Eurographics Symposium on Parallel Graphics and Visualization (2015)
C. Dachsbacher, P. Navrátil (Editors)

Visualization of 2D Wave Propagation by Huygens' Principle

Stefan Heßel[1], Oliver Fernandes[1], Sebastian Boblest[1], Philipp Offenhäuser[2], Malte Hoffmann[1], Andrea Beck[1], Thomas Ertl[1], Colin Glass[2], Claus-Dieter Munz[1], and Filip Sadlo[3]

[1]University of Stuttgart, Germany
[2]High Performance Computing Center Stuttgart, Germany
[3]Heidelberg University, Germany

Abstract

We present a novel technique to visualize wave propagation in 2D scalar fields. Direct visualization of wave fronts is susceptible to visual clutter and interpretation difficulties due to space-time interference and global influence. To avoid this, we employ Huygens' principle to obtain virtual sources that provide a concise space-time representation of the overall dynamics by means of elementary waves. We first demonstrate the utility of our overall approach by computing a dense field of virtual sources. This variant offers full insight into space-time wave dynamics in terms of elementary waves, but it reflects the full problem of inverse wave propagation and hence suffers from high costs regarding memory consumption and computation. As an alternative, we therefore provide a less accurate and less generic but more efficient approach. This alternative performs wave front extraction with subsequent Hough transform to identify potential virtual sources. We evaluate both approaches and demonstrate their strengths and weaknesses by means of a GPU-based prototype and an implementation on a Cray XC40 supercomputer, using data from different domains.

Categories and Subject Descriptors (according to ACM CCS): I.6.6 [Simulation and Modeling]: Simulation Output Analysis—; J.2 [Physical Sciences and Engineering]: Physics—

1. Introduction

Scientific visualization has a long and successful history of revealing essential structures in complex data. While already 3D scalar fields demand advanced visualization techniques for proper analysis, visualization of vector and tensor fields poses even more difficult problems due to the high complexity of the involved phenomena and the structures they cause. Propagation of waves is an omnipresent phenomenon occurring in most of these data, which has been, however, widely neglected in their visualization so far.

While vector fields represent a single direction at each point, and the visualization of n-dimensional tensor fields is typically based on their n-dimensional eigensystem, there are possibly infinitely many waves propagating simultaneously through each point of a domain, each of them potentially having global impact. This complexity is present not only on the structural side—wave propagation phenomena are also computationally expensive to simulate and represent, because very small time steps and high spatial resolutions are required. Together with the high computational cost of inverse wave projection discussed below, this makes visualization of wave propagation exceptionally difficult and costly—and is the reason why this paper addresses wave propagation in 2D domains only. Nevertheless, extension to higher dimensions would be straightforward.

Wave propagation phenomena are traditionally examined by observing displacement (amplitude) in space and time. In 2D fields, mapping amplitude to color and observing the resulting video is commonly used. A drawback with this approach is, however, that already single frames of such video exhibit complexity that is beyond visual interpretation—the massive superposition of wave fronts rapidly exceeds the limits of human perception and reasoning.

Following the basic visualization aim of reducing data to their essential structure, we seek for a more effective representation of wave fronts. A straightforward approach to reduce the time-dependent 2D field to a set of 1D curves, would be to obtain the wave fronts by extracting crease lines, i.e., ridge and valley lines, and observing the resulting animations. To avoid the difficulty of perceiving video, one could additionally employ a space-time representation, i.e., treat time as an additional 'spatial' dimension. In this repre-

sentation, the wave fronts become space-time surfaces, and their extraction by means of crease surfaces becomes more robust because spatio-temporal coherence within the data can be exploited. However, although this approach would provide a static picture of the time-dependent phenomenon, it would typically suffer from occlusion, in particular, because it cannot reduce the massive superposition (interference) of wave fronts. Hence, further reduction is required to achieve expressive visualization.

According to Huygens' principle [BW03], arbitrary wave fronts can be represented by a superposition of elementary waves (Figure 1). Hence, each wave front in space can be represented by a set of virtual sources (i.e., elementary waves), located at respective positions in space and time. A straightforward discretizing approach, that we follow, is to assume a regular spatio-temporal grid of virtual sources, and determine the amplitude of each such that the superposition of the respective elementary waves fits the given dataset. Since this approach is computationally demanding, we also provide an approximating alternative that avoids fitting a large number of elementary waves. Instead, it employs the Hough transform for efficient search of virtual sources and reduces the fitting to their respective, extracted wave fronts.

While our overall approach is generic and could be used for the visualization of waves in vector and tensor fields as well, we focus here on waves in scalar fields. Specifically, our contributions include:

- Employing Huygens' principle for the visualization of wave propagation in terms of elementary waves.
- Introduction of virtual source fields and sets.
- Extraction of wave fronts by space-time crease surfaces.

2. Related Work

Visualization of wave propagation is still a rather small field in scientific visualization. So far, visualization of waves has been concentrating on the visualization of wave fronts, often defined and extracted by means of crease surfaces. A seminal work on the definition and extraction of height ridge surfaces is that by Eberly [Ebe96], based on the Hessian. Deines et al. [DBM*06] visualize acoustic wave fronts including particles and maps, and coloring by incoming intensity. Obermaier et al. [OMD*12] present mesh-free extraction of crease surfaces from 3D acoustic data.

The problem of the inversion of the wave equation is also an important problem from a purely mathematical point of view. Tarantola [Tar05] provides an overview on different inverse problems and a detailed mathematical discussion of their solution. As discussed in Section 4.1, our approach is amenable to common least squares fitting.

