Predicting model deformations for predictive force feedback in haptic bilateral teleoperation applications

Towards complex interactions in haptic bilateral teleoperation

Stefan Stoicescu

Supervisor(s): Ranga Rao Venkatesha Prasad, Kees Kroep

[1]**EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 23, 2024

Name of the student: Stefan Stoicescu
Final project course: CSE3000 Research Project
Thesis committee: Ranga Rao Venkatesha Prasad, Kees Kroep, Qun Song

Abstract

In this paper, we investigate the use of heightmap-based models to predict deformations in granular materials during teleoperation tasks. Our primary objective is to provide real-time cutting feedback and contact predictions for haptic bilateral teleoperation applications, thereby enhancing operator control in remote manipulation scenarios. We developed a simulation model that utilizes heightmaps to represent deformable materials and implemented a cutting algorithm to simulate complex interactions with granular soils.

Through a series of experiments, we demonstrated that higher resolution heightmaps significantly improve the accuracy and stability of cutting feedback. Our findings suggest that this approach is a viable method for keeping track of changes in a deformable object's shape and can be further expanded to include a wider range of teleoperation tasks, potentially improving the safety and effectiveness of remote operations in hazardous environments.

1 Introduction

The ability to manipulate objects at a distance has always seemed "science fiction". It is not difficult to imagine how being able to interact with people and objects in an entirely different location in real-time, via a robotic proxy, could be advantageous. Such an ability could revolutionize fields in which human life is at stake, allowing workers to remotely operate machinery and interact with hazardous environments. Teleoperation, a multidisciplinary field encompassing robotics, psychology, human-computer interaction, and material sciences, addresses this need by enabling human operators to control remote manipulation devices in real-time [1]. This capability could be crucial for scenarios such as space and deep-sea exploration, urban search and rescue missions, or even nuclear plant maintenance [2] [3].

Imagine a scenario where an earthquake has caused a building to collapse, trapping individuals under rubble. Traditional rescue efforts would require sending human responders into potentially dangerous environments, risking their lives in the process. With teleoperation, a highly skilled operator could control a rescue robot from a safe distance, cutting and removing debris in order to locate survivors without exposing anyone to immediate physical danger. This technology not only enhances safety, but also allows robots to harness the wealth of pre-existing human skills, without the need to rely on complex AI-based decision-making.

However, implementing such a system involves solving several challenges. One of the most pressing issues is how to transmit real-time, intuitive feedback back to the human operator. It is easy to imagine that without any real-time knowledge of the remote environment, any action taken by an operator would be reduced to sophisticated guess-work. While the easiest solution to this problem would be to provide a direct video and haptic link to the remote station, the effectiveness of such a setup would be highly dependent on outside factors, such as network delays and packet loss. For example, consider a scenario where an operator is controlling a robot in order to shovel a pile of sand. Any delays, or inaccuracies in force feedback might result in the human missing the pile of sand, misjudging the amount that is picked up, or even misplacing the soil due to erroneous movements.

To address these challenges, Model-Mediated Teleoperation (MMT) proposes the use of local models, which estimate the remote environment, and provide users with near-instantaneous feedback, regardless of any network issues that might arise between the local and remote stations [4]. Such models mostly concern themselves with predicting haptic interactions, as delivering convincing predictive visual feedback remains more complex.

While current research within the field of MMT has made great strides in providing general models of interaction in environments with limited dimensionality or good domain knowledge, models tailored for 3D spaces filled with deformable objects, have proven to be more challenging, due to the innate difficulty of balancing accuracy, and runtime speed [4]. Additionally, while research on haptics for more general models of deformable objects exists [5], papers focused on more specific material types such as granular soils are scarce.

In the real world, robots making use of this technology would have to interact with many discrete material types. As a result, it could be argued that attempting to predict deformation for a certain type of surface, via a tailor-made specialized model, might lead to more compelling results when compared to traditional methods. Therefore in this paper, we propose to answer the following research question:

- **"How can we predict model deformation for cutting interactions in haptic bilateral teleoperation applications?"**

To explore this research question, we need to examine several subquestions:

- "How can deformable materials be represented within a physics simulation?"
- "How can cutting an object be simulated within a physics engine?"
- "How can the physics engine keep track of changes in an object's shape?"
- "Which metrics can be used to approximate feedback for cutting?"

This paper investigates a heightmap-based displacement model as a basis for a cutting soils within an interactive physics simulation. The model aims to enhance teleoperation systems by providing consistent predictions of material shape, and cutting feedback. The paper is structured as follows: We begin with a review of related literature in Section 2, followed by a detailed explanation of the research methodology in Section 3. Next we present the specifics of our model in Section 4, followed up by our experimental setup and results (Section 5), responsible research (Section 6), discussion and future work (Section 7) and conclude with a summary of our findings (Section 8).

2 Related literature

While literature detailing deformable soil interactions and environment modelling for Model-Mediated Teleoperation tasks is scarce, within the realm of Computer Graphics, real-time simulations of sand and other granular media have become relatively commonplace. To this end, the literature that was considered for this paper details several approaches that could be used as a foundation for building a model capable of estimating the feedback generated from cutting deformable objects.

