NORGES TEKNISK-NATURVITENSKAPELIGE
UNIVERSITET

A Perceptual Completion model

by

Per Martin Viddal

PREPRINT
NUMERICS NO. 12/2008

NORWEGIAN UNIVERSITY OF
SCIENCE AND TECHNOLOGY
TRONDHEIM, NORWAY

This report has URL
http://www.math.ntnu.no/preprint/numerics/2008/N12-2008.pdf
Address: Department of Mathematical Sciences, Norwegian University of Science and
Technology, N-7491 Trondheim, Norway.

A Perceptual Completion model

Per Martin Viddal

TMA4500

A Perceptual Completion model

Per Martin Viddal

NTNU
Faculty of Information Technology,
Mathematics and Electrical Engineering
Department of Mathematical Sciences

December 17th, 2008

Preface

This report is written as part of the mandatory project in my 9th semester of the Master's degree programme within Industrial Mathematics at NTNU, Trondheim. The topic is part of the Gala project[1], which is a specific targeted research project within NEST (new and emerging science and technology) activity of the 6th framework programme of the European Union.

During this work, my interests within the topic have grown. Many hours have been spent doing research, and many late nights have been spent in front of the computer, implementing the codes. The interested reader can find the (not well-documented) MATLAB$^{\text{TM}}$-source codes in [14].

In the beginning of October this year, a guest lecture at NTNU was held by one of the authors of my main referred paper, Prof. Giovanna Citti. In addition to Prof. Citti, I would like to thank Prof. Elena Celledoni and Ass. Prof. David Cohen for the following discussions regarding this subject, which was an inspiration in the initial phase of my project.

Last, but not least, I want to thank my supervisor Prof. Brynjulf Owren for helpful ideas and inspirational discussions throughout this semester.

TRONDHEIM, December 2008
Per Martin Viddal

[1] Sub-Riemannian geometric analysis in Lie groups, "http://www.gala.unibo.it/".

A preliminary study of the paper “A Cortical Based Model of Perceptual Completion in the Roto-Translational Space” by G. Citti and A. Sarti [3] is done. The model completes images according to how the area V1 of the visual cortex functions, and the space in which the completion is performed, is the three-dimensional image-oriented manifold.

Various aspects of the model are discussed, including Lie group theory. The first part of the numerical scheme has been analyzed, and a restriction on the time step for the forward Euler method is found.

The model is implemented, and various examples are presented.

Contents

List of Figures

1 Introduction

The perceptual completion phenomenon refers to seeing a figure as complete, when parts of it contains missing information. Objects that are occluding other objects, or when part of the image falls in a blind area of the visual field are examples of missing information.

Image reconstruction has been widely studied in the past. Before the digital age, image reconstruction was done manually by painters, but today, when computers and digital cameras have been a common property, numerous algorithms have been proposed. Many algorithms are designed for noise removal, but others can reconstruct large holes based on the image properties near the hole. Popular ones, such as the Perona-Malik-equation [1], diffuse the original image, and result in noise removal and image enhancement.

Algorithms exploiting main phenomenological properties described by psychology of Gestalt are also studied. This is classically achieved by minimizing an elastica functional. In [2], the elastica functional

$$\int_\gamma |\nabla I| \left(1 + \left|\operatorname{div}\left(\frac{\nabla I}{|\nabla I|}\right)\right|^p\right) dxdy, \quad p > 1 \tag{1}$$

has been minimized. $I(x,y)$ is the function defining the image, p is a constant and the integral is computed on the whole image domain. In images where some object is blocking another object, this procedure will not maintain them both. An example of such an image is shown in Fig. 1.

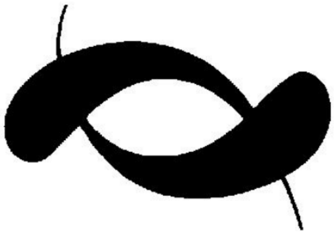


Figure 1: The two fishes of Kanizsa.

This image is an interesting example, since it has no clear depth ordering to distinguish the objects from one another. Such images are difficult to treat, using numerical algorithms.

The objective of this work is to reproduce and extend some results in the paper “A Cortical Based Model of Perceptual Completion in the Roto-Translational Space” by G. Citti and A. Sarti [3]. This is a mathematical model based on psychological and biological models to complete so-called subjective contours, which arise when we look at, e.g., the figure above. Further, their model has the property of maintaining both occluded and occluding objects, by “lifting” the image to three-dimensional space.

The report is organized as follows. Chapter 2 is an introduction to the model, where the basic properties regarding the neuropsychological aspects are introduced. In Chapter 3 we establish the mathematical foundation in Lie group theory, needed to prove that the three-dimensional space the image is lifted to, is a Lie group. In Chapter 4, we derive and present the complete mathematical model. Chapter 5 describes the numerical approximations used in the calculations. It also includes a discussion of the boundary conditions, and a numerical analysis regarding the stability of the first part of the numerical scheme. Numerical results for some chosen images are presented in Chapter 6.

2 The model

The image in Figure 2 is named after the Italian psychologist Gaetano Kanizsa.

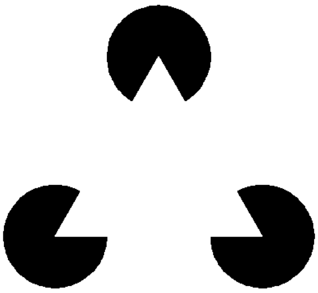


Figure 2: Kanizsa's triangle.

In the center of this figure we perceive a white triangle, even though it is not drawn. This effect is called subjective contours.

Other examples include objects that are occluding other objects, where our "common sense" can figure out the missing information. Figure 3 shows two such examples.

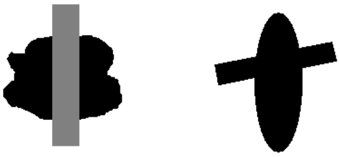


Figure 3: Occluding and occluded objects.

On the left of Figure 3 we perceive the two black sections as one amodally completed surface extending behind the gray occluder. On the right we also have two objects, but the depth of the two objects is not obvious. The objects are alternating on which is the modally completed (in front) and the amodally completed object (in the back).

The last example has some of the same properties as "Kanizsa's fishes" shown in Figure 1 in the introduction. This image shows two fishes that are blocking each other, i.e., we have two objects that are both occluded *and* occluding at the same time. If the two fishes were to be extracted from each other, this would lead to a process that includes modal and amodal completion at the same time.

2.1 Psychological and biological model

According to psychologists and biologists, information that is coming in through our visual system is sent to an area in the back of our brain called the visual cortex [4].

2.1.1 V1 of the visual cortex

The visual cortex consists of several layers or areas, and the first one, called V1, is where information is first processed. V1 is also the first place where one finds simple cells, and these cells alter their properties according to what input arrives. Among these properties are direction, polarity and elongation [5]. We are only going to model the direction-property of the simple cells.

2.1.2 Receptive fields of simple cells

The simple cells of V1 present oriented receptive fields and the orientation is defined by the brightness gradients.

2.1.3 Association fields

If an image contains missing information, our brain tries to complete this information. This is done by actively filling in the information based on the simple cell properties. In regions where there is missing information, the cells direct themselves based on the neighbouring cells. This is called local association fields [6].

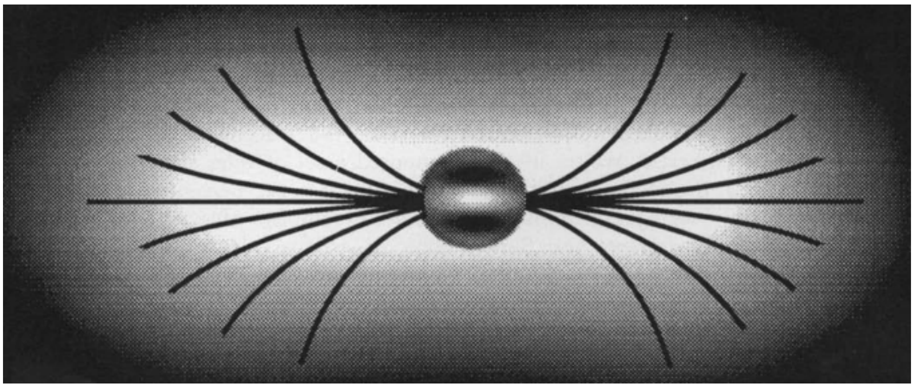


Figure 4: An association field, from the experiments of Field, Hayes and Hess.

In Figure 4 we see an example of an association field. The neighbouring cells of the cell in the center represent the missing information, and their directions are restricted.

2.2 Mathematical model

2.2.1 Extract existing information

In this report we will only consider grayscale digital images, and such an image defines a function:

$$I : (x, y) \to \mathbb{R}. \tag{2}$$

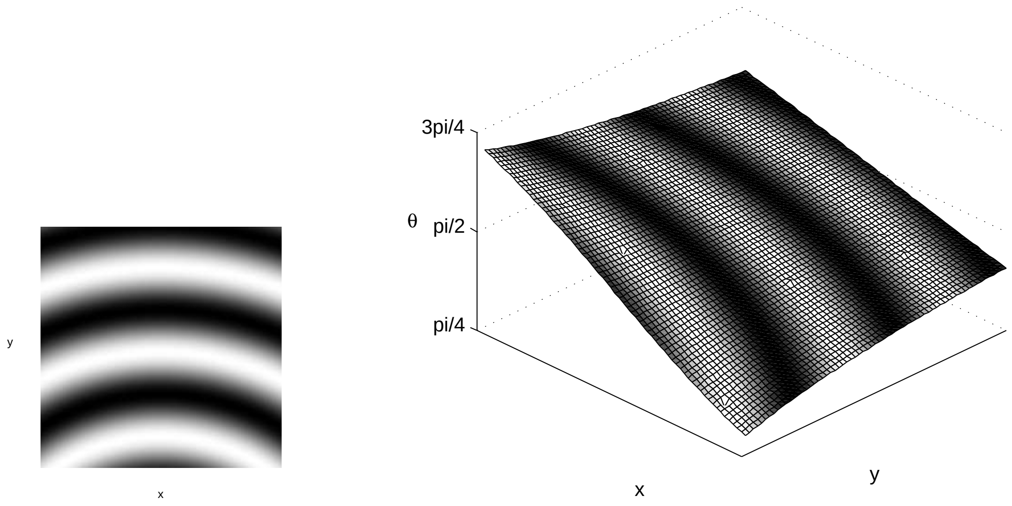
Each discrete point, (x_i, y_i), in the image gives a number which represents the gray-level.

Next, we extract the normal gradient direction at every point as follows:

$$\frac{\nabla I(x,y)}{|\nabla I(x,y)|} = (-\sin\theta, \cos\theta)\,. \tag{3}$$

This is how we model the function of the simple cells in V1, by extracting additional information. Note that the angle θ corresponds to the tangent direction of the level lines of the image I.

We now lift the image to a surface in 3-dimensional space, $(x, y, \theta(x, y))$, which can be seen in Figure 5.



(a) Original image.

(b) Lifted image.

Figure 5: Image lifted in the Roto-translational space.

In this manner the image is represented by a two-dimensional manifold in 3-dimensional space. The 3-dimensional space is called Roto-translational space, and is denoted $\mathbb{R}^2 \times S^1$, where S^1 is the unit circle.

