Computationally Efficient Techniques for Data-Driven Haptic Rendering

Raphael Höver[1]* Massimiliano Di Luca[2] † Gábor Székely[1] ‡ Matthias Harders[1] §

[1]Computer Vision Lab - ETH Zurich, Sternwartstr. 7, CH-8092 Zurich
[2]Max Planck Institute for Biological Cybernetics, Spemannstr. 41, D-72076 Tübingen

Abstract

Data-driven haptic rendering requires processing of raw recorded signals, which leads to high computational effort for large datasets. To achieve real-time performance, one possibility is to reduce the parameter space of the employed interpolation technique, which generally decreases the accuracy in the rendering. In this paper, we propose a method for guiding this parameter reduction to maintain high accuracy with respect to the just noticeable difference for forces. To this end, we performed a user study to estimate this perception threshold. The threshold is used to assess the final error in the rendered forces as well as for the parameter reduction process. Comparison with measured data from real object interactions confirms the accuracy of our method and highlights the reduced computational effort.

Keywords: Virtual Reality, Haptic Rendering, Deformable Models

Index Terms: I.6.5 [Simulation and Modeling]: Model Development—Modeling methodologies; I.2.10 [Artificial Intelligence]: Vision and Scene Understanding—Modeling and recovery of physical attributes H.5.2 [Information Interfaces And Presentation]: User Interfaces—Haptic I/O

1 Introduction

Data-driven rendering in haptics research has recently received attention due to its ability of producing non-linear and complex dynamic effects without employing any explicit physical model. This method is based on the recording of the contact forces during interactions with real objects. Subsequently, virtual force feedback is provided by interpolation of the recorded data. According signal capturing techniques have been used in several related activities. For instance, in [1, 2, 3, 4] parameters of a-priori models are extracted from acquired sensor readings. Due to the fixed model, specific cues in the force signal might not be considered in the fitting step. Furthermore, in [5, 6, 7, 8] rendering techniques are considered where the force feedback is generated by superposing previously acquired recordings. This strategy is appropriate for materials with linear properties. In [9] also nonlinear effects are addressed using more sophisticated interpolation methods. Finally, purely data-driven techniques have been investigated in [10] and [11]. In their methods recorded forces are interpolated by generic functions that are not derived from physical models.

Since the paradigm of pure data-driven rendering requires processing of raw recorded data, the computational effort of the corresponding algorithms is directly related to the number of samples that have been acquired. This affects both possible offline interpolations of the data as well as online evaluations for the rendering. The latter is most critical since the interactive data interpolation needs to be carried out at the typical haptics update rate of $1\,kHz$. Due to the dependency on the amount of captured data, real-time performance can often only be achieved for very limited datasets.

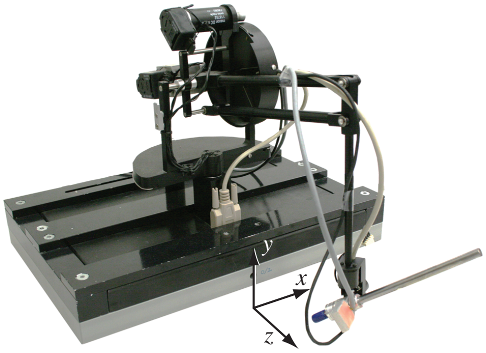

Figure 1: Setup to capture user movements and contact forces between a probing tool and sample objects.

In order to overcome this problem, we followed a precalculation strategy to determine the interpolant offline [11]. However, the computational effort needed grows exponentially with the domain dimension, thus limiting the method to low-dimensional problems.

In this paper, we examine computationally efficient techniques for offline interpolation as well as the online evaluation of data-driven haptic rendering based on radial basis functions (RBFs). To this end special RBF center selection strategies are investigated. These techniques render the precomputation step obsolete and guarantee real-time performance. In order to assess the final error in the data-driven haptic rendering, perception thresholds for force discrimination are applied. To determine the latter, we performed a user study to obtain a conservative estimate of just noticeable force differences. The derived perception thresholds are also used to guide the optimization of one of the proposed algorithms. Comparison with measured data from real object interactions confirms the accuracy of our method and highlights the reduced computational effort.

2 Recording Setup

In [11] we presented a setup for recording user movements and contact forces during the manipulation of real objects. This system is slightly enhanced for the current study by adding an acceleration sensor (ADXL330 from Analog Devices) to the probing tool. The new sensor allows our system to obtain more accurate estimates of tool velocity, which is important for dynamic material effects. The complete setup is shown in Fig. 1.

The signals from all sensors are captured synchronously by a real-time system running under RTAI Linux. For capturing and rendering a sampling frequency of $1\,kHz$ is used. The force output of the device was calibrated using the integrated force sensor. The offline computations are performed on an Intel Xeon Quad-Core with $2.66\,GHz$ and $16\,GB$ RAM.

