Sound Propagation

Group 13
Ruben Bromée, Viktor Carlsson, Hugo Dahl, Uma Eriksson
TNM085

March 17, 2021

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Sound is waves of pressure propagated through different mediums, for example air and water. When an object vibrates, it displaces the mediums particles from the equilibrium position, this causes the particles to collide with the adjacent particles, these particles then collide with the next adjacent particles and so on. This process continues from the source until the sound waves reaches, for example, our ears. The particles do not travel from the source to the listener, rather its the disturbance produced by the source traveling through the medium, meaning the particles oscillate. This disturbance is the sound wave. [1]

In this project, the propagation of sound in a room has been modeled and visualized. The pressure deviations over time is what has been modeled and visualized. The propagation was only modeled in two dimensions. Only a simple reflection was implemented with no consideration for the walls' absorption coefficients. The room was also assumed to be room temperature making the speed of sound 343 *m/s*. These simplifications has been made since the model was iteratively made more complex and this stage of complexity is what was able to be implemented during the time period of the project.

Chapter 2

Method

The acoustic wave equation [2] was used to model the propagation of sound. The acoustic wave equation was used with a two-dimensional laplacian and a second-order central finite difference approximation [3]. The accoustic wave equation is given in 2.1. p is the sound pressure depending on room position and time, t is time, c is the speed of sound, x is the variable describing the width of the room and y is the variable describing the length of the room. Discretization of the partial derivatives in time, the spatial x-direction and the spatial y-direction are given in equations 2.2, 2.3 and 2.4 respectively. i is the index of the width, j is the index of the length, n is the index of the time, Δx is the spatial sampling step for the width of the room, Δy is the spatial sampling step for the length of the room. The discretized version of the acoustic wave equation is shown in 2.5 where $\Delta x = \Delta y$.

$$\frac{\partial^2 p}{\partial t^2} = c^2 \nabla^2 \mathrm{p} \Leftrightarrow \frac{\partial^2 p}{\partial t^2} = c^2 \left(\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} \right) \tag{2.1}$$

$$\delta_t^2 = \frac{p_{i,j}^{n+1} - 2p_{i,j}^n + p_{i,j}^{n-1}}{\Delta t^2} \tag{2.2}$$

$$\delta_x^2 = \frac{p_{i+1,j}^n - 2p_{i,j}^n + p_{i-1,j}^n}{\Delta x^2} \tag{2.3}$$

$$\delta_y^2 = \frac{p_{i,j+1}^n - 2p_{i,j}^n + p_{i,j-1}^n}{\Delta y^2} \tag{2.4}$$

$$p_{i,j}^{n+1} = \frac{c^2 \Delta t^2}{\Delta x^2} (p_{i+1,j}^n + p_{i,j+1}^n + p_{i-1,j}^n + p_{i,j-1}^n - 4p_{i,j}^n) + 2p_{i,j}^n - p_{i,j}^{n-1} \tag{2.5}$$

MATLAB [4] was used to model and visualize the propagation. The modeling was done by first setting initial values which can be seen in appendix A. The sound was modeled as an impulse source which was implemented as an impulse in the first time step at the center of the room. This impulse could be placed at different positions in order to simulate a sound source in different places of the room. A three dimensional matrix was set up which had the two spatial dimensions and time as its dimensions. After the first time step with the impulse sound, each new time step was calculated using 2.5. More impulses were created during the simulation with regards to the initial frequency. To visualize the simulation the values in the room in each time step was color mapped, captured in a frame and stored in a video file. The complete MATLAB code can be seen in appendix B. To run the simulation and animation of the propagation a computer that can run MATLAB is needed. Frequency, sound position and room size can be changed by changing the values of the constants in the code.

Chapter 3

Result

In the first stages of development a simple model that only modeled one impulse was visualized in a way that was good enough for noticing errors in the implementation of the acoustic wave equation. This visualization can be seen in Figures 3.1, 3.2, 3.3. The whiter colors represent high variations in sound pressure and the blacker colors represent low variations in sound pressure.

The project resulted in a simulation and visualization of sound propagating in a two dimensional room, as seen in Figures 3.4, 3.5 and 3.6. The yellow and orange colors represent high variations in sound pressure while the blue colors represent low variations in sound pressure.