3. Fundamentals

3.1. Scalar Waves

A one-dimensional scalar wave is a function $\psi(x,t)$ that solves the partial differential equation

$$\frac{\partial^2 \psi(x,t)}{\partial x^2} - \frac{1}{v^2}\frac{\partial^2 \psi(x,t)}{\partial t^2} = 0, \tag{1}$$

with position x, time t, and phase velocity v. Any function $\psi(x,t) = f(x-vt) + g(x+vt)$ with f and g being arbitrary functions of one variable satisfies Equation (1), particularly, a wave need not be a periodic function in x or t. Equation (1) can be generalized to arbitrary dimensions by replacing the partial derivative $\partial^2/\partial x^2$ with the n-dimensional Laplacian $\Delta_n = \sum_{i=1}^{n} \partial^2/\partial x_i^2$. However, in our case, we focus on elementary waves, i.e., spherical waves $\Psi(\mathbf{x},t)$ that originate at a given point and propagate symmetrically in all directions. It is straightforward to show that all 3D spherical waves can be expressed with the general solution of Equation (1) as

$$\Psi(\mathbf{x},t) = \frac{\psi(r,t)}{r}, \text{ where } r = \sqrt{x^2+y^2+z^2} \quad \text{in 3D}, \tag{2}$$

i.e., Ψ decays $\sim 1/r$ with increasing distance r from its origin [BW03]. Physically, the decay $\sim 1/r$ is a consequence of the conservation of energy. We use this $1/r$ dependency to model the amplitude decay of our elementary waves with $r = \sqrt{x^2+y^2}$ as the underlying physical models are 3D, except for the water basin in Section (6). Here however, we also neglect friction effects.

One example that suggests itself for an application of our method, is acoustic waves in fluids. Such waves are deviations $p(\mathbf{x},t)$ of the pressure from its mean value p_0 and closely related to the generation of sound, or, more specifically, unwanted noise that should be minimized in many industrial applications. To this end, our technique provides a visualization of potential causes for noise. For acoustic waves, the energy density is proportional to p^2 and the decay $p \sim 1/r$ ensures that the flux through the surface $O = 4\pi r^2$ of every sphere around a source is equal.

Please note that we denote the phase velocity 'wave speed' or 'propagation speed' throughout this paper. Further, we assume, and limit our technique to, data with uniform wave speed, i.e., we require the propagation speed to be independent of the frequency, time, and location. This assumption holds reasonably well for many physical phenomena such as sound in air or water, light in air, or waves on free fluid surfaces if the fluid is not too shallow.

3.2. Huygens-Fresnel Principle

Our approach is based on an inversion of the well-known Huygens-Fresnel principle. Here, we give a short overview of this principle. Readers interested in more details are referred to standard textbooks on optics, e.g., to Born and Wolf [BW03]. The propagation of waves through space can

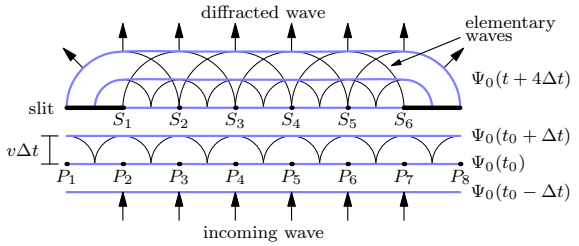


Figure 1: *Huygens' principle applied to the diffraction of a plane wave hitting a single slit. Before the wave reaches the slit, contributions of the elementary waves perpendicular to the direction of propagation cancel each other out. In the slit, the elementary waves at S_1 and S_6 no longer have neighbors with which their orthogonal components could cancel out—the wave is diffracted.*

be described by taking each point P on a wave front at time t_0 as the source for a new elementary wave as in Equation (2). The superposition of all these elementary waves at a later time $t_0+\Delta t$ gives the propagated wave front. A well-known application of Huygens' principle in optics is the diffraction of a plane light wave on a single slit, see Figure 1. To explain why the superposition of the elementary waves only creates a new wave front in the direction of propagation, e.g., $\Psi_0(t_0+\Delta t)$ in Figure 1, and not on the opposite side, e.g., $\Psi_0(t_0-\Delta t)$, Fresnel complemented the Huygens-Fresnel principle with an inclination factor that is 1 in the direction of propagation and 0 in the opposite direction. In Figure 1, the inclination factor is implicitly included by drawing the elementary waves as semicircles. In our inversion of the Huygens-Fresnel principle, however, no such problem arises as we have both spatially and temporally resolved data and hence the direction of propagation is always given.

3.3. Wave Fronts in Space-Time

Any time-dependent 2D scalar field $D(\mathbf{x},t)$ with $\mathbf{x} := (x,y)^\top \in \mathbb{R}^2$ can be interpreted in space-time representation by treating time t as an additional dimension, i.e., as a scalar field $D'(\mathbf{x}')$, with $\mathbf{x}' := (x,y,t)^\top \in \mathbb{R}^3$. Wave front curves in the 2D field turn into wave front surfaces in space-time. To use the definition of elementary waves in a numerical algorithm however, we have to supply them with a finite spatial extent W, where we have to choose W so large that at least two grid nodes are covered by the elementary wave.

For the elementary wave emerging at time t_0 at the point $\mathbf{x}_0$, we then obtain a cone described by the relation

$$\sqrt{\|\mathbf{x}-\mathbf{x}_0\|^2 - v^2(t-t_0)^2} \leq W/2, \quad t > t_0. \tag{3}$$

At each fixed time, the elementary wave is an annulus with inner radius $v(t-t_0)-W/2$ and outer radius $v(t-t_0)+W/2$. Definition (3) also encapsulates that an elementary wave that arises at time t_0 can only contribute to the data field at later times t. The opening angle of this cone $\alpha = \tan v$ directly represents the wave speed, i.e., the slower a wave propagates, the narrower the cone is.

Arbitrary wave fronts represent arbitrary surfaces in space-time. However, since a single wave front, under the aforementioned assumptions, propagates at uniform wave speed, the angle α between the normal of the space-time surface and the time axis is also constant and represents the wave speed. However, even if the wave speed is constant, α can vary due to superposition of wave fronts. During superposition, the fronts of the individual (elementary) waves typically disappear and are replaced by others. This has no impact on our first approach (Section 4.1) that fits a virtual source field, but can, as we will show, affect the alternative approach (Section 4.2) that builds on wave front extraction.

