NORGES TEKNISK-NATURVITENSKAPELIGE
UNIVERSITET

Citti-Sarti model of perceptual completion in roto-translational space. Numerical algorithm development.

by

Elena Celledoni, Kateřina Marková, Brynjulf Owren

PREPRINT
NUMERICS NO. 12/2009

NORWEGIAN UNIVERSITY OF
SCIENCE AND TECHNOLOGY
TRONDHEIM, NORWAY

This report has URL
http://www.math.ntnu.no/preprint/numerics/2009/N12-2009.pdf
Address: Department of Mathematical Sciences, Norwegian University of Science and Technology, N-7491 Trondheim, Norway.

Citti-Sarti model of perceptual completion in roto-translational space. Numerical algorithm development.

Elena Celledoni, Kateřina Marková, Brynjulf Owren

The preceptual completion model by Citti and Sarti is presented, implemented and extesnively testes.

Citti-Sarti Model of Perceptual Completion in Roto-Translational Space Numerical Algorithm Development

Elena Celledoni, Kateřina Marková, Brynjulf Owren

Department of Mathematical Sciences
Faculty of Information Technology, Mathematics and Electrical Engineering
Norwegian University of Science and Technology

Contents

1 Basic Idea of Citti-Sarti model of perceptual completion **3**

2 Numerical Algorithm Development **5**
2.1 Discretization 6
2.2 Diffusion operator 6
2.3 Non-maximal Suppression 7
2.4 Boundary conditions 7

3 Parameters of the perceptual completion method **8**
3.1 Number of iterations within diffusion operator and non-maximal suppression operator 8
3.2 Convergence depending on the time step 15
3.2.1 Constant time steps 15
3.2.2 Time step varying during the computation process 22
3.2.3 Ratio between time step used for diffusion operator and time step for non-maximal suppression 30

4 Conclusion **33**

1 Basic Idea of Citti-Sarti model of perceptual completion

The perceptual completion phenomenon refers to seeing a figure as complete, when there is one or more parts of it missing. Practical example of missing information in the figure is e.g. occluding and occluded objects in one picture or case when part of image falls in the blind area of the visual field.

The main idea of Citti-Sarti model can be sketched as follows:

- Information is extracted from the image. This report assumes working with grayscale digital image and thus the information is already discretized by grayscale level in each pixel of the figure. Let's denote the matrix representing the image I and number of pixels in x- and y-direction N and M, respectively. Then $I \in \mathbb{R}^{M,N}$, where $\mathbb{R}^{M,N}$ is the space of all matrices with M rows and N columns.

- Image is lifted to the roto-translational space $\mathbb{R}^2 \times S^1$ where S^1 denotes the unit circle. It is the same set as $\{(x, y, \theta)|x, y \in \mathbb{R}, \theta \in \langle 0, 2\pi)\}$. The value of θ is defined by
$$\frac{\nabla I(x,y)}{|\nabla I(x,y)|} = (-\sin\theta, \cos\theta) \tag{1}$$
By denoting $\nabla I = (I_x, I_y)$, and using (1), the angle θ can be obtained as
$$\theta = \arctan\left(-\frac{I_x}{I_y}\right) \tag{2}$$

- The tangent vectors to the level lines of the image lie in the plane spanned by vectors
$$\begin{aligned} \vec{X_1} &= (\cos\theta, \sin\theta, 0) \\ \vec{X_2} &= (0, 0, 1) \end{aligned} \tag{3}$$
Asociated directional derivatives are denoted as
$$\begin{aligned} X_1 &= \cos\theta\partial_x + \sin\theta\partial_y \\ X_2 &= \partial_\theta \end{aligned} \tag{4}$$

- Define function $u : \mathbb{R}^2 \times S^1 \to \mathbb{R}$ as
$$u(x, y, \theta) = |\nabla I(x, y)| \tag{5}$$
By using relation (1), the function u can be expressed as
$$u(x, y, \theta) = -\sin\theta\frac{\partial I}{\partial x} + \cos\theta\frac{\partial I}{\partial y} \tag{6}$$

Formerly the angle θ was derived from (1), but the expression for u in (6) can be generalized for all angles θ. This is how u is defined in the whole $\mathbb{R}^2 \times S^1$.

$$u(x, y, \theta) = \left| -\sin\theta \frac{\partial I}{\partial x} + \cos\theta \frac{\partial I}{\partial y} \right| \tag{7}$$

The absolute value is necessary because u should correspond to $|\nabla I|$, and (6) is positive only for $\theta \in (-\frac{\pi}{2} + 2k\pi, \frac{\pi}{2} + 2k\pi) \quad \forall k \in \mathbb{Z}$. Since the function u is evaluated using the absolute value, there is no need of using the whole interval $\langle 0, 2\pi \rangle$, instead $\langle 0, \pi \rangle$ can be used with the same effect.

- The image is now represented in 3-dimensional space, the information needs to be propagated in orientation-specific way. This is accomplished by sub-Laplacian operator. Let's define R-gradient as $\nabla_R = (X_1, X_2)$, then the sub-Laplacian can be defined as $\triangle_R = \nabla_R . \nabla_R^T$. Denote $X_{ij} u := X_i(X_j u)$, then the sub-Laplacian can be written as $\triangle_R = X_{11} + X_{22}$.
  The diffusion equation then becomes

$$\frac{\partial u}{\partial t} = \triangle_R u = X_{11} u + X_{22} u \tag{8}$$

- After the diffusion, the surface has expanded in the θ-direction. A concentration process is needed to recover the surface, so that function u attains maximum on this new surface. New surface can be therefore defined as

$$\left\{ (x, y, \theta) \,\middle|\, \frac{\partial u}{\partial \theta}(x, y, \theta) = 0, \frac{\partial^2 u}{\partial \theta^2}(x, y, \theta) < 0 \right\} \tag{9}$$

  The process of creating new surface is called the non-maximal suppression. Citti and Sarti, however, suggest not to directly create this new surface, but to construct the tangent planes and perform diffusion constrained on these surfaces instead. Detailed derivation and reasoning is omitted in this report, it can be found in [1] or [2]. The final form of the non-maximal suppresion operator is given by the Laplace-Beltrami flow operator.

$$\begin{aligned} v &= X_2 u \\ \frac{\partial u}{\partial t} &= \frac{(X_2 v)^2 X_{11} u + (X_1 v)^2 X_{22} u}{(X_1 v)^2 + (X_2 v)^2} - \frac{(X_{12} u + X_{21} u) X_1 v X_2 v}{(X_1 v)^2 + (X_2 v)^2} \end{aligned} \tag{10}$$

  This expression is taken from [2]. It can be expressed simplier by neglecting the substitution for v:

$$\frac{\partial u}{\partial t} = X_{22} u \frac{X_{11} u X_{22} u - X_{12} u X_{21} u}{(X_{12} u)^2 + (X_{22} u)^2} \tag{11}$$

- The idea of the model is to alternate between performing diffusion and non-maximal suppression.

- Detailed derivation and analysis of the method is shown in [1], [2] and [3].

2 Numerical Algorithm Development

The code written for purpose of this report in Matlab is using the same concept that was used by Per Martin Viddal in [2]. The part that differs the most is the actual implemetnation of the diffusion operator performing (8) and (11). The program offers to choose between applying the 'typical' homogeneous Neumann boundary conditions on boundary i.e.

$$\left.\frac{\partial u(x,y,\theta)}{\partial \vec{n}}\right|_{\partial\Omega} = 0 \tag{12}$$

where $\partial\Omega$ denotes the domain boudary and $\vec{n}$ is the unit outside normal vector to the boundary. When processing image with contour lines not orthogonal to the boundary of image, use of this boundary conditions distorts the image near the boundary. That's why, as default, the roto-translational version of neumann boundary conditions (13) is used instead.

$$\left.\left(\frac{\partial u(x,y,\theta)}{\partial x}\cos\theta + \frac{\partial u(x,y,\theta)}{\partial y}\sin\theta\right)\right|_{\partial\Omega} = 0 \tag{13}$$

To solve the differential equations, finite difference method is used - the forward difference for partial derivative in time and central differences for spatial derivatives. (14) shows the form of finite differences used. The notation is simplified so that $u^n(x,y,\theta) = u(t_n,x,y,\theta)$, Δx, Δy, $\Delta\theta$ are the sizes in step in each axial direction and Δt is the time step, $x_i = x_0 + i\Delta x$, $y_j = y_0 + j\Delta y$, $\theta_k = k\Delta\theta$, $t^n = n\Delta t$, where (x_0, y_0) is the left bottom corner of image. (In the algorithm (x_0, y_0) is set to be the origin $(0,0)$.) $u_{i,j,k}$ denotes $u(x_i, y_j, \theta_k)$.

$$\begin{aligned}
\frac{\partial u}{\partial t}(x_i,y_j,\theta_k) &\cong \frac{u^{n+1}_{i,j,k} - u^n_{i,j,k}}{\Delta t}\\
\frac{\partial u}{\partial x}(x_i,y_j,\theta_k) &\cong \frac{u_{i+1,j,k} - u_{i-1,j,k}}{2\Delta x}\\
\frac{\partial u}{\partial y}(x_i,y_j,\theta_k) &\cong \frac{u_{i,j+1,k} - u_{i,j-1,k}}{2\Delta y}\\
\frac{\partial u}{\partial \theta}(x_i,y_j,\theta_k) &\cong \frac{u_{i,j,k+1} - u_{i,j,k-1}}{2\Delta \theta}\\
\frac{\partial^2 u}{\partial x^2}(x_i,y_j,\theta_k) &\cong \frac{u_{i+1,j,k} - 2u_{i,j,k} + u_{i-1,j,k}}{\Delta x^2}\\
\frac{\partial^2 u}{\partial x \partial y}(x_i,y_j,\theta_k) &\cong \frac{u_{i+1,j+1,k} - u_{i-1,j+1,k} - u_{i+1,j-1,k} + u_{i-1,j-1,k}}{4\Delta x \Delta y}\\
&\dots
\end{aligned} \tag{14}$$

2.1 Discretization

The model is constructed to perform the perceptual completion algorithm on grayscale digital images. The digital image is represented in a discrete way already. There is no need to change its representation. The number of pixels in the x-direction can be denoted as N and in y-direction M. Since the distance between pixels corresponds to the pixel size, the spatial steps Δx and Δy can be without loss of generality assumed to be the same. Let's denote K the number of partial angle intervals for θ in interval $\langle 0, \pi \rangle$. Then $\Delta\theta = \frac{\pi}{K}$.