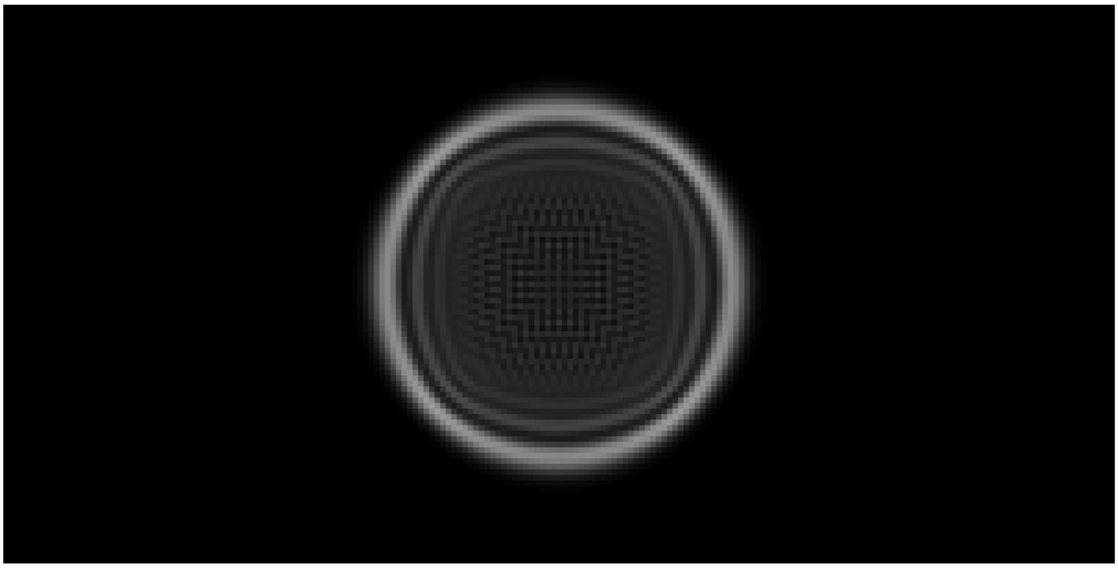

Figure 3.1: 2 dimensional sound propagation at time step 100 with the sound source placed in the center of a 101 x 201 room.

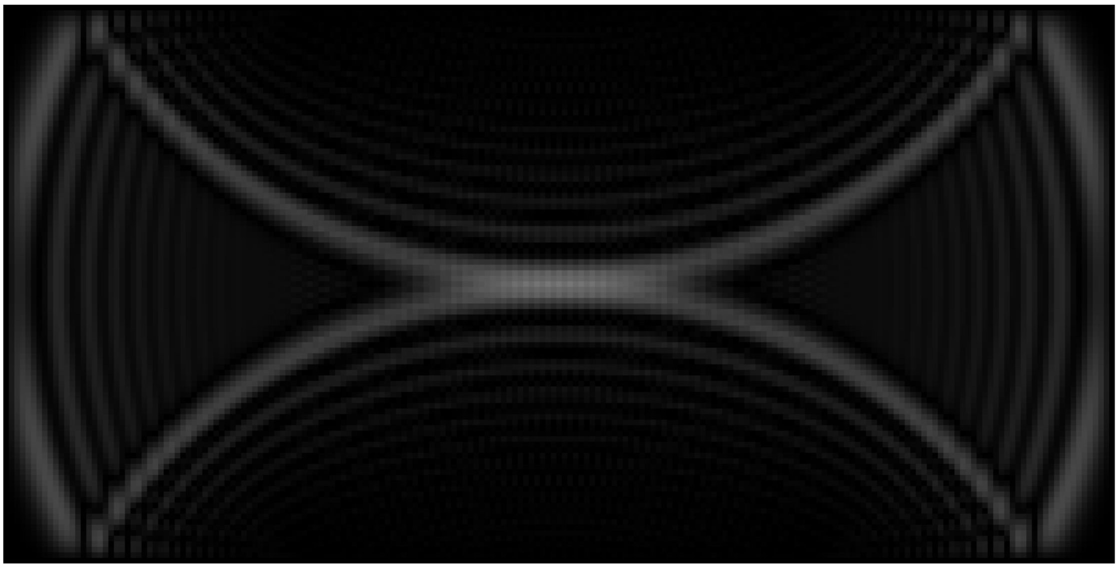

Figure 3.2: 2 dimensional sound propagation at time step 300 with the sound source placed in the center of a 101 x 201 room.

