Illustrating how hydraulic machinery works

submitted by

Garoe Dorta-Perez

for the degree of Master of Science

of the

University of Bath

Department of Computer Sciences

January 2015

Summary

In this research proposal we present a method for automatic depiction of how it works illustrations of hydraulic machinery.

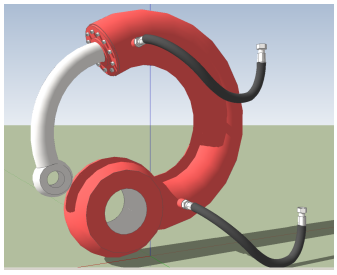
Chapter 1

Introduction

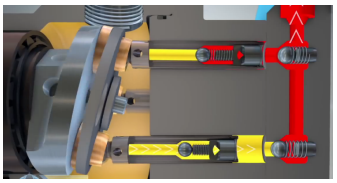
How things work visualizations have been used as an efficient method to explain how a wide range of systems work. This technique usually involves displaying where each part is in relation to the system, showing how force is transmitted from one piece to the next or animating motion. In order to perform this task a range of visual transformations are used. Such as viewing the system from different angles, zoom degrees, transparency levels, as well as displaying only a subset of the parts. Generating material of this sort typically involves manual methods, usually in the form of an expert drawing each illustration by hand or composing a fixed animation using specific software.

Hydraulic machinery is commonly used in our everyday lives. Such as lifting cars with jacks, rams on excavators or gerotors to control fuel intake, as shown in Figure 1-1a. Their popularity is based on their faculty to transmit a force or torque multiplication independently of the distance between input and output. A typical hydraulic equipment has a contained liquid

(a) Cross section of a typical hydraulic cylinder.

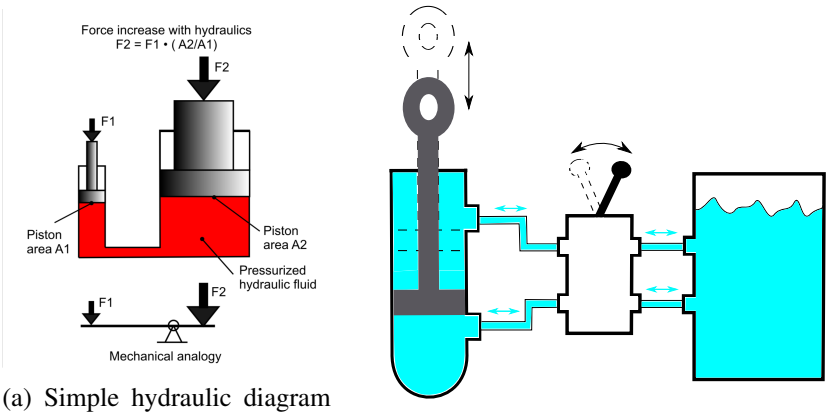


(b) A 3D model of a hydraulic piston.



(c) Frame of manually generated animation of a hydraulic pump.

Figure 1-1: Overview of the intented workflow



(a) Simple hydraulic diagram with a mechanical analogy for easier understanding.

(b) Simplified hydraulic circuit for a hydraulic cylinder.

Figure 1-2: Sample diagrams for hydraulic mechanisms

fluid that becomes pressurised when a force is applied to it, as shown in Figures 1-2a and 1-2b. Then that force is transmitted to the other end of the fluid. In order to grasp how the whole system works, it is essential to understand how the pressure is directed and how it interacts with other parts in the machinery. Therefore, to illustrate the general process, the spatial configuration of each part in the system must be unveiled. As well as the chain of motions that takes place within the gears and the liquid fluids.

In order to to generate *How things work* visualizations, we first need a 3D model of the physical machinery we will be working with, as shown in Figure 1-1b. There are some common visualisation and illustration techniques used for hydraulic machinery, as shown in Figure 1-1c. **Motion arrows** can point out how the solid parts move and they also indicate fluid flow movement. **Frame sequences** display key frames in complex motions and they can highlight temporal interactions between the parts. **Animations** are an useful tool in highly dynamic systems, for example when an excessive number of frame sequences are needed in a particularly complicated motion.

Generating visualisations and illustrations for hydraulic machinery is challenging for designers. They must understand in detail what forces are generated as parts interact among each other and what kind of flow movements are entailed. Furthermore, when 2D illustrations are used, they main disadvantage is the impossibility to change the viewpoint to explore the object from different angle. Moreover, when animations for hydraulic visualizations are generated, they are infrequently updated. Since manual animation is a costly and time consuming task.