Note that in the representation in Figure 5(b), angles that differ by π are identified. In this way, only one half of the unit circle is used, i.e., $\theta \in [0, \pi]$. Later we will see that this procedure can be used in the numerical calculations, but in the theoretical work we use the "whole" of S^1.

If we look at the tangent vector to the level-lines of the image at a point on the manifold, we note that this tangent vector lies in the plane spanned by the vectors

$$\begin{aligned} \vec{X}_1 &= (\cos\theta, \sin\theta, 0), \\ \vec{X}_2 &= (0, 0, 1). \end{aligned} \tag{4}$$

The directional derivatives, or the vector fields, associated to the vectors above are denoted

$$\begin{aligned} X_1 &= \cos\theta\partial_x + \sin\theta\partial_y, \\ X_2 &= \partial_\theta. \end{aligned} \tag{5}$$

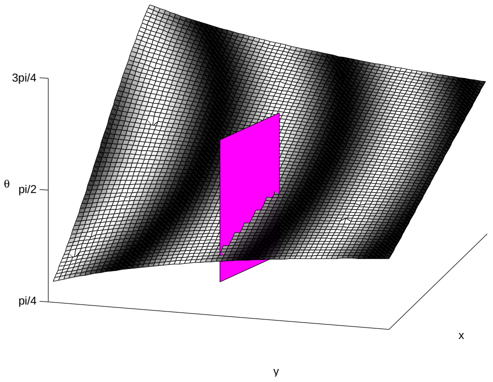


Figure 6: Visualization of the plane spanned by $\vec{X}_1$ and $\vec{X}_2$.

These vector fields generates the Lie algebra of the Lie group $\mathbb{R}^2 \times S^1$, which we will show later.

The vector field X_1, which can be seen in Figure 7, is varying smoothly with θ while X_2 is constant.

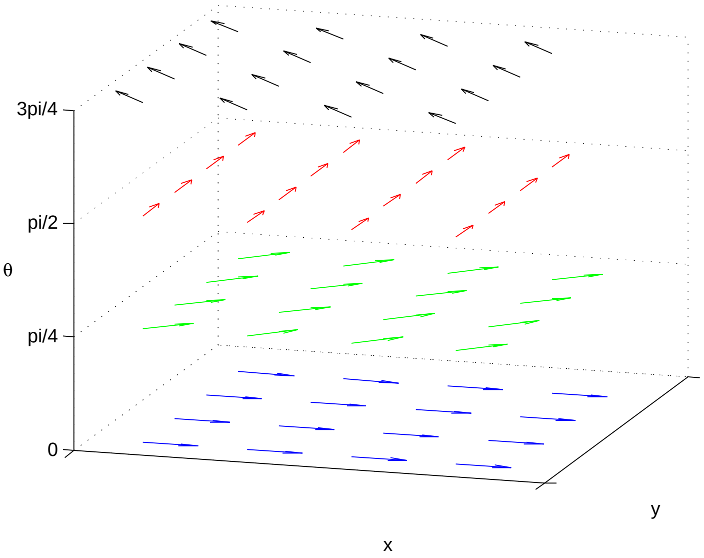


Figure 7: The vector field X_1 is a function of θ.

2.2.2 Filling in the missing information

Diffusion

The image is now represented as a surface in three-dimensional space. The second step in the model is a mechanism that propagates extracted information in an orientation specific way, related to the association fields discussed in

Section 2.1.3. In the mathematical sense, we now diffuse over a certain time interval with the vector fields defined in Equation 5, from the initial surface.

“Non Maximal Suppression”

After the diffusion, the surface has expanded in the θ-direction due to the vector field X_2. Because of this, we need a concentration process to recover a surface. This is achieved by defining a surface, Σ, so that the value of the function u that we diffused will attain its maximum on this surface. The surface can be expressed as

$$\Sigma = \{(x, y, \theta), \partial_\theta u = 0, \partial_\theta^2 u < 0\}. \tag{6}$$

Combined process

The resulting algorithm is then alternating between these two processes, and should allow us to complete missing information based on the geometry of the Roto-translational space. As proved in [3], this procedure will converge to a stationary surface, and no stopping-criterion has to be applied.

3 Lie Groups

In this chapter, we are going to show that the Roto-translational space, $\mathbb{R}^2 \times S^1$, is a Lie group. To this aim, we need some definitions which are all collected from [7].

3.1 Mathematical foundation

Definition 1. *A **Lie group** is a group G which also carries the structure of a smooth manifold in such a way that both the group operation*

$$m : G \times G \to G, \quad m(g,h) = g \cdot h, \quad g, h \in G,$$

and the inversion

$$i : G \to G, \quad i(g) = g^{-1}, \quad g \in G,$$

are smooth maps between manifolds.

Definition 2. *Let C be a smooth curve on a manifold $\mathcal{M}$, parameterized by $\phi : I \to \mathcal{M}$, where I is a subinterval of $\mathbb{R}$. In local coordinates $x = (x^1, ..., x^n)$, C is given by n smooth functions $\phi(\epsilon) = (\phi^1(\epsilon), ..., \phi^n(\epsilon))$ of the real variable ϵ. At each point $x = \phi(\epsilon)$ of C the curve has a **tangent vector***

$$\mathbf{v}|_x = \dot{\phi}(\epsilon) = \dot{\phi}^1(\epsilon)\frac{\partial}{\partial x^1} + ... + \dot{\phi}^n(\epsilon)\frac{\partial}{\partial x^n}.$$

Definition 3. *The collection of all possible tangent vectors to all possible curves passing through a point $x \in \mathcal{M}$ is called the **tangent space** to $\mathcal{M}$ at x, and is denoted $T\mathcal{M}|_x$.*

Definition 4. *A **vector field** $\mathbf{v}$ on $\mathcal{M}$ assigns a tangent vector $\mathbf{v}|_x \in T\mathcal{M}|_x$ to each point in $\mathcal{M}$, with $\mathbf{v}|_x$ varying smoothly from point to point.*

Definition 5. *An **integral curve** of a vector field $\mathbf{v}$ is a smooth parameterized curve $x = \phi(\epsilon)$ whose tangent vector at any point coincide with the value of $\mathbf{v}$ at the same point:*

$$\dot{\phi}(\epsilon) = \mathbf{v}|_{\phi(\epsilon)}$$

for all ϵ.

An integral curve from Definition 5 is called a maximal integral curve if it is not contained in any longer integral curve. I.e., if $\tilde{\phi} : \tilde{I} \to M$ is any other integral curve with the same initial value $\tilde{\phi}(0) = \phi(0)$, then $\tilde{I} \subset I$ and $\tilde{\phi}(\epsilon) = \phi(\epsilon)$ for $\epsilon \in \tilde{I}$.

Definition 6. *If $\mathbf{v}$ is a vector field, we denote the parameterized maximal integral curve passing through $x \in \mathcal{M}$ by $\exp(\epsilon, x)$ and call $\exp$ the **flow** generated by $\mathbf{v}$.*

Definition 7. *For two vector fields $\mathbf{v}$ and $\mathbf{w}$, their **Lie bracket** is defined through their action on functions as*

$$[\mathbf{v}, \mathbf{w}](f) = \mathbf{v}(\mathbf{w}(f)) - \mathbf{w}(\mathbf{v}(f)).$$

Definition 8. *Let G be a Lie group. For any element $g \in G$, we define the **left multiplication map** $L_g : G \to G$ by*

$$L_g(h) = g \cdot h, \quad \forall h \in G,$$

where $\cdot$ denotes the group operation in G.

Definition 9. *Let $\mathcal{M}$ be a smooth manifold, and $F : \mathcal{M} \to \mathcal{M}$ be a smooth map. Each parameterized curve $C = \{\phi(\epsilon)\}$ on $\mathcal{M}$ is mapped by F to a parameterized curve $\widetilde{C} = F(C) = \{\widetilde{\phi} = F(\phi(\epsilon))\}$ in $\mathcal{M}$. Thus F induces a map from the tangent vector $d\widetilde{\phi}/d\epsilon$ to $\widetilde{C}$ at the image point $F(x) = F(\phi(\epsilon)) = \widetilde{\phi}(\epsilon)$. This induced map is called the **differential** of F, and denoted by*

$$dF(\dot{\phi}(\epsilon)) = \frac{d}{d\epsilon}\{F(\phi(\epsilon))\}.$$

If the map F in Definition 9 is one-to-one, the differential then maps the tangent space to $\mathcal{M}$ at x to the tangent space to $\mathcal{M}$ at $F(x)$:

$$dF : T\mathcal{M}|_x \to T\mathcal{M}|_{F(x)}.$$

Further, in local coordinates, the differential of F of a vector field $\mathbf{v}|_x = \sum_{i=1}^{n} \xi^i(x)\partial/\partial x^i$ is

$$dF(\mathbf{v}|_x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \xi^j \frac{\partial F^i}{\partial x^j}(x) \right) \frac{\partial}{\partial x^i}.$$

Definition 10. *A vector field $\mathbf{v}|_x$ on G is called **left-invariant** if*

$$dL_g(\mathbf{v}|_x) = \mathbf{v}|_{L_g(x)}, \quad \forall g, x \in G.$$

Definition 11. *The **Lie algebra** of a Lie group G, denoted $\mathfrak{g}$, is the vector space of all left-invariant vector fields on G.*

3.2 Roto-translational space

Our three-dimensional space $\mathbb{R}^2 \times S^1$ is called Roto-translational space. The infinite set of points (x, y, θ), combined with a composition law that satisfy certain axioms, is then called a group.

3.2.1 Extension to matrix group

The group $\mathbb{R}^2 \times S^1$ has an equivalent matrix form, namely SE(2). This matrix group is called rigid motion, and is defined as

$$\mathrm{SE}(2) = \{g \in \mathbb{R}^{3\times 3} : g = \begin{bmatrix} R & r \\ \mathcal{O} & 1 \end{bmatrix}, R \in \mathrm{SO}(2), r \in \mathbb{R}^2\}, \tag{7}$$

where SO(2) is the special orthogonal group. An element in SO(2) can be expressed as[2]

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix},$$

[2] Brackets are here used to represent matrices and vectors. When Lie brackets are used, this is clear from the context.

which is a rotation in two-dimensional space, and we see that it is only defined by the angle θ.

Since our points on the image-oriented manifold is represented by the triplet (x, y, θ), we can also represent this point as an element $g \in \mathrm{SE}(2)$ as

$$g = \begin{bmatrix} \cos\theta & -\sin\theta & x \\ \sin\theta & \cos\theta & y \\ 0 & 0 & 1 \end{bmatrix}. \tag{8}$$

The special orthogonal matrix, denoted R in (7), is constructed using the vector $\vec{X}_1$ and a vector orthogonal to $\vec{X}_1$.

