. Szabo (supervisor), Concordia University, Montréal, QC, Canada, 2005.

[5] J.V. Blowers, *Matresses, Contra dancing and quilts*, 2001, [online] available: http://jimvb.home.mindspring.com/Matcontra.htm.

[6] F.J. Budden, *Patterns* in *The Fascination of Groups*, Cambridge University Press, 1972, 504-537.

[7] L. Copes, *Counting contra dances: A kinesthetic approach of group theory*, Joint Mathematics Meetings, Mathematical Association of America, Baltimore, 2003, [online] available: http://www.edmath.org/copes/contra/MAA.html.

[8] L. Copes, *Dancing with Mathematics: Mathematics related to contra dancing*, [online] available: http://www.edmath.org/copes/contra/.

[9] *Dance*, [online] available: http://cs.clark.edu/~hum101/Humanities_101/dance.htm.

[10] I. Fox, S. Marion and L. Venable (Organizers), *International Conference Exploring Research and Programming Potential for Labanotation*, Ohio, USA, April 24-26, 2004, [online] available: http://dancenotation.org/DNB/news/Ohio_report/Ohioreport.pdf.

[11] J.A. Gallian, *Contemporary Abstract Algebra*, 3rd ed., Heath and Company, Lexington, Massachusetts, USA, 1994.

[12] D. Herbison-Evans, *Computer Animation of the Merrilyn*, Basser Department of Computer Science, University of Sydney, [online] available: http://www-staff.it.uts.edu.au/~don/vogue/merr.html.

[13] A. Hutchinson, *Labanotation or Kinetography Laban: The system of Analyzing and Recording Movement*, Dance Notation Bureau, Inc., USA, 1954; revised and expanded edition, Theatre Arts Books, USA, 1970, 1977.

[14] A. Hutchinson Guest, *Dance Notation: The Process of Recording Movement on Paper*, Dance Books Ltd., London, England, 1984.

[15] A. Hutchinson Guest, *Choreo-Graphics: A comparison of Dance Notation Systems From the Fifteenth Century to the Present*, Gordon and Breach Science Publishers S.A., London, England, 1989.

[16] M.A. Kinkead, *Elementary Labanotation*, Mayfield Publishing Company, USA, 1982.

[17] R. Kostelanetz (Ed.), *Merce Cunningham: Dancing in Space and Time*, A Cappella Books, USA, 1992.

[18] R. von Laban, *Laban's Principles of Dance and Movement Notation*, Macdonald and Evans Ltd., London, England, 1956, 1975.

[19] I. Peterson, *Contra Dances, Matrices and Groups*, Science News Online, 2003, [online] available: http://www.sciencenews.org/articles/20030308/mathtrek.asp.

[20] Quotations about Mathematics and Education, [online] available: http://www.pen.k12.va.us/Div/Winchester/jhhs/math/quotes.html

[21] R. Ryman, *Labanotation and Life Forms: Computer Animation as a Complement to Dance Notation*, Excerpt of International Circle of Kinetography Laban / Labanotation Proceedings, 2001, [online] available: http://72.14.207.104/search?q=cache:hnVOTY1wOYkJ:www.ickl.org/conf01_proceedings/ICKL01_181_192.pdf+labanotation+computer+animation&hl=en&gl=ca&ct=clnk&cd=3&client=firefox-a

[22] S. Smith, *What is Mathematics?* [online] available: http://www.cate.org/sms99/writ99/whatmth.htm.

[23] F. Szabo, *Linear Algebra: An Introduction Using Mathematica*, Harcourt / Academic Press, San Diego, CA, USA, 2000.

[24] Tucson Friends of Traditional Music, *Step-by-Step Guide to Contra Dancing*, [online] available: http://www.tftm.org/contradancing/Dance%20Moves.pdf.

[25] D.K. Washburn and D.W. Crowe, *Symmetries of Culture: Theory and Practice of Plane Pattern Analysis*, University of Washington Press, USA, 1988.

[26] Wikipedia, The Free Encyclopedia, *Binary numeral system*, [online] available: http://en.wikipedia.org/wiki/Binary_numbers.

[27] Wikipedia, The Free Encyclopedia, *Contra dance*, [online] available: http://en.wikipedia.org/wiki/Contredanse.

[28] Wikipedia, The Free Encyclopedia, *Glossary of partner dance terms*, [online] available: http://en.wikipedia.org/wiki/Glossary_of_ballroom_dance_terms.

[29] Wikipedia, The Free Encyclopedia, *Lead and follow (dance)*, [online] available: http://en.wikipedia.org/wiki/Lead_and_follow_%28dance%29.

[30] Wikipedia, The Free Encycolopedia, *Salsa (dance)*, [online] available: http://en.wikipedia.org/wiki/Salsa_%28dance%29.

APPENDIX

Dance Animation

The following pages present the initial commented source code and frames for computer animation of the matrix notation created in Chapter 5. Recall that the language of choice is OpenGL, and the interface utilized is that of C++.