*e-mail:hoever@vision.ee.ethz.ch
†e-mail:max@tuebingen.mpg.de
‡e-mail:szekely@vision.ee.ethz.ch
§e-mail:mharders@vision.ee.ethz.ch

2.1 Sample Material

A block of silicone serves as our example object for evaluating the fast rendering strategies. A key step of our data-driven rendering technique is the selection of appropriate data dimensions to capture and reproduce the object behavior. Force-displacement curves were obtained in a preliminary experiment to characterize the silicone material behavior. Beside the obvious elastic component, viscous behavior depending on tool velocity as well as stress relaxation effects were observed.

In our data-driven approach we interpolate the measured forces by a generic interpolation method. Hence, the elasticity of the material is considered by using the tool position as one dimension of the interpolation domain. In order to account for the dynamic material behavior, we extend this domain with additional dimensions that correspond to the observed effects. For the viscous behavior velocity is used. To account for stress relaxation, we rely on the generalized Maxwell model. This model incorporates such transient effects via branches consisting of serial spring-damper systems [12]. In the Laplace domain, the relation between the branch force F_b and the velocity V is given by

$$F_b(s) = D_b \frac{1}{\tau_b s + 1} V(s) \tag{1}$$

where $\tau_b = \frac{D_b}{K_b}$ is the relaxation time, D_b the linear damping and K_b the linear stiffness in branch b. According to the term $\frac{1}{\tau_b s+1} V(s)$ in Equation 1, stress relaxation can be captured by using lowpass-filtered versions of the current velocity as additional data dimensions. Therefore, in our investigations we also include two such relaxation branches. Average relaxation times of $\tau_1 = 0.17s$ and $\tau_2 = 1.27s$ were obtained by a fit to the recorded transient signal parts. Restricting the interaction to one degree of freedom results in a four-dimensional domain for interpolating the force signal from the silicone. The domain is given by the position, the velocity, and the two lowpass-filtered velocities of the tool. In the following, we denote a vector in this four-dimensional domain as $\mathbf{u}$. Once we recorded the contact forces $\{\mathbf{f}_i\}$ $(i = 1, \ldots, N)$ and derived the corresponding interaction vectors $\{\mathbf{u}_i\}$ $(i = 1, \ldots, N)$ we perform a first dataset reduction as proposed in our previous work [11]. This is done by removing all data pairs $\{\mathbf{u}_j; \mathbf{f}_j\}$ where the force changes by less than $50mN$ from the previous sample. This data reduction is used to remove clusters in the data set due to slow movement of the user. It is not directly linked to a perception threshold.

3 General Rendering Algorithm

In our data-driven haptic rendering we use radial basis functions to interpolate the raw captured force data. The interpolation is necessary to enable force feedback for arbitrary interactions in a virtual environment, since the user is not restricted to the same trajectory used in the recording. A scalar radial basis function is given by

$$h(\mathbf{x}) = \sum_{j=1}^{N} w_j \phi(\| \mathbf{x} - \mathbf{q}_j \|) + \sum_{k=1}^{L} d_k g_k(\mathbf{x}) \qquad \mathbf{x} \in \mathbb{R}^n, \tag{2}$$

where w_j are the weights and $\mathbf{q}_j$ the centers of the radial basis function $\phi : \mathbb{R}_+ \to \mathbb{R}$. The terms $g_k(\mathbf{x})(k = 1, \ldots, L)$ form a basis of the space $\mathbb{P}_m^n$ of n-variate polynomials with maximum degree m. The d_k are the unknown coefficients of this polynomial. The polynomial term is required since we use polyharmonic splines for ϕ, which are only conditionally positive definite. Without this polynomial term the interpolation problem can be ill-posed for certain datasets [13].

If direct interpolation is considered, the centers $\mathbf{q}_j$ are given by the normalized vectors $\mathbf{u}^\circ$ derived from the captured trajectory. In addition, the constraints

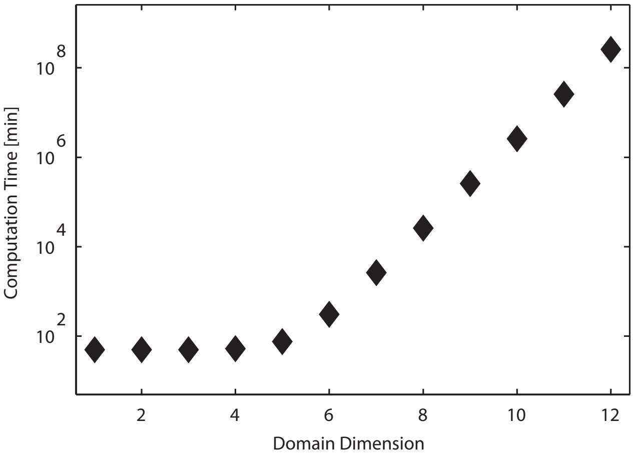

Figure 2: Dependency of computation time on domain dimension for interpolation and precalculation of a dataset with 6,850 data points and 10 voxels resolution.