3.3 The composition law

Since the group operation for matrix groups is the usual matrix-matrix product, we can easily find the composition law in the Roto-translational group by using the matrix representations. For any two points $(x_a, y_a, \theta_a), (x_b, y_b, \theta_b) \in \mathbb{R}^2 \times S^1$, we find the product through the following transformation:

$$\begin{bmatrix} x_a \\ y_a \\ \theta_a \end{bmatrix} +_R \begin{bmatrix} x_b \\ y_b \\ \theta_b \end{bmatrix} \sim g_a \cdot g_b = g_c \sim \begin{bmatrix} x_c \\ y_c \\ \theta_c \end{bmatrix},$$

where g_a, g_b and g_c are the corresponding matrix-elements. The matrix-elements g_a and g_b are defined above and we get:

$$\begin{aligned} g_c = g_a \cdot g_b &= \begin{bmatrix} \cos\theta_a & -\sin\theta_a & x_a \\ \sin\theta_a & \cos\theta_a & y_a \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta_b & -\sin\theta_b & x_b \\ \sin\theta_b & \cos\theta_b & y_b \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} (c_a c_b - s_a s_b) & (-c_a s_b - s_a c_b) & (c_a x_b - s_a y_b + x_a) \\ (s_a c_b + c_a s_b) & (-s_a s_b + c_a c_b) & (s_a x_b + c_a y_b + y_a) \\ 0 & 0 & 1 \end{bmatrix}, \end{aligned}$$

where s_i and c_i denotes $\sin\theta_i$ and $\cos\theta_i$, respectively, for $i = a, b$.

With the trigonometric identities

$$\begin{aligned} \sin(\alpha \pm \beta) &= \sin\alpha\cos\beta \pm \cos\alpha\sin\beta, \\ \cos(\alpha \pm \beta) &= \cos\alpha\cos\beta \mp \sin\alpha\sin\beta, \end{aligned} \tag{9}$$

the matrix g_c simplifies to

$$g_c = \begin{bmatrix} \cos(\theta_a + \theta_b) & -\sin(\theta_a + \theta_b) & \cos\theta_a x_b - \sin\theta_a y_b + x_a \\ \sin(\theta_a + \theta_b) & \cos(\theta_a + \theta_b) & \sin\theta_a x_b + \cos\theta_a y_b + y_b \\ 0 & 0 & 1 \end{bmatrix}. \tag{10}$$

Clearly this matrix is also in SE(2), as expected.

If we now convert back to vector-notation we get the composition-law in the Roto-translational space:

$$\begin{bmatrix} x_a \\ y_a \\ \theta_a \end{bmatrix} +_R \begin{bmatrix} x_b \\ y_b \\ \theta_b \end{bmatrix} = \begin{bmatrix} \cos\theta_a x_b - \sin\theta_a y_b + x_a \\ \sin\theta_a x_b + \cos\theta_a y_b + y_a \\ \theta_a + \theta_b \end{bmatrix}, \tag{11}$$

where we have adopted the notation $+_R$ from [3] as the group operation[3].

[3] The seemingly commutative notation $+_R$ is used. However, our group operation is non-commutative.

3.3.1 Verification of the group axioms

Now that we have our composition law, we can check if it has the necessary properties of a group operation. It has to be associative, there has to exist an identity element and each element should have an inverse.

Associativity means that for elements $a, b, c \in \mathbb{R}^2 \times S^1, a +_R (b +_R c) = (a +_R b) +_R c$. This follows from the transformation, since matrix-matrix multiplication is associative.

Next, the identity element is the trivial triplet (0,0,0), (which gives the identity matrix in the matrix-group):

$$(0,0,0) +_R (x,y,\theta) = (x,y,\theta) = (x,y,\theta) +_R (0,0,0). \tag{12}$$

Last, the inverse of an arbitrary element $g = (x, y, \theta)$ is

$$g^{-1} = (-\cos\theta x - \sin\theta y, \sin\theta x - \cos\theta y, -\theta). \tag{13}$$

So equipped with the composition law we have a group structure.

3.3.2 $\mathbb{R}^2 \times S^1$ is a Lie Group

From above we have that both the group operation and the inversion of elements are smooth maps. Further, since our group carries the structure of a manifold, i.e. the image-oriented manifold, $\mathbb{R}^2 \times S^1$ is a Lie group according to Definition 1.

3.4 Vector fields and flows

In Section 2.2.1 we introduced the two vector fields $X_1 = \cos\theta\partial_x + \sin\theta\partial_y$ and $X_2 = \partial_\theta$. These satisfy the conditions in Definition 4, since X_1 is varying smoothly with θ, and X_2 is constant.

The flow map (Definition 6) applied to the point (x_0, y_0, θ_0) of a linear combination of these vector fields is a curve denoted as

$$\gamma(t) = \exp(t, X_1 + kX_2)(x_0, y_0, \theta_0). \tag{14}$$

This curve is the solution of the system of ordinary differential equations $\dot{x}(t) = \cos\theta(t), \dot{y}(t) = \sin\theta(t), \dot{\theta}(t) = k$. With $k \neq 0$, we get:

$$\begin{aligned} x(t) &= \frac{1}{k}\left(\sin(kt+\theta_0) - \sin\theta_0\right) + x_0, \\ y(t) &= \frac{1}{k}\left(-\cos(kt+\theta_0) + \cos\theta_0\right) + y_0, \\ \theta(t) &= kt + \theta_0. \end{aligned} \tag{15}$$

The parameter k in this representation is the curvature of the integral curve. With $k = 0$, the solution will be a straight line with constant $\theta = \theta_0$. Figure 8 shows an integral curve starting from $(0, 0, 0)$ with $k = 1$.

In Figure 9 we see curves projected to the xy-plane, with k varying. This plot is very similar to the figure from the experiments of Field, Hayes and Hess in Figure 4. So the vector fields X_1 and X_2 are going to play the role of the basis for the association fields discussed in Section 2.1.3.

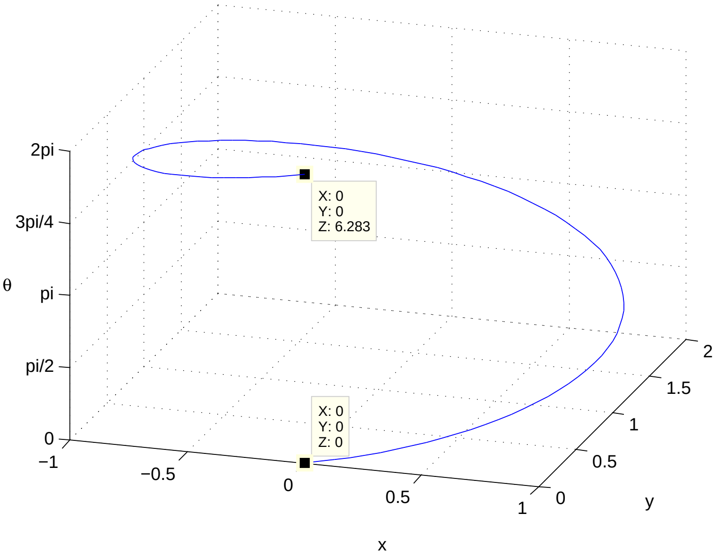


Figure 8: Solution of the integral curve $\dot{\gamma}(t) = \vec{X}_1(\gamma) + k\vec{X}_2(\gamma)$, with $k = 1$.

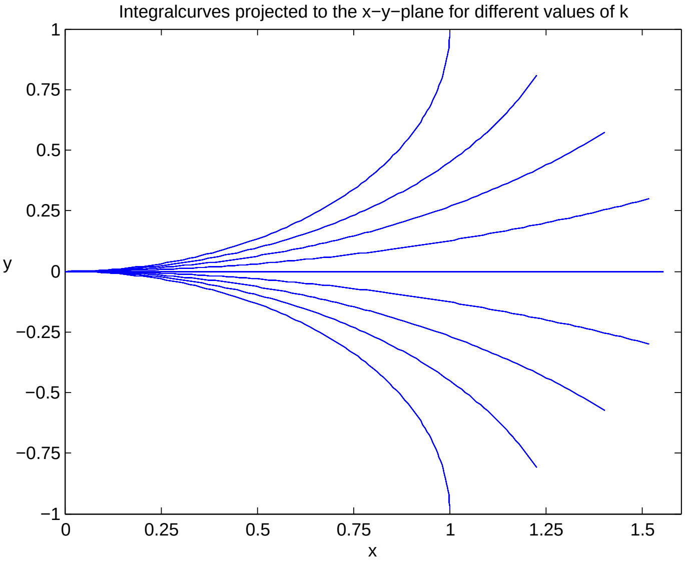


Figure 9: Projected solutions of the integral curves $\dot{\gamma}(t) = \vec{X}_1(\gamma) + k\vec{X}_2(\gamma)$, with $k \in [-1, 1]$.

3.5 Lie brackets

The Lie bracket (Definition 7) of the vector fields X_1 and X_2 can be found through their action on an arbitrary smooth function u, as

$$\begin{aligned} [X_1, X_2]u &= X_1(X_2 u) - X_2(X_1 u) \\ &= (\cos\theta\partial_x + \sin\theta\partial_y)u_\theta - \partial_\theta\left(\cos\theta u_x + \sin\theta u_y\right) \\ &\equiv -X_3 u, \end{aligned} \tag{16}$$

where we have defined the third vector field X_3 as

$$X_3 = -\sin\theta\partial x + \cos\theta\partial y. \tag{17}$$

The other commutators are obtained similarly:

$$\begin{aligned} [X_2, X_3] &= -X_1, \\ [X_1, X_3] &= 0. \end{aligned} \tag{18}$$

This means that the vector fields X_1, X_2 and X_3 form a closed set with respect to the Lie bracket operation. Also, the two first vector fields generate the last. If we denote by $\mathcal{L}(X_1, X_2)$ the set of all linear combinations of X_1, X_2 and the commutators of any order, it is spanned by the three vector fields.

3.6 Left multiplication map

Based on the composition-law of the group $\mathbb{R}^2 \times S^1$, we can define the left multiplication map as in Definition 8:

$$L_g(h) = g +_R h, \qquad \forall g, h \in \mathbb{R}^2 \times S^1. \tag{19}$$

Since every element has an inverse, the left multiplication map is one-to-one:

$$L_g(x) = L_g(y) \Longleftrightarrow x = y.$$

3.6.1 Differential of the left multiplication map

The differential of the map $L_g(x)$ is denoted $dL_g(x)$ and maps the tangent space at $x \in \mathbb{R}^2 \times S^1$ to the tangent space at $L_g(x) \in \mathbb{R}^2 \times S^1$.

Following Definition 9, we also have an explicit expression in local coordinates for the differential of vector fields. In this expression we need the Jacobian of $L_g(x)$. Since the element g is fixed, the Jacobian ($\mathrm{Jac}_{L_g(x)} = \{\partial L_g^i(x)/\partial x^j\}_{i,j=1}^3$) is

$$\mathrm{Jac}_{L_g(x)} = \begin{bmatrix} \cos\theta_g & -\sin\theta_g & 0 \\ \sin\theta_g & \cos\theta_g & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{20}$$

If we now insert X_1 in the expression of the differential of L_g, we get

$$\begin{aligned} dL_{(x_g,y_g,\theta_g)}(X_1|_{(x,y,\theta)}) &= \sum_{j=1}^{3}\left[\left(\cos\theta\frac{\partial L_g^j}{\partial x}\right)|_{(x,y,\theta)} + \left(\sin\theta\frac{\partial L_g^j}{\partial y}\right)|_{(x,y,\theta)}\right] \\ &= \left(\cos\theta\cos\theta_g - \sin\theta\sin\theta_g\right)\partial_x + \left(\cos\theta\sin\theta_g + \sin\theta\cos\theta_g\right)\partial_y \\ &= \cos(\theta+\theta_g)\partial_x + \sin(\theta+\theta_g)\partial_y, \end{aligned} \tag{21}$$

where the last equality follows from the identities in (9). From the composition law we note that this reduces to

$$dL_g(X_1|_x) = X_1|_{L_g(x)}, \tag{22}$$

and that, by Definition 10, X_1 is a left-invariant vector field.