There has been some work done on automatically generating illustrations and visualiza-

tions on mechanical assemblies. Nevertheless, it has been restricted to gear to gear interaction only. I.e. solid parts interacting with other solid parts, as gear movement can be inferred using symmetry information. Whereas work on fluid simulation and visualization has not been applied to hydraulic equipment. As simulations is this field are usually designed to, either generate complex visualizations for engineering purposes, or to produce visually plausible but not physically accurate results for animation or games.

This research proposal aims to introduce a method for generating *How things work* illustrations for hydraulic machinery. Simulations would be used to generate explanatory illustrations for hydraulic machinery. This illustrations would help users understanding how this kind of equipment works.

In summary, the main contributions would be:

- An application for creating how things works illustration for hydraulic machinery 3D models.
- A method for detecting motion and interaction of fluid in the model parts.
- Algorithms to automatically generate illustrations with motion arrows and frame sequences

Chapter 2

Previous Work

IMPORTANT: Make sure the technical/theoretical/practical issues of previous work is clearly stated.

Given a 3D CAD model of some hydraulic machinery we want to generate how things work visualizations. Namely, adding arrows depicting the fluid movement.

The problem can be subdivided into:

1. **Part analysis:** Detecting fluid containers and fluid handling parts.
2. **Fluid simulation:** Simulate how the fluid behaves in the previously detected parts.
3. **Fluid visualization:** Display the fluid simulation data in a intuitive format.

Part analysis involves two steps: part segmentation and part information extraction. Usually 3D models consist of meshes or point clouds were there is no clear distinction between each piece. Consequently, any additional analysis and simulation would be overly complicated without further simplifications. Therefore, a segmentation stage is needed in order to divide the model into its constituent components. Part information extraction consists of detecting the piece type, how it moves and interacts with others. So the information saved for type would be axle, gear, reservoir, fluid conduct, etc. In this classification parts are also organized considering fluid interaction criteria, e.g. if the piece interplays or not with the hydraulic fluid. With respect to types of movement, it will be direction of movement, axis of rotation, axis of translation, etc.

Once the parts have been categorized and given an input force, we will have to simulate how the force is transmitted along the different elements. In the special case where a compo-

nent is a container of a fluid or is in direct contact with one, that force will have to be introduced in a fluid simulation algorithm. The output of the simulation will then carry the information along to the next piece.

Lastly, in order to visualize the fluid simulation data we will need to generate a visual cue that will indicate intuitively the fluid movement. Either generating arrows indicating the overall fluid movement, with an animation or showing illustrative key frames.

2.1 Explanatory illustration

Explanatory illustration has to adequately transmit motion on a still image, consequently transforming from the temporal space to the image domain. This is usually found in comics books and instructions sets.

2.1.1 Image and key frames output

Even though comics books are regularly considered inferior forms for presenting information, each scene contains abundant data and they can be understood by children without even reading the letters. [McC93] cited by [Cut02] reviewed visualization and abstraction techniques used in comics books. Such as, how to depict the motion of single objects and to illustrate noises and speeches bound to time. [Cut02] surveyed traditional techniques for depicting motion. Including, sequences of key frames, blur, squeeze and stretch, unbalanced postures and stroboscopic effects. The author introduced criteria to judge the effectiveness of a particular representation. For example, evocativeness, clarity of object, direction of motion and precision of motion.

[ND05] proposed a technique to depict motion in 3D animations. Scene and behaviour descriptions from specialized scene graphs were analysed in order to create the motion cues. In the author's implementation the user has to select a depiction target and assign labels to it, using the graph information motion lines and simplified sketch images are generated. [JR05] presented a similar approach for volumetric data. Speedlines and flow ribbons are combined to generate illustrations encoding 3D rigid movement and flow dynamics. The flow ribbons use comic inspired line abstraction technique to handle occlusion. Researchers have have look into automatic illustration generation for mechanical assemblies [MYY*10]. The author's work is based only on geometry information, extracted from 3D CAD models. Part analysis is performed using slippage analysis, see section 2.2.7, while the motions are extracted from an interaction graph that is defined by edge contact between the components. The system is able

to generate, motion arrows depicting mechanical assembly interaction, as well as generating key frames of periodic motions.

2.1.2 Video output

Video based illustration is applied to enhance the motion information while maintaining the original video format. Therefore, motions are exaggerated or at the very least some hints to reinforce movement perception are added.

[CRH05] presented a method to embed cartoon-style motion cues in video. Their algorithm has two stages: a motion tracking step, where a feature manually marked will be tracked; and a computer graphics stages, where a cue will be generated and added to the video. The system can add motion lines as well as squash and stretch objects. However, the second case is restricted to videos where the background can be reconstructed from previous frames. Furthermore, [Low14] showed that even though animations have become a generalized tool for visualizing dynamic systems, special care have to be taken as users can fail to extract the necessary information due to the nature of the animation.