Li and Moshell [6] discuss a 2D model for soil slippage and manipulation. At its core, the approach divides the surface of the terrain into discrete "columns" of sand, and then shifts material from one column to another, based on material properties and Newtonian physics. The paper's approach is well suited to height-fields.

Benes, Dorjgotov, Arns, *et al.* [7] present a haptic and visual simulation of sand, in order to facilitate interactive 3D modelling. The model described in the paper provides haptic feedback for a spherical proxy object, and simulates sand displacement and erosion. The project makes use of height-fields for keeping track of alterations in the surface of the sand. Haptic feedback is provided via a simplified physics model, based on a depth, and drag component.

Geiger [8] presents a voxel-based method for simulating interactions with granular media. The approach outlined in the paper enables 3D manipulation of soil, by tracking the velocity of individual sand voxels, and subsequently transferring material to surrounding voxels as needed.

3 Methodology

In developing our soil cutting simulation, we encountered many possible approaches for creating a believable model. In this section, we discuss the intuition behind our choices, the tools which were used for our implementation and experimentation process, and the various challenges that were encountered during development.

3.1 Background

The key techniques used by our method draw inspiration from the realm of Computer Graphics. Two of the papers previously mentioned in the Section 2 propose two different models for simulating granular media, both with entirely different goals (haptics for 3D modelling [7], realistic soil displacement [6]). However, the core technique that underpins both implementations is the use of "heightfields" for the purpose of environment modelling.

Heightfields [1], or heightmaps, are data structures that encode the elevation properties of a surface, often in the form of a 2D texture. Such textures are commonly represented as grayscale images, where the brightness of a pixel denotes the elevation of a mesh at the corresponding texture coordinates.

Since heightmaps are easy to visualize, and similarly easy to create and store, they provide a fast and intuitive way to keep track of changes in an object's shape during the course of a physics simulation, and as a result were chosen to serve as the basis for the soil cutting simulation.

During the early stages of development, voxel-based physics simulations for modeling sand were also considered. However, we ultimately decided against this approach for reasons detailed in Section 3.3.

3.2 Tools

The model has been implemented in a "Bullet Physics SDK"[2]-based physics simulator, coupled to an OpenGL renderer. The core simulation, which was provided by TU Delft's Networked Systems group, has been tailored to work with the "Novint Falcon [3]", a USB controller that provides users with 3D haptic feedback.

The position of the Falcon was initially represented as sphere, with feedback being calculated based on the distance between the desired position, and a "proxy" collision object that interacted with objects in the simulated environment.

However, in the later stages of development, the proxy was replaced with a custom collision object that no longer provided feedback for the Novint Falcon.

All heightmaps used during testing have been created using "GIMP"[4], a commonly used open-source image manipulation software. Similarly, "Blender"[5] was used to create visualisations for experiment 2.

3.3 Implementation

The implementation of the model was created by modifying the pre-existing physics simulation to create a custom heightmap instantiator. This proved to be more challenging than expected, with the OpenGL renderer crashing when creating new rigid body objects. The core of the problem, as observed during the development process, relates to creating a new "collision shape" while the program is running.

As a result, using Bullet's built-in heightfield functionality was not possible, as modifying a heightfield would necessitate the creation of a new collision shape to replace it, invariably resulting in a crash.

The inability to create new objects during runtime also prohibited us from exploring other methods of cutting shapes. One initial idea was to create columns of material based on height information from a heightmap, and then dynamically split each column when colliding with the simulated knife. However, as before, cutting a column in half would necessitate the creation of two new collision shapes, thus crashing the renderer.

As a result, the final model (seen in fig. 1) utilised an approach which entirely avoided the creation of any new objects during runtime, via the creation a new terrain-shape class, and new terrain controller.

A voxel-based approach also had the possibility of entirely avoiding the problem of creating new shapes, however, due to the time limitation inherent to the research project, had

[1] https://en.wikipedia.org/wiki/HeightmapHeightfields

[2] https://github.com/bulletphysics/bullet3

[3] https://delfthapticslab.nl/device/novint-falcon/

[4] https://www.gimp.org/

[5] https://www.blender.org/

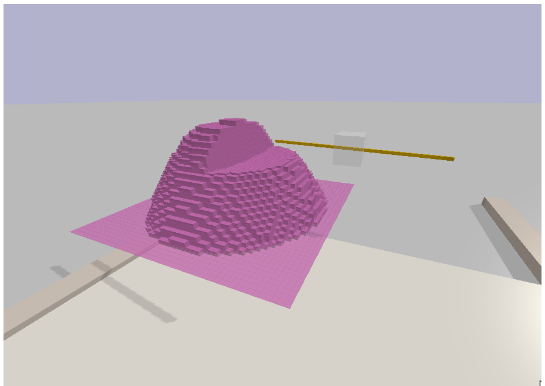
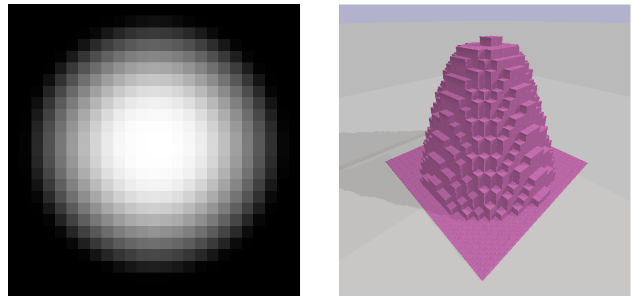


Figure 1: An in-engine screenshot of the terrain system, showing a partial horizontal cut. The blade is represented by a yellow box that is aligned with the blade's cutting axis. The position of the Novint Falcon's proxy is shown by the transparent cube.