The article [1] suggests using time step and spatial steps so that $\Delta x = \Delta y$, $\Delta\theta = \Delta x^2$, $\Delta t = \Delta x^2$. The proposed program, however, chooses $\Delta\theta$ independedently on the spatial step. Most experiments were actually done with $\Delta\theta > \Delta x$. Reason why the suggestion by Citti and Sarti was neglected is that the number of subintervals for θ in $\langle 0, \pi \rangle$ has big influence on the computational cost of the program. The program stores function u in a 3-dimensional array (a tensor of 2nd order), and to represent each "θ level" matrix with MN elements is required. The default time step used in this method is $\Delta t = \frac{1}{4}\min\{\Delta x^2, \Delta\theta^2\}$. It is the maximal time step within the stability region of the diffusion operator as is shown in [2].

2.2 Diffusion operator

The diffusion equation shown in (8) can be rewritten in terms of partial derivatives in x-, y-, θ-directions by using relation for X_1 and X_2 in (4).

$$\frac{\partial u}{\partial t} = (\cos\theta)^2 \frac{\partial^2 u}{\partial x^2} + 2\cos\theta\sin\theta \frac{\partial^2 u}{\partial x \partial y} + (\sin\theta)^2 \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial \theta^2} \tag{15}$$

Denote $\tilde{u} \in \mathbb{R}^{M,N,K}$ the discrete representation of the function u so that

$$\tilde{u}_{i,j,k} = u(x_i, y_j, \theta_k) \tag{16}$$

and D_{ij} the discrete form of operator X_{ij}

$$X_{ij} u \cong D_{ij} \tilde{u} \tag{17}$$

Then define $\tilde{u}_{\theta_k}$ to be the matrix representing values of $\tilde{u}$ for fixed $\theta = \theta_k$. Then the discrete form of $X_{11}u$ can be computed using matrix multiplications as:

$$D_{11}\tilde{u}_{\theta_k} = \frac{1}{\Delta x^2}\left[(\sin\theta_k)^2 \mathbf{A}\tilde{u}_{\theta_k} + (\cos\theta_k)^2 \tilde{u}_{\theta_k}\mathbf{B} + \frac{1}{2}\sin\theta_k\cos\theta_k \mathbf{D}\tilde{u}_{\theta_k}\mathbf{E}\right] \tag{18}$$

where $\mathbf{A}, \mathbf{D} \in \mathbb{R}^{M,M}$ and $\mathbf{B}, \mathbf{E} \in \mathbb{R}^{N,N}$.

Similarly define $\tilde{u}_{x_i}$ to be the matrix representing values of $\tilde{u}$ for fixed $x = x_i$. Then $D_{22}\tilde{u}$ can be expressed as

$$D_{22}\tilde{u}_{x_i} = \frac{1}{\Delta\theta^2}\tilde{u}_{x_i}\mathbf{C} \tag{19}$$

where $\mathbf{C} \in \mathbf{R}^{K,K}$.

$$\mathbf{A} = \begin{pmatrix} -2 & 1 & \dots & 0 \\ 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 0 & \dots & 1 & -2 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} -2 & 1 & \dots & 0 \\ 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 0 & \dots & 1 & -2 \end{pmatrix}$$

$$\mathbf{C} = \begin{pmatrix} -2 & 1 & \dots & 0 \\ 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 0 & \dots & 1 & -2 \end{pmatrix} \tag{20}$$

$$\mathbf{D} = \begin{pmatrix} 0 & 1 & \dots & 0 \\ -1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 0 & \dots & -1 & 0 \end{pmatrix} \quad \mathbf{E} = \begin{pmatrix} 0 & -1 & \dots & 0 \\ 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -1 \\ 0 & \dots & 1 & 0 \end{pmatrix}$$

2.3 Non-maximal Suppression

The differential equation (11) describing the Laplace-Beltrami flow is nonlinear, and combines operators $X_{11}u$, $X_{12}u$, $X_{22}u$ and $X_{33}u$.

$$\begin{aligned} X_{11}u &= (\cos\theta)^2 \frac{\partial^2 u}{\partial x^2} + (\sin\theta)^2 \frac{\partial^2 u}{\partial y^2} + 2\sin\theta\cos\theta \frac{\partial^2 u}{\partial x \partial y} \\ X_{12}u &= \cos\theta \frac{\partial^2 u}{\partial x \partial \theta} + \sin\theta \frac{\partial^2 u}{\partial y \partial \theta} \\ X_{21}u &= -\sin\theta \frac{\partial u}{\partial x} + \cos\theta \frac{\partial u}{\partial y} + X_{12}u \\ X_{22}u &= \frac{\partial^2 u}{\partial \theta^2} \end{aligned} \tag{21}$$

2.4 Boundary conditions

The main difference between program described in [2] and program described in this report is the application of boundary conditions. Per Martin Viddal's code creates special form of difference schemes for spatial derivatives on the boundary, that satisfy the boudnary conditions. This approach is convenient if more differential equations were tested. The disadvantage is that allocating the values of differences is highly computationally expensive. For each of the difference substitute for spatial derivative, there has to be allocated new array of the same size as the one used to store values of $\tilde{u}$, which is a 3-dimensional array in $\mathbb{R}^{M,N,K}$.

The program described in this report applies boundary conditions when defining the difference scheme for the differential equation. Imaginary boundary cells are used to implement the boundary conditions. The roto-translational boundary conditions are described in (13), the boundary conditions in θ are set to be periodic. That is

why acutally instead of using an array of $M \times N \times K$ elements to describe function $\tilde{u}$, the array of $(M+2) \times (N+2) \times (K+2)$ elements is used.

3 Parameters of the perceptual completion method

3.1 Number of iterations within diffusion operator and non-maximal suppression operator

The model derived in [1] suggests varying between applying the diffusion operator and non-maximal suppression. There is, however, no suggestion on how to vary between these processes.

That is why several ratios between number of iterations of each process were tested to find the optimal one. Similar comparison was done in [2] with the conclusion that the optimal choice is to use 3 iterations of diffusion process and 3 iterations of non-maximal suppression process in each "global" iteration of the program.

Two different pictures were used for the comparison of results gained when using different ratios of number of individual iterations. First image is in Fig. 1, its projection to roto-translational space is in Fig. 2, ideally completed image is sketched in Fig. 3. Corresponding presentations of the second image are shown in Fig. 4, Fig. 5 and Fig.6. The second example is artificially made. The picture is lifted into the roto-translational space and then part of data is deleted in the middle of the image. Reason why this image was used is that it illustrates very well the process of filling in the missing information.

The comparison is made with spatial step $\Delta x = \Delta y = 0.01$, angular step $\Delta\theta \cong 0.1$ and time step set to $\Delta t = 0.25\Delta x^2$. The results are shown after 50 global iterations of the algorithm.

The pictures illustrating the results are Fig. 7 Fig. 8 Fig. 9 Fig. 10. The number of iterations within diffusion process is denoted as N_1 and number of iterations of non-maximal suppression is N_2.

Figure 1: 1^{st} image

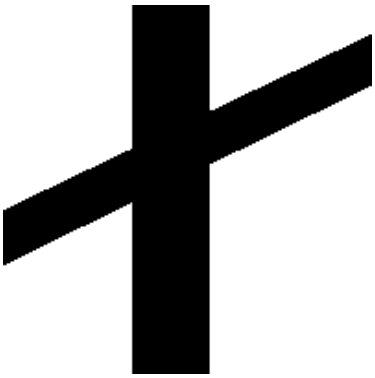


Figure 2: Roto-translational projection of the 1^{st} image

Figure 3: Ideal completion of the 1^{st} image in roto-translational space

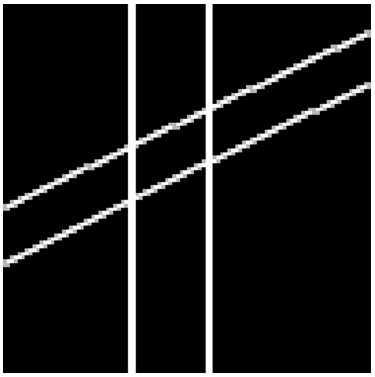


Figure 4: 2^{nd} image

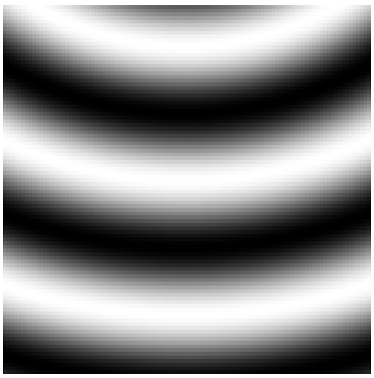


Figure 5: Roto-translational projection of the 2^{nd} image with the hole

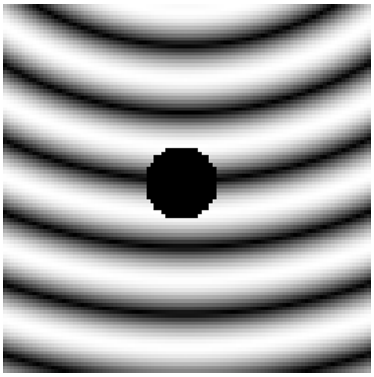


Figure 6: Ideal completion of the 2^{nd} image in roto-translational space

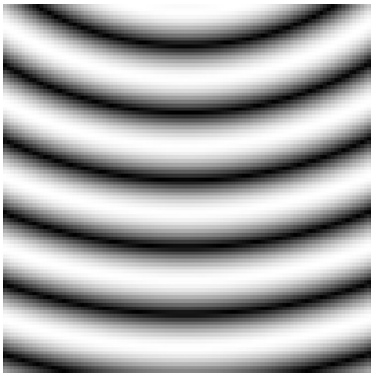


Figure 7: Comparison of results for 1^{st} image for different ratios of diffusion steps N_1 and non-maximal supression steps N_2