Similar to above, we find that X_2 and X_3 are also left invariant vector fields.

3.7 Lie algebra of the Lie group

Since the vector fields X_1, X_2 and X_3 are all smooth, left-invariant vector fields, and since the dimension of $\mathbb{R}^2 \times S^1$ is three, these vector fields form a basis of the Lie algebra, $\mathfrak{g}$, of the Roto-translational group by Definition 11. We denote this as

$$\mathfrak{g} = \mathcal{L}(X_1, X_2). \tag{23}$$

4 Mathematical Model

Here we build up our mathematical model based on the neurophysiologic aspects discussed in Chapter 2, and using the geometric space described by our Lie algebra of the Roto-translational group discussed in Chapter 3.

4.1 Extraction of existing information

Receptive fields of simple cells are usually modeled as convolution kernels of filters, such as Gabor filters [8],

$$G(x, y, \theta) = \frac{1}{2\pi s} \exp\left(-\frac{\tilde{x}^2 + \tilde{y}^2}{s^2} + i\tilde{y}/s \right),$$

where

$$\begin{aligned} \tilde{x} &= x\cos(\theta) + y\sin(\theta), \\ \tilde{y} &= -x\sin(\theta) + y\cos(\theta), \end{aligned}$$

and the parameter s is the standard deviation in the Gaussian envelope. The authors of [3] have shown that up to a multiplicative constant, the odd part of this Gabor filter can locally be approximated as

$$\approx -\partial_{\tilde{y}} \exp(-(\tilde{x}^2 + \tilde{y}^2)/s^2). \tag{24}$$

This is equivalent to the directional derivative in the direction $(-\sin(\theta), \cos(\theta))$, which is the direction of the vector field we called X_3 above.

A three-dimensional function, $u_0(x, y, \theta)$, is constructed, where the two first dimensions correspond to the image's domain. We now set each value in u_0 as

$$u_0(x, y, \theta) = -X_3(\theta)I_s, \tag{25}$$

where I_s denotes the discrete convolution of the original image with the convolution kernel $\exp(-(\tilde{x}^2 + \tilde{y}^2)/s^2)$.

By defining the surface

$$\Sigma = \{(x, y, \bar{\theta}), |X_3(\bar{\theta})I_s| = \max_{\theta} |X_3(\theta)I_s)|\}, \tag{26}$$

we set $u = u_0\delta_\Sigma$. That is, we set u to be u_0 concentrated on the surface Σ.

This is how we model the simple cells property to choose the direction that results in the maximum of activity. It is also equivalent to the procedure we did in Section 2.2.1, where we found the θ-value that corresponded to the normal gradient direction.

Note that since the function u is constructed using derivatives of the original image, some information is lost through this transformation. If the original image contains domains of constant gray level, the interior of these domains will all be represented with the value 0 in u, since the derivative here is 0 in every direction.

4.2 Completion of the missing information

4.2.1 Diffusion

Now we diffuse our function $u(x, y, \theta)$ with the vector fields X_1 and X_2. In the Roto-translational space, we denote the R-gradient and the sub-Laplacian, respectively, as

$$\begin{aligned} \nabla_R &= (X_1, X_2)^T, \\ \Delta_R &= \nabla_R^T \cdot \nabla_R = X_1^2 + X_2^2, \end{aligned} \tag{27}$$

where X_i^2 represents the i'th vector field applied to itself. The diffusion then becomes

$$u_t = \Delta_R u = X_1(X_1 u) + X_2(X_2 u) = (X_1^2 + X_2^2)u. \tag{28}$$

4.2.2 “Non Maximal Suppression”

After we have diffused for a certain time, the surface has expanded in the θ-direction. Now we apply a concentration process, which the authors of [3] have called “Non Maximal Suppression”, which is needed to explain the sharp tuning exhibited by the simple cells in V1.

At each point, (x, y), we want to find the θ-value for which our function u attains its maximum. This can be achieved by demanding that $X_2 u = 0$ and $X_2 (X_2 u) < 0$ at each point. Our new surface can then be defined with the function $v \equiv X_2 u$ as:

$$\Sigma = \{(x, y, \theta) : v = 0, X_2 v < 0\}, \quad v = X_2 u. \tag{29}$$

We note that

$$\Sigma \subseteq \{(x, y, \theta) : v = 0, X_2 v \neq 0\}, \tag{30}$$

and by assuming only maxima, they are equal. From [9], Σ is a regular surface, and we are guaranteed the existence of a tangent plane at all points in Σ. Further, Σ is the 0-level set of the function v.

4.2.3 Projection on tangent planes

At each point, $\mathbf{x}$, on a two-dimensional manifold M in $\mathbb{R}^3$ we have a normal-vector ν. If we denote the projection matrix P_ν as

$$P_\nu = I - \frac{\nu \otimes \nu}{|\nu|^2},$$

this matrix will project any vectors at $\mathbf{x}$ onto the tangent plane of the surface at $\mathbf{x}$. Note that the elements of P_ν can be expressed as

$$(P_\nu)_{ij} = \delta_{ij} - \frac{\nu_i \nu_j}{|\nu|^2},$$

where δ_{ij} is the Kronecker delta function. For a vector field $X \in \mathbb{R}^3$, we define the differential operator $P_X \nabla$ component-wise as

$$(P_X \nabla)_i = \sum_{j=1}^{3} \left(\delta_{ij} - \frac{X_i X_j}{|X|^2} \right) \partial_{x_j}. \tag{31}$$

4.2.4 Diffusion constrained on the surface

By using our previous notation, we find $P_X\nabla_R$ by exchanging the ∂_{x_j} with X_j in (31). Now we use the geometry of our space to derive a level set method in $\mathbb{R}^2 \times S^1$.

From the level-set function in (30), we define the unit-vector

$$\nu = \frac{\nabla_R v}{|\nabla_R v|} = \frac{(X_1 v, X_2 v)}{\sqrt{(X_1 v)^2 + (X_2 v)^2}}. \tag{32}$$

Note that we can expand ∇_R to be a three dimensional vector, $\nabla_R = (X_1, 0, X_2)^T$, and obtain the same results.

We now know that ν (expanded to a three-dimensional vector) is a normal vector to Σ, and from Section 4.2.3 we have that the matrix P_ν projects vectors onto the tangent plane in Σ.

For a function u on Σ, we define the Eulerian surface gradient as

$$\nabla_\nu u = P_\nu \nabla u, \tag{33}$$

and the Eulerian Laplace-Beltrami operator as the tangential divergence of the tangential gradient [10]:

$$\Delta_\nu u = \nabla_\nu \cdot \nabla_\nu u. \tag{34}$$

When going from Eulerian- to Roto-translational-space, we have to change all the ∇-operators to $\nabla_R = (X_1, X_2)^T$. The surface R-divergence of the surface R-gradient of the function u, is

$$\operatorname{div}_R^\nu (\nabla_R^\nu u) = (\nabla_R^\nu)^T \cdot \nabla_R^\nu u = (P_\nu \nabla_R)^T \cdot P_\nu \nabla_R u. \tag{35}$$

A simple calculation shows that

$$P_\nu = P_{(X_1 v, X_2 v)} = \begin{bmatrix} \left(1 - \frac{(X_1 v)^2}{|Xv|^2}\right) & -\frac{X_1 v X_2 v}{|Xv|^2} \\ -\frac{X_2 v X_1 v}{|Xv|^2} & \left(1 - \frac{(X_2 v)^2}{|Xv|^2}\right) \end{bmatrix}, \tag{36}$$

and since our projection matrix satisfies $P_\nu^2 = P_\nu = P_\nu^T$, we find that

$$\begin{aligned} \operatorname{div}_R^\nu (\nabla_R^\nu u) &= \left(\begin{bmatrix} \left(1 - \frac{(X_1 v)^2}{|Xv|^2}\right) & -\frac{X_1 v X_2 v}{|Xv|^2} \\ -\frac{X_2 v X_1 v}{|Xv|^2} & \left(1 - \frac{(X_2 v)^2}{|Xv|^2}\right) \end{bmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \right)^T \cdot \begin{pmatrix} X_1 u \\ X_2 u \end{pmatrix} \\ &= \left(1 - \frac{(X_1 v)^2}{|Xv|^2}\right) X_1(X_1 u) - \frac{X_1 v X_2 v}{|Xv|^2} X_2(X_1 u) \\ &\quad - \frac{X_1 v X_2 v}{|Xv|^2} X_1(X_2 u) + \left(1 - \frac{(X_2 v)^2}{|Xv|^2}\right) X_2(X_2 u). \end{aligned} \tag{37}$$

When inserting $|Xv|^2 = (X_1 v)^2 + (X_2 v)^2$, and using the notation $X_i(X_j u) \equiv X_{ij} u$ we get the final equation:

$$\Delta_R^\nu u = \operatorname{div}_R^\nu (\nabla_R^\nu u) = \frac{(X_2 v)^2 X_{11} u + (X_1 v)^2 X_{22} u}{(X_1 v)^2 + (X_2 v)^2} - \frac{(X_{12} u + X_{21} u) X_1 v X_2 v}{(X_1 v)^2 + (X_2 v)^2}. \tag{38}$$

After the diffusion $u_t = \Delta_R u$, we then calculate the time evolution

$$u_t = \Delta_R^\nu u. \tag{39}$$

The two equations are then combined in an alternating fashion. First we diffuse (28) over a certain time interval t_1, and then integrate (39) over a time interval t_2. Then we do it all over again, and solve the pair of equations for a total of N times.

4.3 Post processing

After applying the mechanisms described above on an image, the completed image is represented as a three dimensional function. To convert it back to an image again, we set:

$$I(x,y) = \max_\theta \left(|u(x,y,\theta)|\right). \tag{40}$$

This procedure will give us the direction that results in maximum response in the simple cells of V1.

5 Numerical Discretization

5.1 Discretized function

Since the images we are considering are digital (and thus discrete), the dimensions in the x- and y-directions are given by the pixel resolutions.

The third dimension is found when $\Delta\theta$, the step size in the θ-direction, is chosen. The interval $[0, 2\pi]$ is divided into approximately $2\pi/\Delta\theta$ points.

5.2 Step size

When approximating the equations with finite differences, we need a step size in all the directions. In the results below, only square images have been considered, so the step size in the y-direction, Δy, has been set equal to Δx.

There is no natural way to define the step size Δx though. The size of a digital image is only defined by the pixel resolution, and since Δx is supposed to represent the distance between two pixels, this must be chosen.

In all the results below, the step size in the θ-direction has been chosen to be equal to Δx for simplicity. This imposes restrictions on the choice of Δx. Since the dimension in the θ-direction and Δx are inverse proportional, Δx must be chosen wisely.

The time step Δt is restricted by the Courant-Friedrichs-Lewy condition, which we will discuss below in Section 5.5.

5.3 The differential equations

We approximate the equations with finite differences. The derivatives in x, y and θ are approximated using second order centered differences, and differentiation in time is approximated by a first order forward difference. This means that we use the forward Euler method for the time integration.