3.4. Hough Transform and Waves

If we now take a complementary look—i.e., instead of looking at elementary waves and how they superimpose over time, we look at a given point in space-time and search for all points that can influence this point in terms of wave propagation—we obtain a similar cone as for the elementary wave in Equation (3), but now extending backwards in time with $t < t_0$. This cone contains *wave source candidates*, i.e., all space-time points that have to be considered as possible sources for D at the given point. We employ the Hough transform for extracting these source candidates.

The Hough transform is a tool widely used in computer vision for detecting parametric shapes [DH72]. It expects binary input corresponding to potential features. The basic idea is to establish a parameter space representing all shapes that can possibly be detected. For each input point, all points in parameter space that represent a possible shape containing this point are voted for, i.e., their value is increased. Subsequently, the accumulation points are extracted from the parameter space, giving the parameters that describe the matched shapes. Circles in images, e.g., can be extracted by using an edge detector for input generation and voting in a parameter space with center coordinates and radius as axes.

For our method, since we assume constant wave speed, the virtual source candidates have their space-time origin as the only parameter while their amplitudes at this point are still undetermined. Therefore, the parameter space for our method is space-time itself. This means marking all parameter configurations for a given input feature point can be achieved by voting for all points lying on the reverse cone extending from that particular feature point.

4. Method

Our technique consists of two alternative approaches: a direct, but computationally expensive approach providing vir-

tual source fields (Section 4.1), and a more efficient, but approximate and limited approach extracting individual virtual sources (Section 4.2).

4.1. Virtual Source Field Fitting

Our goal is to invert Huygens' principle, i.e., to find a set of weighted virtual sources that give rise to the original field if their respective weighted elementary waves are superimposed. To facilitate computation and to keep memory consumption within reasonable limits, we do not solve for the positions of the virtual sources in this first approach. Instead, we assume a user-defined 3D space-time regular grid of virtual sources and limit the problem to finding a weight (amplitude) for each of the sources such that the superposition of the respective elementary waves approximates the original field as accurately as possible. We denote this space-time grid of weights the *virtual source field*.

This problem represents a projection problem, i.e., we project the original space-time dataset D onto a regular space-time grid of basis functions. Each of the basis functions represents an elementary wave in the form of a space-time cone as in Equation (3), with its tip located at the origin of the respective virtual source in space and time. In this formulation, the amplitudes (weights) $\mathbf{S}$ follow from a least squares problem of the form

$$\|A\mathbf{S} - \mathbf{D}\| \rightarrow \min. \tag{4}$$

Here, A is an $m \times n$ matrix, where m is the number of samples from the original dataset and n the number of virtual sources, i.e., basis functions, with $m > n$. We obtain the form

$$A(\mathbf{x}_D, t_D, \mathbf{x}_S, t_S) = \frac{1 - \left[(t_D - t_S)v - |\mathbf{x}_D - \mathbf{x}_S|\right]/W}{|\mathbf{x}_D - \mathbf{x}_S|} \tag{5}$$

for the component of A that describes the influence of the elementary wave located at $(\mathbf{x}_S, t_S)$ on the data point $(\mathbf{x}_D, t_D)$. This definition takes the amplitude decay $\sim 1/r$ in Equation (2) into account. Please note that all components outside of the cones (3) vanish, especially all components with $t_D < t_S$. We can easily ensure the requirement $m > n$ by introducing, if necessary, more sample points by interpolation of the original space-time field. In all our results, we have $m = r_x \times r_y \times (r_t - 1)$, with r_i being the dataset resolution in the respective direction. This way, we ignore the source candidates with no influence, i.e., those in the latest time slice.

Generally, problems similar to our least squares formulation can suffer from being not well-posed, which can lead to substantial noise in the solution. In such cases Tikhonov regularization [TGSY95] is an often used approach to give preference to a particular solution. We experimented with including such a regularization in our algorithm. However, due to the physically motivated amplitude decay $\sim 1/r$ that we apply, we found an ordinary least squares to be sufficient.

4.2. Virtual Source Extraction

Due to the high computational demands of the virtual source field extraction explained so far, we now present an alternative approach that is less demanding, but also less generic and less accurate. The computational cost of the technique from Section 4.1 basically scales cubically with the number n of virtual sources to fit, and its memory requirement scales with $mn + n^2$. Hence, a primary strategy to reduce the overall cost is to reduce the number n of virtual sources. We achieve this by reducing the problem at different stages. First, instead of taking the whole space-time dataset into account, we only consider the wave fronts it contains. We extract these in terms of space-time crease surfaces (Section 4.2.1). These surfaces represent feature candidates for elementary waves. For efficiency, we search for the respective virtual sources using the Hough transform, i.e., we vote in the parameter space (Section 3.4) of the space-time cones and detect and filter accumulation points in this space (Section 4.2.2). Finally, we fit the obtained source point candidates to their influence range in the original space-time dataset (Section 4.2.3) to obtain their amplitude, and visualize them (Section 4.2.4).

4.2.1. Extraction of Space-Time Wave Fronts

As introduced in Section 3.3, we can extract the wave fronts by means of crease surface extraction. We use the height ridge definition by Eberly [Ebe96] and employ a modified marching cubes algorithm for their extraction, according to Sadlo and Peikert [SP07]. Extracting the wave fronts as crease surfaces in space-time, instead of time series of crease lines in space, makes the extraction more robust by exploiting spatio-temporal coherence and thus suppressing noise.

The extraction of height ridges (and valleys) according to Eberly involves the computation of the Hessian of the original data, which requires second derivatives, and is therefore subject to noise amplification, which typically results in spurious ridges. These issues are usually addressed by filtering. Besides the widely used filtering by value (amplitude), second derivatives (eigenvalues of the Hessian), and component size, we additionally exploit the wave speed, i.e., we reject crease surface parts whose normal angle α deviates too much from the angle associated with the wave speed. Figures 2(a) and 2(b) show an example of the resulting space-time crease surfaces, before and after filtering, respectively.