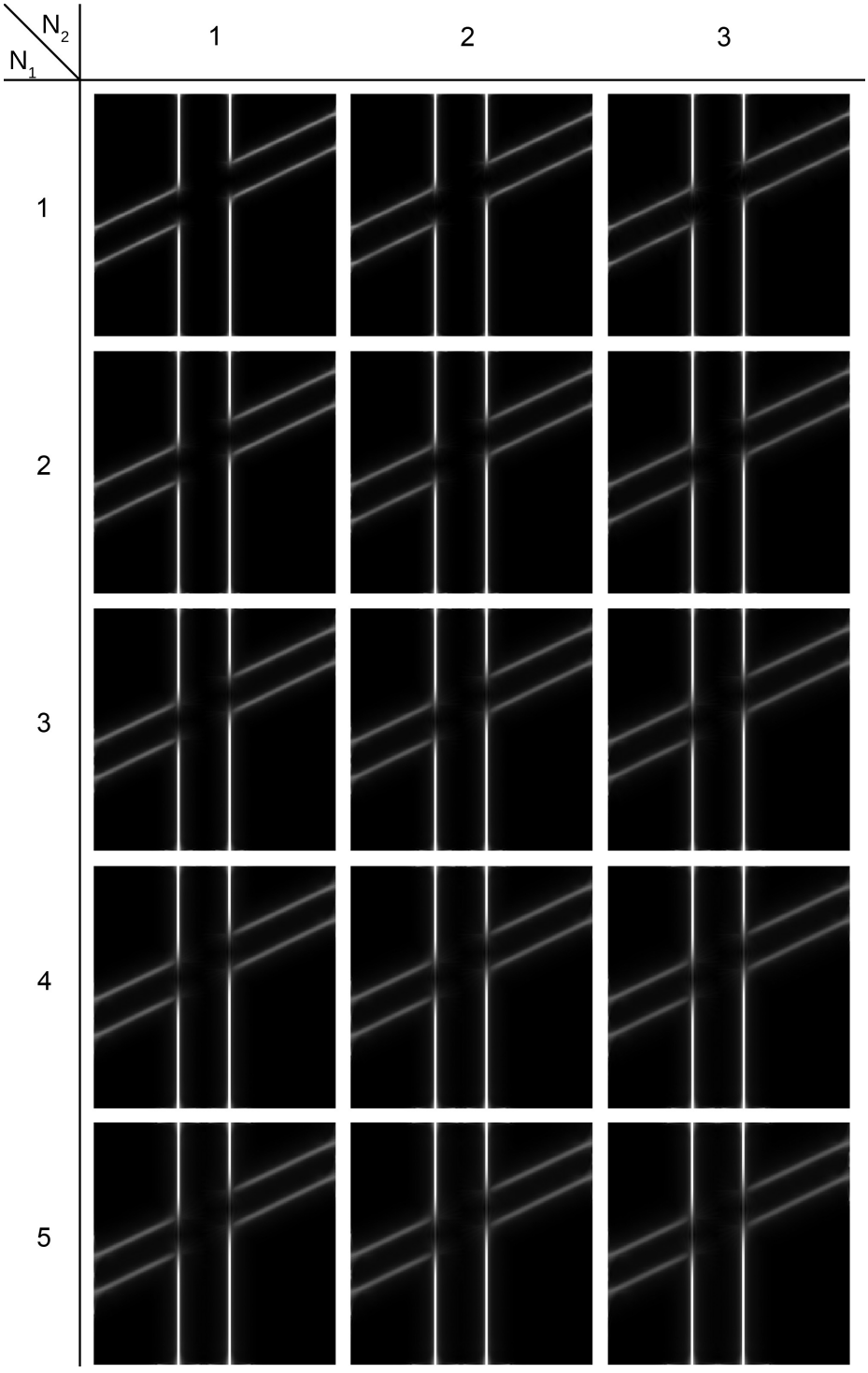


Figure 8: Comparison of results for 1^{st} image for different ratios of diffusion steps N_1 and non-maximal supression steps N_2

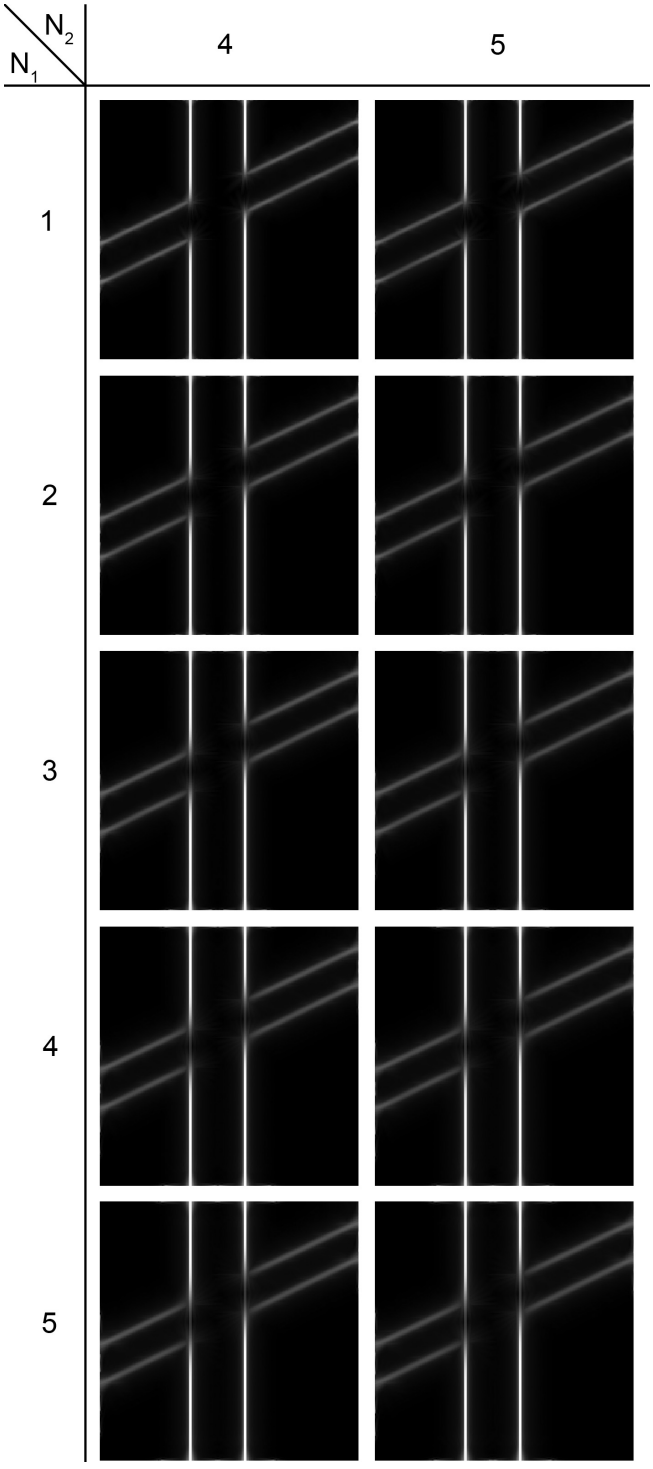


Figure 9: Comparison of results for 2^{nd} image for different ratios of diffusion steps N_1 and non-maximal supression steps N_2