2.2 Part analysis

Libraries of 3D mesh models are quite common [Tri14], [Gra14], [Aut14], in addition they contain vast quantities of models. Usually the models are also represented using points clouds, which can be obtained easily from the meshes. However, clouds and mesh models lack middle and high level information such as: symmetry, parallelism or part segmentation. Therefore we must extract this information from either representation. Since solving for general shapes is quite challenging, a number of constrained approaches have been proposed. For a more in depth knowledge there are a number of surveys on the topic [VMC97], [APP*07] and [Sha08].

2.2.1 Region growing

Region growing algorithms start with a seed element for the sub-mesh and then perform a local greedy element addition to the current cluster. [MDKK06] proposed a mesh segmentation technique based on curvature estimation with sharp edge recognition. The author's method performs well on mechanical objects and it is also robust to noise. However, oversegmentation occurs in models with soft edges.

2.2.2 Iterative clustering

Iterative clustering is based on the k-means clustering algorithm. Representatives for each cluster are chosen and then using some metric the elements are assigned to the best cluster and the representatives are recalculated. [LZHM06] presented a cluster based segmentation algorithm. The method performs better with hierarchical models composed of regions at multiple scales. Therefore, it gives the same results regardless of model scale or the coarseness of the mesh. A quadratic surface fitting algorithm was presented by [YWLY12]. Which combines several distance metrics to perform the quadratic minimization.

2.2.3 Random walk

With the random walks approach, n faces are chosen as seeds, with n been the number of desired segments. Then for each unlabelled face, the probabilities that a random walker will reach each labelled face are calculated. Finally, each face is assigned to the highest probability label. [Gra06] proposed a first random walks implementation constrained into the imaged domain. Thereafter, [LHMR09] extended it to for 3D mesh models, the author's technique also included an automatic seeding mode were seeds quantity and placement is based on a spring-like energy function. Seeds placement can greatly affect the final segmentation. Nevertheless, the authors tackle the issue with a post processing stage were the clusters are merged if oversegmentation occurred and the edges are smoothed.

2.2.4 Data driven

[GF09] also demonstrated a clustering technique with special emphasis on edge and face consistency. The algorithm is based on a graph construction which encodes triangle neighbour information. The method then performs the clustering on the graph space. The main advantage is consistent segmentation of similar models.

2.2.5 Global energy function

[dCdLL14] presented an segmentation technique based on calculating a global discrepancy function, based on the Lambert illumination model, for each triangle in the mesh. Then directly using that value to segment the model in several parts.

2.2.6 Hybrid

[WY11] approach for mesh segmentation functions with a combination of curvature estimation, Gauss mapping and B-spline surface fitting.

2.2.7 Slippage analysis

However all the previous methods cannot be directly applied to segment mechanical parts. [GG04] proposed a method for segmenting mechanical objects based on local slippage. This is quite useful as it gives for each part its degrees of freedom, e.g. sphere (3 rotations), cylinder (1 rotation and 1 translation), plane (1 rotation and 2 translations), etc. [YLT*14] improved [GG04] method making it more robust to noise and giving extra primitive information(e.g. normal of a plane, center of a sphere, etc).

2.3 Fluid simulation

The current paradigm in fluid simulation consist of solving the Navier-Stokes equations of fluid dynamics, shown in Equations 2.1 and 2.1.

$$\nabla \cdot \mathbf{u} = 0 \tag{2.1}$$

$$\mathbf{u}_t = -(\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla \cdot (v\nabla\mathbf{u} - \frac{1}{\rho}\nabla p + \mathbf{f}) \tag{2.2}$$

The first equation enforces mass conservation and ensures incompressibility, while the second encodes momentum conservation it is derived from Newton's Second Law. Where $\mathbf{u}_t$ is the time derivative vector field of the fluid velocity, p is the scalar pressure field, ρ is the density of the fluid, v is the kinematic viscosity and $\mathbf{f}$ represents the body force per unit mass, usually gravity.

Along the years several methods have been proposed, although all of them are based on the same Navier-Stokes equations.

- **Fourier Transform** techniques uses several superimposed sinusoidal waves to model the fluid behaviour.
- **Grid** based methods track the fluid features at fixed points in space.
- **Smooth Particle Hydrodynamics** track a large number of particle in the fluid.
- **Hybrid** algorithms are combinations of the aforementioned methods.

Regardless of the chosen method, to update the particles in the for the next frame must be computed. However, it is common for the algorithms to have a upper time step, if the update

is computed after the limit there is no guaranty that the output would be reasonable.

Another important concept in fluid simulation is whether the simulation implements interplay within the fluid and rigid bodies that come in contact with it (solid-fluid coupling). And how flexible is this interaction, one way solid fluid(e.g. a rock drops on a small pool of water, so the water moves but the rock is almost not affected by the and moves it, but the fluid does not affect the rock), or fluid solid(e.g. a small buoy floating in the ocean has little effect on the surrounding water), and two way solid fluid interaction(e.g. a flexible object drops into a pool of fluid).