We denote u^n_{lmq} as the function u evaluated in the discrete node $(x, y, \theta) = (l\Delta x, m\Delta y, q\Delta\theta)$ at time $t = n\Delta t$. With an image resolution of $L \times M$ pixels, l can be in the range $[0, ..., L-1]$, and m in the range $[0, ..., M-1]$. Further, q is in the range $[0, ..., Q-1]$, where $Q = \lceil 2\pi/\Delta\theta \rceil$ is the dimension in the θ-direction. The various derivatives are approximated as

$$\begin{aligned}
D_1 u^n_{lmq} &= \cos\theta_q \delta_x u^n_{lmq} + \sin\theta_q \delta_y u^n_{lmq}, \\
D_2 u^n_{lmq} &= \delta_\theta u^n_{lmq}, \\
D_{11} u^n_{lmq} &= \cos^2\theta_q \delta_x^2 u^n_{lmq} + 2\sin\theta_q\cos\theta_q \delta_x\delta_y u^n_{lmq} + \sin^2\theta_q \delta_y^2 u^n_{lmq}, \\
D_{22} u^n_{lmq} &= \delta_\theta^2 u^n_{lmq}, \\
D_{12} u^n_{lmq} &= \cos\theta_q \delta_x\delta_\theta u^n_{lmq} + \sin\theta_q \delta_y\delta_\theta u^n_{lmq}, \\
D_{21} u^n_{lmq} &= -\sin\theta_q \delta_x u^n_{lmq} + \cos\theta_q \delta_x\delta_\theta u^n_{lmq} + \cos\theta_q \delta_y u^n_{lmq} + \sin\theta_q \delta_y\delta_\theta u^n_{lmq},
\end{aligned} \tag{41}$$

where $D_i u$ is the approximation of the vector field X_i applied to u, and $D_{ij}u$ is the approximation of the vector field X_i applied to $X_j u$. Furthermore, δ_i denotes the second order centered difference in the direction i, for $i = x, y, \theta$. For example, $\delta_x^2 u^n_{lmq} = 1/\Delta x^2 (u^n_{l-1,m,q} - 2u^n_{lmq} + u^n_{l+1,m,q})$.

The combined numerical scheme is then:

```
FOR $n = 0, ..., N_1 - 1,$ DO:
    $u_{lmq}^{n+1} = u_{lmq}^{n} + \Delta t \left( D_{11} u_{lmq}^{n} + D_{22} u_{lmq}^{n} \right)$
END FOR
$v_{lmq}^{N_1} = D_2 u_{lmq}^{N_1}$
FOR $n = N_1, ..., N_1 + N_2 - 1,$ DO:
    $u_{lmq}^{n+1} = u_{lmq}^{n} + \Delta t \Big[ \frac{(D_2 v_{lmq}^{n})^2 D_{11} u_{lmq}^{n} + (D_1 v_{lmq}^{n})^2 D_{22} u_{lmq}^{n}}{(D_1 v_{lmq}^{n})^2 + (D_2 v_{lmq}^{n})^2} - \frac{D_1 v_{lmq}^{n} D_2 v_{lmq}^{n} \left( D_{12} u_{lmq}^{n} + D_{21} u_{lmq}^{n} \right)}{(D_1 v_{lmq}^{n})^2 + (D_2 v_{lmq}^{n})^2} \Big]$    (42)
    $v_{lmq}^{n+1} = D_2 u_{lmq}^{n+1}$
END FOR.
```

In this way we diffuse u for a time $N_1 \Delta t$ and then apply the "Non Maximal Suppression" for a time $N_2 \Delta t$. This numerical scheme is applied for a total of N times.

5.4 Boundary conditions

To apply the numerical scheme in (42), we need boundary conditions. In the θ-direction, it is natural to define periodic boundary conditions.

In the x- and y-directions, the authors of [3] use Neumann boundary conditions. In the results below, regular homogeneous Neumann boundary conditions was first used. This means that we set $\partial u / \partial \vec{n} = 0$ on the boundaries, where $\vec{n}$ is the outward normal direction. Since the image is a rectangular (quadratic) domain, the directional derivatives in the normal directions are easy to obtain.

The Neumann boundary conditions did not give satisfactory results, the image was distorted near the boundaries. Thus the Neumann boundary conditions were adjusted to the Roto-translational setting. Instead of using outward normal directional derivatives, we used the vector field X_1. This can be expressed as

$$\begin{aligned} X_1 u_{0,m,q} = X_1 u_{L-1,m,q} = 0, \quad & \forall m, q, \\ X_1 u_{l,0,q} = X_1 u_{l,M-1,q} = 0, \quad & \forall l, q. \end{aligned} \tag{43}$$

5.4.1 Discretizing the boundary conditions

When using fictive nodes in the derivation of the derivatives at the boundaries, the periodic boundary conditions are imposed by setting

$$\begin{aligned} u_{l,m,-1} = u_{l,m,Q-1} \quad & \forall l, m, \\ u_{l,m,Q} = u_{l,m,0} \quad & \forall l, m. \end{aligned} \tag{44}$$

Further, for the regular Neumann boundary conditions in x- and y-directions, we find the fictive nodes from a second ordered centered difference on the boundaries:

$$\begin{aligned} u_{-1,m,q} &= u_{1,m,q} & \forall m,q,\\ u_{L,m,q} &= u_{L-2,m,q} & \forall m,q,\\ u_{l,-1,q} &= u_{l,1,q} & \forall l,q,\\ u_{l,M,q} &= u_{l,M-2,q} & \forall l,q. \end{aligned} \tag{45}$$

When imposing “Roto-translational” Neumann boundary conditions, we get more complicated equations, since the vector field X_1 includes derivatives in both the x- and y-direction. For example, at the left boundary we get

$$u_{-1,m,q} = \frac{\sin\theta_q}{\cos\theta_q}\Big(u_{0,m+1,q} - u_{0,m-1,q}\Big) + u_{1,m,q}, \tag{46}$$

where $\theta_q = q\Delta\theta$. Obviously, if the θ-value associated to a node is a multiple of $\pi/2$ somewhere on the boundary, some problems will occur. If $\theta = \pi/2$ in the example above, the fraction in the equation will have a zero denominator. In this case, $u_{0,m-1,q} = u_{0,m+1,q}$ when imposing $X_1 u = 0\,(u_{l+1,m,q} - u_{l-1,m,q}) + (u_{l,m+1,q} - u_{l,m-1,q})$
$= 0$. Secondly, derivatives in the x-directions have been obtained using a first order finite difference into the domain, leading to an absence of boundary conditions.

5.4.2 Corners

We also experienced a problem with boundary conditions at the corners. If the θ-value at a corner in the xy-plane represents a direction that never enters the domain (see Figure 10), this can lead to problems. With regular Neumann boundary conditions this problem does not occur, since the normal direction is either vertical or horizontal, and both directions can be imposed at corners.

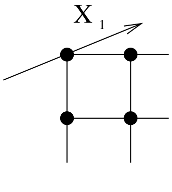


Figure 10: $\vec{X}_1$ can point out of the domain.

Also, using second order centered differences for the mixed derivative, u_{xy}, at e.g. the lower left corner, we need the fictive node $u_{-1,-1,q}$. To avoid these problems, we use a first order difference approximation for the mixed derivatives at these nodes.

5.5 The associated diffusion-matrix

The numerical approximation of the diffusion (first part of the scheme in (42)) with the forward Euler method can be seen as an iteration from a starting vector. In practice, the solution u^n_{lmq} is not a vector, but the iteration can be viewed in terms of a matrix operating on a vector as

$$u^{n+1}_{lmq} = Au^n_{lmq}, \tag{47}$$

since the diffusion equation is linear.

To ensure convergence, we have from the Courant-Friedrich-Lewy-condition (CFL) [11] that the spectral radius of A must be less than or equal to 1. Otherwise, the diffusion part of the iteration in (47) can produce wrong results.

The equation of an “inner node”, u_{lmq}, in the diffusion is as follows

$$\begin{aligned} u^{n+1}_{lmq} &= u^n_{lmq} + \Delta_t \left(X_1^2 u^n_{lmq} + X_2^2 u^n_{lmq}\right) \\ &\approx u^n_{lmq} + \frac{\Delta_t}{\Delta_x^2}\Big[\quad \cos^2\theta_q \Big(u^n_{l+1,m,q} - 2u^n_{lmq} + u^n_{l-1,m,q}\Big) \\ &+ \frac{1}{2}\cos\theta_q \sin\theta_q \Big(u^n_{l+1,m+1,q} + u^n_{l-1,m-1,q} - u^n_{l+1,m-1,q} - u^n_{l-1,m+1,q}\Big) \\ &\qquad\qquad + \sin^2\theta_q \Big(u^n_{l,m+1,q} - 2u^n_{lmq} + u^n_{l,m-1,q}\Big) \qquad \Big] \\ &\qquad\qquad + \frac{\Delta_t}{\Delta_\theta^2}\left[u^n_{l,m,q+1} - 2u^n_{lmq} + u^n_{l,m,q-1}\right]. \end{aligned} \tag{48}$$

In Figure 11 we see the “molecule” of this diffusion equation. The value of a node at the next time step depends on 11 nodes, including itself, at the previous time step.

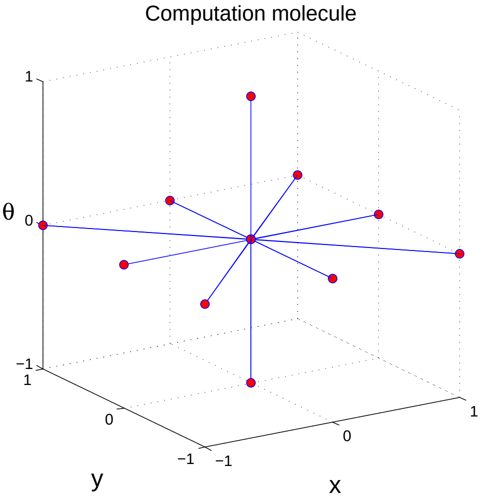


Figure 11: The computation molecule for the diffusion term in the numerical scheme.

From (48), and with $\Delta\theta = \Delta x$, the iteration matrix A can be split into

$$A = I + \frac{\Delta t}{\Delta x^2}\widetilde{A}. \tag{49}$$

We can then find the restrictions on Δt by looking at the eigenvalues of $\widetilde{A}$.

The associated matrix $\widetilde{A}$ is a sparse matrix with at most 11 non-zero entries on a single row. The band width of the matrix depends on the labeling of our u when we map it from $\mathbb{R}^{L\times M\times Q}$ to $\mathbb{R}^{LMQ}$.

If we label the vector as $u = [u_{0,0,0}, u_{1,0,0}, ..., u_{L-1,0,0}, u_{0,1,0}, ..., u_{L-1,1,0}, u_{0,2,0}, ..., ..., u_{L-1,M-1,0}, u_{0,0,1}, ..., ..., u_{L-1,M-1,Q-1}]^T$, the structure of the matrix $\widetilde{A}$ will be as seen in Figure 12 below.

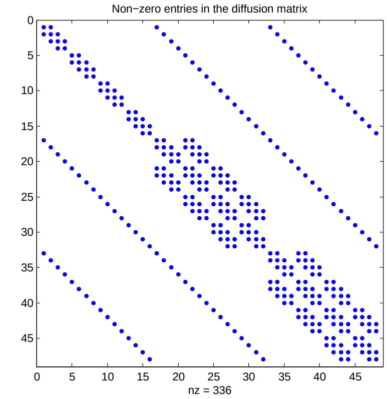


Figure 12: A visualization of the sparse structure of the matrix $\widetilde{A}$.