$$\sum_{j=1}^{N} w_j g_i(\mathbf{u}_j^\circ) = 0 \quad \text{for } i = 1, \ldots, L, \tag{3}$$

have to be met to guarantee a well posed interpolation problem [13]. These equations can be combined in the linear system

$$M \begin{pmatrix} \mathbf{w} \\ \mathbf{d} \end{pmatrix} = \begin{pmatrix} \Phi & G \\ G^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{w} \\ \mathbf{d} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix} \tag{4}$$

$$\Phi_{ij} = \phi(\| \mathbf{u}_i^\circ - \mathbf{u}_j^\circ \|) \qquad i, j = 1, \ldots, N \tag{5}$$

$$G_{ij} = g_j(\mathbf{u}_i^\circ) \qquad i = 1, \ldots, N; j = 1, \ldots, L \tag{6}$$

$$\mathbf{f} = (f_1, \ldots, f_N)^T \tag{7}$$

with the weight vector $\mathbf{w} = (w_1, \ldots, w_N)^T$ and the coefficient vector $\mathbf{d} = (d_1, \ldots, d_L)^T$. The solution of this system determines the interpolation function. However, in our case the system is usually ill-conditioned and cannot be solved directly since the data sites lie close together in the domain [14]. In [11], we used a truncated singular value decomposition to address this issue. However, due to the large number of data points N, the interpolant has to be precomputed offline at fixed domain discretizations, in order to render online evaluation during the simulation feasible.

Unfortunately, this technique becomes computationally intensive for larger recordings and domain dimensions. The problem is illustrated for an example dataset in Fig. 2, where the required computation time is depicted for a dataset of 6,850 data points at a resolution of 10 voxels along each dimension of the interpolation domain. As shown, for a small number of dimensions, the inversion of M dominates the overall computation time, which takes about 50 minutes. However, the inversion of the matrix M has the complexity $O(N^3)$ in the number of data points, which limits the interpolation approach to problems with a few thousand samples. Moreover, the precomputation requires the specification of the number of precomputed sites along each domain dimension and is only possible on a finite subspace of the interpolation domain.

In addition to this, the number of voxels grows exponentially with increasing domain dimensions. Thus, if more than four dimensions are used for the interpolation, the computation time is dominated by the exponential increase of the precalculation step.

For instance, if 6 dimensions are considered, the computation requires 308 minutes. Hence, for larger domain dimensions a precalculation step is not feasible. Therefore, we examined methods to reduce the computational effort of the offline and online steps, which are discussed in the following.

4 Computationally Efficient Techniques

Considering that an exact solution of Equation 4 is not feasible, the computational effort for solving the linear system and evaluating the interpolant can be decreased by reducing the center set $\{\mathbf{q}_j\}$.

Each center corresponds to one column in the matrix of the linear system. Thus, if we remove specific centers we directly affect the range of the column space of this matrix. Therefore, the original interpolation problem is changed to an approximation problem. Various efficient techniques have been examined to follow this strategy. In [14] it is shown that the approximation order directly depends on the maximum distance between the center set and the data points. Hence, a center selection strategy based on scattered data filtering is proposed that maintains a small distance between the centers and data points. More recently, another selection strategy was proposed in [15]. A quad-tree is used to decompose the domain and define the center positions. By withdrawing the restriction that the centers have to be a subset of the given data sites, superior results in the approximation accuracy are achieved. In [16], the centers are selected based on a farthest point strategy to assure a minimum condition number of the linear system.

In our case, we strive to optimize the set of selected centers in order to minimize a residual error of the approximation, thus enhancing the haptic rendering quality. Our method minimizes the mean squared relative force error that is generated by the approximation to Equation 4. This is achieved by scaling the matrix M and the right-hand side of the system with the matrix

$$\begin{pmatrix} diag\left(\frac{1}{f_i}\right) & 0 \\ 0 & I \end{pmatrix} \tag{8}$$

where I is the $L \times L$ identity matrix. The scaled matrix is denoted by M'. Only the columns of this matrix that correspond to the selected centers are then used to calculate the pseudo inverse and minimize the residual error of the linear system. In the following we will outline the associated perceptual user study.

4.1 Determination of Perceptual Thresholds

In order to examine the perceivability of the residual error in the force signal we relate the relative error of each rendered force sample to the just noticeable difference (JND) of forces in our specific setup. Note, that these JNDs can be used for both, the evaluation and guiding the center selection. In the literature, the JND is only examined for forces larger than $2.5N$. Moreover, the experimental configuration and task is different from a typical interaction with a PHANToM. The results in [17] show that for a pinching task, using thumb and index finger, the JND is roughly 7% and remains constant for different reference forces. This conforms to the Weber fraction, which describes a constant relation between a stimulus intensity and the just noticeable difference. However, in [17] it is mentioned that the JND should increase for very small forces, thus deviating from Weber's law. This behavior has already been observed in [18].

