Perceptually Augmented Simulator Design

Timothy Edmunds, Dinesh K. Pai

Abstract— Training simulators have proven their worth in a variety of fields, from piloting to air-traffic control to nuclear power station monitoring. Designing surgical simulators, however, poses the challenge of creating trainers that effectively instill not only high-level understanding of the steps to be taken in a given situation, but also the low-level "muscle-memory" needed to perform delicate surgical procedures. It is often impossible to build an ideal simulator that perfectly mimics the haptic experience of a surgical procedure, but by focussing on the aspects of the experience that are perceptually salient we can build simulators that effectively instill learning.

We propose a general method for the design of surgical simulators that augment the perceptually salient aspects of an interaction. Using this method, we can increase skill-transfer rates without requiring expensive improvements in the capability of the rendering hardware or the computational complexity of the simulation. In this paper, we present our decomposition-based method for surgical simulator design, and describe a user-study comparing the training effectiveness of a haptic-search-task simulator designed using our method vs. an unaugmented simulator. The results show that perception-based task decomposition can be used to improve the design of surgical simulators that effectively impart skill by targeting perceptually significant aspects of the interaction.

Index Terms—Haptic I/O; Artificial, augmented, and virtual realities; Life and Medical Sciences; Surgical simulation.

1 Introduction

Surgical procedures typically require dextrous neuro-muscular control as well as high-level cognitive awareness of the steps necessary to perform the procedure. The need to develop this low-level neuro-muscular control is the reason that repeated practice is usually necessary to learn to competently execute the surgical procedure. Since obtaining repeated opportunities to perform the real procedure is often costly or dangerous, interactive surgical simulation can be a valuable training tool. In a surgical simulator that is intended to effectively instill the required neuro-muscular control, the haptic modality is especially important, but high-fidelity haptic simulation and rendering can be one of the more prohibitively expensive aspects of a simulator design.

Whereas increasing the overall haptic fidelity of a surgical simulator may require the costly development of novel rendering hardware, and computationally expensive updating of high-precision dynamical models, the *training effectiveness* of the simulator may be improved by selectively allocating resources to certain aspects of the simulation and rendering.

In previous work [1] we demonstrated that by augmenting the rendering of perceptually salient features of the interaction that makes up a surgical procedure, the effectiveness of a haptic simulator can be improved without requiring improved hardware. We used this augmentation approach to design a simulator for a surgical task (bone-pin insertion), and showed a statistically significant performance improvement among subjects who trained on that simulator. In that work we left open the problem of how to design such an augmented simulator in the general case of surgical tasks that involve a sequence of steps, each of which may require a distinct neuro-motor control strategy. In preliminary work in this direction [2] we described a decomposition approach and user-study using a SensAble PHANToM [3] haptic device, in which we found indications that the decomposition approach could be used to build more effective haptic simulators. In this paper we describe how we modified the experimental task to use a high performance magnetically levitated haptic device [4], and conducted a more thorough user-study in which we found that the augmented simulation designed with our decomposition approach was a more effective training simulator than the basic unaugmented simulator.

As a surgeon performs a procedure, various different interaction features are encountered. For example, when placing a suture, the interaction will involve transient features such as making/breaking contact between the holder and the needle or the needle and the tissue, and changes to the tissue's structure as the needle pierces it. But the features of the interaction that are pertinent to the user depend on what aspect of the task is being performed. For the surgeon, the forces on the needle when it is being prepared for insertion may not be important, but the forces and torques applied indirectly to the needle-holder once the needle is being inserted are critical to properly

• *The authors are with the Sensorimotor Systems Lab, Department of Computer Science, University of British Columbia, Vancouver, BC. E-mail: {tedmunds, pai@cs.ubc.ca}*

10.1109/ToH.2011.42 © 2012 IEEE

guiding the needle through the tissue. The high-level task can be decomposed into subtasks that correspond to different contexts for interaction. In the suturing example these subtasks might be: manipulating the needle-holder to acquire a secure grasp with the needle in the necessary orientation; maneuvering the needle to the insertion area while keeping the thread clear; placing the needle tip at the point of insertion; piercing the tissue; guiding the needle smoothly along its naturally curved trajectory to the tissue interface; etc. The subtask being performed determines which interaction features are most perceptually pertinent and need to be effectively rendered by the simulation (e.g., when the surgeon is guiding the needle along its trajectory, effective rendering of the torques exerted by the tissue is critical).

If we can identify the subtasks that make up a surgical procedure, then we can selectively augment the features of the interaction that are perceptually pertinent for learning to perform those subtasks. This reduces the problem of holistically assessing a surgical procedure and generating appropriate augmentation to three subproblems: decomposing the overall task into subtasks; determining what augmentation is appropriate for the perceptual context of each subtask; and detecting throughout the interaction what type of subtask is being performed.