Figure 12 shows a dot for each non-zero value in $\widetilde{A}$. This particular matrix is constructed using only 4 points in the spatial directions, and 3 points along the θ-direction. We notice the symmetry pattern, and that there are missing some points in the upper left corner because of the sine squared and the mixed term involving a product with sine, since the associated θ-value in this area is zero.

In Table 1, the spectral radius for some different dimensions is shown. When constructing the matrix $\widetilde{A}$, homogeneous boundary conditions are imposed for simplicity.

Dim. in x- and y-direction	Dim. in θ-direction	Calculated spectral radius
10	7	7.7218
20	13	7.9198
40	26	7.9942
60	38	7.9974
80	51	7.9947
100	63	7.9966
120	76	7.9993
155	98	7.9996

Table 1: Table of calculated spectral radii for different dimensions.

The eigenvalues of $\widetilde{A}$ are found to be $\lambda \in (-8, 0)$. I.e. the spectral radius seems to converge towards 8 for increasing spatial resolution.

The eigenvalues of the discrete one-dimensional Laplace operator

$$T = \begin{bmatrix} -2 & 1 & & & 0 \\ 1 & -2 & 1 & & \\ & & \ddots & & \\ & & 1 & -2 & 1 \\ 0 & & & 1 & -2 \end{bmatrix} \tag{50}$$

are known to be on the form

$$\lambda_i = -2\left(1 - \cos\left(\frac{i\pi}{N+1}\right)\right). \tag{51}$$

Here N denotes the dimension, and λ_i is the i'th eigenvalue ($i = 1, ..., N$). The eigenvalues are thus in the range $(-4, 0)$, and the minimum converges to -4 for high dimensions. Since we find the spectral radius of our problem to converge towards 8, this shows clear resemblances with diagonalization techniques used for solving the two-dimensional Poisson problem [12]. Such techniques involves a scaling with the summation of two eigenvalues, where each of them is the corresponding eigenvalues of the one-dimensional discrete Laplace operator in (50). The minimum of the sum is thus converging towards -8, as in our problem.

From (49), we see that the eigenvalues of A can be expressed as

$$\mathrm{eig}(A) = \mathrm{eig}\left(I + \Delta t/\Delta x^2 \widetilde{A}\right) = 1 + \Delta t/\Delta x^2 \mathrm{eig}\left(\widetilde{A}\right). \tag{52}$$

The CFL-condition then gives the following restrictions on Δt:

$$\begin{aligned} 1 - 8\frac{\Delta t}{\Delta x^2} &\leq 1, \\ 1 - 8\frac{\Delta t}{\Delta x^2} &\geq -1. \end{aligned} \tag{53}$$

The first equation is unconditionally satisfied, since both Δt and Δx have to be positive. The second is satisfied when $\Delta t \leq 0.25\Delta x^2$.

6 Results

6.1 Stability

Above, we found a restriction on the time step, Δt. In Figure 13 we see an image before and after applying the numerical scheme in (42) 10 times with $N1 = N2 = 3$ and $\Delta t = 0.4$.

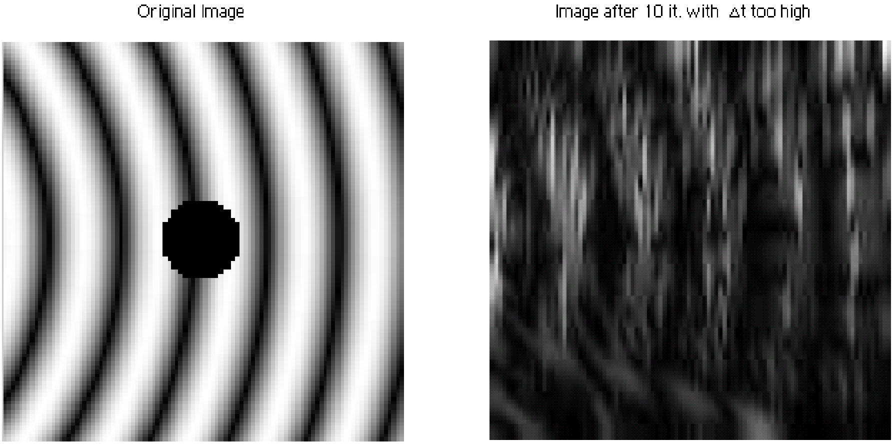


Figure 13: Image before and after 10 iterations with $\Delta t = 0.4$.

The iteration obviously diverged, and the image is no longer recognizable. All the following results, have thus a time step $\Delta t = 0.25$.

6.2 Dependence on N_1 and N_2

In the numerical scheme, the parameters N_1 and N_2 represent the length we integrate in time for the diffusion and "Non-maximal Suppression", respectively. The authors of [3] do not specify which values they use.

In Figure 14 we see a comparison of different combinations of the parameters. 3 values have been tested for the two parameters, making a total of 9 combinations. All calculations were done with regular Neumann b.c. and 20 iterations.

It looks like there is no clear advantage by choosing a specific combination. Since the original image contained a hole in the center, we also see how this hole is closing in the calculations. By following the diagonal downwards, the hole decreases. This is because we integrate in a longer time interval $N\Delta t(N_1 + N_2)$, and not a reason to favour the lower right combination.

In all the following results, the combination $N_1 = N_2 = 3$ has been used.

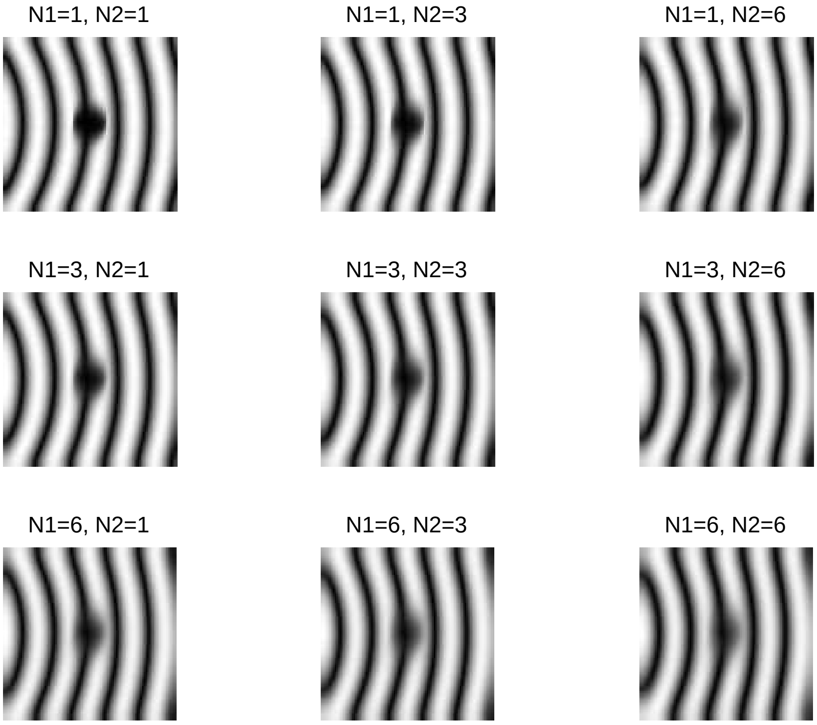


Figure 14: Calculations with $N = 10$ and different combinations of N_1 and N_2.

6.3 Boundary conditions

Here we compare the different boundary conditions used in the calculations.

6.3.1 Regular Neumann boundary conditions

With regular Neumann boundary conditions, we impose restrictions on the derivative of the solution at the boundaries. These are set to zero, and the result can be seen in Figure 15. This figure shows the original image (upper left corner) with missing information in the center, and the solutions after different choices of iterations lengths.

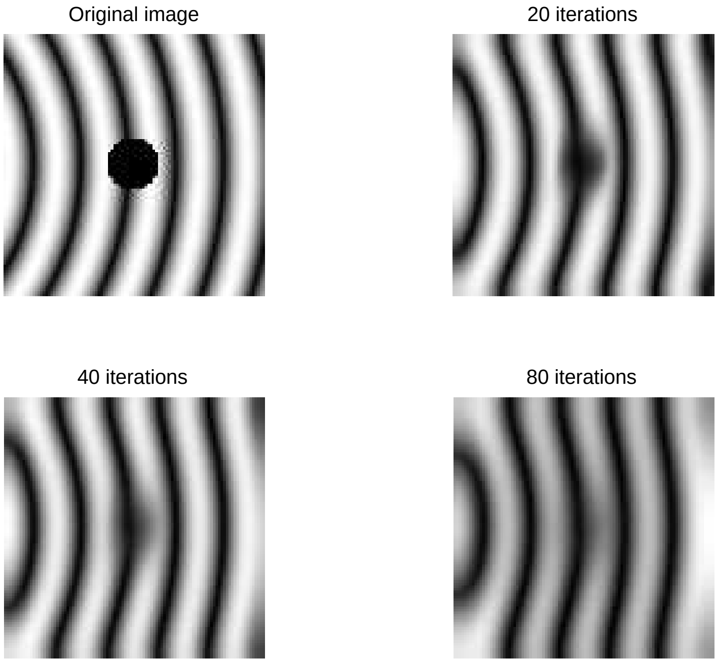


Figure 15: The solution after different number of iterations with Neumann boundary conditions.

After many iterations the hole closes, but the image is changing character near the boundaries. Since the computations impose homogeneous Neumann boundary conditions, the level lines of the image are “straightened out” according to which boundary is the closest. In the top and bottom, the curved level lines are bent to straight vertical lines. On the left, the level lines at the boundary are bent even more, and are turned into horizontal lines.

The development around the hole looks promising, and after enough iterations, it closes. If we know where the missing information “exists”, and if it is not near the boundaries, this procedure could work. The known domain could be kept, and a suitable post processing could insert the missing information after calculations. A stopping criterion would also have to be imposed, since in the limit $t \to \infty$, the surface will not converge towards the "correct" stationary surface.

6.3.2 "Roto-translational" Neumann boundary conditions

Calculations on the same image as above, but with $X_1 u = 0$ imposed on the boundaries are shown in Figure 16. The original image is shown complete, but the calculations started with a hole in the center. The behaviour around the hole is, as expeced, similar to the case above. Around the boundaries though, we now get better results, and the curved lines are maintained.

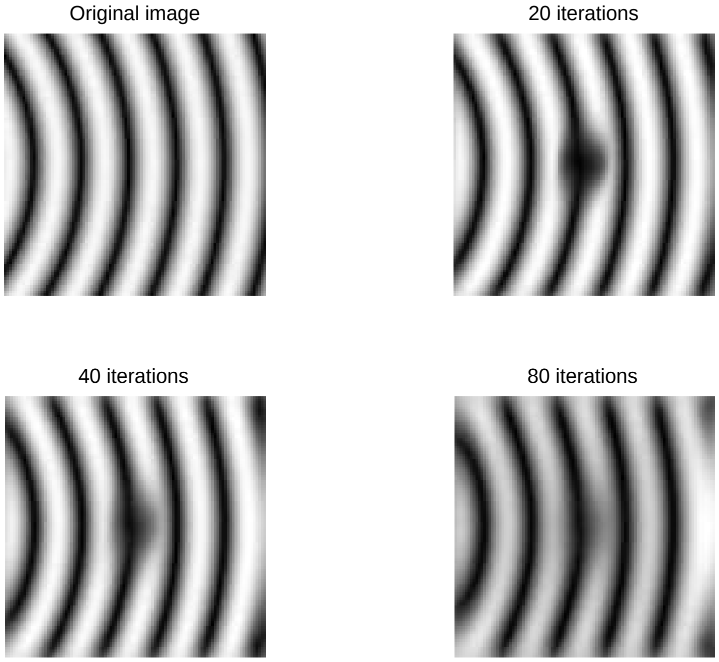


Figure 16: Solution with "Roto-translational" Neumann boundary conditions.