For more information on real time fluid simulations see [VLM12] survey. While for survey specific to SPH fluid simulation see [IOS*14].

2.3.1 Fourier Transform

When simulating fluid with periodic boundary conditions, procedural simulation can be applied with a low computational cost. Usually this method is used with large masses of water (ocean simulation), as they are an approximation to periodic boundary conditions fluids. The waves can be modelled as a superimposition of sinusoidal waves, which can be efficiently decomposed using Fast Fourier Transform methods [MWM87]. Waves produced in this fashion are visually plausible but physical accuracy is not enforced inherently by the model.

Each wave has a wave number k and a wave vector $\mathbf{k}$. [Tes01] presented the basic technique in this area. The authors modelled the ocean surface as a summation of complex sinusoids with different wave vectors. Visually pleasing results were achieved generating random Fourier amplitudes.

The previous method was further improved by [CGG03], with the addition of solid (ships, coast line, etc). The model developed more complex wave shape and motion equations, with differentiated equations in open sea and near the shore. And by [CC06] with adaptive surface tessellation. Furthermore, the method was implemented on GPUs, and also included shallow water effects and spray dynamics.

2.3.2 Grid

Grid methods were among the first techniques to solve fluid simulation and so the firsts ones implement intuitive solutions. Grid based methods solve Navier-Stokes Equations 2.1 and 2.2, in a fixed position in space (i.e. grid). As fluid flows, the equations are solved in each position,

giving a value for speed and pressure at each time step. This approach is advantageous as many numerical methods can be applied easily on grids and with they are more easily adaptable to GPUs implementations than other methods.

One of the firsts papers in this area introduced a Grid method [FM96] to solve Navier-Stokes equations by applying forward Euler time integration. However, their solid fluid coupling was one way only, the solids do not affect the fluid behaviour. [Sta99] extended this method in order to overcome stability issues. The author's implemented a stable solver which allowed for larger time steps, trough an operator splitting scheme and more importantly a semi-Lagrangian method to calculate the advection term.

[CMT04] proposed solid-fluid coupling algorithm for grids models using distributed Lagrange multipliers. Nevertheless, the author's method uses a non adaptive grid and multiphase fluids and bubbles are not covered by it.

2.3.3 Smooth Particle Hydrodynamics

Smooth Particle Hydrodynamics(SPH) has been established as one the major breakthroughs for fluid simulations in computer graphics, positioning itself as the most popular method nowadays. Compared to grid based methods that need high resolution grids to produce foam and splashes, SPH can achieve the same with less computational load. Two extra advantages are that mass conservation is automatically satisfied if the amount and mass of the particles is keep constant; and as the particles move with the fluid, the advection term does not need to be calculated explicitly.

SPH were first introduced by [Des96], where each particle encodes its position, velocity, mass and the forces that act on it.

While, [MST*04] presented a method for computing solid fluid coupling. Where boundary particles are created on the surface of the solid and then particle coupling is calculated. [AIA12] further improved the previous method with the inclusion of irregular particle distributions as well as friction and dragging. [SZMTW14] also solved stability issues in the previous SPH solid-fluid coupling techniques, using a correction scheme. Furthermore their algorithm was implemented on GPUs, thus achieving better performance than previous methods.

2.3.4 Hybrid

Hybrid methods are combinations of the aforementioned approaches. As with other hybrid techniques, the objective is to capitalize the advantages of each and discard the weaknesses.

A common hybrid approach is the particle-in-cell (PIC) method, [Har62]. It solves the advection using SPH and the incompressibility with an Euler grid. In more detail, the particles are averaged onto a grid where all the terms in the Navier-Stokes Equations 2.1 and 2.2 are calculated, except advection. Next, the particles are translated using the grid velocity. This method has been used in a variety of situations, for example [ZB05] used it for sand simulation, while [HG09] explored fire simulations. A further improvement is an implicit approach, namely the fluid implicit particle (FLIP) [JH86]. Where the particles are not updated directly with grid data. The main strength of FLIP is that the advection term calculation is always stable. However, the particles can sometimes clamp or have voids. FLIP was extended by [RWT11] to be able to solve the equations in a coarse grid.

2.4 Flow visualization

Extensive work has been done in this area as visualizing fluid movement has a broad range of applications. However, this is a challenging task as it has to effectively display complex and copious amounts of data. Seeing that fluid simulation is generally solved by means of highly divided grids or with large number of particles, as explained in section 2.3.