For the interested developer who wishes to continue the programming aspect further, one simply has to complete the body parts, and then create a global class to draw all the parts together, such as drawBody() which would contain drawHead(), drawNeck(), and so forth. Afterwards, it is just a matter of making the rotations and/or translations of the specific body part work either through animation or manually.

In terms of animation, the user would have to enter the matrices as steps. The program would then have to store these matrices in an array or some type of head memory. When a certain flag or bool is activated, the dancer would then perform these actions from the matrices saved in the array. On the other hand, the manual technique would involve the user entering the matrix and seeing the model change position in real-time. This can be achieved if the user enters special codes such that the program understands which body part is to be moved, and how. For example, the program would prompt the user for input. The user can then enter a string of the form

[BodyPart][translate x][translate y][translate z][rotate x][rotate y][rotate z]

where:

- BodyPart = 2 characters;
- Translate x,y,z = 2 characters = [bit][number] = [0/1][0 – 9], where [0] = positive and [1] = negative;
- Rotate x,y,z = 4 characters = [bit][number] = [0/1][0 – 999], where [0] = clockwise and [1] = counterclockwise.

So for example, entering the string RA000100002200001105 would signify:

- [BodyPart] = RA, so move the right arm;
- [translate y] = [01], so translate 1 unit in the positive y direction;
- [rotate x] = [0022], so rotate 22° clockwise along the x –axis;
- [rotate z] = [1105], so rotate 105° counterclockwise along the z –axis.

```
/*****************************************************************************************/
/*
/* A wireframe/solid model of a dancing figure using OpenGL.
/*
/*****************************************************************************************/

    #include <stdlib.h>
    #include <GL/glut.h>
    #include <stdio.h>              // Other necessary program related includes.
    #include <math.h>
    #include <C:\Documents and Settings\Denz\My Documents\tempor\texture.h>

//  typedef unsigned bool;     // Better to exclusively typedefine bool. Else gcc might throw error.

    #define SIZE 600                // To deal with the view volume
    #define NEAR_Z 1.0
    #define FAR_Z 100.0

    #define PI 3.14159265       // An excessively abused used constant !!

    #define ROOM_SIDE 15.0
    #define FLOOR -10.5

//  #define ROOM 1

    static GLdouble viewer[3];    // For keeping track of location of camera.
    static GLdouble center[3];    // For keeping track of the look at position of the camera.
    static GLdouble up[3];        // For keeping track of the up Vector of the camera.

    double theta_camera;         // Polar coordinates for the camera.
    double phi_camera;
    double rad_camera;

//  enum {false, true};
    bool bAxis;                      // To show or not to show the reference axis.
    bool bWire;                      //Wireframe of Solidframe


    /* These are for the animations*/
    bool bWalk;                      //Animate walking
    /*End of animation bools*/



    enum {YAW, PITCH, ROLL} Ypr;        //States of the head movement


    static float temp_angle3y=0.2;        //for yaw
    static float temp_angle3p=0.2;        //for pitch
    static float temp_angle3r=0.2;        //for roll
    static float yAngle=0.0;                  //starting angles for
    static float pAngle=0.0;                  //the head's yaw,
    static float rAngle=0.0;                  //pitch, and roll
```



```
    static float wAngle=0.0;                  //angle for leg walking

    static float incr=0.005;                  //for eye animation
    static float leftz = 2.0;                 //for animation
    static float rightz= -2.0;                //of the shifting eye

    static float movey=0.005;                 //for dancer
    static float tempmove=0.005;          //for dancer
    static float tempUp   = 0.25;        //incrementer for upAngle
    static float tempSide  = 0.5;        //incrementer for sideAngle
    static float sideAngle = 0.0;        //to move around

    static float body_size=1;                 //To make the body size dynamic


    float fovy = 90.0;            // For Perspective projections.
    int H,W;
    float fViewVol;               // For Orthographic projections.
    enum {PERSPECTIVE, ORTHOGRAPHIC, EYEVIEW} ePrjn = PERSPECTIVE;

    typedef GLfloat Point[3];
    Point pt[] = {
    { -15.0, -15.0, 15.0 },
    { -15.0, 15.0, 15.0 },
    { 15.0, 15.0, 15.0 },
    { 15.0, -15.0, 15.0 },
    { -15.0, -15.0, -15.0 },
    { -15.0, 15.0, -15.0 },
    { 15.0, 15.0, -15.0 },
    { 15.0, -15.0, -15.0 } };      // Definition of the vertices of the cube.

    GLfloat ceiling_light_position[] = { 0.0, ROOM_SIDE/2, 0.0, 1.0 };

/* This section is if the developer would like to add texture mapping to the dancer model*/
/* To do this they would just have to include the picture files and indicated the format */
/* in the following function: g_apcTextureFiles[]*/


#define TEXTURES_AVAIL 1
// All available textures' integer "names"
// to be associated with generatd images
// of textures via binding
static GLuint g_aiTexNames[TEXTURES_AVAIL];

// Loaded *.rgb images are stored in this
// array as sequences of bytes for all available
// textures
GLubyte* g_apubLoadedImages[TEXTURES_AVAIL] = {NULL};

// Filenames for the texture images, taken
// from the course web page. First six are
// used for the cube faces; the last one is
// for the sphere