A challenge in investigating the problem of surgical simulator design is the same one that creates the need for simulators in the first place: experimentation requires a laboratory task that captures aspects of real-world surgical procedures while being easily repeatable and allowing detailed analysis of the interaction. To address this challenge, we designed an artificial haptic search task that requires some of the same fine neuro-motor control as do many surgical procedures: the subject has to scan the haptic environment to find textured surface patches; identify the surface patch with the correct texture; and precisely locate its centre. We use this task as the basis for an evaluation of our design approach.

1.1 Our Contributions

In this paper, we address the question of how surgical simulators can be designed to improve training effectiveness without requiring improvements in the capabilities of the simulator rendering hardware. Specifically, we examine the hypothesis that subjects who train on a simulator designed using our decomposition-based augmentation approach will exhibit greater performance improvement on the real task than control subjects who train on an unaugmented simulator.

1.2 Background and Approach

One approach to the problem of automatically generating augmentation for haptic simulations is to measure the haptic properties of the real task, find the differences between those properties and the properties rendered by the simulation, and augment the simulation with the "difference" between the two. Acceleration matching for impact augmentation [5] is an example of this type of approach. However, while this approach might help achieve greater fidelity with a real environment, that criterion is not always the only or even the best one for judging the effectiveness of a surgical simulation; when the goal of the simulation is to improve transfer of training, controlled *deviation* from the real dynamics can improve the simulation's effectiveness [6]. Instead of a purely fidelity-based evaluation criterion, we need to consider what augmentations will achieve the desired training effect in a surgical simulator. For example, to evaluate the effectiveness of a surgical trainer, the procedure success rate after training is more important than the amount of error in the forces rendered.

The problem of augmenting haptic simulations is complicated by the large number of different dimensions that are perceived haptically. In their early work on haptic exploration (focussing primarily on haptic identification), Klatzky and Lederman [7] identified a set of haptic dimensions that are directly sensed and aid in the haptic identification of objects. (The term *haptic dimension* [8] refers to a domain of variation that is accessible to the perceptual system (e.g., roughness of a surface being scraped); the value that a particular situation has in that dimension (e.g., very smooth) is a *haptic property*.) The dimensions that they identified are texture, hardness, temperature, weight, global volume, exact shape, part motion, and specific function. They also identified a set of *exploratory procedures* that are typically used to assess an object's value along each of these dimensions.

In the broader context of general tasks, we can define the set of *interactive procedures* as a parallel to (and superset of) the set of exploratory procedures described above. Where the goals of all exploratory procedures are to investigate and assess a haptic dimension, not all recognizable actions made while performing a surgical procedure fall into this category. Some actions are taken to have an effect on the environment in order to accomplish the goal of the procedure (e.g., making an incision in tissue). Such actions are not exploratory, but they conform to the schema that an interactive procedure is performed with a specific intent, be it to explore a specific characteristic of the environment (like material stiffness) or to change a specific property of the environment (as by cutting an object into parts).

One well studied category of interactive procedures is the exploratory procedures used in *haptic identification* which allows the interactor to identify the current haptic situation. Awareness of the haptic situation can involve such factors as identifying an object being explored with the hand [7], assessing the geometric properties of an object [9], or determining the material properties of a surface being tapped or scraped with a tool [5, 10]. Identifying the haptic situation can involve the assessment of both *object features* (such as texture, shape, and compliance) and *interaction features* (such as the onset of impact, the transition from sticking to slipping, or object-part motion). The classifying characteristic of these interactive procedures is that they all involve performing an action intended to reveal a property of the environment.

Another identifiable class of interactive procedures involve *haptic localization* — i.e., actions whose goal is to spatially locate a haptic feature. This can involve determining which finger is touching a material with a specific haptic property [11], noticing where amongst a line of distractors a haptic target lies [12], or finding where on an object a specific haptic feature has been placed [13]. Haptic localization involves both the construction of mental models of spatial location of features and the execution of exploratory motions to obtain the necessary sensory input.

Some interactive procedures involve not only the discovery of properties of the environment, but also the intent to change the environment. This class of interactive procedure is particularly pertinent in the context of surgical simulators designed to train surgeons at a particular procedure. Such procedures often require the user to explore an environment, and then modify it to conform to a goal configuration (e.g., make an incision, or insert a suture). This type of interactive procedure still involves assimilating sensory information, since accomplishing the desired action is often assisted by haptic interaction with the environment; the environment can influence the trajectory of a planned motion, and the interactor can, in turn, make use of this influence to achieve a desired trajectory. This haptic guidance of the user by the environment is critical in performing many tasks, for example inserting a catheter into a vessel [14] or inserting a peg into a hole [15].

A given task may involve the exercise of any combination of interactive procedures, either simultaneously or sequentially, in separate subtasks. By focussing on the set of interactive procedures used in performing a task, we can guide the automatic generation of augmentation to improve the performance of the task.