(a) A 25x25 pixel heightmap.

(b) The structure created by loading image (a) into the simulation, made up of numerous boxes of varying heights.

Figure 2: Heightmap and its corresponding terrain, as seen in-engine. As mentioned before, brighter pixels correspond to taller "columns" of material. In this case, the heightmap shown in image (a) creates a mound-like structure.

to be abandoned in favour of more performance-oriented approaches. The use of a 100x100 heightmap texture could result in more than 10.000 shapes being displayed on the screen, based on the height values in the texture. As a result, while the simulation framerate would remain steady at 1 kilohertz, the renderer framerate would be below 10 frames per second, resulting in juddery inputs and frequent crashes.

4 Terrain Cutting Simulation

This section will detail the implementation of the heightmap terrain system, as well as the algorithm used to simulate cutting objects.

4.1 Terrain System

As mentioned in Section 3.3, a limitation in the C++ physics engine necessitated the implementation of a new, simplified heightmap terrain instantiator, and controller.

The instantiator used to derive terrain from a heightmap creates one box for each pixel in a heightmap, with the height of each box corresponding to each pixel's brightness. The result is a shape that matches the heightmap, comprised of "columns" of material. Within the simulation, each column is a separate, kinematic rigid-body object, that carries special behaviour in order to react to a collision with a blade.

Whenever the simulated blade collides with one of the columns, the affected column's rigid-body object undergoes a downward translation, such that the top of the box is below the line representing the blade (shown in fig. 3). This approach allows the system to keep track of two critical pieces of information: the remaining material (height of the column above the terrain's ground level), and the removed material (depth of the column below ground level). Tracking changes in the terrain's shape in this manner enables easy updating and storage of the heightfield's state: The heights of the columns above and below ground level are recorded into two different heightmaps.

4.2 Cutting Algorithm

Since the aim of the model is to provide feedback for MMT, the cutting algorithm must, at the very least, provide a metric

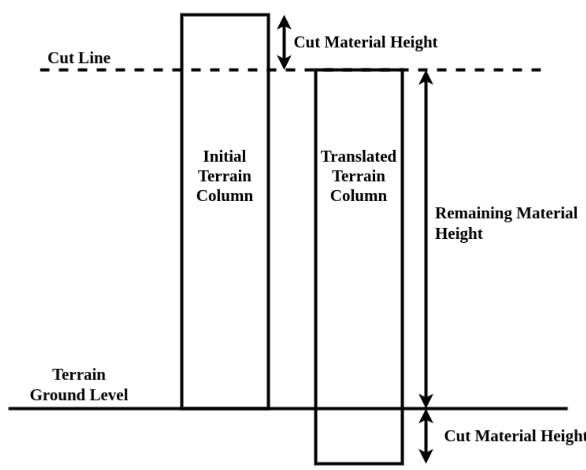


Figure 3: Diagram showing the behaviour of a terrain column when cut. The column is translated downwards such that the top of the shape is below the cut line. This approach allows the physics engine to simulate removing material, without creating any new rigid-body objects or collision shapes.

that is able to be used in feedback calculations for a haptic input device, such as the "Novint Falcon".

In it's current state, the model described in this paper does not provide haptic feedback to an end user. What it is capable of however, is the calculation of the length of a cutting blade's contact with terrain. While this metric is not necessarily indicative of the forces a user might feel when cutting, we believe that it could lead to believable force feedback, when used in other models.

To that end, in order to facilitate the calculation of the aforementioned metric, the simulation must be able to reliably calculate collisions with a heightmap in real-time. The algorithm used in this paper models the cutting surface as a line (similar to a hotwire cutter in concept). In order to measure the contact length, and trigger collisions with terrain columns, the cutting object calculates ray-box intersections with all rigid-body objects that make up the terrain. Since in our model all terrain boxes are axis-aligned, the simulation utilizes the very popular "Slab Method" for ray-box intersections, which yields two collision points (ray entry and exit) for boxes that lie in the path of the cutting axis. [9][10]. The model then sums the distance between the two collision points for each box that is intersected by the blade, once every millisecond.

5 Experimental Setup and Results

5.1 Experimental Setup

In order to gauge the accuracy and usefulness of our proposed model, two experiments were drawn up. On the whole, the testing done in this paper aims to show that heightmaps can be used to accurately predict object deformation for future interactions.