For the optimization that we want to carry out, an estimate of the threshold for the whole range of possible contact forces is required. Thresholds should also be determined for typical interactions used with our haptic setup. Therefore, we performed a user study following a two alternatives forced-choice design to determine the JND for small forces. Subjects held the stylus of the PHANToM horizontally with the arm stretched out while the elbow position was fixed. In each trial, two forces were rendered consecutively via the device to the stylus held by the user. The forces were directed towards the user roughly oriented along the stylus. Note that in this pre-study we only examined force discrimination without user movement. Future work will also target active arm movements, however, research in [19] already indicates that JNDs might be higher due to effects of tactile suppression. After each trial the subject had to provide feedback as to which of the two forces felt larger. In the trials, reference forces were $0.3N$, $0.5N$, $1.0N$ and $2.5N$. Comparison forces were presented in a range from 60% to 140% of the corresponding reference force. Each pair of forces was presented 10 times, thus resulting in 320 trials. Twelve subjects participated in the study, 8 male, 4 female.

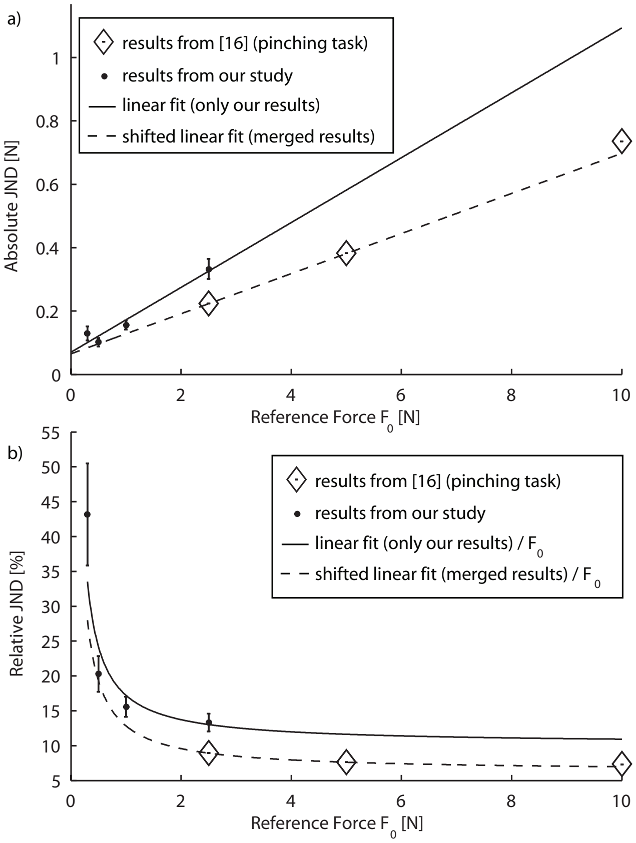

Figure 3: ((a) Absolute force JND over the reference force. A linear function is fitted to the results of our study (solid line). The dashed line is a shifted conservative fit so that the data points from our study and from [17] lie above the linear function. (b) Relative force JND. The hyperbole curves were obtained by dividing the corresponding linear functions from (a) by the reference force F_0.

Note that with this experiment we try to obtain perception thresholds for our particular device and interaction, and that the results are only valid for this specific configuration. The outcome of our study is shown in Fig. 3. In the upper part, the absolute JND is plotted for the different reference forces. For comparison we also include the results from [17]. A linear fit to our data is shown in the plot by a solid line. As can be seen, the fitted function deviates from the previously reported values. A possible reason for this behavior is the difference in the experimental design. In our case the stylus was held at a fixed position and the forces pointed towards the subject. In [17] the subjects performed a pinching task, which also provides proprioceptive cues. Moreover, in our study only naive subjects were used while in the other study more than half of the

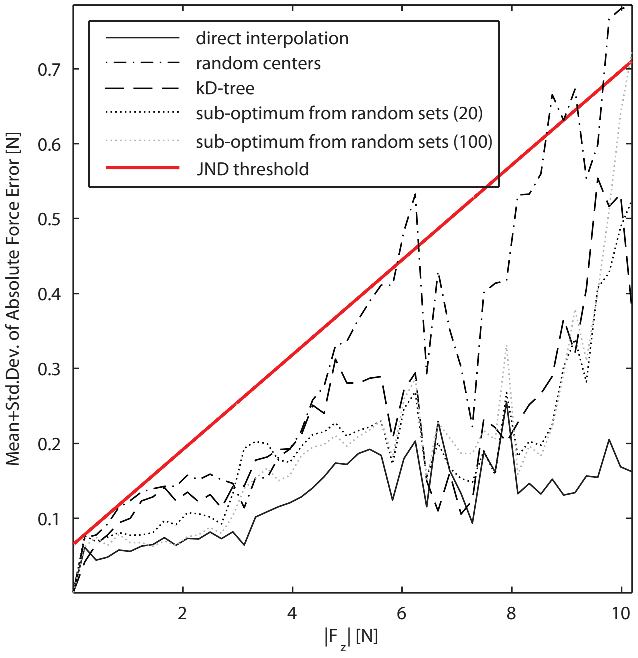

Figure 4: Mean plus standard deviation of the absolute force error plotted over the absolute value of the force for the different center selection strategies. Also, the threshold from Fig. 3 (a) is shown.