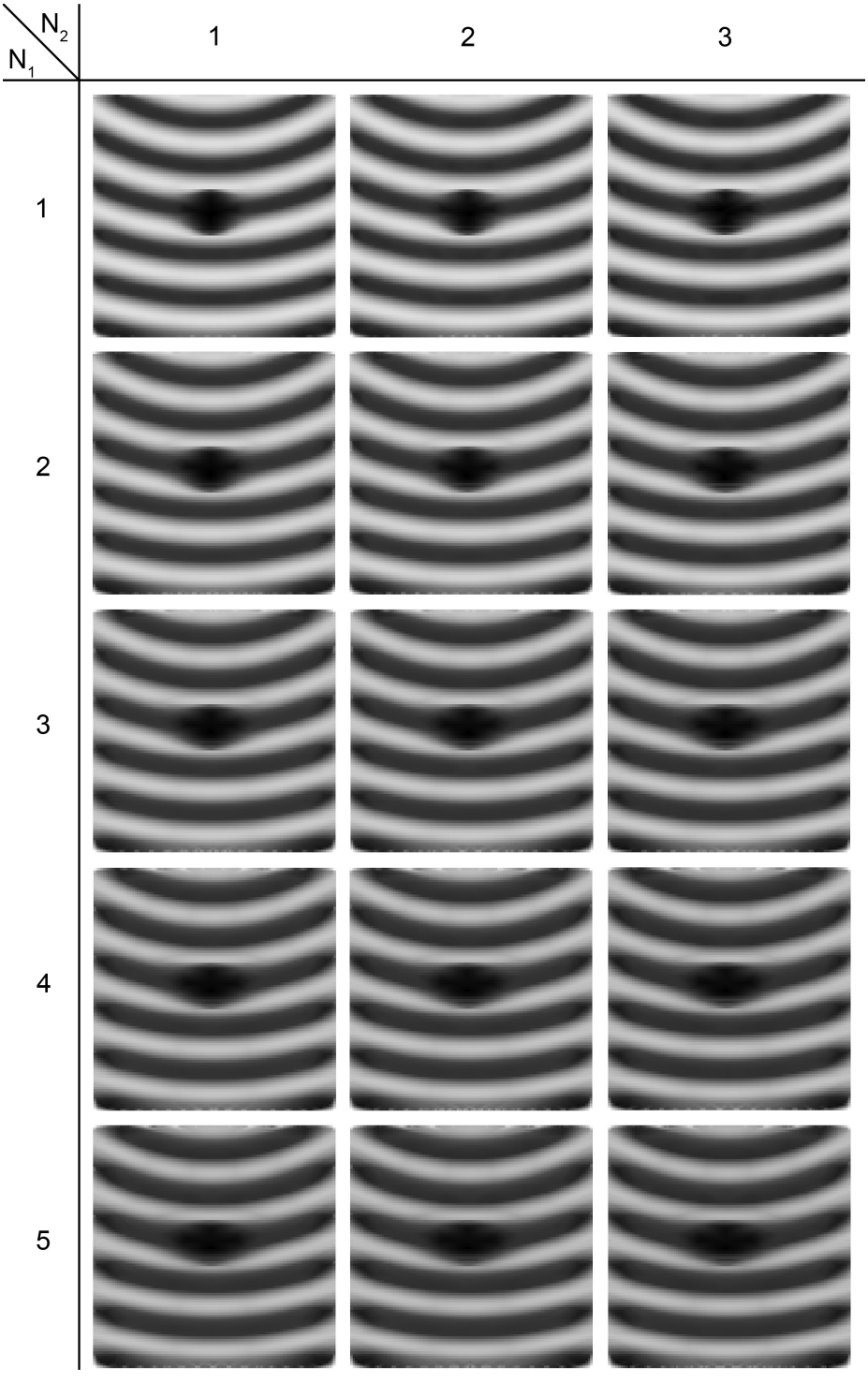
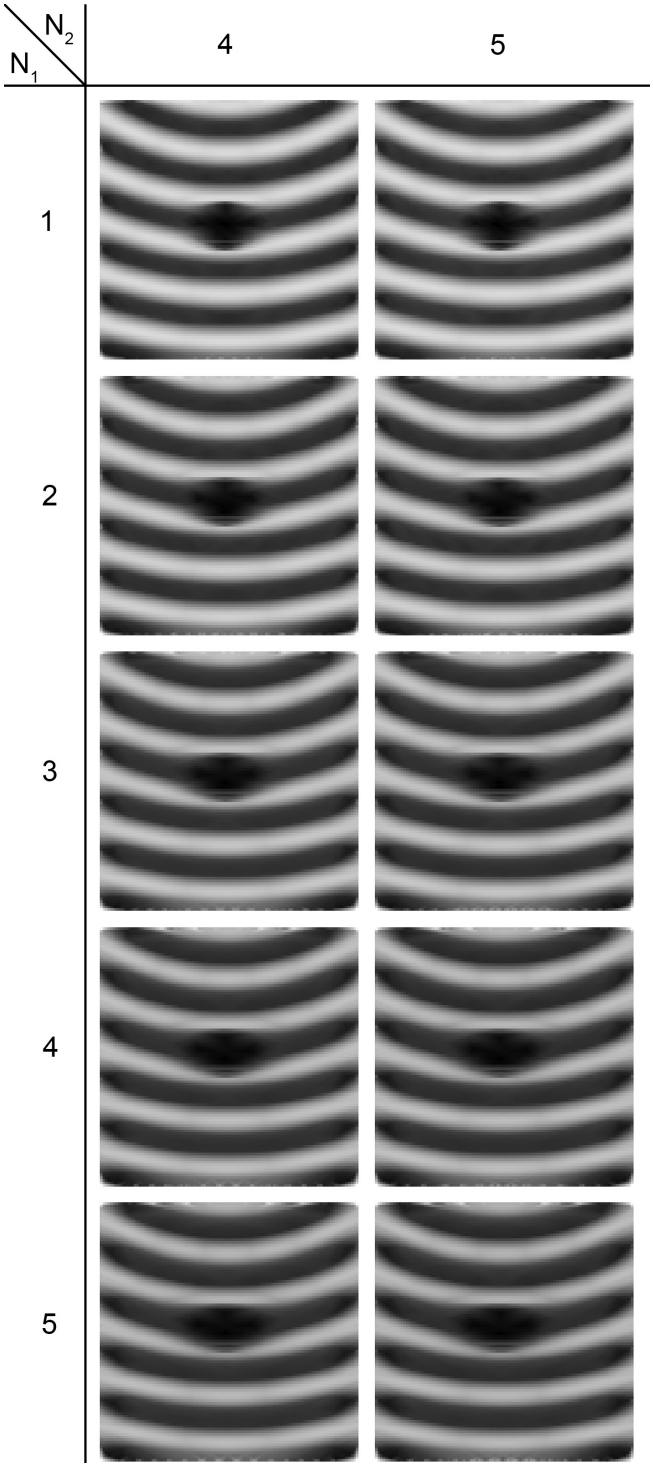


Figure 10: Comparison of results for 2^{nd} image for different ratios of diffusion steps N_1 and non-maximal supression steps N_2



The effect of increasing number of diffusion iterations is well illustrated by the images. The higher number of diffusion steps is taken in each global iteration of the algorithm, the more effectively is filled in the missing information. The cost of higher efficiency is the loss of accuracy. The more diffusion steps, the more are the contours of the image blurred. In conclusion, the choice suggested in [2] of 3 diffusion steps per iterations seems to be a good compromise between efficiency and accuracy.

Deciding how many steps should be taken within non-maximal suppression is much more difficult. The effect of varying step number is not so obvious. With close study of the images, the effect of increasing number of non-maximal suppression steps is the increase in the speed in which are the contours filled in, but the drawback is that contour lines are getting less recognizable.

It seems that the choice of 3 steps in both the parts of the algorithm is a reasonable choice for keeping the performance and accuracy at high level at the same time. In following discussion all the experiments will be therefore done using this setting.

3.2 Convergence depending on the time step

Another area of interest is the influence of time step on the efficiency of the perceptual completion model.

There are several approaches to running the perceptial completion model. Either it can be run with constant time step for the whole program-execution period, or the time step can be changed during the evaluation process. Also the ratio of time step for diffusive and non-maximal suppresion part of the algorithm can be changed.

All results demonstarted earlier in this report have been obtained using the stability estimate made in [2], which sets maximal time step to be

$$\Delta t_{max} = 0.25 \min\{\Delta x^2, \Delta\theta^2\} \tag{22}$$

The used time step is set to be Δt_{max}, the time step for non-maximal suppresion is the same as for the diffusion. Following text compares using smaller time step or varying time step during the computation.

3.2.1 Constant time steps

Let's first compare using different constant time steps during the whole evaluation period. The images Fig.11, Fig.12 and Fig.13 represent the perceptual completion applied to the first picture as shown in Fig.1 with time steps Δt_{max}, $0.1\Delta t_{max}$ and $0.05\Delta t_{max}$, respectively. Figures Fig.14, Fig.15 and Fig.16 are results for the second image Fig.4 with the same time steps. The results are shown after overall time period $T_1 = 50\Delta t_{max}$.

To better illustrate the difference in results, which is very difficult to be seen for time period T_1, tests for another time period $T_2 = 100\Delta t_{max}$ were run. Due to high computational cost of this computations, only results for the first image for time steps Δt_{max} and $0.1\Delta t_{max}$ are presented. First image was chosen because the effects of perceptual completion are better remarkable. The resulting pictures are Fig.17 and Fig.18.