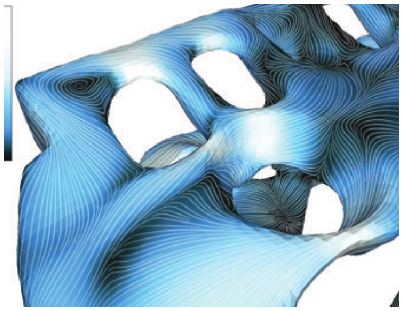


Figure 2-1: Streamlines on a 3D surface [SLCZ09].

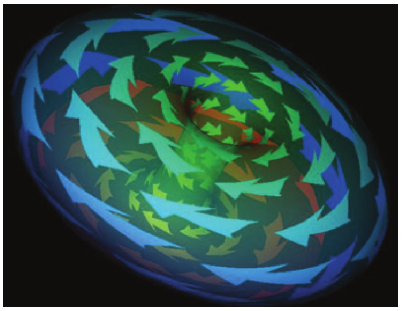


Figure 2-2: Arrows placed on streamlines paths in a 3D surface [L̈98].

Streamlines are convenient tools to describe and visualize flow, as shown in figure 2-1. A streamline is defined as a curve that is everywhere tangent to flow field, i.e. it is parallel to the local velocity vector. Therefore, they provide an intuitive mechanism to show the fluid travel direction. Furthermore, they have properties such as: streamlines will not cross each other

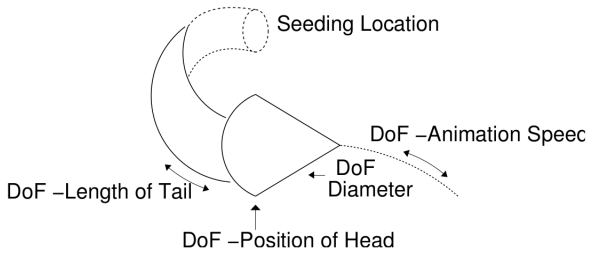


Figure 2-3: Streamcomet depiction with its degrees of freedom outlined [LH05].

(flow will not go across them), or a particle in the fluid starting on one streamline will not leave it. Once the streamlines has been calculated, instead of displaying them as such, they can be replaced by arrows arranged using some criteria. For instance, on the path of each streamline or after clustering some streamlines together, as shown in Figure 2-2.

In the following sections we focus only on steady flow visualization. For more information on the subject we refer the readers to [MLP*10] survey.

2.4.1 2D visualization

On the 2D image domain, an image-guided algorithm for visualizing 2D flow in images was proposed by [TB96]. Which [LHS08] improved, by only generating the fewest possible number of streamlines.

2.4.2 Streamlines on surfaces

Seeding techniques for curves on 3D surfaces were explored by [WST09], who developed a technique to combine model reduction with grid based methods. And by [SLCZ09], whose method generates streamlines only for visible parts of the surface, thus providing a significant gain in efficiency.

2.4.3 Streamline rendering and placement

Rendering too many streamlines can result in clutter and too few can lead to omit important characteristics. There has been plenty research into optimizing streamline production and placement. However for our hydraulic machinery the issue is less critical than in other areas.

[LH05] presented the streamcomet and a fast animating technique. A streamcomet, as shown in Figure 2-3, is an extension of a streamline, they move along the path of a stream line, with adjustable head position, length and translucency of tail, and variable animation speed. While the fast animating technique applies a stipple pattern to the streamline path, so

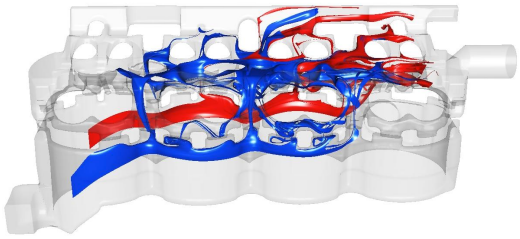


Figure 2-4: Streamsurface illustrating fuel flow in an engine [LGD*05].

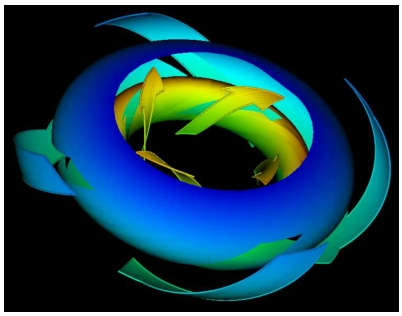


Figure 2-5: Streamarrows in an occluded flow situation [LMG97].

the pattern is shifted at rendering time to add animation. [RPP*09] proposed a dual seeding technique for streamline placement. Dual streamlines that are orthogonal to the fluid flow are also calculated. However they are not rendered but rather used to better decide which and how the primal streamlines are going to be drawn. Considering the occlusion problems that normally arise in this area, [MCHM10] tackle the streamline rendering from a view dependant perspective. The authors precompute a random pool of streamlines. Then, occluded areas are pruned and empty areas are reseeded for the current view via occupancy buffers. What is more, a GPU implementation of their method is provided. Clustering is a common technique for reducing the amount of rendered streamlines. To address slow euclidean distance tests in clustering, [MJL*13] presented a signature based metric. The streamline attributes for the signature are curvature, torsion and tortuosity. The authors presented two variations, a fast simpler method and a hierarchical one that scarifies speed for superior results. However, both provide a performance increase over euclidean distance calculations.