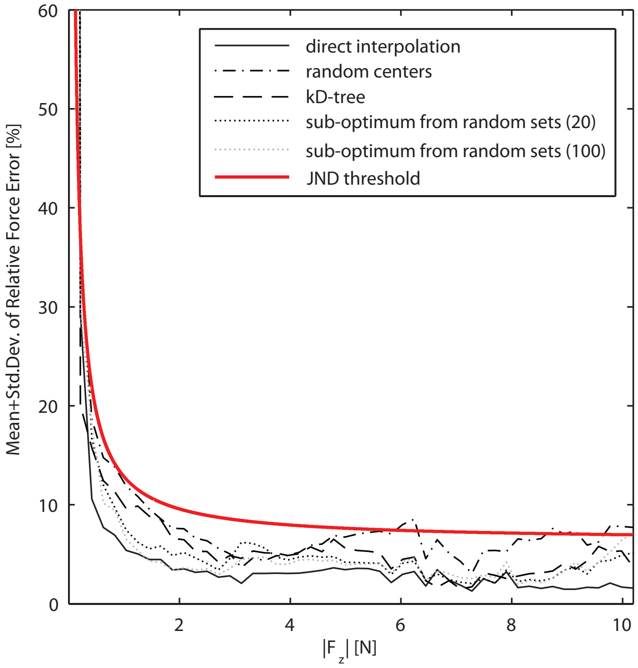

Figure 5: Mean plus standard deviation of the relative force error plotted over the absolute value of the force for the different center selection strategies. Also, the threshold from Fig. 3 (b) is shown.

subjects were familiar with such setups. The deviations are also evident for the reference force $2.5N$, which was considered in both studies. In our experiment the average relative JND for this force level is 13.3% while in the related work it is reported as 9% (t-test against 9% $t(11) = 3.57$, $p = 0.0044$, two tails).

In order to obtain a conservative estimate of the overall threshold curve that we will use for the optimization, we first fit a linear function to the data of both studies. This function is then shifted so that all measurements finally lie above this line (dashed line). The shifted line has a slope of $\nabla JND_{abs} = 6.33\%$ and $JND_{abs}(F_0 = 0N) = 0.065N$. This function is finally applied in our algorithm to estimate the perceivability of the resulting force errors.

In Fig. 3 (b), the relative threshold for the different reference forces is plotted. These are obtained by dividing the absolute values by their reference forces. The data points from our study confirm the increase of the relative JND for small forces as it is predicted in [17].

4.2 RBF Center Selection Strategies

Finding an optimal point set of a predefined size that is used for the centers of the RBF approximation is NP complete [14]. Thus, for large datasets as in our case, this optimization is not feasible. Therefore, we examine three alternative algorithms for selecting the center positions. While the first two strategies focus on minimal computation time for the center selection, the third method is designed to maintain a high accuracy of the resulting approximation. Note that, independent from the employed selection strategy, the resulting approximation can be evaluated in a haptic loop in real-time. This is achieved by fixing the number of RBF centers N_c, used for the approximation. This directly controls the time necessary to evaluate the fitted function at runtime as the complexity for this task is $O(N)$. In our experiments we use $N_c = 100$, which results in an evaluation time of $0.06ms$ on our real-time system.

To demonstrate the approximation quality for each center selection strategy, we use recordings of user interaction with the silicone block. Data from one recording are used to compute the approximation function. A different recording is then used to test the accuracy of the fitted function. The trajectory of the latter serves for calculating the virtual force feedback based on the determined approximation function. Finally, the calculated forces are compared to the measured forces. For each computed force sample we compare its error with the perception threshold given in Fig. 3.

The applied example recording consists of $41,499$ data samples and is reduced to $N = 6,850$ following the steps explained in Section 2.1. The remaining data points are used for the center selection strategies. The second recording consists of $45,272$ data points, which are all used for the evaluation.

The sum of mean errors plus standard deviations of the error distributions for each force magnitude is plotted in Fig. 4 and 5 together with the threshold curves derived in Section 4.1. In addition, in Fig. 6 and 7 the maximum error for each force magnitude is plotted. Note that in the latter plots one single outlier could be located above the threshold while the bulk of the error values could lie below the perception curves. However, since a single erroneous value could result in perceived differences between real and virtual renderings we also examine these maximal errors. To compare the result to the accuracy achieved by a direct interpolation using a precalculated grid of $25 \times 11 \times 11 \times 11$ voxels, the resulting error curves are also plotted in the figures.