Figure 11: Completion of the 1^{st} image in time T_1 with time step Δt_{max}

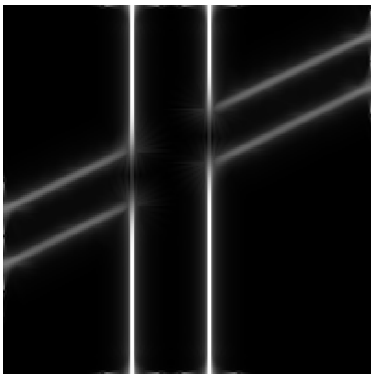


Figure 12: Completion of the 1^{st} image in time T_1 with time step $0.1\Delta t_{max}$

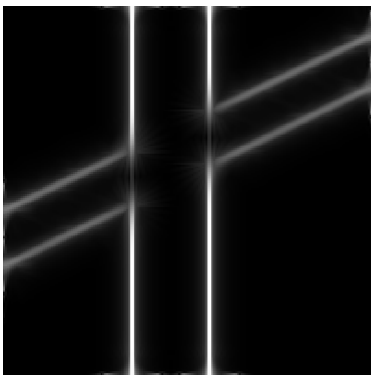


Figure 13: Completion of the 1^{st} image in time T_1 with time step $0.05\Delta t_{max}$

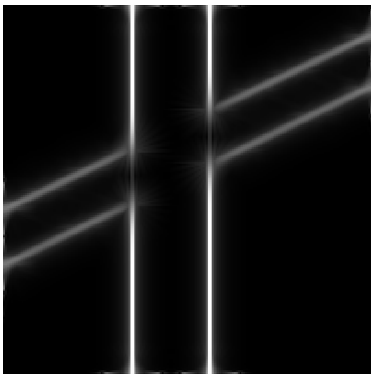


Figure 14: Completion of the 2^{nd} image in time T_1 with time step Δt_{max}

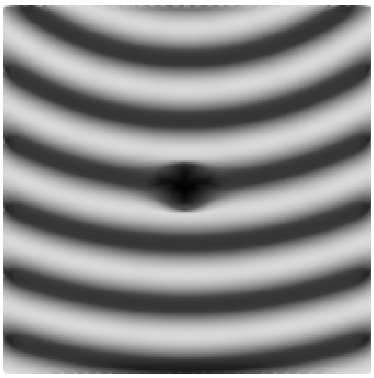


Figure 15: Completion of the 2^{nd} image in time T_1 with time step $0.1\Delta t_{max}$

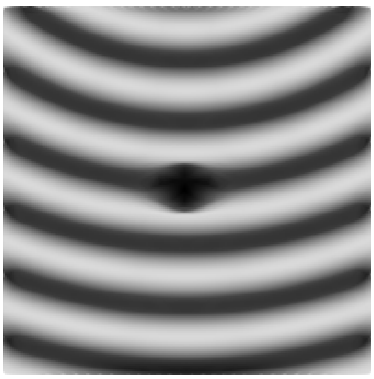


Figure 16: Completion of the 2^{nd} image in time T_1 with time step $0.05\Delta t_{max}$

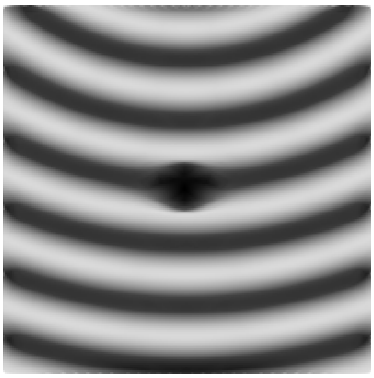


Figure 17: Completion of the 1^{st} image in time T_2 with time step Δt_{max}

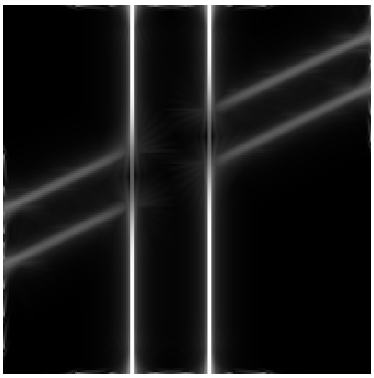
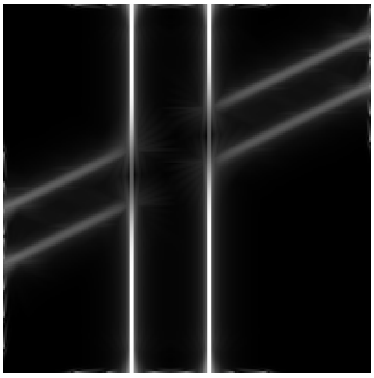


Figure 18: Completion of the 1^{st} image in time T_2 with time step $0.1\Delta t_{max}$



Only minor differences can be observed in the resulting images. The refinement in time step leads to improvement of the method (the contours are projected little faster), but the computational cost increase does not seem to be compensated. When the time step is scaled by factor $s < 1$, then the number of iterations needed to acquire desired final time T is multiplied by $\frac{1}{s}$. The improvement in the method is almost neglectable compared to increased time needed to perform the computation.

The perceptal completion model described in this report is based on separate application of diffusion operator and Laplace-Beltrami flow operator. For small time step the overall error of numerical solution should be dominated by the splitting error. That is why for following experiments the time interval for diffusion operator and the time interval for non-maximal suppression were kept constant, both set to $3\Delta t_{max}$, and different number of steps was used to perform diffusion and non-maximal suppression in this time period. The resulting sets of images are shown in Fig.19 and Fig.20. The total of 50 global iterations was computed. All the experiments used the same number of steps for the diffusion operator and for the

Laplace-Beltrami flow operator $N_1 = N_2 = n$.

Figure 19: Comparison of completion of the 1^{st} image with changing $N_1 = N_2 = n$

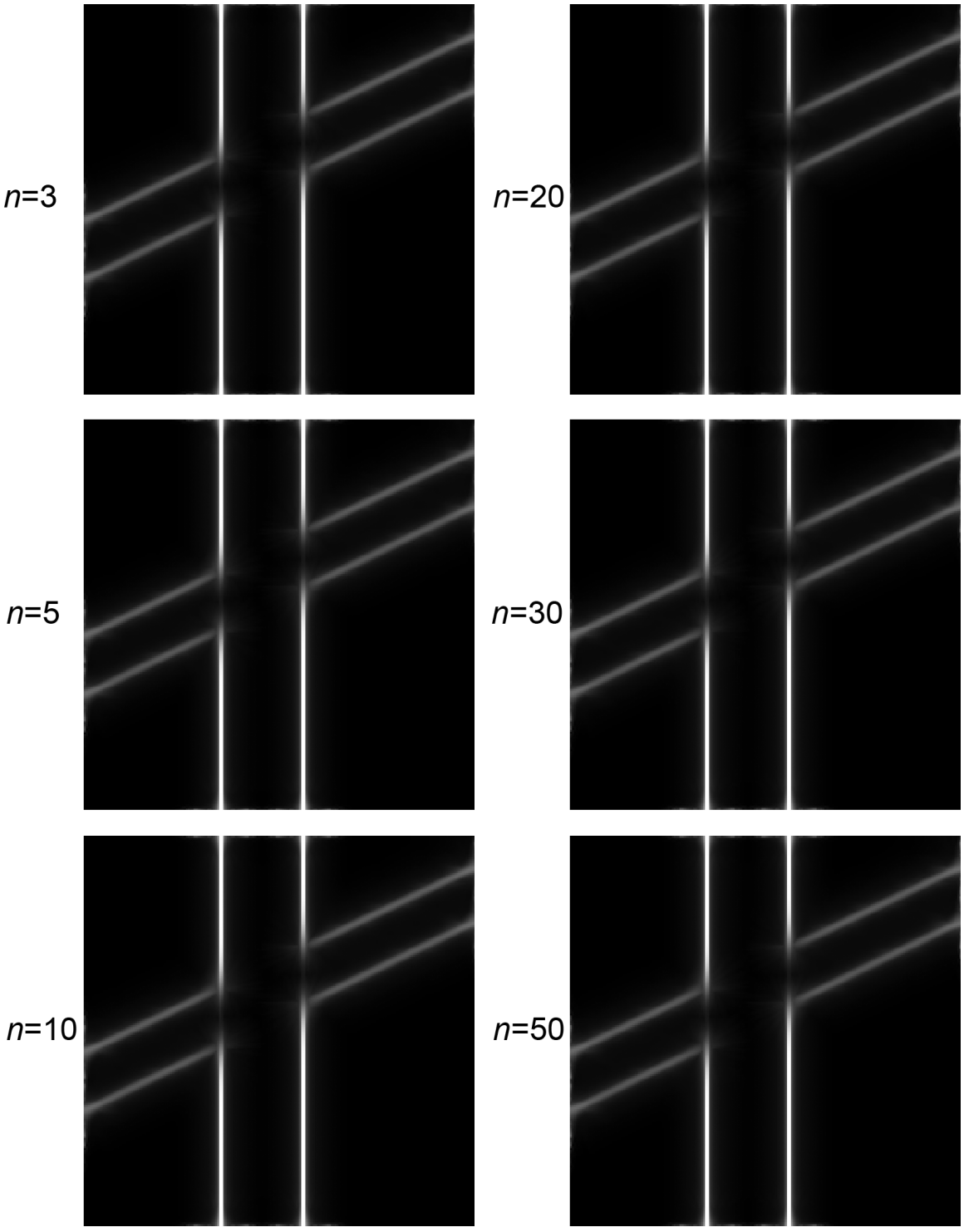
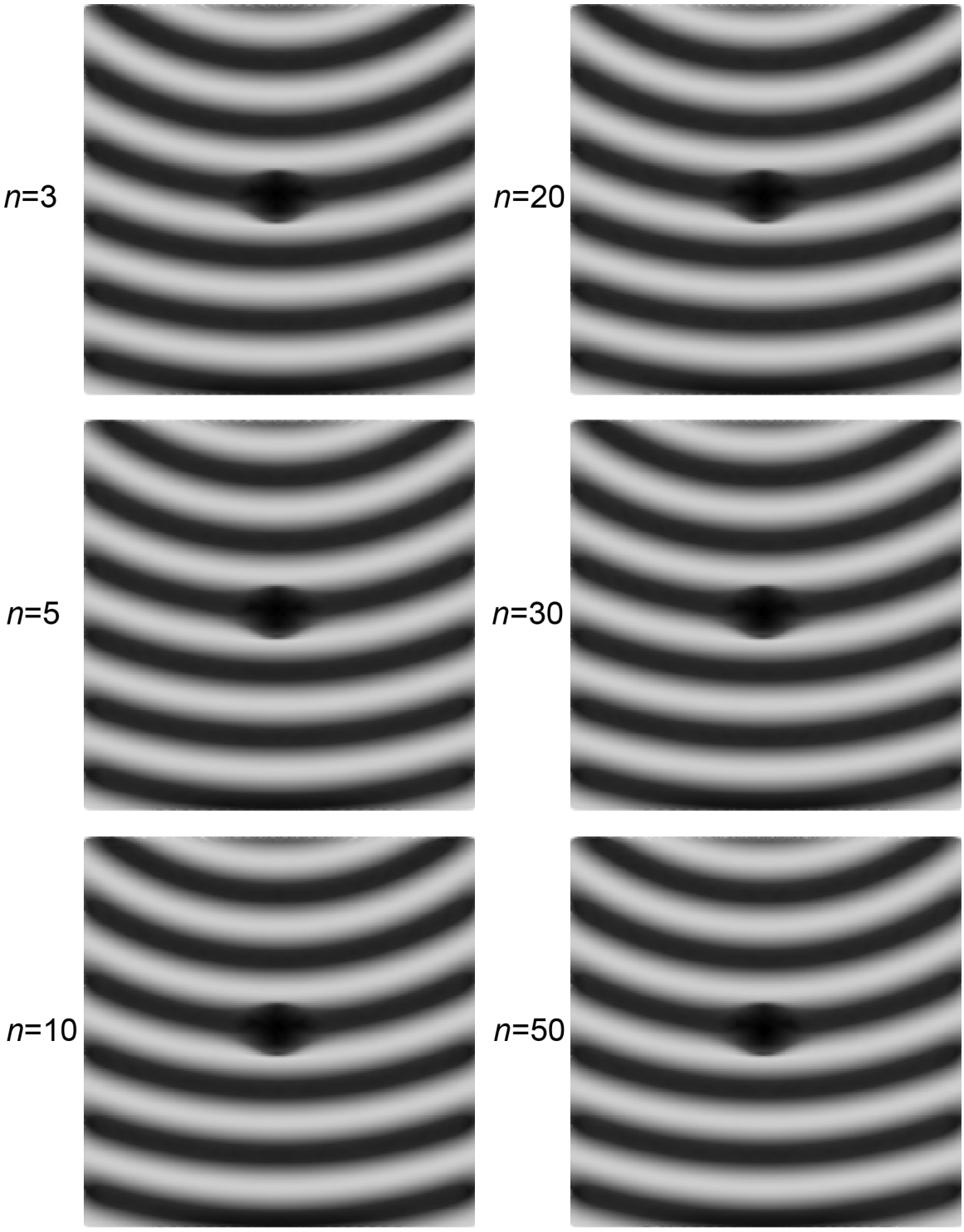