2.4.4 Surface based integral objects

Streamlines can be extended dimensionally in order to work with surfaces instead of lines, an example of a streamsurface is shown in Figure 2-4. Evidently, following this path leads to more complex methods as a new dimension in added. Nevertheless, surfaces can provide better visualization results. For example, the use of shading gives an improved depth insight.

Streamsurfaces were introduced by [Hul92]. The author's method seeds streamlines from a curve and those streamlines are advanced through the vector field. New streamlines are seeded in a the points reach a certain distance. They are also terminated if another streamline is too close. The streamlines points are used to create the streamsurface mesh. To aid in the interpretation of flow, [LMG97] extended streamsurface model with the addition of streamarrows, as shown in Figure 2-2. The author's technique involves adding arrows to show flow direction,

and removing the arrow space on the surface to help with occlusion. A hybrid method with streamlines and streamtubes was proposed by [ZDL03]. Depicting linear anisotropy regions with streamtube and planar anisotropy ones with streamsurfaces. The colors used for rendering encode additional anisotropy information. More recently, [BW10] presented extensions in streamsurface rendering to address occlusion issues, as well as a GPU implementation of their technique. Contour lines, halftoning, silhouetts, contour arrows, cuts and slabs are applied to a raw streamsurface in order to be able to depict occluded areas.

Chapter 3

Proposed methodology

How are we actually going to solve the problem. What is the proposed approach? Here we talk about what we are going to use. Still not clear what are the actual contributions of the proposal.

3.1 Part detection

How are we going to detect the parts. Heuristics to detect were the liquid is? The slippage thing, with manual input when needed. I.e. to specify pump input or to say where the liquid is.

3.2 Fluid simulation

Using SPH, Grid or Hybrid simulation. Looks like grid since we are constrained to a small area and we do care about foams or other extra realism stuff.

3.3 Flow visualization

We are going to use stream lines on surfaces and then replace them by arrows.

3.4 Fixed parts visualization

Rip off gears papers

3.5 Animation

Simulation data is already the animation

3.6 Key frames visualization

sample the animation For uniform translation only one in between, sample the animation at critical points

3.7 Time line

Table with a time line, diagram?

Chapter 4

Evaluation

we plan to test it on a variety of models, other way is comparison with previous work, user study -¿ like doing and study of how well they understood how the machinery works, try think about more ideas to test the approach, comparison of frame sequence, animation and arrows, which one does the user think is more useful

Chapter 5

Applications

generating visualizations for museums, schools, education in general

Chapter 6

Conclusions

Bibliography

[AIA12] Akinci N., Ihmsen M., Akinci G.: Versatile rigid-fluid coupling for incompressible SPH. *ACM Transactions on Graphics* (2012).

[APP*07] Agathos A., Pratikakis I., Perantonis S., Sapidis N., Azariadis P.: 3D Mesh Segmentation Methodologies for CAD applications. *Computer-Aided Design and Applications 4*, 6 (Jan. 2007), 827–841.

[Aut14] Autodesk: Autodesk 123D, 2014. URL: `http://www.123dapp.com/3d-model-library`.

[BW10] Born S., Wiebel A.: Illustrative stream surfaces. *Visualization and Computer Graphics, IEEE Transactions on 16*, 6 (2010), 1329–1338.

[CC06] Chiu Y.-f., Chang C.-f.: GPU-based Ocean Rendering. *2006 IEEE International Conference on Multimedia and Expo* (July 2006), 2125–2128.

[CGG03] Cieutat J., Gonzato J., Guitton P.: A general ocean waves model for ship design. *Virtual Concept 4* (2003), 5–7.

[CMT04] Carlson M., Mucha P. J., Turk G.: Rigid fluid: animating the interplay between rigid bodies and fluid. *ACM Transactions on Graphics (TOG)* (2004), 377–384.

[CRH05] Collomosse J., Rowntree D., Hall P.: Rendering cartoon-style motion cues in post-production video. *Graphical Models 67*, 6 (Nov. 2005), 549–564.

[Cut02] Cutting J. E.: Representing motion in a static image: constraints and parallels in art, science, and popular culture. *Perception 31*, 10 (2002), 1165–1194.