After many iterations, there are still some minor differences between the original and the calculated image. We observe thicker lines on the calculated one. Also, the right corners differ, and could be a consequence of the low order finite differences used here.

Since this type of boundary conditions gives the best results for this image, it has been used throughout the rest of the images below.

6.4 Completion of selected images

6.4.1 Image containing hole

The first example is an image already visited above in the report. In Figure 17 we see the original on the left, and how this image is represented as the function u in (25) on the right, both with a hole in the center. Our perception of what is missing is clear; the level lines should be continuous through the hole. We notice that we get more waves on the right, because the derivative is zero both on the "peaks" (white) and in the "valleys" (black).

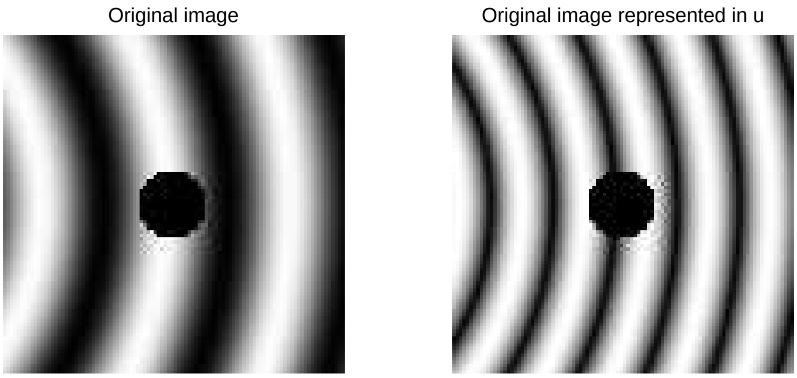


Figure 17: Original image (left), and original image represented in Roto-translational space.

In Figure 16 above, we have already seen that the algorithm completes the missing information. Figure 18 shows a surface plot of the three dimensional array after a calculation with 20 iterations. The height corresponds to the θ-value of maximum activity, and the colour represent the value (gray level) of that node.

6.4.2 Occluding object

The next example we are going to consider is when one object is occluding another object. In Figure 19 we see the original image (left), and how this is represented as the function u. The two white sections are perceived as one object extending behind the gray bar.

Figure 20 shows the result after applying the algorithm for 4 different iteration lengths. We see that the level lines for the original white object are diffusing through the occluder. For long iteration lengths, level lines from each side meet in the center, but the result is a bit blurred. By increasing the resolution in θ, this can be improved.

In Figure 21 we see a slice plot of the three dimensional array after 100 iterations. The plot shows three different planes, which corresponds to different θ-values. The colors represent the values in these xy-planes. In this way it shows where the different objects of the image are located in the θ-range. Since the derivatives in directions that differ by π have the same magnitude, but opposite signs, the three planes shown are for $\theta < \pi$.

We see that the gray bar is now located at the plane $\theta = 0$ (parallel to the x-axis). The white object "lives" above at higher θ-values, and in this way, both the objects are maintained. The gray bar is also located in the planes where θ is a multiple of π, since its directed along the x-axis.

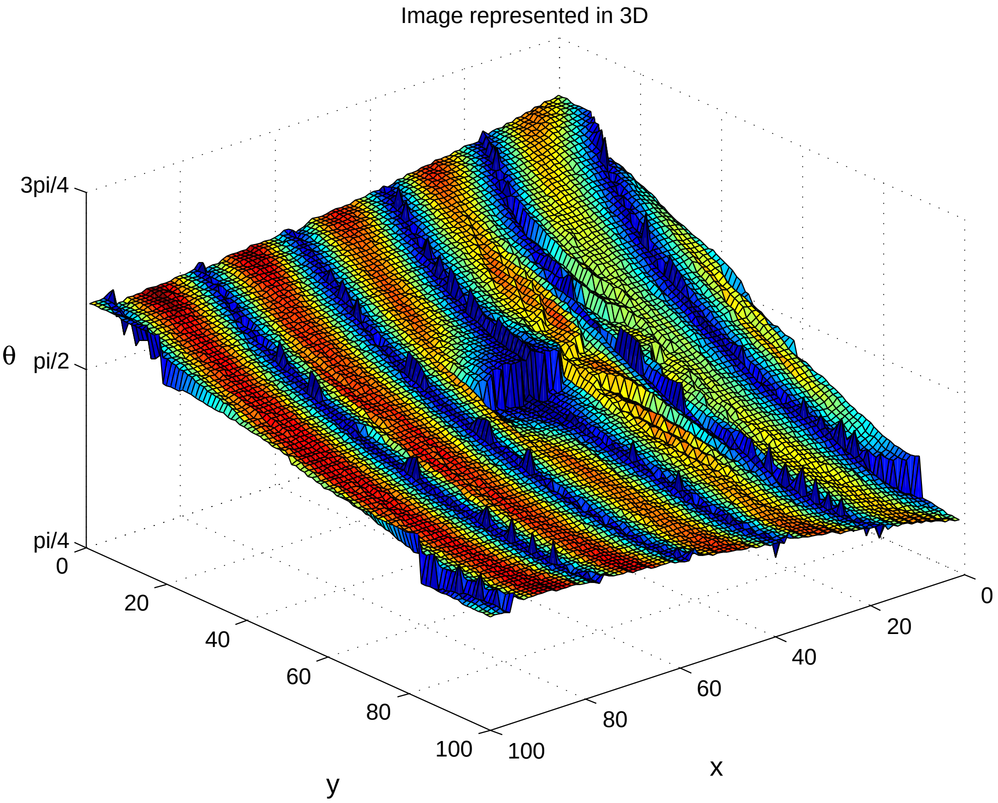


Figure 18: Slice plot of the image in three-dimensions after 20 iterations.

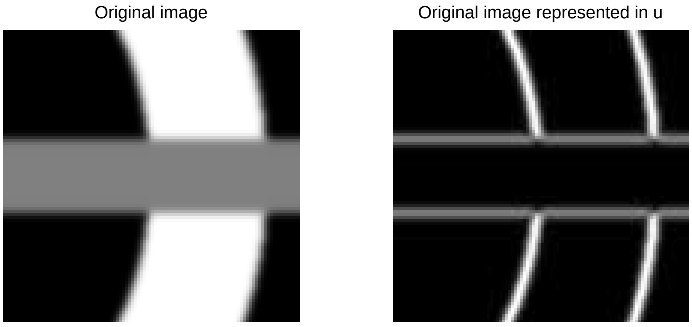
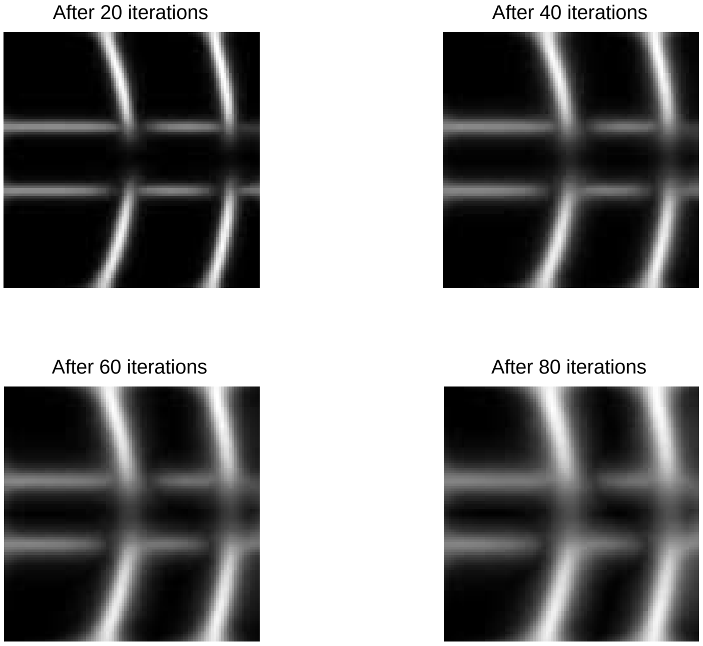


Figure 19: Original image (left), and original image in Roto-translational space.



After 20 iterations

After 40 iterations

After 60 iterations

After 80 iterations

Figure 20: Image after 20, 40, 60 and 80 iterations.

6.4.3 Kanizsa's fishes

The last example is "Kanizsa's fishes", which can be seen in Figure 22. This is an image showing two fishes, where both of them block eachother. We see that only the contours are shown in the Roto-translational representation.

Calculated results by applying the algorithm for 10 and 50 iterations can be seen in Figure 23. The right side of each fish head is completed first, since the tails here are thinnest. After 50 iterations, the whole heads are completed. This is how the original image is interpreted (at least the authors interpretation), that both fishes block each others tails and not the opposite.

In Figure 24 a slice plot of the three dimensional array is shown for "Kanizsa's fishes" after 120 iterations. Here we see that the heads of the fishes are represented at the level $\theta < \pi/2$, while the tails are located higher.

By iterating long enough, the processes should complete the tails, as well as the heads. The combined diffusion/"Non-maximal Suppression"-process can operate independently on objects separated in three dimensions.