1. The first experiment aims to measure the impact of heightmap resolution on the resulting cutting feedback metric. During the testing process we consider the length of blade's contact with the terrain to be a good proxy for force feedback calculations. Described below is a step-by-step description of the way the experiment was conducted:
   (a) Create a heightmap in GIMP.
   (b) Scale the original heightmap to five different resolutions (5x5, 15x15, 25x25, 35x35, 50x50).
   (c) For each of the resulting heightmaps:
      - Load the heightmap into the simulation.
      - Perform a horizontal cut across the terrain.
      - Log the resulting contact patch information.

   For this experiment, the selected base heightmap generates a mound-like structure, as illustrated in fig. 2. This specific heightmap was chosen because it produces predictable variations in the length of the blade's contact patch, which is crucial for evaluating the system's performance. When the heightmap is sliced horizontally, it creates circular cross-sections of varying radii. These consistent geometric shapes allow us to precisely measure and analyze the blade's contact length across different resolutions.

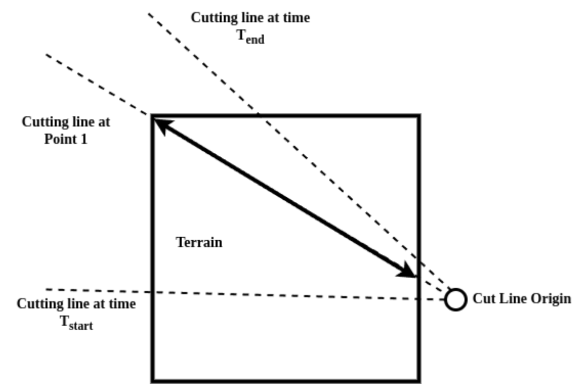


Figure 4: Diagram showing the interaction between the cutting line and terrain at three different moments in time. The image views the terrain along the Z axis. The behaviour seen in fig.10 becomes more apparent.

2. The second experiment aims to show that the proposed heightmap-based model can efficiently keep track of changes in an object's shape throughout a series of successive cuts, and that changing the order of the cuts results in different contact readings. Below is a is step-by-step description of the proposed experiment:
   (a) Create a base heightmap in GIMP.
   (b) Define three cuts across the terrain, referred to as A, B, and C.
   (c) For three different sequences of cuts comprised of A, B and C:
      - Load the base heightmap into the simulation.
      - Perform the sequence of cuts.
      - Log the contact information alongside the changes in shape.

   To properly gain an understanding of how this test might perform, we have to examine each of the three cuts individually first. Because the test performs successive cuts on the same initial object, the initial heightmap was chosen to be a white 35x35 square, in order to fill all terrain columns to the maximum amount (since all pixels in the texture are at maximum brightness).

   In **Cut A**, the blade moves along the the Z axis from one end of the object to the other. The initial cutting axis is X. As the blade moves across the terrain, the cutting line rotates 45 degrees along the Z axis. In order to cut a larger chunk, the blade moves slightly downwards. The contact patch data that results from the interaction shows a peak (visible in fig. 10, 5 and 4 as "Point 1") when the cutting line collides with the edge of the cube. A visualization of this cut can be seen in fig 5.

   Cut B is a simple horizontal cut, through the middle of the object. The blade moves along the Y axis, with the cutting axis being equal to the X axis. The direction of the cut is reversed when compared to cuts A and C. The contact information for this cut can be seen in fig.11 in appendix Section A.2.

   Cut C is similar to the first cut, but mirrored on the X

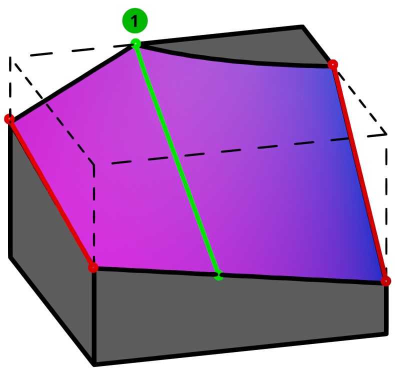


Figure 5: Cut A visualized in Blender. The image shows the terrain as a grey box, and the surface of the cut as colored gradient. The red lines show the position of the cutting line as the blade enters and exits the object. The dotted lines show the shape of the removed material. Point 1 corresponds to the longest contact length, and is shown as a line coloured in green (cutting line).

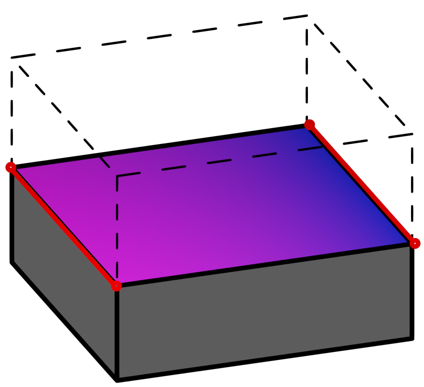


Figure 6: Cut B visualized in Blender. The image shows the terrain as a grey box, and the surface of the cut as colored gradient. The red lines show the position of the cutting line as the blade enters and exits the object. The dotted lines show the shape of the removed material.