As discussed in Section 3.3, wave fronts can exhibit deviating propagation speeds due to superposition. For non-periodic fields, this does not represent a severe problem because the wave fronts superimpose, i.e., cross, only at comparably few locations. After rejecting those parts due to deviating α, there will still remain enough ridge parts that represent wave front propagation at wave speed and lead to good results. However, in case of periodic wave phenomena that undergo massive interference, it can happen that almost all wave front regions are subject to interference and hence

only few crease regions exhibit the required α, leading to an insufficient number of feature points. Although the virtual source extraction approach fails in such cases, these cases can be detected by having too few feature points. Such data need to be analyzed with our virtual source field fitting approach, however, at high computational cost.

4.2.2. Extraction of Source Candidates

Using the filtered crease surface parts, we can now search for appropriate virtual source candidates. To this end, the centroid of each triangle of the crease surface parts represents a feature point. Each elementary wave can influence other space-time positions only within the spatio-temporal cone pointing forward in time. Hence, searching for virtual sources that can contribute to a given wave front feature point can be achieved by a Hough transform that detects these cones. This is accomplished by voting with the reverse cone in space-time parameter space. In our examples, we discretize this parameter space using a regular grid with resolution equal to the original dataset.

To obtain the virtual source candidates, we extract the local maxima from the Hough field. We achieve this by extracting critical points of type sink from its gradient field (Figures 2(c), 2(d)). To suppress spurious local maxima, we reject from these candidates those where the largest eigenvalue of the Hessian is not sufficiently negative (given by a user-defined threshold) and subsequently merge the remaining candidates that are closer to any other candidate than a prescribed threshold. While this approach provided good results in our applications, employing topological persistence [ELZ02] could be a worthwhile alternative.

4.2.3. Amplitude Fitting

Having the source candidates identified, we have to derive quantitative information, i.e., how each of them contributes to the extracted wave fronts under consideration. We accomplish this by the least squares fitting from Section 4.1, but this time we fit only the $\tilde{n}$ obtained source candidates instead of all n virtual sources on a grid. Since each of the $\tilde{n}$ source candidates can influence only the data within its forward-time cone, we only need to fit the candidates to the node values of the original grid within these cones, which also reduces m and allows us to visualize substantially larger datasets. After fitting, each source candidate has an amplitude (weight), with which the elementary wave has to be released to fit the data in the vicinity of the filtered crease surface parts, obtained in Section 4.2.2. Similar to Section 4.1, this can be used to reconstruct the field, but only at the grid points that are in the influence region of the elementary waves. The residual between the resulting field and the original can serve for validation of the visualization.

4.2.4. Virtual Source Visualization

The virtual sources obtained so far not only provide a notion about the quantitative propagation of the elementary waves and how they contribute to the extracted space-time ridges, but their amplitude also reflects the importance of a virtual source within the determined set. Virtual sources with low amplitude have lower impact and are therefore of inferior importance. Thus, we visualize the obtained virtual sources with glyphs (such as spheres and crosses) and map the amplitude to both color and radius, to reflect both the value and the importance of each obtained virtual source (Figure 2(d)).

4.3. Comparison

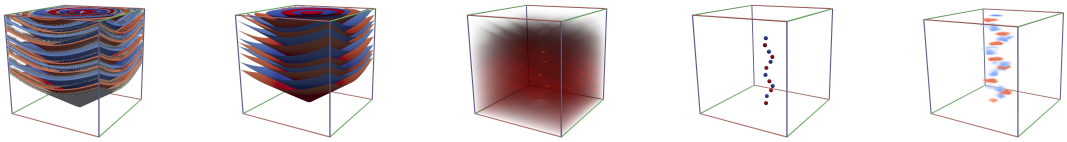
To conclude, we favor the approach of fitting virtual source fields because it is not affected by interference (in contrast to the virtual source extraction) and provides a full representation of the original field (the superposition of the elementary waves reconstructs the full field). Furthermore, the residual field of the least squares method directly provides a measure of fitting error and hence the uncertainty of the representation. On the other hand, inverting Huygens' principle is a hard problem that requires substantial compute power for problems at the real-world scale.

5. Implementation

5.1. Least Squares Fitting

The least squares fitting is demanding both concerning computation time and memory. We investigated a single-CPU-based implementation using the Eigen library [GJ*10], a GPU-based implementation that employs LU decomposition using CULA [HPS*10], and a supercomputer implementation that uses the ScaLAPACK library [BCC*97]. The bottleneck of the GPU implementation is memory. We further experienced noise in the virtual source field when using single precision arithmetic, possibly related to the limited resolution, and we therefore had to employ double precision for all shown results. We used a nVidia GeForce GTX 680 and were able to treat, as our largest example, a $17 \times 17 \times 25$ dataset in less than one minute but with almost using the entire memory of the GPU. On the other hand, our single-CPU implementation took over an hour on a i7 2600 with 3.4 GHz for this dataset.

Our massively parallel implementation was executed on the Cray XC40 "Hornet" at HLRS. The Hornet supercomputer consists of 3944 compute nodes, each having 24 cores of Intel(R) Xeon(R) E5-2680 v3 CPUs with 2.50 GHz and 128 GB memory. In our implementation, we subdivided the $m \times n$ matrix A in blocks of size $m_{\text{sub}} \times n_{\text{sub}}$ and distributed them on $(m/m_{\text{sub}}) \cdot (n/n_{\text{sub}})$ processors. We chose m_{sub} and n_{sub} as divider of m or n, respectively. Although this technique provides a concise and robust visualization, the limited resolution represents a major downside regarding applicability in non-high-performance computing environments. There, however, its results are convincing (Section 6).