[dCdLL14] de Castro P. M. M. a., de Lima L. A. P., Lucena F. L. A.: Invariances of single curved manifolds applied to mesh segmentation. *Computers & Graphics 38* (Feb. 2014), 399–409.

[Des96] Desbrun, Mathieu and Gascuel M.-P.: *Smoothed particles: A new paradigm for animating highly deformable bodies*. Springer, 1996.

[FM96] Foster N., Metaxas D.: Realistic Animation of Liquids. *Graphical Models and Image Processing 58*, 5 (Sept. 1996), 471–483.

[GF09] Golovinskiy A., Funkhouser T.: Consistent segmentation of 3D models. *Computers & Graphics 33*, 3 (June 2009), 262–269.

[GG04] Gelfand N., Guibas L. J.: Shape segmentation using local slippage analysis. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing - SGP '04* (New York, NY, USA, 2004), ACM Press, pp. 214–223.

[Gra06] Grady L.: Random walks for image segmentation. *IEEE transactions on pattern analysis and machine intelligence 28*, 11 (Nov. 2006), 1768–1783.

[Gra14] GrabCAD: GrabCAD, 2014. URL: `http://grabcad.com/library`.

[Har62] Harlow F. H.: *The particle-in-cell method for numerical solution of problems in fluid dynamics*. Tech. rep., Los Alamos Scientific Lab., N. Mex., 1962.

[HG09] Horvath C., Geiger W.: Directable, high-resolution simulation of fire on the GPU. *ACM Transactions on Graphics 28*, 3 (July 2009), 41:1–41:8.

[Hul92] Hultquist J. P.: Constructing stream surfaces in steady 3D vector fields. In *Proceedings of the 3rd conference on Visualization'92* (1992), IEEE Computer Society Press, pp. 171–178.

[IOS*14] Ihmsen M., Orthmann J., Solenthaler B., Kolb A., Teschner M.: SPH fluids in computer graphics. In *Eurographics 2014-State of the Art Reports* (2014), no. 2, The Eurographics Association, pp. 21–42.

[JH86] J.U. Brackbill, H.M. Ruppel: FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. *Journal of Computational Physics 65*, 2 (1986), 314–343.

[JR05] Joshi a., Rheingans P.: Illustration-inspired techniques for visualizing time-varying data. In *Visualization, 2005. VIS 05. IEEE* (2005), Ieee, pp. 679–686.

[LÖ98] Löffelmann H.: *Visualizing Local Properties and Characteristic Structures of Dynamical Systems*. PhD thesis, Technical University of Vienna, 1998.

[LGD*05] Laramee R. S., Garth C., Doleisch H., Schneider J., Hauser H., Hagen H.: Visual analysis and exploration of fluid flow in a cooling jacket. In *Visualization, 2005. VIS 05. IEEE* (2005), IEEE, pp. 623–630.

[LH05] Laramee R., Hauser H.: Geometric flow visualization techniques for CFD simulation data. In *Proceedings of the 21st spring conference on Computer graphics* (2005), vol. 1, pp. 221–224.

[LHMR09] Lai Y.-K., Hu S.-M., Martin R. R., Rosin P. L.: Rapid and effective segmentation of 3D models using random walks. *Computer Aided Geometric Design 26*, 6 (Aug. 2009), 665–679.

[LHS08] Li L., Hsieh H.-H., Shen H.-W.: Illustrative Streamline Placement and Visualization. *2008 IEEE Pacific Visualization Symposium* (Mar. 2008), 79–86.

[LMG97] Löffelmann H., Mroz L., Gröller E.: *Hierarchical streamarrows for the visualization of dynamical systems*. Springer, 1997. URL: `http://link.springer.com/chapter/10.1007/978-3-7091-6876-9\_14`.

[Low14] Lowe R. K.: Dynamic Visualizations: A Two-Edged Sword? In *Handbook of Human Centric Visualization*, Huang W., (Ed.). Springer New York, New York, NY, 2014, pp. 581–604.

[LZHM06] Lai Y.-K., Zhou Q.-Y., Hu S.-M., Martin R. R.: Feature sensitive mesh segmentation. In *Proceedings of the 2006 ACM symposium on Solid and physical modeling - SPM '06* (New York, NY, USA, 2006), ACM, pp. 17–25.

[McC93] McCloud S.: *Understanding Comics*. 1993.

[MCHM10] Marchesin S., Chen C.-K., Ho C., Ma K.-L.: View-dependent streamlines for 3D vector fields. *IEEE transactions on visualization and computer graphics 16*, 6 (2010), 1578–86.

[MDKK06] Mizoguchi T., Date H., Kanai S., Kishinami T.: Segmentation of Scanned Mesh into Analytic Surfaces Based on Robust Curvature Estimation and Region Growing. In *Geometric Modeling and Processing-GMP 2006* (2006), Springer, pp. 644–654.