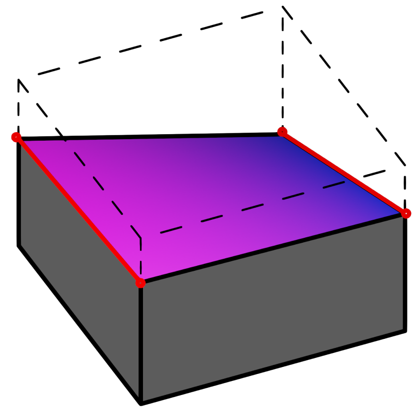


Figure 7: Cut C visualized in Blender. The image shows the terrain as a grey box, and the surface of the cut as colored gradient. The red lines show the position of the cutting line as the blade enters and exits the object. The dotted lines show the shape of the removed material.

axis. In addition, the rotation of the blade is less extreme, with Z rotation of about 26 degrees. The initial state of the blade is identical to cut A, thus oriented along the X axis. The contact information for this cut can be seen in fig.12 in appendix Section A.3.

5.2 Results

Experiment 1

The results of **experiment 1** suggests that while resolution can have a significant impact on the performance of the model, contact path measurements begin to display consistent behaviour at relatively low texture sizes.

The plot in Figure 8 illustrates the blade's contact length over time for different heightmap resolutions (5x5, 15x15, 25x25, 35x35, and 50x50). This graph helps us understand how varying the resolution of the heightmap affects the precision and stability of the blade's interaction with the terrain.

From the plot we can summarize that resolutions above 5x5 quickly converge to a consistent pattern, thus providing a stable and reliable measure of the blade's contact length. Higher resolutions, such as 50x50, lead to a reduction in jitter, which could enhance the smoothness of force interactions with deformable terrain. The graph also shows a linear increase in the minimum step size of the contact length as the resolution increases, indicating finer detail and accuracy in the blade's interactions with the terrain.

On the other hand, the 5x5 resolution displays abrupt changes in contact length and temporal inaccuracies when predicting a collision. It predicts that the start of the contact with the shape occurs 0.5 seconds earlier than any other resolution, due to the coarseness of the heightmap. Similarly, the predicted end of the collision is also offset from all other measurements, however the amount is different.

Experiment 2

In **experiment 2**, we analyzed how different sequences of cuts influence the contact length data in our terrain simulation. The aim was to assess the model's ability to track

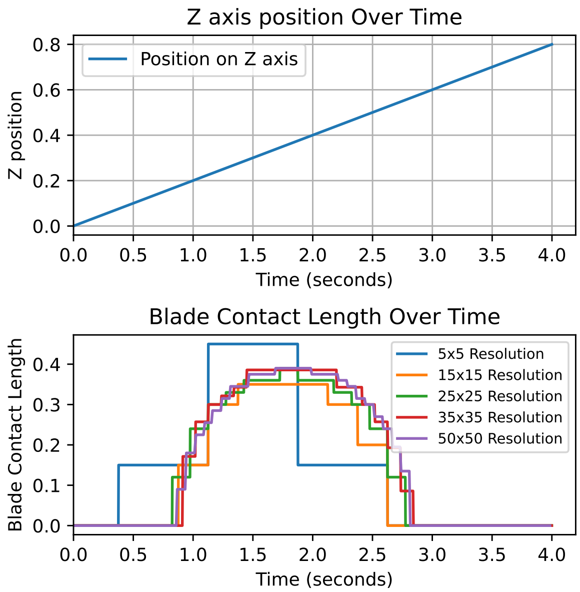


Figure 8: Graphs showing the performance of the terrain cutting system under different heightmap resolution. The figure logs contact length for 5 different texture sizes. The 5x5 resolution stands due to its temporal offset from all other measurements. Position is shown through a graph logging the blade's position on the Z axis over time (X and Y remain constant, and are left out for clarity).

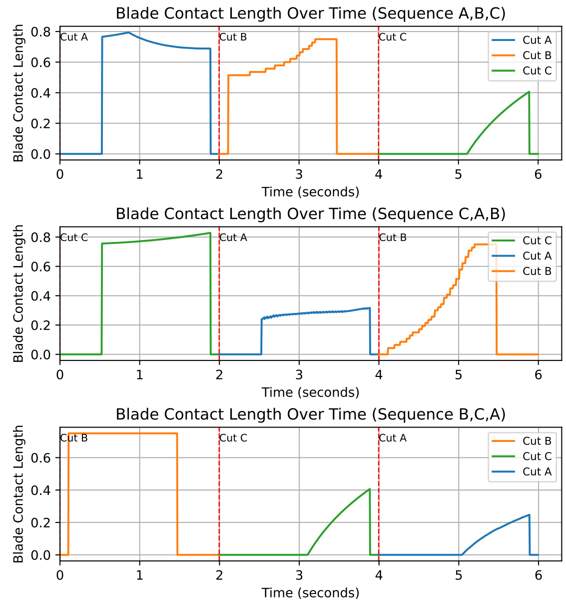


Figure 9: Graphs showing the performance of the terrain cutting system, for three sequences of consecutive cuts. Each sequence of cuts has its own graph, logging the length of the blade's contact with terrain during cutting.

changes in the terrain following sequential cutting operations. Additionally, the test showcases the terrain system's ability to handle complex movements, involving simultaneous rotation and translation.