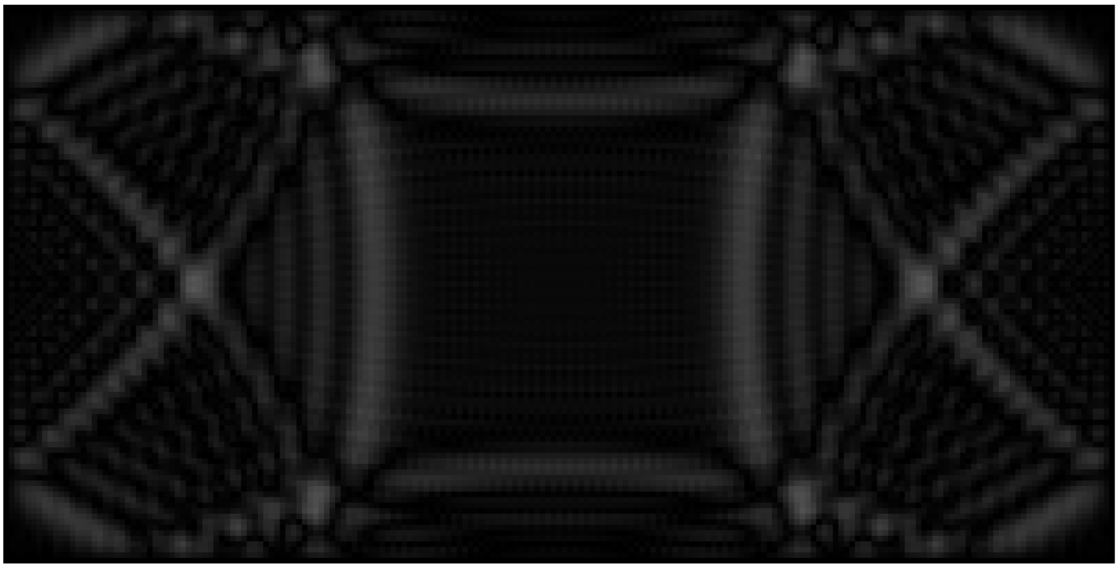

Figure 3.3: 2 dimensional sound propagation at time step 500 with the sound source placed in the center of a 101 x 201 room.