Figure 20: Comparison of completion of the 2^{nd} image with changing $N_1 = N_2 = n$



3.2.2 Time step varying during the computation process

One aproach to refining the time step during the computation is to lower the time step after each global iteration is done (i.e. diffusion process and non-maximal suppression are computed with given number of steps). Following figures illustrate the effect of scaling time step during computation. Time was scaled so that $\Delta t^{n+1} = s\Delta t^n$, where $s = 0.999$ and 0.99 respectively. Starting time step is set to be Δt_{max}. For better comparison, also images with constant time steps are presented. The computation time period is set to be $T_1 = 50\Delta t_{max}$. Results for the first image with time scaling factors 1, 0.999, 0.99, are shown in figures Fig.21, Fig.22, Fig.23, respectively. The completion of second image is illustrated by figures Fig.24, Fig.25 and Fig.26, respectively.

Scaling time step while running the program increases the efficiency of the completion, but at the same time the time period necessar to evaluate the result is increasing. The choice whether to scale or not depends on the final user and used equipment. Probably better than scaling time step while running the model is to choose smaller but constant time step.

Figure 21: Completion of the 1^{st} image in time T_1 with scaling factor 1

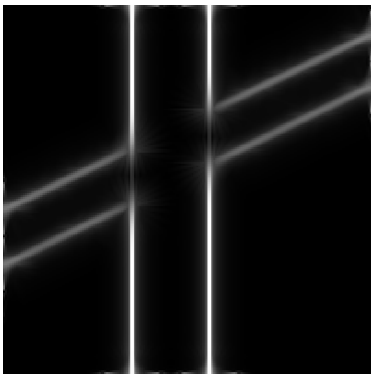


Figure 22: Completion of the 1^{st} image in time T_1 with scaling factor 0.999

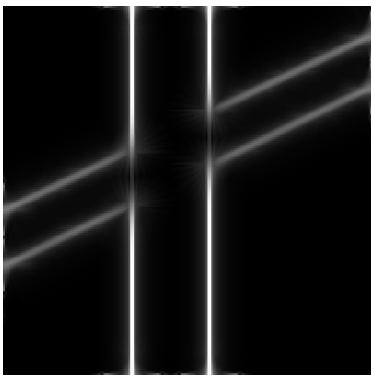


Figure 23: Completion of the 1^{st} image in time T_1 with scaling factor 0.99

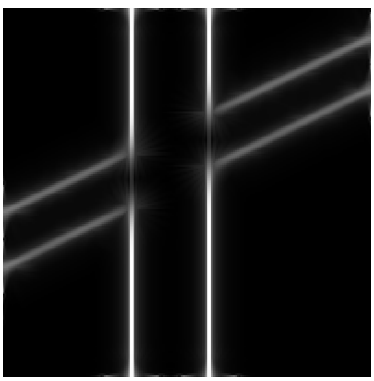


Figure 24: Completion of the 2^{nd} image in time T_1 with scaling factor 1

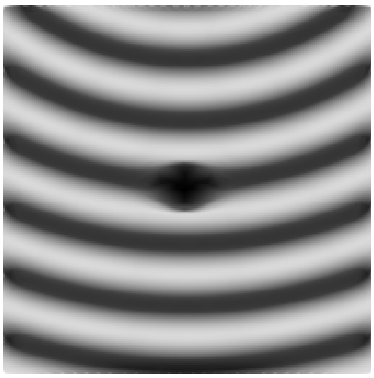


Figure 25: Completion of the 2^{nd} image in time T_1 with scaling factor 0.999

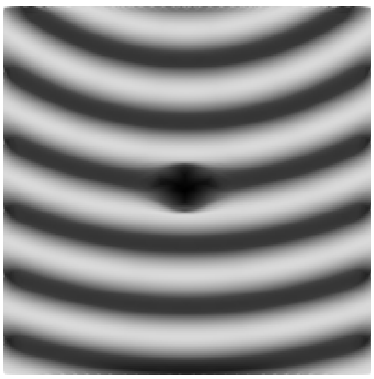
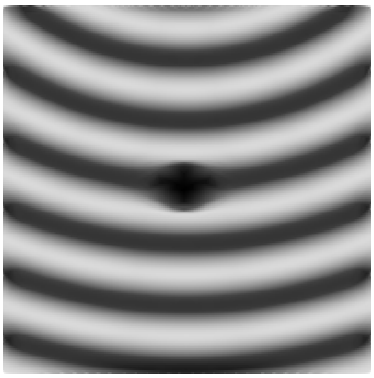


Figure 26: Completion of the 2^{nd} image in time T_1 with scaling factor 0.99



The other approach to refining time step during computation is to set the size of time intervals in which the diffusion part and non-maximal suppression part of algorithm should be computed and then change number of steps within each of these intervals (so that the period for which the diffusion or non-maximal supression is the same in each iteration, but the time step used for computation is getting smaller). The number of global iterations is not going to change, but in each step, the number of steps in diffusion and non-maximal suppresion operator will increase. This method of refining time step is demonstrated in following figures. If the number of steps within diffusion and non-maximal suppression was to be increased after each global iteration, the computation would be very time consuming and that is why a step was added in every tenth or fifth iteration. The results are shown after 50 global iterations. The results showed to be dependent on initial number of steps and thus two sets of images were produced. Again also picture with constant number of steps in all the iterations is shown for comparison.

The resulting images are quite strongly affected by the initial setting. The completion starting at 1 step per diffusion and non-maximal suppression do not reach the same level of completion as the results with constant number of 3 steps per iteration, even though the computation is more time consuming. Figures completed starting with 3 steps per diffusion and non-maximal suppression seem to acquire approximately the same level of completion. Detailed observation of the images shows that the results obtained by increasing the number of steps does not lead to increased quality of completion. The images are not completed as efficiently as when using number of steps fixed to be 3 during the whole computation. That is why the extra computation cost required by this time step refinement does not seem to be justified.