(a) (b) (c) (d) (e)

Figure 2: *Crease surfaces (red: ridges, blue: valleys) of the Circulating Source dataset, before (a), and after (b) filtering, and volume rendering of the Hough transform of the ridges (c), using a black body radiation color map. Accumulation points are visible as yellow blobs, and are extracted to obtain the final reconstructed sources (d). The result of the virtual source field fitting for resolution* $17 \times 17 \times 25$ *is shown in (e) for comparison.*

5.2. Crease Surface Extraction

The raw extraction of the space-time crease surfaces is accomplished on the GPU, however, the angle filter and the connected component labeling to filter crease parts by their size is employed on the CPU subsequently.

5.3. Hough Transform

As detailed in Section 4.2.2, we need to convolve all space-time feature points with a reverse space-time cone. We tested both a GPU and a CPU approach for that purpose.

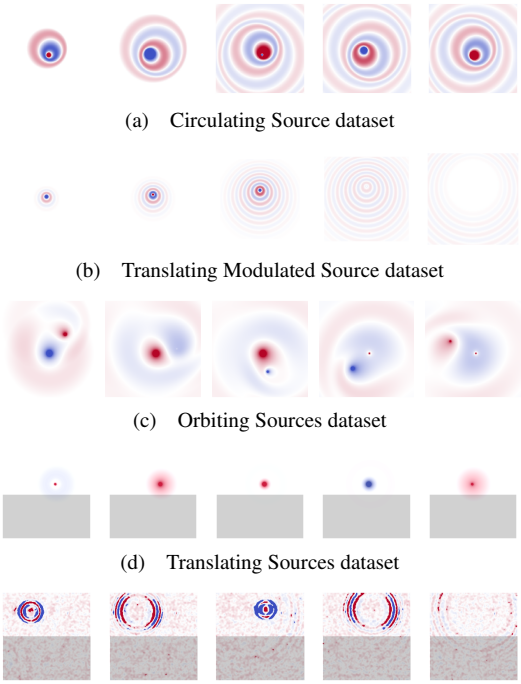
To carry out the Hough transform on the GPU, we process every point in parameter space in parallel. For each parameter point, we evaluate whether it lies within a given neighborhood of the space-time cone extending from a feature point. If it does, the value at the given parameter point is incremented. This operation is carried out for all feature points.

Our CPU implementation of the Hough transform, on the other hand, employs both forward and inverse Fourier transform using the FFTW library [FJ05] to accelerate the convolution. For that, we first need to sample the feature points on a regular grid, which we accomplish by sampling a disc with radius $r_D = v/h$ extending in both spatial dimensions. Here, v is the wave speed and h the spatial resolution of the sampling grid. The disc is centered at the feature point, with value 1 at its center and linearly decaying to zero at the disc boundary. This preprocessing step is carried out on the GPU.

We then employ the convolution theorem by transforming both the sampled discs and the cone to the frequency domain using the Fourier transform, performing a point-wise multiplication there, and transforming the result back to space-time by the inverse Fourier transform, which results in the Hough parameter space. Due to its high memory requirements, this approach is not feasible on the GPU.

6. Results

We exemplify our technique using five 2D time-dependent datasets, of which four are synthetic and one measured with a custom setup. Figure 3 provides an overview. For our synthetic datasets, we assumed the sources to have an extent of half a cell. Within that region, their amplitude is set constant.



(a) Circulating Source dataset

(b) Translating Modulated Source dataset

(c) Orbiting Sources dataset

(d) Translating Sources dataset

(e) Water Basin dataset

Figure 3: *Overview of used datasets.*

Circulating Source (CS) This dataset (Figure 3(a)) consists of a temporally sine-oscillating source that moves uniformly along a circle with constant radius. This data has been sampled on a uniform grid of dimension $150 \times 150 \times 150$. Figure 2 shows the complete source reconstruction process for this dataset.

Translating Modulated Source (TMS) This dataset (Figure 3(b)) is a linearly moving source with sine-modulated

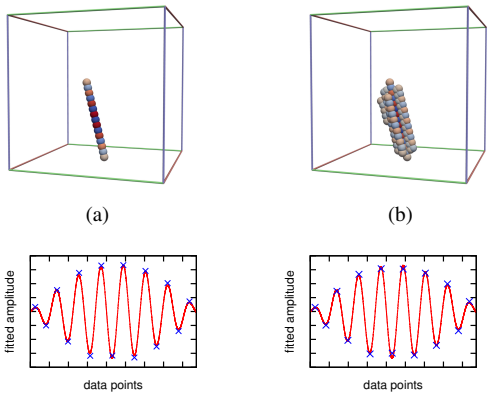


Figure 4: *Extracted virtual source set for the Translating Modulated Source dataset with adapted (a), and with relaxed filtering parameters (b), leading to additional spurious sources (TMSAC), with respective amplitude graphs ((c), (d)) of the original source variation (red), and fitted amplitudes of the central 15 best fits (blue crosses).*

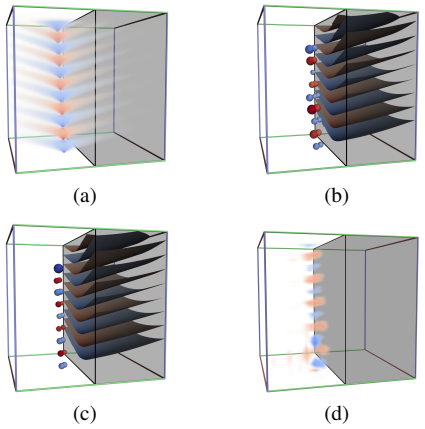


Figure 5: *Translating Source dataset. Volume rendering (a) and results for resolutions* 150^3 *(b) and* 300^3 *(c) with the glyph radii of the virtual sources scaled according to their importance. Figure (d) shows the corresponding virtual source field fitting result with a resolution of* $17 \times 9 \times 24$*. Virtual source clusters and fitting are better for higher resolution. The moving source is located outside of the input sub set (gray) but is still sufficiently well detected.*

amplitude. As can be seen in Figure 4(a), our crease-based technique successfully extracts all 15 peaks of the source. We validated this result by comparing the amplitudes of the extracted virtual sources with the original modulated signal of the translating source, as shown in Figure 4(c).