In order to gain an understanding of the result's significance, we have to analyze each sequence in depth. All three sequences begin with a different initial cut, and last 6 seconds (with each cut taking 2 seconds).

1. **Sequence (A, B, C)**

   The graph for sequence (A,B,C), seen in fig. 9, shows the blade's contact length over time for each cut.

   - **Cut A** creates an initial deformation, producing a complex shape that influences the next cut.
   - **Cut B**'s contact length data reflects the changes introduced by Cut A. This is evident on the graph, as the convex shape between X coordinates 2.1 and 3.2 closely mirrors the shape of the terrain (as seen in fig.13). This suggests that the model can accurately track and represent the object's modified shape.
   - **Cut C** then interacts with the already altered terrain object. Due to the significant amount of material removed by the previous cuts, Cut C has a smaller contact length, and produces less significant changes when compared to the initial cuts.

2. **Sequence (C, A, B)**

   In the sequence (C, A, B), cuts C and A remove material from opposite sides of the object resulting in the following behaviour:

   - **Cut C** initiates the process by cutting a shape not to dissimilar from Cut A.
   - **Cut A** carves the other side of the object, but experiences a reduced contact patch due the Cut C's effect on the other side of the cube, significantly reducing the amount of terrain encountered.
   - **Cut B**'s horizontal slice logs a contact length shape similar to the one in sequence (A, B, C), however due to the addition of Cut C, the blade encounters very little terrain at the start of the cut (can be seen in fig. 14, the shape left after cut B).

3. **Sequence (B, C, A)**

   In the sequence (B, C, A), the following observation can be made:

   - **Cut B** effectively slices the cube in half. This cut significantly affects the following cuts' interaction with the terrain
   - **Cut C** follows, but only registers contact with the terrain after the blade starts to rotate. This is due to the fact that cut C starts out as a horizontal cut, thus effectively skimming surface created by the previous slice.
   - **Cut A** experiences the same behaviour as Cut C, but on the opposite side of the object.

The final shape of the object after executing any of the previously mentioned sequences of cuts remains the same, and can be seen in heightmap form in fig. 14.

6 Responsible Research

During the course of the research project, multiple factors regarding responsible research, such as transparency, reproducibility, ethical and safety considerations have been addressed.

First and foremost, from a transparency standpoint, we are committed to sharing comprehensive information about our research implementation. This includes providing detailed documentation of the image textures used in our tests and a thorough description of the codebase developed for this project. However, it's important to note that our model builds on code from TU Delft's Networked Systems group. Due to an embargo related to their ongoing research, we are unable to release the full code repository simultaneously with the publication of this paper. Once the embargo is lifted, we will make the complete codebase publicly accessible to support transparency and allow for further examination and replication of our work.

Subsequently, as the project's main purpose is to provide environment-modeling for force-feedback tasks, any erroneous behavior stemming from our method may put a human operator in physical danger. To that end, the terrain cutting simulation created in the paper has been made with the Novint Falcon's specifications in mind. Since the controller cannot provide forces that are so great as to risk injuries, we believe that our code cannot cause any harm, when used in the manner described in the paper.

Lastly, we are committed to providing reproducible and unbiased results, by meticulously documenting our research methods, and by including detailed descriptions of the teleoperation algorithms, hardware, and procedures that were used. All experimental results have been included, regardless of whether or not they support the feasibility of our method. Our experimental setup has been thoroughly explained, and external tools used have been mentioned in the paper.

7 Discussion and Future Work

7.1 Discussion

The primary aim of this study was to investigate the feasibility of using a heightmap-based model to predict deformation in teleoperation applications, by using cutting as a benchmark for more complex interactions. Our results demonstrate that heightmaps can effectively represent the deformation seen in granular soils, providing a robust framework for future force feedback models.

However, during the course of the project, multiple limitations were identified. Firstly, while heightmaps are efficient in representing the elevation of a surface, they simplify the three-dimensional complexities of the material being cut. In our simulation, all material above the blade vanishes. This simplification could affect the accuracy of force feedback, particularly in scenarios involving complex object shapes. While complex, a voxel-based model might have some advantages in this particular area, by being able to represent cavities in a shape.

Secondly, the implementation of the terrain, as described in this paper, does not support high-resolution heightmaps (above 200x200). We believe this to be linked to inefficiencies in the terrain implementation, rather than a drawback of the general approach outlined in this paper.

Finally, the impact of network delays, and real-time responsiveness of the system were not fully explored in our current experiments. These factors are critical in real-life teleoperation scenarios, where timely and accurate haptic feedback is critical from a safety standpoint.