4.2.1 Random Center Set

The first strategy performs a random selection of N_c centers drawn from the set of normalized interaction vectors $\{\mathbf{u}_i^\circ\}\,(i = 1,\ldots,N)$. Using this approach the matrix of the over-determined system might be ill-conditioned since some of the randomly selected centers can lie close together. Hence, we use a truncated singular value decomposition to determine the pseudo inverse. The complete data processing takes $10s$ from which $4s$ are used to compute the RBF approximation.

While this approach requires the least computational effort, the

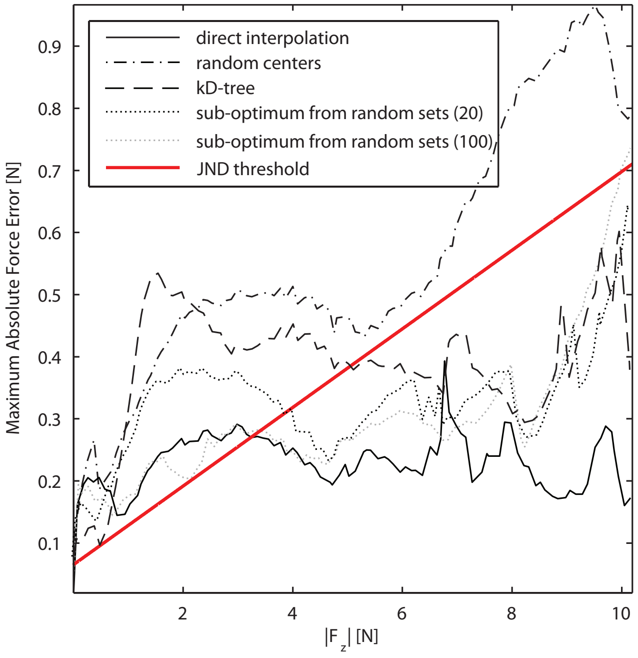

Figure 6: Maximum absolute force error plotted over the absolute value of the force for the different center selection strategies. Also, the threshold from Fig. 3 (a) is shown.

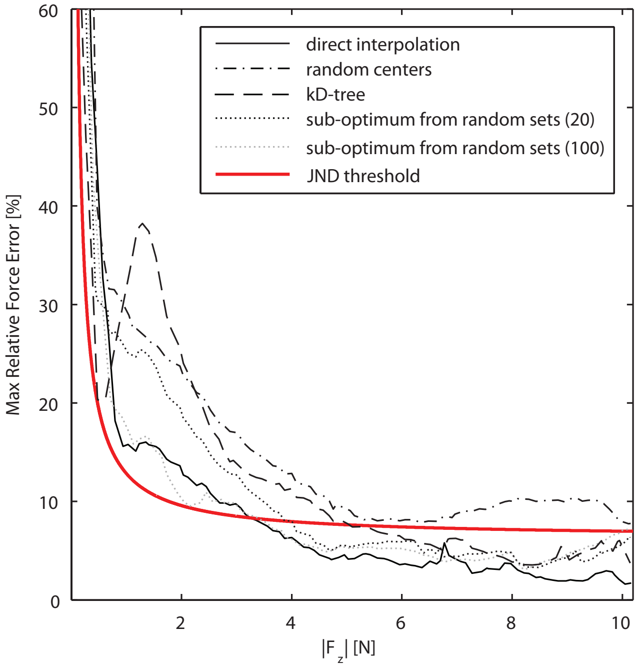

Figure 7: Maximum relative force error plotted over the absolute value of the force for the different center selection strategies. Also, the threshold from Fig. 3 (b) is shown.

error plots reveal that for each force magnitude there exist computed samples which have a force error that lies above our threshold curve. This result is also observed with other recordings so it is not specific to this particular dataset.

4.2.2 KD-Tree Selection

The second center selection strategy is motivated by [15]. As our data is embedded in a four-dimensional domain, each regular subdivision of a rectangular cell, as proposed in [15], would result in $2^4 - 1 = 15$ new cells and thus 15 new centers for the RBF approximation. To achieve a better control of the center selection, we use a kD-tree [20] instead, where each subdivision only generates $2^1 - 1 = 1$ additional cell. This way, we have more control where to place the centers.