To examine the robustness of our crease-based approach, we reduced the filtering thresholds in the Hough extrema extraction, leading to additional virtual source candidates (denoted as TMSAC). As can be seen in Figure 4(b), the additional 84 spurious off-axis candidates were assigned low amplitudes because the 15 central sources already represented the original data sufficiently well. We evaluated the impact of the spurious sources by comparing the amplitudes of the 15 central sources with the original signal, which were of inferior quality, but still reasonably well represented the original variation of the source, indicating the low impact of spurious sources (Figure 4(d)).

Orbiting Sources (OS) With our next dataset, we investigate the robustness of our ridge-based virtual source set extraction technique regarding wave front interference. As discussed in Section 3.3, superposition can change the apparent wave front propagation speed and hence pose problems for our ridge-based extraction technique.

To investigate this, we use a dataset with a stationary periodic source near the center and superimpose a periodic source moving along a circle around it. The linear source was not placed exactly at the center of this circle to avoid symmetry effects (Figure 3(c)).

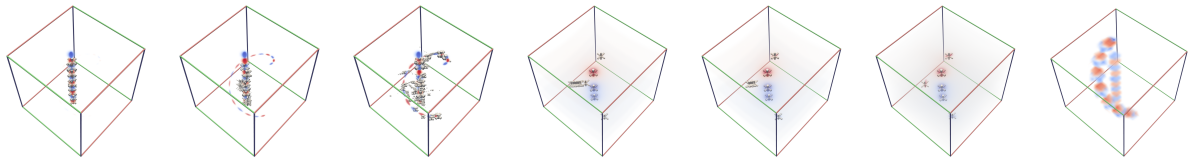
One can see the quality of our extracted extrema deteriorating, from a good fit in Figure 6(a), where the additional source's amplitude is 10% of the original source's amplitude, to a rather insufficient fit in Figure 6(c), where the additional source's amplitude is 50% of the original source's amplitude.

For the 10% superposition, the source candidates still lie where the sources in the original dataset reside, but are a little offset. For 50%, the candidates can only be extracted with substantial noise, while the sources on the circle are too weak to be extracted reliably. This can be attributed to the deformed ridges due to interference, leading to a blurring of the accumulation points in Hough space due to variation of the apparent wave speed, which in turn leads to inferior extrema extraction. Using higher temporal and spatial resolution of the Hough space did not help with this problem. As can be seen in Figures 6(d)–(f), aperiodic signals do not suffer from these issues.

We also use this dataset to demonstrate our virtual source field fitting approach. From Figures 2(e) and 6(g) it becomes clear that both the stationary source and the circularly moving source are captured, however, at the respective cost.

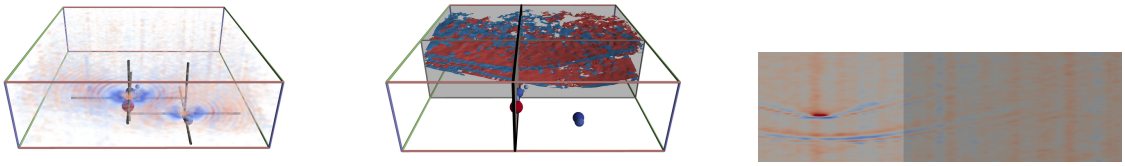
Translating Source (TS) To increase the difficulty for visualization, we now test our technique with a dataset where the source resides outside the domain. This corresponds to typical real-world cases where measurements are physically confined and the desired sources are located somewhere outside the measured domain. To this end, we use a temporally sine-oscillating source that translates at constant speed (see Figures 3(d) and 5(a)).

Our approach was able to reconstruct the original sources



(a) $a = 0.1$ (b) $a = 0.3$ (c) $a = 0.5$ (d) $a = 0.1$ (e) $a = 0.3$ (f) $a = 0.5$ (g)

Figure 6: *Two types of Orbiting Source datasets. Figures (a)–(c) show periodic data, whereas (d)–(f) show six singular events. The stationary source oscillates with $a = 1$ throughout, while we assume different oscillation amplitudes for the moving source. Sources are depicted as crosses, the original field by volume rendering. One can see that our ridge-based approach fails if the orbiting source has low magnitude with periodic sources, but not for discrete (singular) sources. Finally, Figure (g) shows the result of the virtual source field fitting method with $a = 1.0$ for the periodic moving source for comparison, where this method can still accurately extract the sources.*



(a) (b) (c)

Figure 7: *Water Basin dataset. (a) Original dataset with our virtual source result (spheres colored with fitted amplitude), and, for comparison, ground truth space-time positions of the droplet impacts (crosses) determined visually from the video. (b) Extracted ridges in the observed region (gray) of the dataset, together with our virtual sources. (c) A slice of the original space-time dataset, highlighting the observed region (gray). The location of the slice is also shown in (b) with a black outline.*

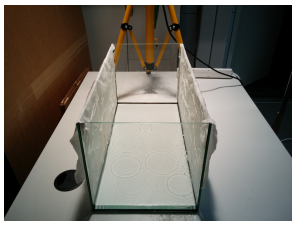


Figure 8: *Experimental setup. Fish tank coated with white paper on a white table, camera above, light source from above (for the acquisition of the dataset we used sun light).*

reasonably well in the form of concentrated clusters, although they were located outside of the available data. Only the topmost source is missing, as its starting time is too late to generate a sufficient amount of space-time creases within the space-time domain. As can be seen in Figures 5(b) and 5(c), increasing the resolution of the dataset results in a better location of the extracted sources and improved weighting.