7.2 Future Work

In order to build on current findings, we suggest multiple avenues for future research:

- **Enhanced Soil Modelling**: The current model could be expanded to support more complex granular soil properties, such as soil erosion and flow. The methods shown by Benes, Dorjgotov, Arns, *et al.* [7] and Li and Moshell [6] lend themselves particularly well to the heightfield terrain system created in the paper.
- **Broader Range of Interactions**: Apply the heightmap-based model to a wider range of teleoperation tasks beyond cutting, such as scooping, or other complex manipulations. This will help prove the versatility and robustness of such an approach.
- **Haptic Model**: As previously mentioned, the simulation described in this paper has the potential to act as a solid framework for a real-time haptic model of cutting. For instance, the length of the blade's contact with the soil could be linked to a drag parameter on the movement of the haptic controller. Research into this area could also benefit from a user study, measuring the usability of the force feedback.
- **Network Simulation**: Future work could explore the effects of network latency on a cutting simulation. This would involve testing the system under different network conditions (different delay settings).

8 Conclusion

In this paper, we explored the development of predictive models to simulate model deformations for the purpose of providing accurate and real-time force feedback in haptic bilateral teleoperation systems. Our research focused on leveraging heightmap-based models to represent and predict interactions with deformable materials, specifically granular soils, which are often encountered in complex remote operations.

The experiments conducted demonstrated that heightmaps can effectively model the deformation of materials during cutting tasks, providing a reliable basis for calculating predictive force feedback. We observed that higher resolution heightmaps lead to smoother and more precise feedback, crucial for maintaining effective and safe teleoperation. However, lower resolutions tend to introduce significant inaccuracies and jitter, reinforcing the importance of balancing computational efficiency with the need for detailed simulation.

Moreover, our model is able to handle sequential complex interactions, as shown in our second cutting experiment, suggesting that heightmap-based models are well suited for more intricate teleoperation tasks.

These findings pave the way for further advancements in teleoperation technologies, enabling more complex and intuitive remote manipulations. As teleoperation continues to evolve, the integration of such predictive models will be crucial for enhancing the safety, accuracy, and effectiveness of remote operations in diverse and challenging environments.

8.1 Acknowledgements

First and foremost, I would like to thank my supervisors, for providing insightful feedback and guidance over the course of the project. Their knowledge of the field of teleoperation has profoundly influenced my work.

I would also like to acknowledge the significant contributions of the previous developers and team members who worked on the codebase. In particular I would like to thank both Koen Wösten, and Kees Kroep for their readiness to share information about any previous work on the simulation software.

References

[1] C. Yang, J. Luo, and N. Wang, "Chapter one - introduction," in *Human-In-the-loop Learning and Control for Robot Teleoperation*, C. Yang, J. Luo, and N. Wang, Eds., Academic Press, 2023, pp. 1–15, ISBN: 978-0-323-95143-2. DOI: https://doi.org/10.1016/B978-0-32-395143-2.00005-X. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B978032395143200005X.

[2] V. Pruks, K.-H. Lee, and J.-H. Ryu, "Shared teleoperation for nuclear plant robotics using interactive virtual guidance generation and shared autonomy approaches," in *2018 15th International Conference on Ubiquitous Robots (UR)*, 2018, pp. 91–95. DOI: 10.1109/URAI.2018.8441814.

[3] Q. Lin and C. Kuo, "Virtual tele-operation of underwater robots," in *Proceedings of International Conference on Robotics and Automation*, vol. 2, 1997, 1022–1027 vol.2. DOI: 10.1109/ROBOT.1997.614269.

[4] X. Xu, B. Cizmeci, C. Schuwerk, and E. Steinbach, "Model-mediated teleoperation: Toward stable and transparent teleoperation systems," *IEEE Access*, vol. 4, pp. 425–449, 2016. DOI: 10.1109/ACCESS.2016.2517926.

[5] X. Xu and E. Steinbach, "Towards real-time modeling and haptic rendering of deformable objects for point cloud-based model-mediated teleoperation," in *2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2014, pp. 1–6. DOI: 10.1109/ICMEW.2014.6890578.

[6] X. Li and J. M. Moshell, "Modeling soil: Real-time dynamic models for soil slippage and manipulation," *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, 1993. [Online]. Available: https://api.semanticscholar.org/CorpusID:207176936.

[7] B. Benes, E. Dorjgotov, L. Arns, and G. Bertoline, "Granular material interactive manipulation: Touching sand with haptic feedback," Jan. 2006.

[8] A. D. Geiger, "A voxel-based approach to the real-time simulation of sands and soils," AAI28141638, M.S. thesis, 2015, ISBN: 9798662598959.

[9] T. L. Kay and J. T. Kajiya, "Ray tracing complex scenes," *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, 1986. [Online]. Available: https://api.semanticscholar.org/CorpusID:624126.

[10] A. Majercik, C. Crassin, P. Shirley, and M. McGuire, "A ray-box intersection algorithm and efficient dynamic voxel rendering," vol. 7, no. 3, pp. 66–81, Sep. 2018. [Online]. Available: https://jcgt.org/published/0007/03/04/.