char* g_apcTextureFiles[] =
```

```
{
    "test.bmp"
};


void readAllTextures(void);

void drawGrid()
{
    glTranslatef(0, -35, 0);                    // Create the Floor
    glScalef(2,1,2);
    glBegin(GL_QUADS);
        glColor3f(1.0, 1.0, 1.0);
        glVertex3f( 20.0, 20.0,-20.0);
        glVertex3f(-20.0, 20.0,-20.0);
        glVertex3f(-20.0, 20.0, 20.0);
        glVertex3f( 20.0, 20.0, 20.0);
    glEnd();
    glTranslatef(0, 35, 0);
    glScalef(0.5,1,0.5);
}


/*****************************************************************************************/
void showMenu(void)
                    // Self Explanatory.
{
    printf("\n\n\n\n SAMPLE ILLUSTRATION OF DANCER. ");
    printf("\n-----------------------------------------------------------------");
    printf("\n Operations: ");

    printf("\n\n ******** VIEWS AND CAMERA MOTION ******** ");
    printf("\n Use o/O for ortho views.");
    printf("\n Use 1/3 to toggle between 1st person and 3rd person");
    printf("\n Use w/W to switch between wireframe and solid frame.");
    printf("\n Use z/Z to zoom in and out");
    printf("\n Use f/F to move the camera fwd/bkwd.");
    printf("\n Use up/down to rotate the camera around the X-axis of the At point.");
    printf("\n Use left/right to rotate the camera around the Y-axis of the At point.");
    printf("\n Use a/A to show/hide the reference axis <x = RED, y = GREEN, z = BLUE>.");
    printf("\n Use m/M to see the menu again.\n\n");
    printf("\n Use b/B to activate movement.\n\n");
    printf("\n Use y/Y to rotate the body of the dancer along the y-axis.\n\n");
    printf("\n Use p/P to rotate the arms and head of the dancer along the z axis.\n\n");
    printf("\n Use r/R to rotate the head of the dancer along the x-axis.\n\n");

}

/*****************************************************************************************/
    void init (void)
                            // Initializes the gl Graphics env and the program variables.
    {
        viewer[0] = 0.0;        // For use in the 'eye' parameter for the gluLookAt(..). Also the initial camera position.
        viewer[1] = 0.0;
```



```
        viewer[2] = 25.0;

        center[0] = 0.0;        // For use in the 'center' parameter for the gluLookAt(..). Also the initial orientation.
        center[1] = 0.0;
        center[2] = 0.0;

        up[0] = 0.0;            // For use in the 'up' parameter for the gluLookAt(..). Also the up direction of the camera.
        up[1] = 1.0;
        up[2] = 0.0;

        phi_camera = -270.0 * (PI/180);
        theta_camera = 0.0;
        rad_camera = sqrt(viewer[0] * viewer[0] + viewer[1] * viewer[1] + viewer[2] * viewer[2]);
                                // Polar coordinates for the camera

        W = H = SIZE;       // global variables used to keep track of window size.

        glMatrixMode(GL_PROJECTION);
                                // Set the current openGL matrix to GL_PROJECTION ie projection matrix.
        glLoadIdentity();
                                // Load identitiy values in the above.

        if(ePrjn == PERSPECTIVE)
        {
            fovy = 90.0;
            gluPerspective(fovy, (GLfloat)W/(GLfloat)H, NEAR_Z, FAR_Z);
                                // This sets the view volume to be a Perspective one.
        }
        else if (ePrjn == ORTHOGRAPHIC)
        {
            fViewVol = 50.0;
            glOrtho(-fViewVol, fViewVol, -fViewVol, fViewVol, NEAR_Z, FAR_Z);
                                // This sets the view volume to be a Orthographic one.
        }

        glMatrixMode(GL_MODELVIEW);
                                // Change the current matrix mode to Model-View matrix.
        glLoadIdentity();


        glClearColor (0.0, 0.0, 0.0, 0.0);
                            // set the background color to black.

        bAxis = true;
        showMenu();
    }

/*****************************************************************************************/
    void keyboardCallbackProc(unsigned char key, int x, int y)
                            // This is the callback procedure for capturing OpenGL Keyboard events.
    {

        switch(key)
```

```
    {

    case 'a':                // Show axes.
        if (bAxis == true)          bAxis = false;
        else                        bAxis = true;
        break;

    case 'b':                // Start walking animation
    case 'B':
        if (bWalk == true)          bWalk = false;
        else                            bWalk = true;
        break;

    case 'f':                // Move Camera forward.
        rad_camera -= 0.1;
        break;

    case 'F' :               // Move Camera backward.
        rad_camera += 0.1;
        break;


    case 'w' :               //Show wired or solids
        if (bWire==true)        bWire = false;
        else                    bWire = true;
        break;

    case 'z' :               // Zoom in.
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
        if(ePrjn == PERSPECTIVE)
        {
            fovy -= 1.5;
            gluPerspective(fovy, (GLfloat)W/(GLfloat)H, NEAR_Z, FAR_Z);
        }
        else
        {
            fViewVol -= 1.0;
            glOrtho(-fViewVol, fViewVol, -fViewVol, fViewVol, NEAR_Z, FAR_Z);
        }
        glMatrixMode(GL_MODELVIEW);
        break;

    case 'Z' :               // Zoom out.
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
        if(ePrjn == PERSPECTIVE)
        {
            fovy += 1.5;
            gluPerspective(fovy, (GLfloat)W/(GLfloat)H, NEAR_Z, FAR_Z);
        }
        else
        {
            fViewVol += 1.0;
```



```
            glOrtho(-fViewVol, fViewVol, -fViewVol, fViewVol, NEAR_Z, FAR_Z);