The algorithm initializes the tree by sorting the data sites $\{\mathbf{u}_i^\circ\}\,(i = 1,\ldots,N)$ into a kD-tree with 2 leafs, each containing the same number of data sites. The barycenters of the subsets in each leaf are used as center positions for the RBF. Note that this generates center positions that do not necessarily coincide with a data point. Then, the RBF is fitted to the data using these 2 centers. To guide the refinement of the tree, the relative error of the fitted RBF is evaluated for all data sites. The sample $\mathbf{u}_i^\circ$ for which the relative error deviates the most from the threshold curve is located in the tree and the corresponding leaf is subdivided. Thus, the refinement of the tree is based on the determined perceptual threshold. The center generated by the original leaf is removed and two new centers are created by the two new leafs. After that, the RBF is fitted to the data using the new center set.

This process is repeated until the maximum number of centers N_c is reached. The algorithm finishes after roughly $5\,min$. Again, we test the final approximation using the test recording. Due to the adaptive refinement of the approximation the overall accuracy is better than the one from the random selection strategy, especially for large forces. The mean plus standard deviation curves are always below the threshold, which confirms the high accuracy of this method. Similar results are obtained with other datasets.

4.2.3 Sub-optimum from Random Sets

The third strategy for the center selection iteratively selects the locally optimal data site $\mathbf{u}_i^\circ$ that minimizes the mean squared relative error of the linear system. The algorithm initializes M' with the scaled polynomial part. The center set is initially empty. In each iteration, a random subset of the data sites $\{\mathbf{u}_i^\circ\}\,(i = 1,\ldots,N)$ is generated and each element of this random subset is tested separately by extending M' with the corresponding column that is generated by the data point. The data point that minimizes the residuum of the least squares problem is added to the center set and removed from the set of data sites. M' is permanently extended by the column corresponding to the data point. This greedy center selection is continued until N_c centers have been selected.

We tested random subsets with 20 and 100 elements for each iteration. In the former case, the computation needs $9\,min$, in the later case it requires $41\,min$. While the maximum error for the small forces is considerably lower if 100 samples are used in each iteration, the expected error given by the mean plus standard deviation curves is almost identical and always lies under the threshold.

This approach achieves the highest accuracy of all presented approximation techniques. Using random sets of 100 elements in each iteration leads to an accuracy that is almost identical to the accuracy of the interpolant. These results do not vary considerably for multiple repetitions using other random selections.

4.3 Computation Time

To demonstrate the improved speed of the discussed methods we measured the computation times with respect to the number of data points used for the RBF fitting. The results are shown in Fig. 8. For our previous pure interpolation a precalculation is performed on a $25 \times 11 \times 11 \times 11$ grid. As mentioned in Section 3 the inversion of M has the complexity $O(N^3)$, which causes the nonlinear increase in computation time for the direct interpolation. In contrast to this,

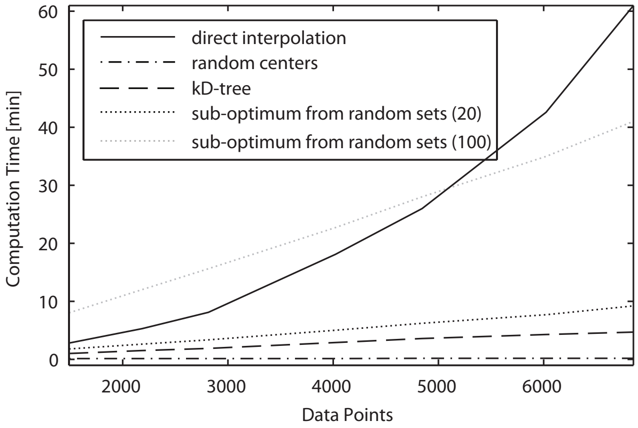

Figure 8: Measured computation time for the interpolation and the different approximation techniques with respect to the number of data points used for the fitting. In case of the interpolation the time also incorporates the precalculation of the interpolant on a $25 \times 11 \times 11 \times 11$ grid.

all presented fast approximation techniques exhibit a linear complexity with respect to the number of data points. The selection strategies that lead to more accurate approximations require more computation time. However, for large datasets, all proposed methods outperform the pure interpolation.

Assuming that the required number of RBF centers does not grow exponentially with the domain dimension to maintain a specific accuracy, the exponential increase of the computation time for higher dimensions, as shown in Fig. 2, is avoided. Hence, these techniques can be used in high-dimensional domains since no precomputation step is necessary.

5 Conclusion

In this paper we presented a method for reducing the computational complexity of data-driven haptic rendering. The method applies to radial basis functions for the interpolations of measured data and focuses on the reduction of the number of interpolation centers. We presented three different center selection strategies that trade off performance against computational complexity. All proposed methods reduce the required computation time considerably.

We showed that the proposed techniques exhibit linear complexity in the number of used data samples. Moreover, the computational effort for evaluating the approximation function can be directly controlled by prescribing a maximum number of centers used for the RBF. In our example we set this parameter to 100, which results in an evaluation time of $0.06\,ms$. This leaves enough time for other computations like collision detection in the haptic simulation.