A Contact Length Graphs for Cuts A, B, C

A.1 Cut A

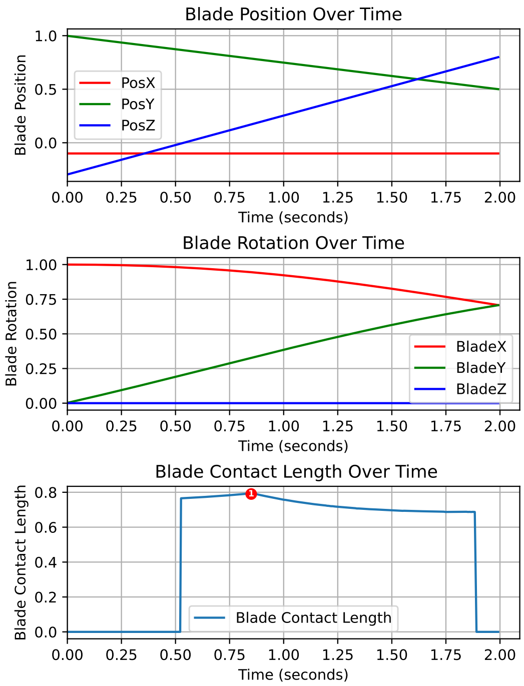


Figure 10: Graphs showing the position, rotation, and blade contact information for cut A. Cut A performs cuts the terrain while performing a rotation over the Z axis. The contact patch length peaks at Point 1, and slowly decreases until the blade abruptly exits the object.

A.2 Cut B

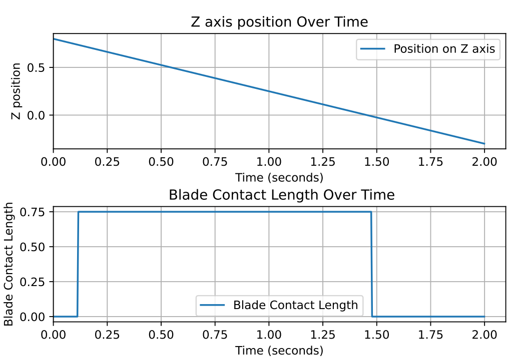


Figure 11: Graphs showing the position, rotation, and blade contact information for Cut B. Cut B results in a an abrupt increase in the length of the contact patch, as the blade collides with the terrain. Cut rotation has been omitted since the cutting axis is identical to the X axis for the duration of the cut. The X and Y positions have similarly been left out for clarity

A.3 Cut C

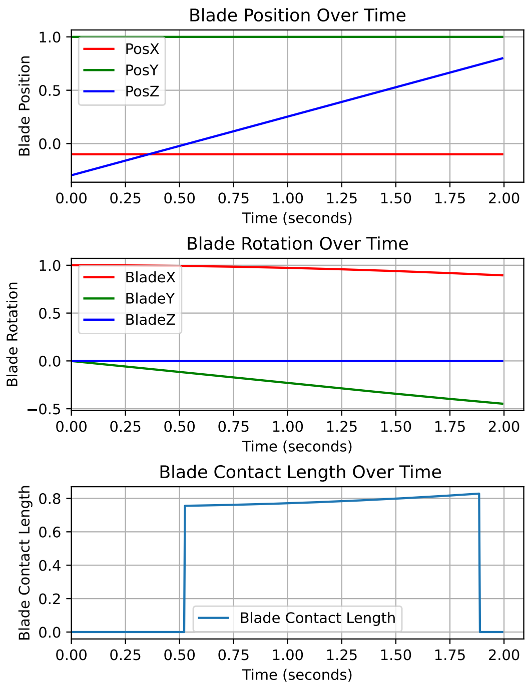


Figure 12: Graphs showing the position, rotation, and blade contact information for cut C. Cut C slices the terrain while performing a rotation around the Z axis.

B Heightmaps

B.1 Heightmap Resulting from Sequence (A,B)

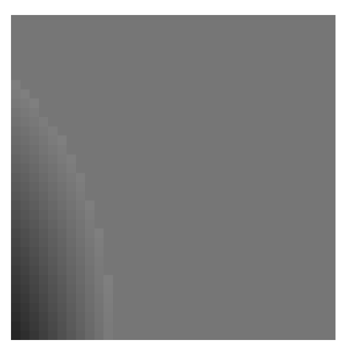


Figure 13: Heightmap created during experiment 2, after the initial object has been subjected to cut A, and cut B. Since cut B is a simple horizontal slice, most of the terrain is flat. Due to the fact that the image's vertical axis corresponds to the simulation's Z axis, the border of the flat plane determined by cut B closely resembles the contact patch information in fig. 9 (for sequence A,B,C)

B.2 Heightmap After Executing All Cuts

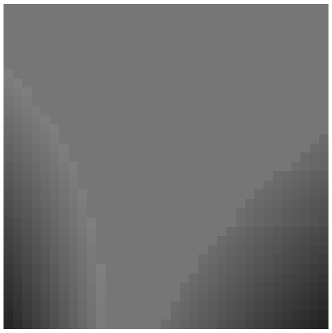


Figure 14: Heightmap created during experiment 2, after all cuts have been performed. The shape visible on the left is caused by Cut A, and the one on the right by Cut C. The flat grey area visible towards the top is caused by Cut B's horizontal slice.