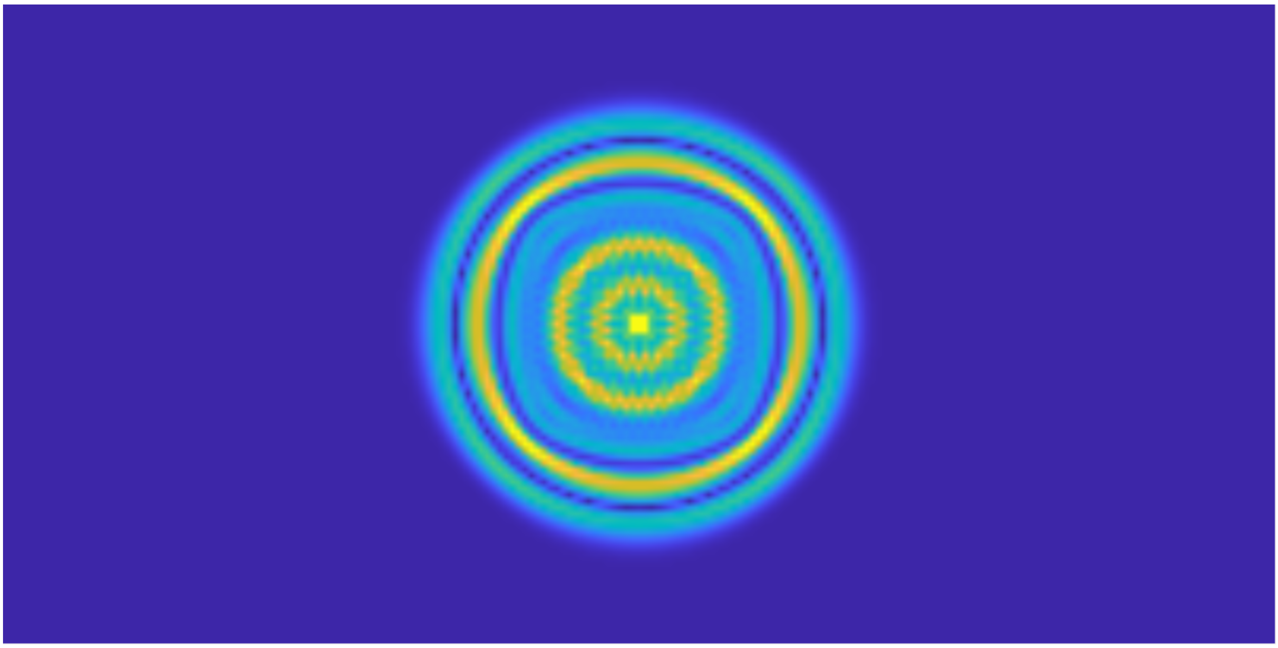

Figure 3.4: 2 dimensional sound propagation at time step 100 with the sound source placed in the center of a 101 x 201 room with a sound frequency of 500 Hz.

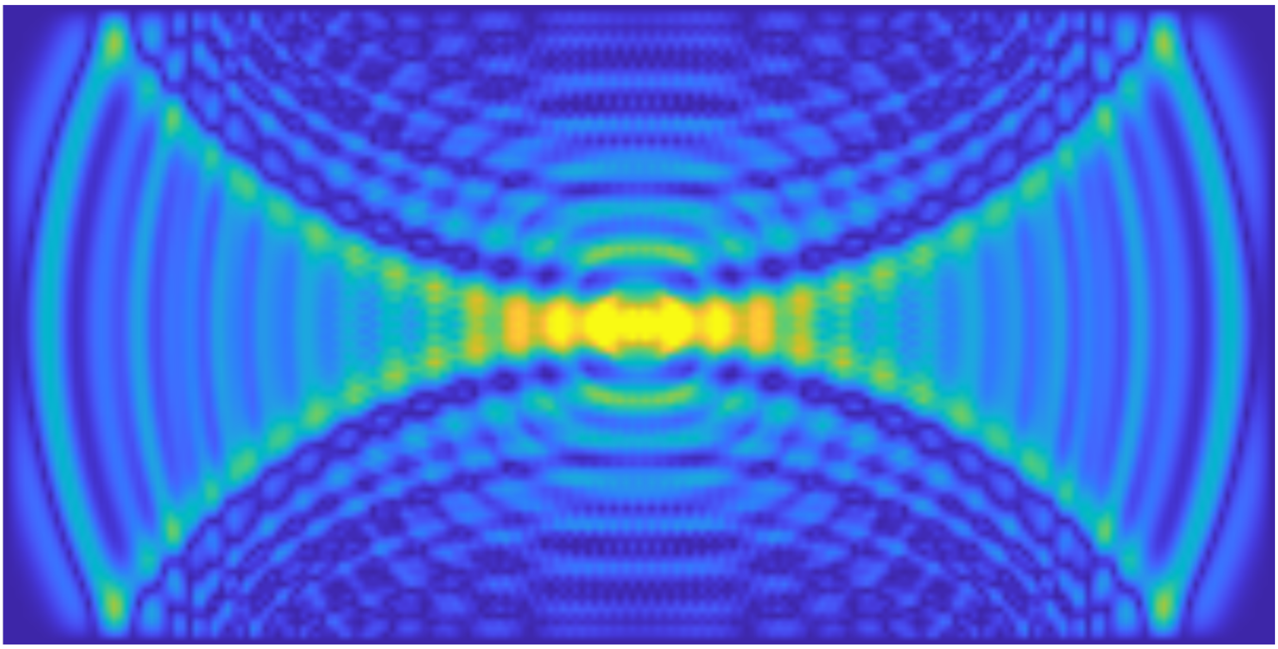

Figure 3.5: The same sound propagation as Figure 3.4 but at time step 300.