Water Basin (WB) As a final example to test our approach, we acquired a real-world dataset of waves originating from droplets falling into a basin filled with water (Figure 3(e)). We put some water in a fish tank and coated its walls with white paper, to avoid reflection of light. The tank was placed on a white table surface and a high-speed camera was positioned straight above the surface. Sun light was used to avoid temporal interference with artificial light sources, due to the 50 Hz frequency of alternating current. The whole setup is depicted in Figure 8.

We then created waves by sprinkling water droplets into the tank. The waves produced by the droplets caused caustics which allowed us to capture them in terms of brightness variations on the white table. We captured a background image with a still water surface and employed background subtraction to get rid of uneven illumination and stains. Subsequently, we applied Gaussian smoothing to reduce noise.

Similarly to the Translating Source dataset, we applied our method only to a subset of the dataset in this case. We chose a part where two nearby drops of water created waves shortly after another, which then entered the observed part. Again, using only data that does not contain the original sources, we are able to obtain an estimate in the form of concentrated clusters of virtual sources. Figure 7 shows a volume rendering of the original data, together with our source estimates. The observed part is highlighted in gray. As can be seen in Figure 7(b), the ridges are more uneven and incomplete than in the synthetic datasets, due to noise and and inaccuracies of measurement. Our technique yields two clusters of source candidates representing the two original drops that caused the waves. For comparison, we provide a vol-

ume rendering of the original data and two large crosses that indicate the manually-determined impacts of the drops.

Supercomputer Implementation of Field Fitting Figure 9 shows results for the the virtual source field fitting applied to the Circulating Source dataset that we obtained with our implementation on the Hornet supercomputer using different resolutions for the dataset. Figure 9(a) shows the results for a dataset with resolution $17 \times 17 \times 25$, the largest one that we could also solve with our GPU implementation. The results for higher resolutions show that our field fitting method scales and can accurately localize wave sources.

Time Measurements We conclude our results with a discussion of the performance achieved for the different approaches and for different parameter settings.

For the virtual source field fitting approach, as described in Section 4.1, we performed a small problem size N scaling test, going from $N \approx 1500$ to $N \approx 7300$ data points, yielding execution times ranging from 2 s to 49 s on the GPU (Table 1). For the same problem size range, the CPU implementation executed in 29 s to 3974 s, and was hence only used for validation. For the GPU, both memory consumption and execution time scaled with $\mathcal{O}(N^2)$ (as expected), easily reaching the GPU memory constraint at a $17 \times 17 \times 25$ grid.

Table 2 shows the corresponding timings for the CS dataset on the Hornet supercomputer. As each processor builds only its subblock of A, all other contributions to the runtime expect T_S are negligible and hence not shown. Due to limitations on the available computing time on the Hornet, we could not perform a detailed scaling test. However, even with the limited data available, we can see that our implementation scales approximately linearly with the number of processors for constant problem sizes but the runtime increases more than linearly with the size of A.

For the virtual source extraction as described in Section 4.2, we measured key quantities shown in Table 3. The generation of ridges (Gen.) scales linearly with the number of detected features, which mainly reflects the cost of storing produced geometry during the marching cubes step. For an additional step, filtering the geometry to wave speed, an unoptimized step was executed on the CPU as post process to the marching cubes (Filt.). This step could easily be integrated into the marching cubes algorithm, but was kept separated for validation reasons. The disc generation and padding execution time during the Fourier convolution step is small enough to be negligible ($< 1\%$ of the Fourier step execution time). The Fourier transformation would scale linear with the problem size, however padding the data was necessary to execute the algorithm. This leaves a piecewise constant, linearly increasing scaling property for the Fourier transformation step, while the convolution step execution time is proportional, at roughly 5% of the time.

Table 1: *GPU-based least squares timings for different datasets with the number of features N_F and the number of data points N_P. We compare the time T_M to establish the matrix and T_S to solve for the inverse.*

Dataset	N_F	N_P	T_M [s]	T_S [s]
CS	12	316108	0.434	0.111
TMS	15	507489	0.593	0.222
TMSN	99	792718	2.299	2.841
TS	41	90465	2.379	0.082
OS	234	287028	3.590	2.288
OSH	496	1867424	N/A	N/A
WB	12	58152	0.205	0.025

Table 2: *Timings for the Circulating Source dataset in the massively parallel implementation.*

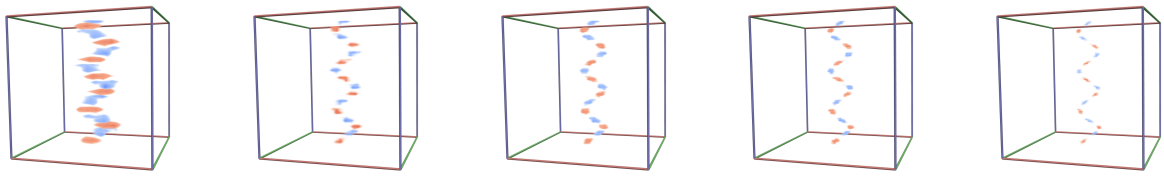
Resolution	N_{proc}	Matrix size	T_S [s]
$17 \times 17 \times 25$	5×2	7225×6936	45
$17 \times 17 \times 25$	5×8	7225×6936	6
$26 \times 26 \times 38$	8×13	25688×25012	174
$32 \times 32 \times 48$	16×16	49152×48128	632
$40 \times 40 \times 60$	16×16	96000×94400	5354
$40 \times 40 \times 60$	32×32	96000×94400	1185
$52 \times 52 \times 76$	52×52	205504×202800	5272

7. Conclusion

We introduced the inversion of Huygens' principle for the visualization of wave dynamics in terms of elementary waves. Because the inversion is computationally costly, we provided two approaches: one solving the full problem on a predefined grid of virtual sources, i.e., a virtual source field, and the other based on explicit extraction of wave fronts and their reconstruction by selected virtual sources. To further limit the computational cost, both approaches require datasets with uniform wave propagation speed, which, however, is the case in many applications in science and engineering.