        }
        glMatrixMode(GL_MODELVIEW);
        break;

    case '3':
        ePrjn = PERSPECTIVE;
        init();
            break;

    case 'O':
    case 'o':
        ePrjn = ORTHOGRAPHIC;
        init();
        break;

    case '1':                        //For 1st person view
        ePrjn = EYEVIEW;
        break;


    case 'p':
    case 'P':
        if (pAngle>=16)              //does something
        {temp_angle3p=-0.5;}
        else if (pAngle<=-16)
        {temp_angle3p=0.5;}
        pAngle+=temp_angle3p;
        break;

    case 'y':
    case 'Y':
        if (yAngle>=16)              //Rotation of entire body
        {temp_angle3y=-0.5;}
        else if (yAngle<=-16)
        {temp_angle3y=0.5;}
        yAngle+=temp_angle3y;
        break;

    case 'A':                    //rotate body clockwise
        yAngle+=0.5;
        break;
    case 'D':                    //rotate body counterclockwise
        yAngle-=0.5;
        break;

    case 'S':                    //translate body backward
        movey-=0.5;
        break;

    case 'W':
        movey+=0.5;              //translate body forward
        break;
```

```
        case 'r':
        case 'R':
            if (rAngle>=16)          //Roll tilting head side-side
            {temp_angle3r=-0.5;}
            else if (rAngle<=-16)
            {temp_angle3r=0.5;}
            rAngle+=temp_angle3r;
            break;


        case 'C':
        case 'c' :                     // Clear viewing parameters and return to initial view.
            init();
            break;

        case 'M' :
        case 'm' :
            showMenu();
            break;

        case 27 :                      //ESCAPE Code for exiting program.
            exit(1);
            break;

            }
        glutPostRedisplay();       // Direct the glut system to redisplay the screen.
    }

/*******************************************************************************************/

void specialCallbackProc (int key, int x, int y)
                    // This is the callback procedure for capturing special charater events.
{
    switch (key)
    {

        case GLUT_KEY_LEFT:        // Rotate Camera in an anticlockwise direction about the Y axis of the At (center) point.
            theta_camera -= 1.0 * (PI/180);
            break;

        case GLUT_KEY_RIGHT:   // Rotate Camera in an clockwise direction about the Y axis of the At (center) point.
            theta_camera += 1.0 * (PI/180);
            break;

        case GLUT_KEY_UP :         // Rotate Camera in an anticlockwise direction about the X axis of the At (center) point.
            phi_camera -= 1.0 * (PI/180);
            break;

        case GLUT_KEY_DOWN :   // Rotate Camera in an clockwise direction about the X axis of the At (center) point.
            phi_camera += 1.0 * (PI/180);
            break;
    }
    glutPostRedisplay();
}
```



```
/*******************************************************************************************/
void reShapeCallbackProc(int w, int h)
                    // This is the callback procedure for capturing reShape event for window resizing.
{
    glViewport(0, 0, w, h);
                    // Set the viewport to current window size.
    W = w;
    H = h;

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    if(ePrjn == PERSPECTIVE)
        gluPerspective(fovy, (GLfloat)W/(GLfloat)H, NEAR_Z, FAR_Z);
    else
    {
        if (w > h)
        {
            W = (fViewVol * w) / h;
            H = fViewVol;
        }
        else
        {
            W = fViewVol;
            H = (fViewVol * h) / w;
        }
        glOrtho(-W, W, -H, H, NEAR_Z, FAR_Z);
    }               // Change the view volume to maintain the aspect ratio wrt to viewport.

    glMatrixMode(GL_MODELVIEW);

    glutPostRedisplay();
}
/*******************************************************************************************/

void drawSphere(float fRadius)
                    // Used to generate a Sphere shape.
{
    GLUquadricObj* pObj;
    glPushMatrix();

    pObj =  gluNewQuadric();
                    // Creates a new quadrics object and returns a pointer to it.

    if (bWire==false)
    gluQuadricDrawStyle(pObj, GLU_FILL);

    else
    gluQuadricDrawStyle(pObj, GLU_LINE);

    gluSphere(pObj, fRadius, 20, 20);
                    // Draw the sphere with a radius : fRadius.
    gluDeleteQuadric(pObj);
```

```
                                // Free the Quadric
    glPopMatrix();

}
/*******************************************************************************************/

void drawCylinder(float fTopR, float fBottomR, float fHeight)
                                // Used to generate a cylinder shape.
{
    GLUquadricObj* pObj;
    glPushMatrix();

    pObj = gluNewQuadric();
                                // Creates a new quadrics object and returns a pointer to it.
    if (bWire==false)
    gluQuadricDrawStyle(pObj, GLU_FILL);

    else
    gluQuadricDrawStyle(pObj, GLU_LINE);

    gluCylinder(pObj, fTopR, fBottomR, fHeight, 20, 20);
                                // Draw the cylinder with a radius : fRadius.
    gluDeleteQuadric(pObj);
                                // Free the Quadric
    glPopMatrix();

}

/*******************************************************************************************/


void showReferenceAxis(void)
                            // Draw the reference axis
{
    if(bAxis)
    {
     glPushAttrib(GL_ALL_ATTRIB_BITS);
                            // Pushes attributes like current color to attribute stack.
     glBegin(GL_LINES);
                            // X axis red
   glColor3f(1, 0, 0);
   glVertex3f(0, 0, 0);
   glVertex3f(10, 0, 0);