[MJL*13] McLoughlin T., Jones M. W., Laramee R. S., Malki R., Masters I., Hansen C. D.: Similarity measures for enhancing interactive streamline seeding. *IEEE transactions on visualization and computer graphics 19*, 8 (Aug. 2013), 1342–1353.

[MLP*10] McLoughlin T., Laramee R. S., Peikert R., Post F. H., Chen M.: Over Two Decades of Integration-Based, Geometric Flow Visualization. *Computer Graphics Forum 29*, 6 (Sept. 2010), 1807–1829.

[MST*04] Müller M., Schirm S., Teschner M., Heidelberger B., Gross M.: Interaction of fluids with deformable solids. *Computer Animation and Virtual Worlds 15*, 34 (July 2004), 159–171.

[MWM87] Mastin G. A., Watterberg P. A., Mareda J. F.: Fourier synthesis of ocean scenes. *Computer Graphics and ... 7*, 3 (1987), 16–23.

[MYY*10] Mitra N. J., Yang Y.-L., Yan D.-M., Li W., Agrawala M.: Illustrating How Mechanical Assemblies Work. *ACM Trans. Graph. 29*, 4 (July 2010), 58:1–58:12.

[ND05] Nienhaus M., Dollner J.: Depicting dynamics using principles of visual art and narrations. *Computer Graphics and Applications, IEEE 25*, June (2005), 40–51.

[RPP*09] Rosanwo O., Petz C., Prohaska S., Hege H.-C., Hotz I.: Dual streamline seeding. In *2009 IEEE Pacific Visualization Symposium* (Apr. 2009), Ieee, pp. 9–16.

[RWT11] Raveendran K., Wojtan C., Turk G.: Hybrid smoothed particle hydrodynamics. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics symposium on computer animation* (2011), pp. 33–42.

[Sha08] Shamir A.: A Survey on Mesh Segmentation Techniques. *Computer Graphics Forum 27*, 6 (2008), 1539–1556.

[SLCZ09] Spencer B., Laramee R. S., Chen G., Zhang E.: Evenly Spaced Streamlines for Surfaces: An Image-Based Approach. *Computer Graphics Forum 28*, 6 (Sept. 2009), 1618–1631.

[Sta99] Stam J.: Stable fluids. *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), 121–128.

[SZMTW14] Shao X., Zhou Z., Magnenat-Thalmann N., Wu W.: Stable and Fast Fluid-Solid Coupling for Incompressible SPH. In *Computer Graphics Forum* (Oct. 2014), vol. 00.

[TB96] Turk G., Banks D.: Image-guided streamline placement. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), pp. 453–460.

[Tes01] Tessendorf J.: Simulating ocean water. *Simulating Nature: Realistic and Interactive ... 1* (2001). URL: `http://www-evasion.imag.fr/people/Fabrice.Neyret/images/fluids-nuages/waves/Jonathan/articlesCG/simulating-ocean-water-01.pdf`.

[Tri14] TRIMBLE NAVIGATION LIMITED: 3D Warehouse, 2014. URL: https://3dwarehouse.sketchup.com/.

[VLM12] VINES M., LEE W.-S., MAVRIPLIS C.: Computer animation challenges for computational fluid dynamics. *International Journal of Computational Fluid Dynamics 26*, 6-8 (July 2012), 407–434.

[VMC97] VÁRADY T., MARTIN R. R., COX J.: Reverse engineering of geometric modelsan introduction. *Computer-Aided Design 29*, 4 (Apr. 1997), 255–268.

[WST09] WICKE M., STANTON M., TREUILLE A.: Modular bases for fluid dynamics. *ACM Transactions on Graphics 28*, 3 (July 2009), 1.

[WY11] WANG J., YU Z.: Surface feature based mesh segmentation. *Computers & Graphics 35*, 3 (June 2011), 661–667.

[YLT*14] YI B., LIU Z., TAN J., CHENG F., DUAN G., LIU L.: Shape recognition of CAD models via iterative slippage analysis. *Computer-Aided Design 55*, 0 (Oct. 2014), 13–25.

[YWLY12] YAN D.-M., WANG W., LIU Y., YANG Z.: Variational mesh segmentation via quadric surface fitting. *Computer-Aided Design 44*, 11 (Nov. 2012), 1072–1082.

[ZB05] ZHU Y., BRIDSON R.: Animating sand as a fluid. *ACM Trans. Graph. 24*, 3 (2005), 965–972.

[ZDL03] ZHANG S., DEMIRALP C., LAIDLAW D. H.: Visualizing diffusion tensor MR images using streamtubes and streamsurfaces. *Visualization and Computer Graphics, IEEE Transactions on 9*, 4 (Oct. 2003), 454–462. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1260740.