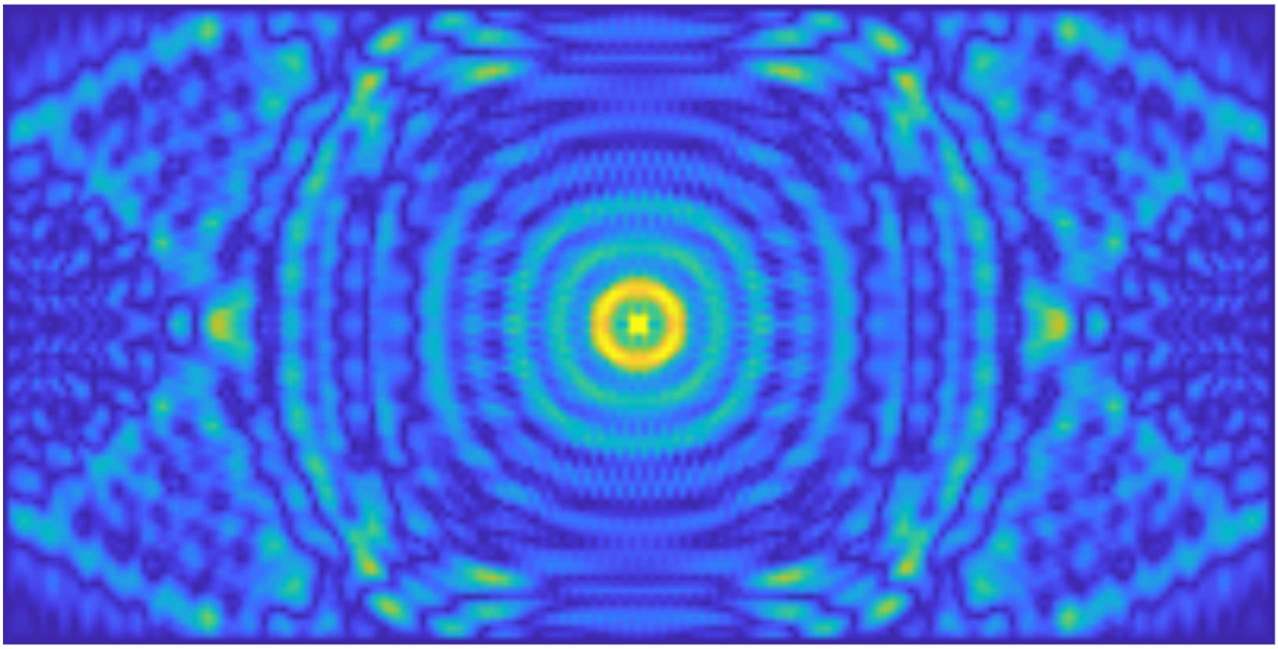

Figure 3.6: The same sound propagation as Figure 3.4 and Figure 3.5 but at time step 500.

Chapter 4

Discussion

The visualization shows how sound propagates in a very clear and straightforward way, the colors are especially helpful in separating the sound waves which in turn makes it easier to see how the waves interact and move around. The black and white images sufficiently show the waves to a certain extent, but without the color mapping it becomes a bit harder to observe things such as interference and reflection. The interpolation in the colored visualization also helps with creating nicer looking and less jagged waves. A similar pattern emerges in the wave propagation from both just one impulse as in figures 3.1, 3.2, 3.3 and in the propagation that has a frequency, which was shown in figures 3.4, 3.5 and 3.6. The colored propagation that has several impulse signals created in the given frequency has more interference interactions than the uncolored one.

A relatively small amount of code was needed to simulate a simple model of sound propagation. The algorithm can become quite computationally demanding if the amount of time steps are increased or if another dimension is added to the simulation matrix since the code already contains a triple nested for-loop.

Further complexity that could have been implemented is a model in three dimensions. Since the acoustic wave contains a laplacian it can easily be expanded into three dimensions since the laplacian just becomes three dimensional. Another expansion that could have been made, would be to have a more complex model for reflection which would consider the absorption coefficient of the material of the walls. An interface to change things like sound frequency, sound position and room size without having to recompile the program could also be an expanded implementation.

Originally the plan was to implement the animation with the help of *WebGL* [5] and the library *Three.js* [6], however after further discussion the group decided that this method would just create unnecessary complexity. It was decided to simply implement the animation in MATLAB, which worked without issue. Had the group implemented further complexity like 3 dimensions, MATLAB would still suffice for the animation implementation, as it supports 3D graphs and animations. If the visualization would have been in the browser, *Three.js* would have been a better option since it is optimized to implement 3D graphics on the web.

Chapter 5

Conclusion

In conclusion sound propagation can be simulated and visualized in a clear way using a relatively small amount of code. Complexity could be added to both the simulation and the visualization but a consideration should be made towards how much computing power is needed for a more complex simulation.