                            // Y axis green
   glColor3f(0, 1, 0);
   glVertex3f(0, 0, 0);
   glVertex3f(0, 10, 0);
                            // Z axis blue
   glColor3f(0, 0, 1);
   glVertex3f(0, 0, 0);
   glVertex3f(0, 0, 10);
      glEnd();
      glPopAttrib();
```



```
                                // // Pops attributes like current color from the attribute stack and sets as current.
    }
}


/*******************************************************************************************/
void drawDancerBody(float body_size)

{
    glPushMatrix();
    glTranslatef(0.0, ceiling_light_position[1], 0.0);
    glScalef(1.0, -1.0, 1.0);
    glPopMatrix();

    glTranslatef(0.0, -9.0, 0.0);  // Set the dancer to the ground

    glPushMatrix();                 //body of man
    glScalef(body_size, body_size, body_size);
    glTranslatef(-.3, 6.5, 0.0);
    glColor3d(1,0,1);
    glRotatef(90.0, 1.0, 0.0, 0.0);
    drawCylinder(4.0, 3.5, 7);
    glPopMatrix();


}

/*******************************************************************************************/

void drawDancerEye(float rightz, float leftz)
{
    glPushMatrix();                 //outer eye
    glScalef(0.5, 0.5, 0.5);
    glTranslatef(18, 20.5 , -2);
    glColor3d(1,1,1);
    drawSphere(0.9);
    glPopMatrix();

    glPushMatrix();                 //outer eye
    glScalef(0.5, 0.5, 0.5);
    glTranslatef(18, 20.5 , 2);
    glColor3d(1,1,1);
    drawSphere(0.9);
    glPopMatrix();

    glPushMatrix();                 //MY right eye
    glScalef(0.5, 0.5, 0.5);
    glTranslatef(18.5, 20.6 , rightz);
    glColor3d(0,0,0);
    drawSphere(0.5);
    glPopMatrix();

    glPushMatrix();                 //MY left eye
```

```
    glScalef(0.5, 0.5, 0.5);
    glTranslatef(18.5, 20.6 , leftz);
    glColor3d(0,0,0);
    drawSphere(0.5);
    glPopMatrix();
    }


/*******************************************************************************************/

void drawDancerHead(void)
{

glPushMatrix();

    glRotatef(rAngle,1,0,0);
    glRotatef(yAngle,0,1,0);
    glRotatef(pAngle,0,0,1);
    glTranslatef(0, -10, 0.0);

    glPushMatrix();                    //head
    glRotatef(90.0, 0.0, 1.0, 0.0);
    glTranslatef(0, 10, 0.0);
    glColor3d(0.95,0.87,0.75);
    drawSphere(3);
    glPopMatrix();


    glPushMatrix();                    //eyes
    glTranslatef(-6.4, 1.0, 0.0);
    drawDancerEye(rightz, leftz);
    glPopMatrix();

glPopMatrix();


}
/*******************************************************************************************/

void drawDancerNeck(void)
{

    glPushMatrix();                    // Neck of man using cylinder
    glTranslatef(0, 8, 0.0);
    glRotatef(90.0, 1.0, 0, 0.0);      //(angle, x, y, z) rotates angle w/reference to the axis
    glColor3d(0.95,0.87,0.75);
    drawCylinder(1.5, 2.5, 2.5);
    glPopMatrix();

}


/*******************************************************************************************/
```



```
void drawLegLeft(void)
{
    glPushMatrix();                    // Leg 2.1 (rear)
    glTranslatef(-7.0, 0, -2.25);
    glScalef(0.4, 0.5, 0.4);
    glRotatef(90, 1, 0, 0);
    drawCylinder(4, 2.5, 3.5);
    glPopMatrix();


    glPushMatrix();                    //Leg 2.2 (rear)
    glScalef(0.5, 0.5, 0.5);
    glTranslatef(-14, -4, -4.62);
    drawSphere(2);
    glPopMatrix();

    glPushMatrix();                    // Leg 2.3 (rear)
    glTranslatef(-7, -2, -2.25);
    glScalef(0.5, 0.5, 0.5);
    glRotatef(90, 1, 0, 0);
    drawCylinder(1.5, 1.5, 7);
    glPopMatrix();
}
/*******************************************************************************************/

void drawLegRight(void)
{

    glColor3d(0.95,0.87,0.75);
    glPushMatrix();                    // Leg 1.1 (rear)
    glTranslatef(-7.0, 0, 2.25);
    glScalef(0.4, 0.5, 0.4);
    glRotatef(90, 1, 0, 0);
    drawCylinder(4, 2.5, 3.5);
    glPopMatrix();

    glPushMatrix();                    //Leg 1.2 (rear)
    glScalef(0.5, 0.5, 0.5);
    glTranslatef(-14, -4, 4.62);
    drawSphere(2);
    glPopMatrix();

    glPushMatrix();                    // Leg 1.3 (rear)
    glTranslatef(-7, -2, 2.25);
    glScalef(0.5, 0.5, 0.5);
    glRotatef(90, 1, 0, 0);
    drawCylinder(1.5, 1.5, 7);
    glPopMatrix();

}

/*******************************************************************************************/
```

```
void drawDancerLeg(void)
{

    if (bWalk == true)                      //for animation purposes
    {


        if (wAngle>=30)             //Pitch for the head  (USE 16/-16)
            {tempUp=-0.25;}
            else if (wAngle<=-30)
            {tempUp=0.25;}
            wAngle+=tempUp;
    }


    glPushMatrix();
    glRotatef(yAngle,0,1,0);                //attempting movement
    glRotatef(wAngle,0,0,1);
    glTranslatef(7,0,-0.7);
    drawLegRight();
    glPopMatrix();

    glPushMatrix();
    glRotatef(yAngle,0,1,0);                //attempting movement
    glRotatef(-wAngle,0,0,1);               //legs have to walk opposite
    glTranslatef(7,0,0.7);
    drawLegLeft();
    glPopMatrix();

}
/*************************************************************************************/

void drawArm(void)
{
glPushMatrix();
glTranslatef(0, -3, -4);                    //to focus shoulder at origin for rotation

    glPushMatrix();
        glTranslatef(0, 6, 0);

        glPushMatrix();                     //Shoulder
            drawSphere(2);
        glPopMatrix();
    glPopMatrix();

    glPushMatrix();
        glTranslatef(0, 6, 0);
            glPushMatrix();                 // upper arm
                glRotatef(90, 1, 0, 0);
                drawCylinder(1.5, 1.5, 7);
            glPopMatrix();
    glPopMatrix();
```