Furthermore, in a user study, we examined the human discrimination threshold for small forces considering a typical configuration using the PHANToM haptic display. The results confirm an increase of the relative JND for small forces. The choice to minimize the mean squared relative force error in the fitting step is based on the observation that the relative perception threshold is approximately constant for a wide range of force magnitudes. Also, the kD-tree based approximation directly employs the derived threshold to guide the center selection. Finally, we apply the estimated threshold to evaluate the accuracy of our rendering strategies.

In our future work we will extend the recording scenario to consider more degrees of freedom of the tool. This will allow for lateral movements and eventually sliding contacts during the recording. To capture such situations appropriately, the sensing capabilities of the tool need to be further extended in order to detect the slip between the tool and the object. We also will further investigate the perceptional thresholds for active arm movements. Finally, we will perform studies where users directly compare the feedback from real and virtual objects to scrutinize the perceivability of force errors in a real-time simulation.

Acknowledgements

This work was supported by the EU project Immersence IST-2006-27141.

References

[1] K. MacLean, "The 'haptic camera': A technique for characterizing and playing back haptic properties of real environments," in *Proc. of ASME Dynamic Systems and Control Devision*, vol. 58, Nov. 1996, pp. 459–467.

[2] C. Richard, M. Cutkosky, and K. MacLean, "Friction identification for haptic display," in *Proc. of the ASME Dynamic Systems and Control Division*, vol. 67, 1999, pp. 327–334.

[3] A. Okamura, R. Webster, J. Nolin, K. Johnson, and H. Jafry, "The haptic scissors: Cutting in virtual environments," in *Proc. of the ICRA*, vol. 1, Sep. 2003, pp. 828– 833.

[4] M. Colton and J. Hollerbach, "Reality-based haptic force models of buttons and switches," in *Proc. of the ICRA*, Apr. 2007, pp. 497–502.

[5] D. Pai, J. Lang, J. Lloyd, and R. Woodham, "Acme, a telerobotic active measurement facility," in *Experimental Robotics VI*, vol. 250, 2000, pp. 391–400.

[6] D. Pai, K. van den Doel, D. James, J. Lang, J. Lloyd, J. Richmond, and S. Yau, "Scanning physical interaction behavior of 3d objects," in *ACM SIGGRAPH 2001 Conference Proc.*, Aug. 2001, pp. 87–96.

[7] M. Sedef, E. Samur, and C. Basdogan, "Visual and haptic simulation of linear viscoelastic tissue behavior based on experimental data," in *Haptic Symposium*, Mar. 2006, pp. 201– 208.

[8] E. Samur, M. Sedef, C. Basdogan, L. Avtan, and O. Duzgun, "A robotic indenter for minimally invasive characterization of soft tissues," in *Proc. of the Computer Assisted Radiology and Surgery*, vol. 1281, May 2005, pp. 713–718.

[9] I. Peterlik and L. Matyska, "Haptic interaction with soft tissues based on state-space approximation," in *EuroHaptics*, Jun. 2008, pp. 886–895.

[10] M. Mahvash and V. Hayward, "Haptic simulation of a tool in contact with a nonlinear deformable body," in *Surgical Simulation and Soft Tissue Deformation*, vol. 2673, 2003, pp. 311–320.

[11] R. Hoever, M. Harders, and G. Szekely, "Data-driven haptic rendering of visco-elastic effects," in *IEEE Haptic Symposium*, Mar. 2008, pp. 201–208.

[12] M. Shaw and W. MacKnight, *Introduction to Polymer Viscoelasticity*, 3rd ed. John Wiley & Sons, Inc., 2005.

[13] M. Buhmann, "Radial basis functions," in *Acta Numerica*, 2000, pp. 1–38.

[14] A. Iske, "Reconstruction of smooth signals from irregular samples by using radial basis function approximation," in *International Workshop on Sampling Theory and Applications*, 1999, pp. 82–87.

[15] A. Iske and J. Levesley, "Multilevel scattered data approximation by adaptive domain decomposition," in *Numerical Algorithms*, vol. 39, Jul. 2005, pp. 187–198.

[16] M. Botsch and L. Kobbelt, "Real-time shape editing using radial basis functions," in *Computer Graphics Forum*, vol. 24, 2005, pp. 611–621.

[17] X. Pang, H. Tan, and N. Durlach, "Manual discrimination of force using active finger motion," in *Perception and Psychophysics*, vol. 49, 1991, pp. 531–540.

[18] H. Pongrac, B. Faerber, P. Hinterseer, J. Kammerl, and E. Steinbach, "Limitations of human 3d force discrimination," in *Human-Centered Robotics Systems*, Oct. 2006.

[19] M. P. Vitello, M. O. Ernst, and M. Fritschi, "An instance of tactile suppression: Active exploration impairs tactile sensitivity for the direction of lateral movement," in *EuroHaptics*, 2006, pp. 351–355.

[20] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*, 3rd ed. Springer-Verlag, 2008.