Bibliography

[1] Dr David Sharp, *What is sound?*, OpenLearn, 2019-08-30, retrieved: 2021-03-11
https://www.open.edu/openlearn/science-maths-technology/science/physics-and-astronomy/physics/what-sound

[2] J. Saarelma and G. Greco, *SOUND FIELD VISUALIZATION USING THE FINITE-DIFFERENCE TIME-DOMAIN METHOD AND MEASUREDSPATIAL ROOM IMPULSE RESPONSES*, Institute of Acoustics, 2015, hämtad: 2021-03-11
https://users.aalto.fi/ ktlokki/Publs/p68_2015.pdf

[3] Peter Olver, *Introduction to Partial Derivative Equations*, Springer Science & Business Media, 2013

[4] MATLAB, The MathWorks, Inc. 1994, retrived: 2021-03-11
https://www.mathworks.com/products/matlab.html

[5] WebGL, Khronos Group, 2011, retrived: 2021-03-17
https://www.khronos.org/webgl/

[6] Three.js, Three.js Authors, 2010, retrived: 2021-03-17
https://threejs.org/

Appendix A

Initial simulation values

Initial values	
Name	Value
Spatial sampling step	0.1 m
Wave velocity	343 m/s
Room length	10.1 m
Room Width	20.1 m
Sound frequency	500 Hz
Number of time steps	1001
Time step	0.0001 s
Period	0.002 s

All initial values used in the implementation of the sound propagation.

Appendix B

MATLAB code

```
delta_x = 0.1; % Spatial sampling step
c = 343; % Wave velocity for sound in room temperature air
m = 101; % Room length
n = 201; % Room width
origo_m = ceil(m/2); % Length midpoint
origo_n = ceil(n/2); % Width midpoint
f = 500; % Sound frequency
n_t = 1001; % Number of time steps
delta_t = 0.0001; % Time step
P = zeros(n_t, m, n); % Room visualization matrix
period = 1 / f; % Wave time period
period_n_t = round(period / delta_t); % Wave time period expressed ...
% in amount of time steps

% Placement of sound source. Orgio of room as default.
% origo_m = 100;
% origo_n = 200;

% Starting conditions, impulse source.
P(1, origo_m, origo_n) = 1;
P(1, origo_m + 1, origo_n + 1) = 1;
P(1, origo_m - 1, origo_n + 1) = 1;
P(1, origo_m - 1, origo_n - 1) = 1;
P(1, origo_m + 1, origo_n - 1) = 1;
P(1, origo_m + 1, origo_n) = 1;
P(1, origo_m - 1, origo_n) = 1;
P(1, origo_m, origo_n + 1) = 1;
P(1, origo_m, origo_n - 1) = 1;
```

```


% Accoustic wave equation implemented numerically
for j = 2:n_t-1 % Iteration over time
    % Create new impulse according to frequency
    if mod(j, period_n_t) == 0
        P(j, origo_m, origo_n) = 1;
        P(j, origo_m + 1, origo_n + 1) = 1;
        P(j, origo_m - 1, origo_n + 1) = 1;
        P(j, origo_m - 1, origo_n - 1) = 1;
        P(j, origo_m + 1, origo_n - 1) = 1;
        P(j, origo_m + 1, origo_n) = 1;
        P(j, origo_m - 1, origo_n) = 1;
        P(j, origo_m, origo_n + 1) = 1;
        P(j, origo_m, origo_n - 1) = 1;
   end
    for k = 2:m-1 % Iteration over length
        for l = 2:n-1 % Iteration over width
            P(j+1, k, l) = c^2 * (delta_t^2 / delta_x^2) * ...
            (P(j, k + 1, l) - 4 * P(j, k, l) + P(j, k - 1, l ) + ...
            P(j, k, l + 1) + P(j, k, l - 1)) + ...
            2 * P(j, k, l) - P(j - 1, k, l);
        end
    end
end

% Animation and video write loop
movie_obj = VideoWriter('wave_prop_impulse_flashing.avi');
open(movie_obj);
fig = figure(1);
for i = 1:n_t
    i_p(:, :) = P(i , :, :);
    s = abs(i_p(:,:));
    imshow(s(:, :), 'InitialMagnification', ...
    600, 'Interpolation', 'bilinear');
    colormap default;
    drawnow;
    F = getframe(fig);
    writeVideo(movie_obj, F);
end
close(movie_obj);
```