file:///C|/Documents%20and%20Settings/Claud/Desktop/Thesis/dancer%20code.txt

```
    glPushMatrix();                         //Arm Joint
        drawSphere(2);
    glPopMatrix();


    glPushMatrix();                         // lower arm
        glRotatef(90, 1, 0, 0);
        drawCylinder(1.5, 1.5, 5);
    glPopMatrix();


glPopMatrix();

}
/*************************************************************************************/

void drawLArms(void)
{

    glColor3f(0.85, 0.69, 0.64);        //set skin color


glPushMatrix();
    glRotatef(pAngle,0,0,1);
        glPushMatrix();                     // Used for both Arms (set for left arm)
                glPushMatrix();
                glScalef(0.55, 0.55, 0.55);
                drawArm();
                glPopMatrix();
        glPopMatrix();
glPopMatrix();
}
/*************************************************************************************/

void drawDancer(void)
{           //draws all parameters of dancer

    glPushMatrix();

        glRotatef(yAngle,0,1,0);            //To rotate


        if (bWalk == true)
        {movey+=tempmove;                       //translation of dancer
         if (movey > 10)                    //how fast and how far he moves
            tempmove=-0.005;
         else if (movey < -10)
            tempmove=0.005;


        }
```

```
        glTranslatef(movey, 0, 0);
        drawDancerBody(body_size);

        drawDancerNeck();

        glPushMatrix();
        glTranslatef(0,10,0);
        drawDancerHead();
        glPopMatrix();

/**********The below code concerns the legs of the person **********************/
        drawDancerLeg();



/**********The below code concerns the arms of the person **********************/

        glPushMatrix();                 //this is the left arm
        glTranslatef(0, 2.5, -3);
        glRotatef(40, 1, 0, 0);
        drawLArms();
        glPopMatrix();

        glPushMatrix();                 //this is the right arm
        glTranslatef(0, 5.5, 6);      //location is a bit different
        glRotatef(-40, 1, 0, 0);      //have to rotate -40 to mirror
        drawLArms();
        glPopMatrix();

/**********END OF CODE CONCERNING ARMS **********************/

glPopMatrix();

};

/****************************************************************************************/



    void displayCallbackProc (void)
                            // This is the callback procedure for capturing OpenGL Display events.
                            // All the 'happening' things happen here :)
    {
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
        glColor3f(1.0, 1.0, 1.0);

        glMatrixMode(GL_MODELVIEW);
        glLoadIdentity();

        viewer[1] = rad_camera * cos(phi_camera);
        viewer[2] = rad_camera * sin(phi_camera) * cos(theta_camera);
        viewer[0] = rad_camera * sin(phi_camera) * sin(theta_camera);
```