Figure 27: Completion of the 1^{st} image with 3 steps within each algorithm module

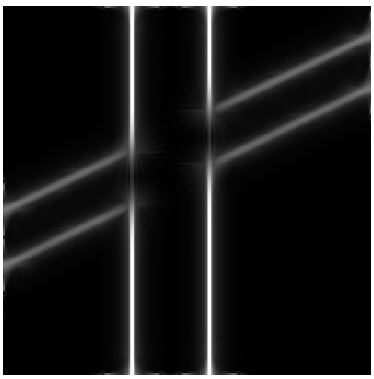


Figure 28: Completion of the 1^{st} image with a step added in every 10^{th} iteration starting from 1

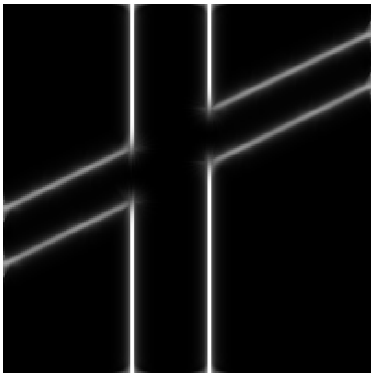


Figure 29: Completion of the 1^{st} image with a step added in every 5^{th} iteration starting from 1

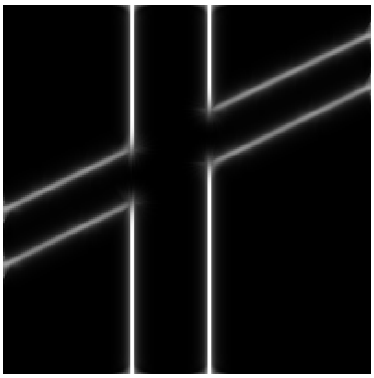


Figure 30: Completion of the 1^{st} image with a step added in every 10^{th} iteration starting from 3

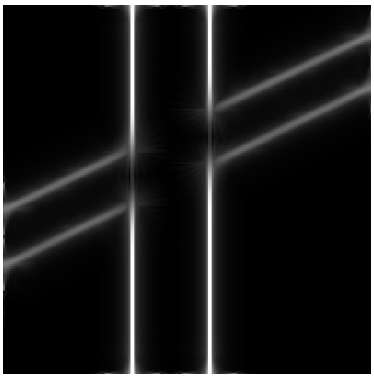


Figure 31: Completion of the 1^{st} image with a step added in every 5^{th} iteration starting from 1

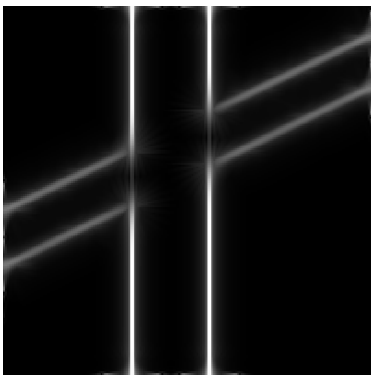


Figure 32: Completion of the 2^{nd} image with 3 steps within each algorithm module

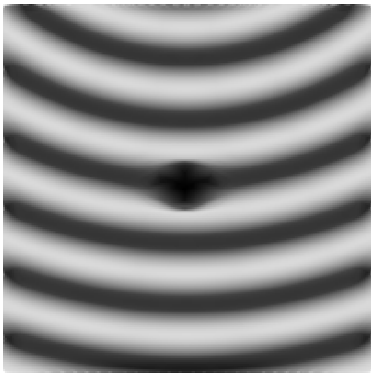


Figure 33: Completion of the 2^{nd} image with a step added in every 10^{th} iteration starting from 1

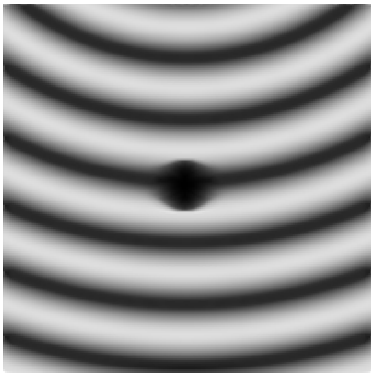


Figure 34: Completion of the 2^{nd} image with a step added in every 5^{th} iteration starting from 1

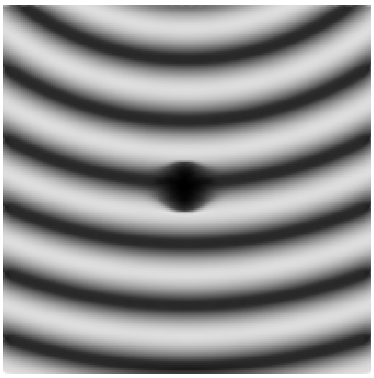


Figure 35: Completion of the 2^{nd} image with a step added in every 10^{th} iteration starting from 3

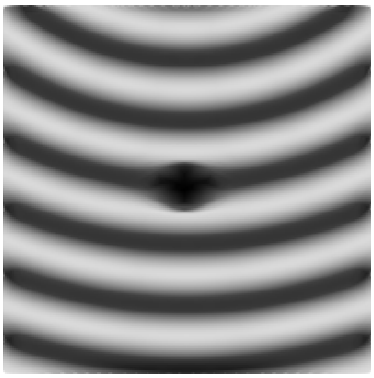
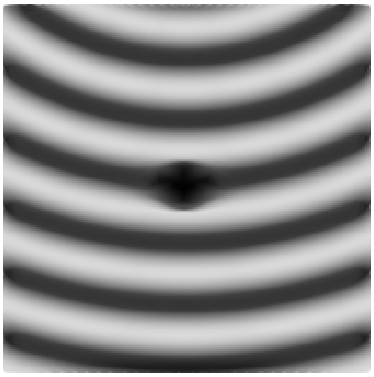


Figure 36: Completion of the 2^{nd} image with a step added in every 5^{th} iteration starting from 3



3.2.3 Ratio between time step used for diffusion operator and time step for non-maximal suppression

The diffusion operator is a linear differential operator, a stability analysis was made in [2] that suggested optimal time step for diffusion part - let's denote this time step Δt_D - in dependance on size of spatial and angular step.

The differential equaiton expressing the non-maximal suppression (using the Laplace-Beltrami flow operator) is, however, nonlinear. It is thus very difficult to create any stability estimates for this part of perceptual completion algorithm. As default choice, the time step for the Laplace-Beltrami flow Δt_{LB} was chosen to be the same as Δt_D. This proved itself to be a reasonable choice, because the stability is preserved for both parts of the algorithm. But it may not be the optimal choice for the Laplace-Beltrami flow operator. Different ratios between Δt_D and Δt_{LB} were therefore tested and results are shown in Fig.37 and Fig.38. The time step for the diffusion part was in according with previous discussion chosen to be $\Delta t_D = \Delta t_{max}$. In each iteration 3 steps were performed in both diffusion and non-maximal suppression. The total of 50 global iterations was computed.

In both figures Fig.37 and Fig.38, the stability estimates are similar. For the time step for Laplace-Beltrami flow operator set to be $\Delta t_{LB} = 7\Delta t_{max}$, it is easy to see that the instability occurs. From Fig.38 it seems that the instability starts for time step five times as big as Δt_{max}. The same conclusion can be made by close observation of results representred in Fig.37 (even though this observation can not be made in the magnification of the printed version of this paper). By this rough method, the stability region of Laplace-Beltrami flow could be estimated so that the operator is stable for time step satisfying

$$\Delta t_{LB} \leq 4\Delta t_{max} \tag{23}$$

For $s = 0.5$, the completion progresses in slower rate, by heightening the scale, the completion rate is increasing, but with improving completion efficiency, the image gets more "blurred". The optimal choice of scaling factor depends on priorities of the final user.