The former approach requires both high computational power and a sufficient memory. But it is insensitive to arbitrary interference and results in a full representation of the original field. We demonstrated the power of this approach with a supercomputer implementation. The latter approach is an approximation to speed up the computation and allow for larger datasets. This approach suffers from interference and does not provide a full reconstruction—it provides only selected virtual sources at previously extracted wave fronts.

It is clear, that our virtual source field can only be an approximation of the true sources that caused the original field. Further, we observed, at least in some ambiguous configurations, that several virtual sources assume the role of a real source or that a single virtual source aggregates several real sources. Notwithstanding these limitations, our method yields a concise representation and can reveal important fea-



(a) $17 \times 17 \times 25$ (b) $26 \times 26 \times 38$ (c) $32 \times 32 \times 48$ (d) $40 \times 40 \times 60$ (e) $52 \times 52 \times 76$

Figure 9: *Circulating Source dataset in massively parallel implementation.*

Table 3: *Timings (in seconds) for crease surface extraction (generation and filtering), Hough transform and extrema extraction. Triangle count of the input crease surfaces, disc generation (DG), data Padding, Fourier transform, Convolution (Con), inverse Fourier transform for convolved field (IF), unpadding of the result (Un), and extraction both for positive and negative extrema.*

	Gen.	Filt.	Δ-Count	DG	Padding		Fourier		Con	IF	Un.	Extrema	
					Cones	Discs	Cones	Discs				pos.	neg.
CS	0.42	15.3	463559	0.053	0.549	0.514	2.06	2.05	0.567	1.89	0.022	16.92	16.79
TMS	0.96	25.9	619035	0.064	0.571	0.521	2.05	2.06	0.568	1.89	0.020	17.57	17.34
TMSAC	0.95	20.7	619035	0.068	0.578	0.524	2.08	2.08	0.571	1.91	0.020	17.48	17.16
TS	1.25	20.2	576047	0.094	4.292	4.081	40.77	40.84	4.529	39.45	0.191	145.91	146.33
OS	0.44	9.8	252118	0.056	0.551	0.541	2.08	2.09	0.581	1.92	0.026	17.94	17.76
OSH	2.72	45.6	1149897	0.111	4.298	4.135	40.80	40.50	4.509	39.29	0.185	143.83	145.42
WB	0.58	21.6	94932	0.045	4.342	4.152	41.47	41.48	4.617	40.23	0.017	12.78	12.71

tures contained in wave propagation data. A prominent application are of course configurations, where the real sources are located outside of the measured region. As future work, we plan to extend the approach to data with varying wave propagation speed for wider applicability.

Acknowledgements

This work was supported in part by grants from Baden Württemberg Stiftung, the German Research Foundation (DFG) under grant SPP 1648 (ExaScale FSA), the Collaborative Research Center SFB-TRR 75 "Droplet dynamics under extreme ambient conditions", and BMBF HPC III HONK "Industrialisierung von hochauflösender Numerik für komplexe Strömungsvorgänge in hydraulischen Systemen". Stefan Heßel and Filip Sadlo are indebted to the Baden-Württemberg Stiftung for the financial support of this research project by the Eliteprogramme for Postdocs.

References

[BCC*97] BLACKFORD L. S., CHOI J., CLEARY A., D'AZEVEDO E., DEMMEL J., DHILLON I., DONGARRA J., HAMMARLING S., HENRY G., PETITET A., STANLEY K., WALKER D., WHALEY R. C.: *ScaLAPACK Users' Guide*. SIAM, Philadelphia, PA, 1997. 5

[BW03] BORN M., WOLF E.: *Principles of optics*, 7 ed. Cambridge University Press, 2003. 2

[DBM*06] DEINES E., BERTRAM M., MOHRING J., JEGOROVS J., MICHEL F., HAGEN H., NIELSON G. M.: Comparative visualization for wave-based and geometric acoustics. *IEEE Trans. Vis. Comput. Graphics 12*, 5 (2006), 1173–1180. 2

[DH72] DUDA R. O., HART P. E.: Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM 15*, 1 (1972), 11–15. 3

[Ebe96] EBERLY D.: *Ridges in Image and Data Analysis*. Computational Imaging and Vision. Kluwer Academic Publishers, 1996. 2, 4

[ELZ02] EDELSBRUNNER H., LETSCHER D., ZOMORODIAN A.: Topological persistence and simplification. *Discrete comput. Geom. 28* (2002), 511–533. 5

[FJ05] FRIGO M., JOHNSON S. G.: The design and implementation of FFTW3. *Proc. IEEE 93*, 2 (2005), 216–231. 6

[GJ*10] GUENNEBAUD G., JACOB B., ET AL.: Eigen v3. http://eigen.tuxfamily.org, 2010. 5

[HPS*10] HUMPHREY J. R., PRICE D. K., SPAGNOLI K. E., PAOLINI A. L., KELMELIS E. J.: CULA: hybrid GPU accelerated linear algebra routines. In *Modeling and Simulation for Defense Systems and Applications V* (2010). 5

[OMD*12] OBERMAIER H., MOHRING J., DEINES E., HERING-BERTRAM M., HAGEN H.: On mesh-free valley surface extraction with application to low frequency sound simulation. *IEEE Trans. Vis. Comput. Graphics 18*, 2 (2012), 270–282. 2

[SP07] SADLO F., PEIKERT R.: Efficient visualization of Lagrangian coherent structures by filtered AMR ridge extraction. *IEEE Trans. Vis. Comput. Graphics 13*, 6 (2007), 1456–1463. 4

[Tar05] TARANTOLA A.: *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, Philadelphia, PA, 2005. 2

[TGSY95] TIKHONOV A. N., GONCHARSKY A. V., STEPANOV V. V., YAGOLA A. G.: *Numerical methods for the solution of ill-posed problems*. Kluwer/Academic Press, Dordrecht, 1995. 4