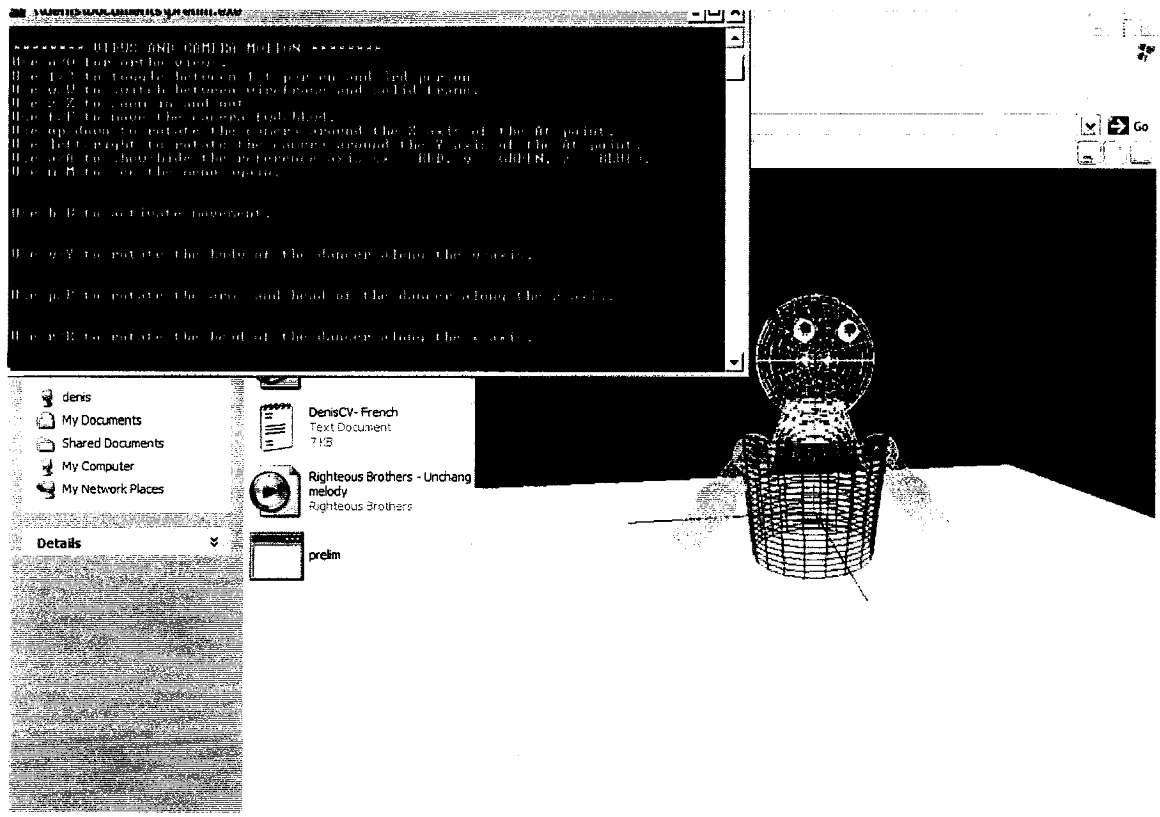


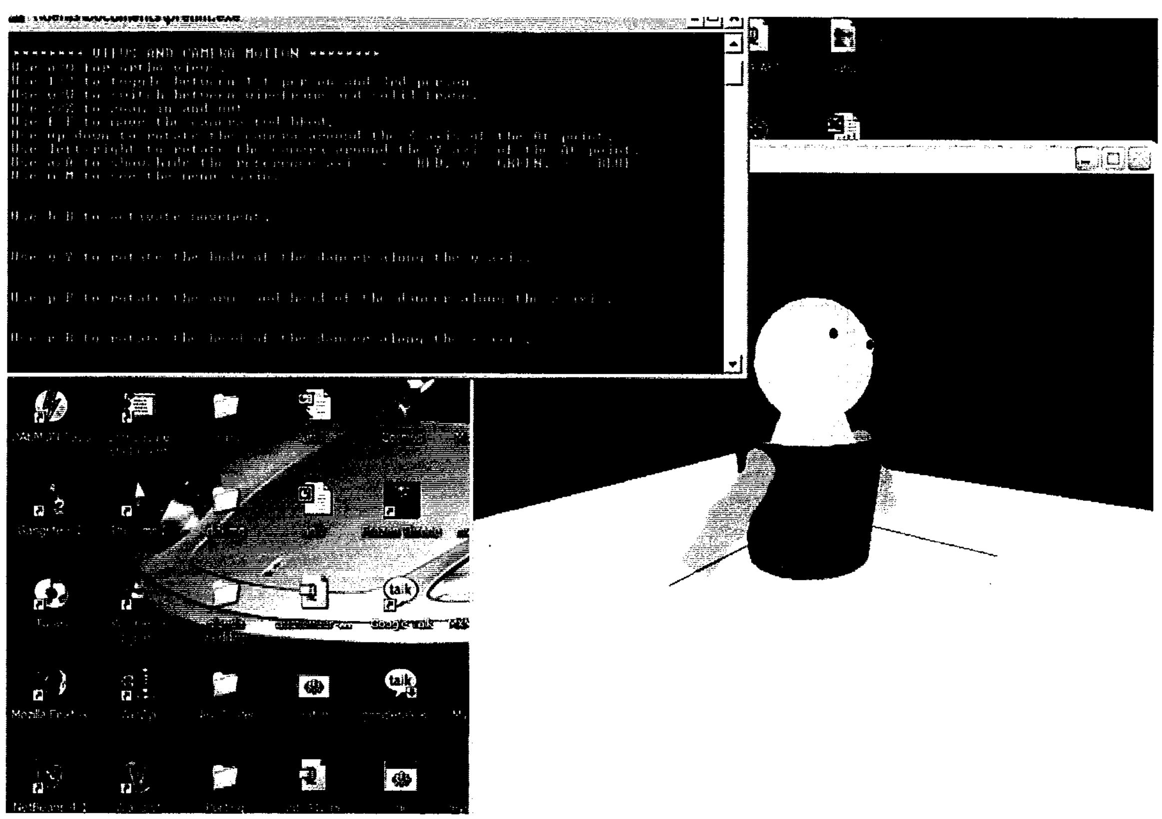
```
        if (ePrjn != EYEVIEW) //if its Ortho or Perspective
        {   gluLookAt(viewer[0],viewer[1],viewer[2], center[0], center[1], center[2], up[0], up[1], up[2]);}
                            // This function positions the camera to view your world.

        else if (ePrjn == EYEVIEW)          //For 1st person view
        {
            glTranslatef(0,-10.4, 11);                  //Position camera between
            glRotatef(-yAngle, 0, 1, 0);              //the eyes
            glRotatef(-pAngle, 1, 0, 0);
            glRotatef(rAngle, 0, 0, 1);
            gluLookAt(0.0, -3.5, 0.0, 20, 0, 0, 0, 1, 0);
        }

        /****************************************************************************************/
                                    // Next we draw our world.
        showReferenceAxis();

        glPushMatrix();                 //to extend the floor
        glTranslatef(15,7,0);
        drawGrid();
        glPopMatrix();

        glPushMatrix();                 //Added to center dancer at the axis
        glTranslatef(0,7,0);        //Raise everything by 7 y-values
        drawGrid();
        drawDancer();                       //Draws the dancer
        glPopMatrix();                  //The popmatrix from the pushmatrix that
                                        //raised everything by 7 y-values.

        glutSwapBuffers();              // Use of double buffering to avoid flicker.
        glutPostRedisplay();
    }

/****************************************************************************************/

void readAllTextures()
{
    // To be returned by read_texture() from textures.h
  int l_iWidth;
    int l_iHeight;
    int l_iComponents;

    // Genreate integer names for textures
  glGenTextures(TEXTURES_AVAIL, g_aiTexNames);

    // Load images
  for(int i = 0; i < TEXTURES_AVAIL; i++)
  {
        // Actual loading
        g_apubLoadedImages[i] = (GLubyte *)read_texture
        (
            g_apcTextureFiles[i],
            &l_iWidth,
            &l_iHeight,
```

```