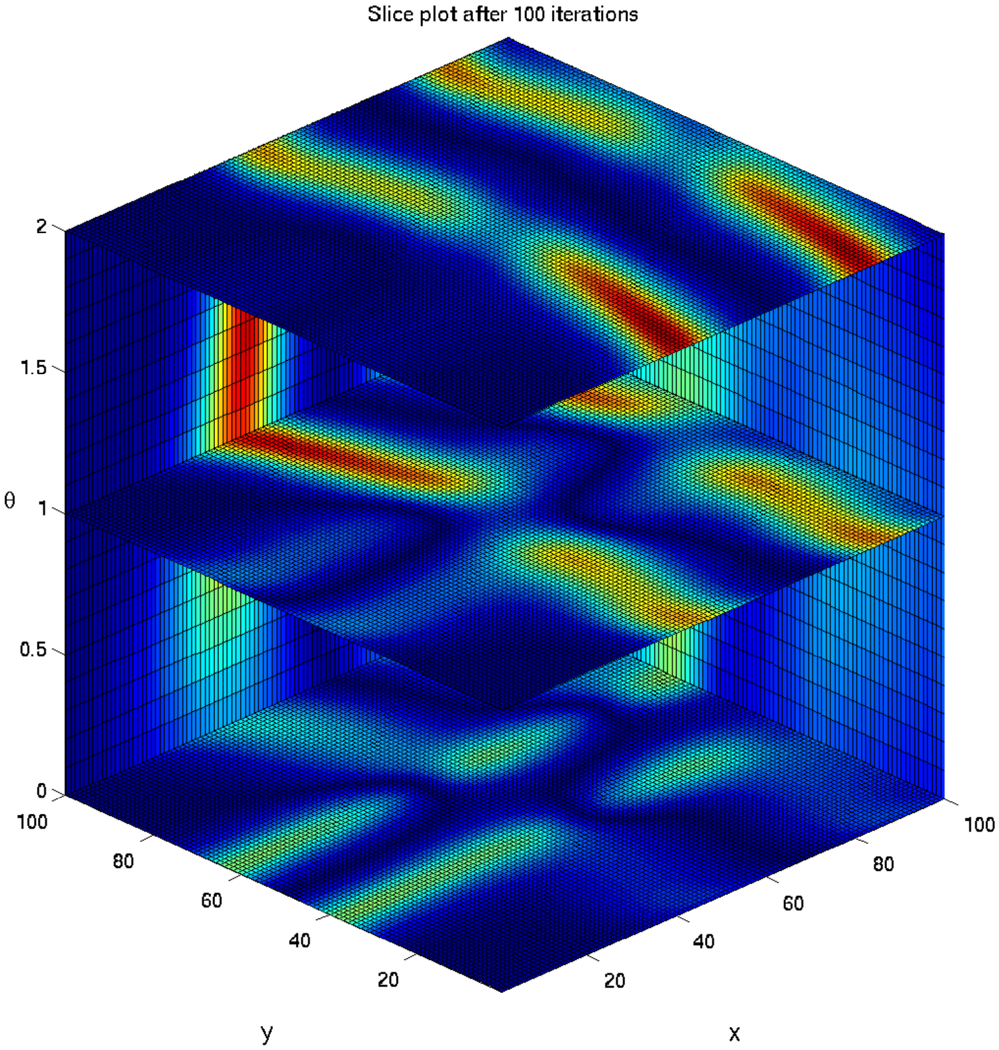


Figure 21: Slice plot of the three-dimensional array.

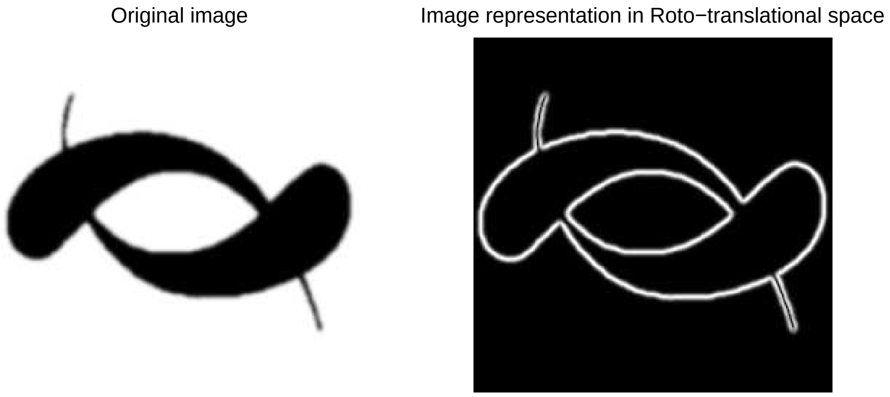


Figure 22: Original and Roto-translational representation of Kanizsa's fishes.

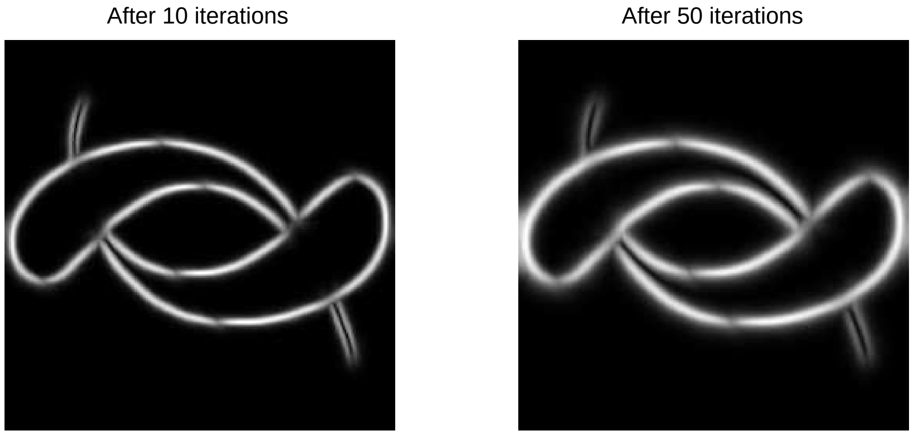


Figure 23: Kanizsa's fishes after 10 and 50 iterations.

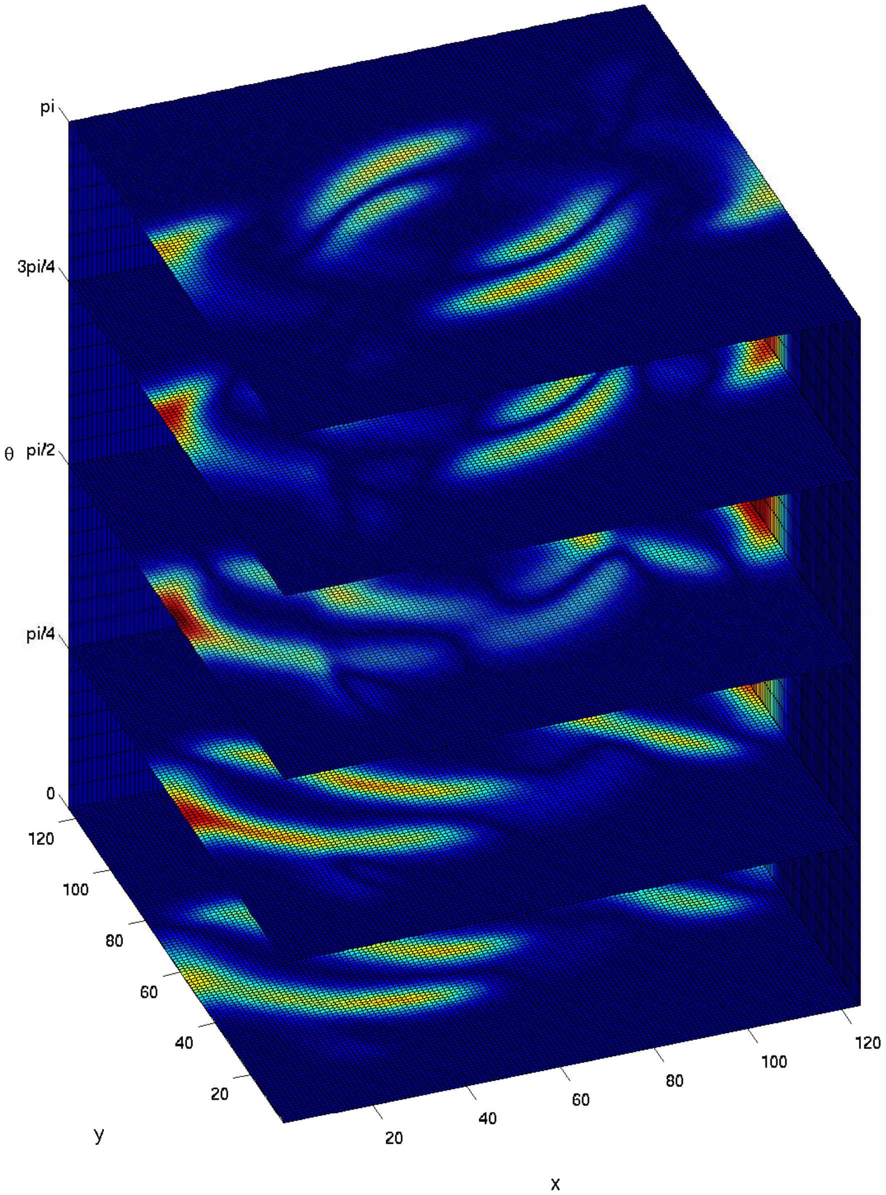


Figure 24: Slice plot of Kanizsa's fishes after 120 iterations.

7 Conlusion

The perceptual completion model in [3] has been studied and the numerical scheme was implemented. The various aspects in the model have been discussed, and some theory was established.

The diffusion in the first part of numerical scheme was analyzed more extensively, and stability conditions were found. Based on the results, the second part of the scheme, which approximates the solution of a non-linear differential equation, seems to work. This can imply that the restriction we found on the time step from the diffusion part is a conservative estimate, also valid for this part.

Chosen images were completed and discussed. These are great examples of the perceptual completion phenomena, and a verification that the model works.

The overall strategy regarding the implementation was simplicity. For instance, a simple forward Euler method was used for time integration. The step sizes have also been chosen to maintain a small to mediocre problem-size. Some difficulties regarding the boundary conditions were observed, but still, the model work very well.

Future work could include more advanced numerics, for instance a more sophisticated time integrator. Further, the Lie group structure of the problem can be exploited. Since the second part of the algorithm is a time evolution constrained on a surface, this can be solved using a Lie group method.

An entirely different approach could involve a generated grid suited for the space in which we complete the images. In [13] a grid is generated based on the structure of the Heisenberg group, and, if possible, a similar extension to the Roto-translational group could be beneficial.

References

[1] P. Perona and J. Malik, “Scale-space and edge detection using anisotropic detection”, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (7), pp. 629-639, 1990

[2] S. Masnou and J-M. Morel, “Level lines based disocclution”, IEEE International Conference on Image Processing 3, pp. 259-263, 1998

[3] G. Citti and A. Sarti, “A Cortical Based Model of Perceptual Completion in the Roto-Translation Space”, Journal of Mathematical Imaging and Vision 24 (3), pp. 307-326

[4] R. R. Seeley, T. D. Stephens and P. Tate, “Essentials of Anatomy & Physiology”, McGraw-Hill 6th edition, 2006

[5] M. Carandini, D. J. Heeger and J. A. Movshon, “Linearity and Gain Control in V1 Simple Cells”, Cerebral Cortex vol. 13. Cortical Models, 1998

[6] D. J. Field, A. Hayes, R. F. Hess, “Contour Integration by the Human Visual System: Evidence for a Local “Association Field”, *Vision* Res. Vol. 33, No. 2, pp. 173-193, 1993

[7] Peter J. Olver, “Applications of Lie Groups to Differential Equations”, second edition, Springer-Verlag

[8] S. Marcelja, “Mathematical description of the responses of simple cortical cells”, Journal of the Optical Society of America 70 (11), pp. 1297-1300

[9] B. Franchi, R. Serapioni and F. S. Cassano, “On the Structure of Finite Perimeter Sets in Step 2 of Carnot Groups”, J. geom. Anal. 13, no. 3, pp. 421-466, 2003a

[10] G. Dziuk and C. M. Elliot, “An Eulerian level set method for partial differential equations on evolving surfaces”, Computing and Visualization in Science, pp. 1-12

[11] R. Courant, K. Friedrichs and H. Lewy, “On the Partial Difference Equations of Mathematical Physics”, Mathematische Annalen 100, pp. 32-74, 1928

[12] E. M. Rønquist, “The Poisson problem in $\mathbb{R}^2$: diagonalization methods”, Lecture notes, Numerical Linear Algebra, 2007

[13] Y. Achdou and N. Tchou, “A finite difference scheme on a non commutative group”, Numer. Math. 89, pp. 401-424, 2001

[14] MATLAB$^{\text{TM}}$source codes, “http://folk.ntnu.no/permarti/PerceptualCompletion”