Figure 37: Completion of the 1^{st} image with $\Delta t_{LB} = s\Delta t_{max}$

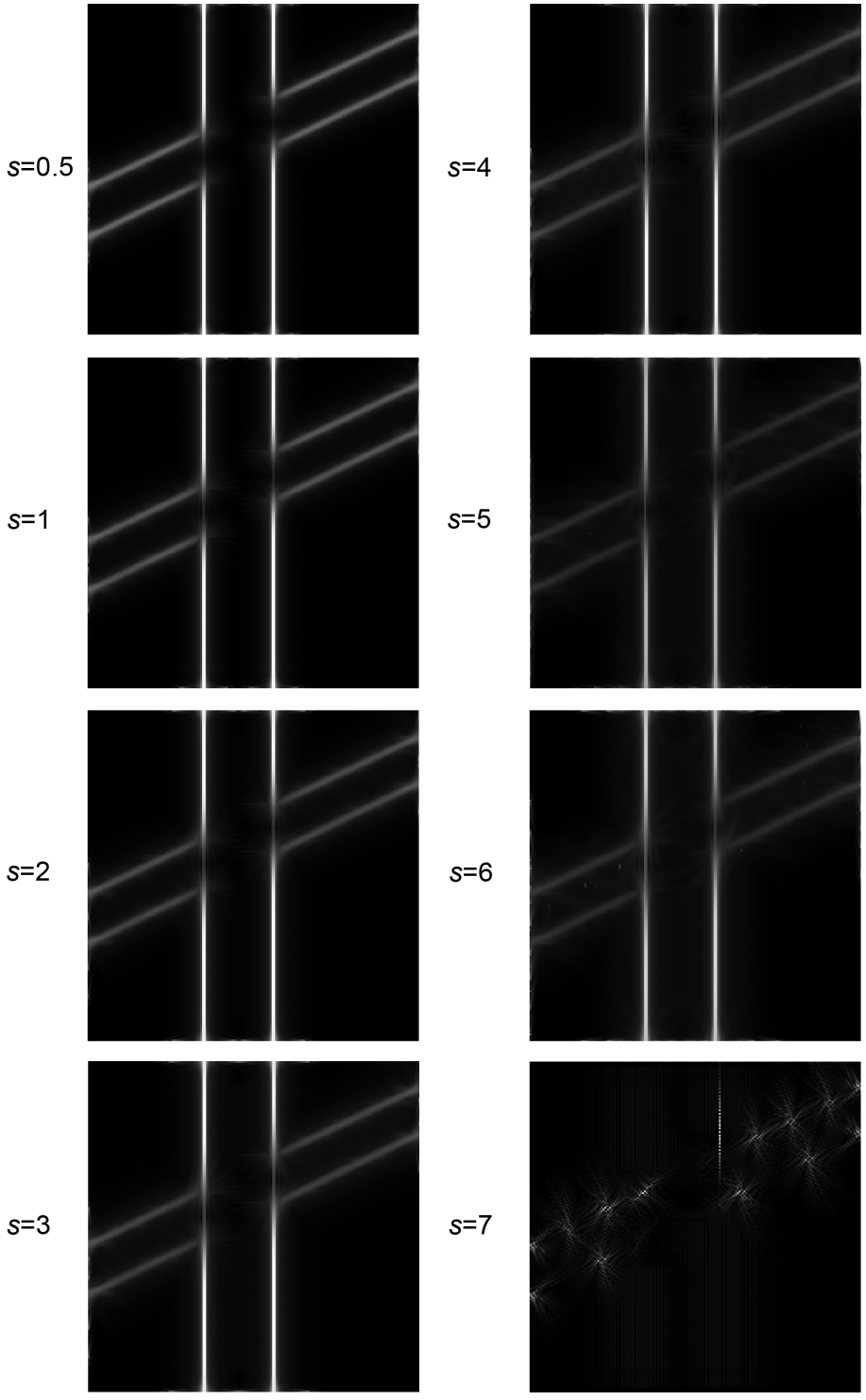
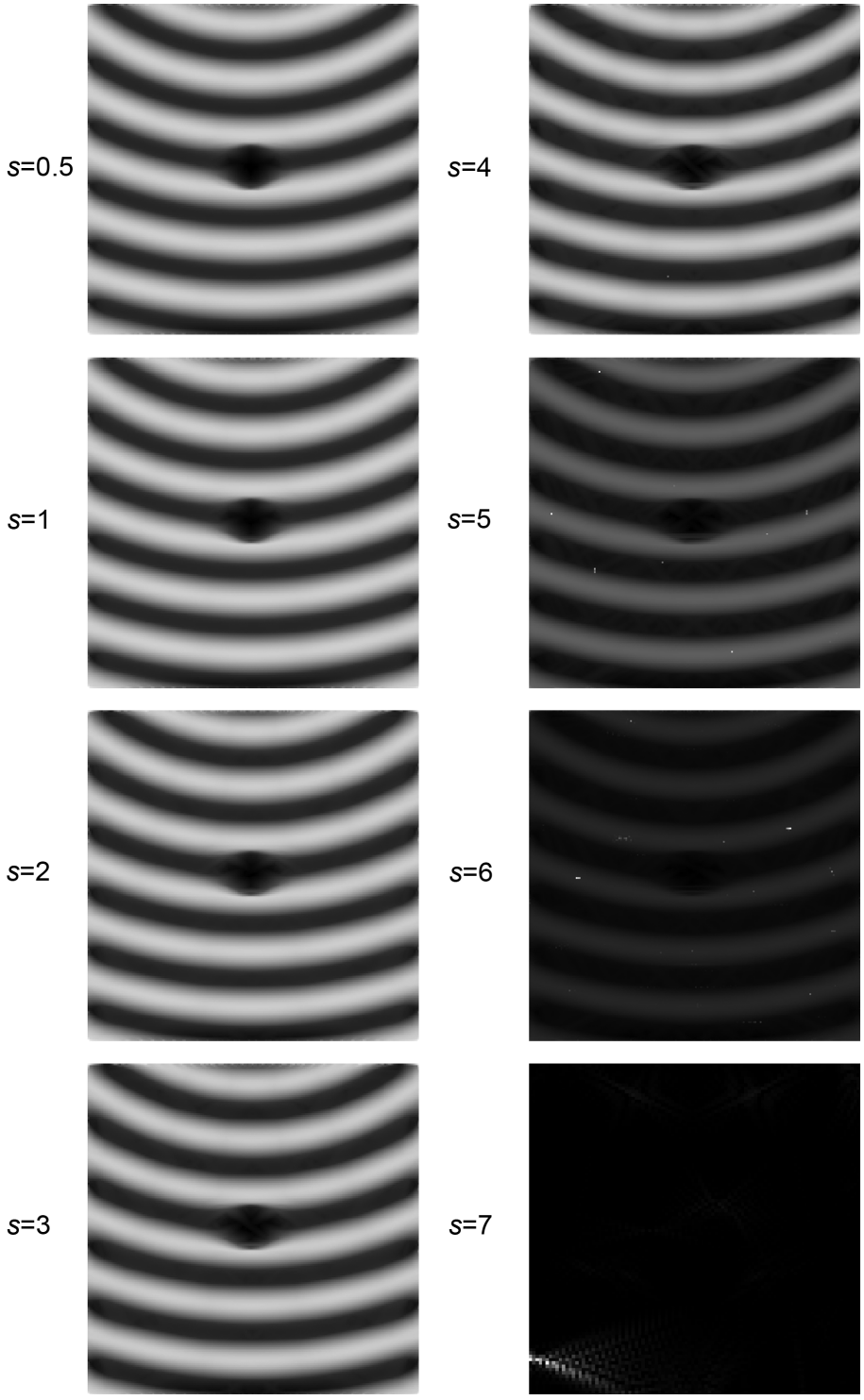


Figure 38: Completion of the 2^{nd} image with $\Delta t_{LB} = s\Delta t_{max}$



4 Conclusion

During work on this report, several settings of the algorithm for perceptual completion constructed as suggested in article [1] were tried. Concluded from these experiments, the most effective setting seems to be using 3 steps of diffusion and 3 steps of Laplace-Beltrami flow in each global iteration, time step for diffusion set to the maximal stable time step given in [2] by $\Delta t_{max} = 0.25 \min\{\Delta x^2, \Delta\theta^2\}$, time step for Laplace-Beltrami should be less or equal to $4\Delta t_{max} = \min\{\Delta x^2, \Delta\theta^2\}$ to preserve the stability of non-maximal suppression procedure. These settings seem to be the best compromise between computational costs and computational accuracy.

It is difficult to find a way how to express the accuracy of the numerical solution of perceptual completion model in a mathematical way and thus the results are represented in the form of the "completed" images projected in the roto-translational space.

The results are presented mostly after 50 global iterations of the algorithm. Part of the reason is the time cost of the computation, another reason is that due to the diffusion process, if the algorithm was run for too long time, the original data would be diffused until all the information would be lost.

List of Figures

1 1^{st} image 9
2 Roto-translational projection of the 1^{st} image 9
3 Ideal completion of the 1^{st} image in roto-translational space 9
4 2^{nd} image 10
5 Roto-translational projection of the 2^{nd} image with the hole 10
6 Ideal completion of the 2^{nd} image in roto-translational space 10
7 Comparison of results for 1^{st} image for different ratios of diffusion steps N_1 and non-maximal supression steps N_2 11
8 Comparison of results for 1^{st} image for different ratios of diffusion steps N_1 and non-maximal supression steps N_2 12
9 Comparison of results for 2^{nd} image for different ratios of diffusion steps N_1 and non-maximal supression steps N_2 13
10 Comparison of results for 2^{nd} image for different ratios of diffusion steps N_1 and non-maximal supression steps N_2 14
11 Completion of the 1^{st} image in time T_1 with time step Δt_{max} 16
12 Completion of the 1^{st} image in time T_1 with time step $0.1\Delta t_{max}$ . . 16
13 Completion of the 1^{st} image in time T_1 with time step $0.05\Delta t_{max}$ . . 16
14 Completion of the 2^{nd} image in time T_1 with time step Δt_{max} 17
15 Completion of the 2^{nd} image in time T_1 with time step $0.1\Delta t_{max}$ . . 17
16 Completion of the 2^{nd} image in time T_1 with time step $0.05\Delta t_{max}$. 17
17 Completion of the 1^{st} image in time T_2 with time step Δt_{max} 18
18 Completion of the 1^{st} image in time T_2 with time step $0.1\Delta t_{max}$ . . 18
19 Comparison of completion of the 1^{st} image with changing $N_1 = N_2 = n$ 20
20 Comparison of completion of the 2^{nd} image with changing $N_1 = N_2 = n$ 21
21 Completion of the 1^{st} image in time T_1 with scaling factor 1 23
22 Completion of the 1^{st} image in time T_1 with scaling factor 0.999 . . 23
23 Completion of the 1^{st} image in time T_1 with scaling factor 0.99 . . . 23
24 Completion of the 2^{nd} image in time T_1 with scaling factor 1 24
25 Completion of the 2^{nd} image in time T_1 with scaling factor 0.999 . . 24
26 Completion of the 2^{nd} image in time T_1 with scaling factor 0.99 . . . 24
27 Completion of the 1^{st} image with 3 steps within each algorithm module 26
28 Completion of the 1^{st} image with a step added in every 10^{th} iteration starting from 1 26
29 Completion of the 1^{st} image with a step added in every 5^{th} iteration starting from 1 26
30 Completion of the 1^{st} image with a step added in every 10^{th} iteration starting from 3 27
31 Completion of the 1^{st} image with a step added in every 5^{th} iteration starting from 1 27
32 Completion of the 2^{nd} image with 3 steps within each algorithm module 28
33 Completion of the 2^{nd} image with a step added in every 10^{th} iteration starting from 1 28
34 Completion of the 2^{nd} image with a step added in every 5^{th} iteration starting from 1 28

35 Completion of the 2^{nd} image with a step added in every 10^{th} iteration starting from 3 29
36 Completion of the 2^{nd} image with a step added in every 5^{th} iteration starting from 3 29
37 Completion of the 1^{st} image with $\Delta t_{LB} = s\Delta t_{max}$ 31
38 Completion of the 2^{nd} image with $\Delta t_{LB} = s\Delta t_{max}$ 32

References

[1] G. Citti and A. Sarti. A cortical based model of perceptual completion in the roto-translation space. *J Math Imaging Vis*, 24:307–326, 2006.

[2] P. M. Viddal. A perceptual completion model. Pre-master at the Master's degree programme within Industrial Mathematics at NTNU, Trondheim, 2008.

[3] P. M. Viddal. Minimal surfaces in a sub-riemannian geometry with applications to a perceptual completion model. Master's Thesis, Faculty of Information Technology, Mathematics and Electrical Engineering, NTNU, Trondheim, 2009.

[4] Peter J. Olver and Chehrzad Shakiban. *Applied Linear Algebra*. Pearson Education, Inc., 2006.