            &l_iComponents
        );

        // Binding loaded image to generated integer names
        glBindTexture(GL_TEXTURE_2D, g_aiTexNames[i]);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER,GL_NEAREST);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER,GL_NEAREST);

        gluBuild2DMipmaps
        (
            GL_TEXTURE_2D,
            l_iComponents,
            l_iWidth,
            l_iHeight,
            GL_RGBA,
            GL_UNSIGNED_BYTE,
            g_apubLoadedImages[i]
        );
    }
}

/*****************************************************************************************/
    int main (int argc, char *argv[])
                                    // The main program.
    {

        /* All customary glut env initializations. */
        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
        glutInitWindowSize(SIZE, SIZE);
        glutInitWindowPosition(350, 200);
        glutCreateWindow("View of Dancer.");
        glEnable(GL_DEPTH_TEST);

        /* glut env initializations done. */


        init();             // Initialize the openGL env. variables and the application global variables.


        /* Callback registrations with the OpenGL env are done here */

        glutDisplayFunc(displayCallbackProc);
        glutKeyboardFunc(keyboardCallbackProc);
        glutSpecialFunc(specialCallbackProc);
        glutReshapeFunc(reShapeCallbackProc);

    //  readAllTextures();

        /* Callback registrations done.*/

        glutMainLoop();
```



```
                                    // Inside glutMainLoop(); all the mouse/KB events pertaining to the
                                    // application window are dispatched. This loop is never exited so the
                                    // statements after glutMainLoop(); are never executed !
        return 1;
    }

/************************** END OF PROGRAM ***********************************************/
```


denis
My Documents
Shared Documents
My Computer
My Network Places
Details
DenisCV- French
Text Document
Righteous Brothers - Unchang
melody
Righteous Brothers
prelim
Go


talk
Google talk