As well as determining which subtasks are performed in executing a task, we need to consider what *augmentations* are appropriate to improve the skill transfer for the subtasks. We can describe the domain of possible haptic augmentation as the addition of haptic features to the environment or the modification of features already present. For example, a smooth surface could be made rough, a shallow groove could be made deeper, or a light object could be made heavier. Simulated haptic environments allow for a broader range of features than real environments. Force pulses can be applied in response to certain events (e.g., braking pulses applied in response to collision events [16, 17, 18]). Surfaces can be made to vibrate on command. Features like grooves can be made to move or disappear.

If we identify what sensory percepts are being generated in performing a subtask, we can attempt to generate augmentation that will target those percepts (for example exaggerating the roughness of a surface to make it easier to identify). By taking this approach, we decompose the problem of generating the augmentation into two problems: determining which interactive procedures are used in performing the subtask, and choosing haptic augmentations that improve the transfer of training in those procedures. The first of these problems can in many cases be solved wholly or in part by high level domain knowledge of the task, particularly with a focus on recognizing known interactive procedures. To simplify the second problem, we can leverage existing psychophysical findings that illuminate how different haptic dimensions are perceived.

By measuring response time in a target/distractor surface identification search task, experimenters found that haptic dimensions that are discernible without a spatial reference frame (termed *intensive* dimensions) are available for processing earlier in the neural pathway than those that require spatial encoding [8]. Hence, variation in a surface's magnitude of roughness might make a better cue than changes in the orientation of an anisotropic surface (like a grating) for training the haptic identification capability. For haptic localization, there is an inherent need for a spatial reference frame. Even when the space across which localization takes place is the set of fixed fingertip sites, rather than a continuous 2- or 3-D domain, there is a processing cost incurred in determining the location of a particular haptic property [11]. Thus, for a localization task, a cue that contributes to the assembly of a spatial reference frame should be more effective than one that does not. For example, if the task is to locate a small haptic feature that lies at the centre of a circle, then an inherently spatial feature (such as a groove around the periphery of the circle) is more helpful than an intensive feature (such as a circular patch of uniform roughness).

The remainder of this paper is organized as follows. In Section 2 we describe the haptic search task that we developed to facilitate a concrete investigation of our approach to simulator design. We give the details of how we applied our decomposition-based approach to generate an augmented training simulator for the haptic search task in Section 3. In Section 4, we describe the user study we conducted to compare the training effectiveness of our augmented simulator to that of an unaugmented simulator. We analyzed the results of that user study in terms of different performance metrics — these metrics and the analysis are described in Section 4.1. In Section 5, we discuss the implications of our findings in the context of surgical simulation, including specific guidelines for how our proposed decomposition-based approach can be used in the design of perceptually-augmented haptic training simulators.

2 Haptic Search Task

To allow for a concrete investigation of the design of virtual simulators for real tasks, we need to test our approach on a specific task. We created a haptic search task that parallels a visual search task used in psychophysical experiments: the subject attempts to locate a target stimulus that is presented in the company of distractor stimuli that have similar (but distinguishable) characteristics [19]. In our search task, the subject must: haptically scan the environment to search for a target (or distractor) texture patch; discriminate between the target and distractors based on texture properties; and finally locate the precise centre of the target patch.

This synthetic search task is useful in the laboratory setting because it captures key aspects of real surgical tasks (such as including a sequence of actions that must be performed to allow later phases of the task to be completed) while allowing detailed recording of the subject's interaction to support rigorous analysis of the task performance along multiple dimensions. This task is also easily repeatable for user studies because it can be implemented entirely in a virtual environment.

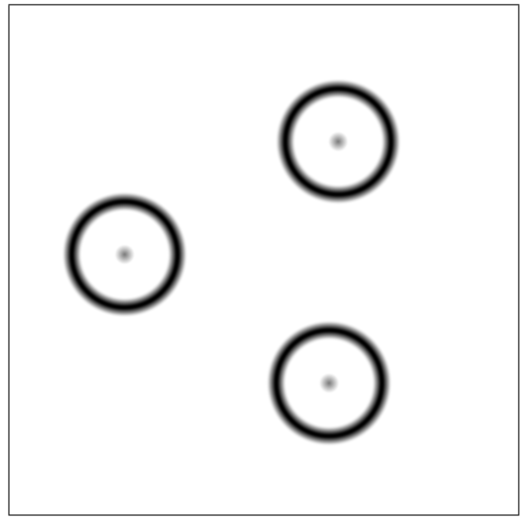
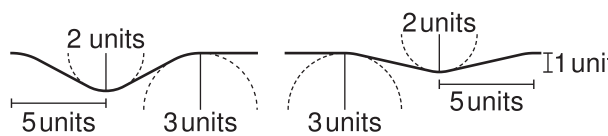


Fig. 1. Example stimulus height field (without texture). The height of the surface ranges from the base floor level (white) to 1 unit below that (black).



(a) Groove Cross-Section (b) Pit Cross-Section

Fig. 2. Environment geometry profiles. (a) The cross-section of the groove surrounding each scene element. (b) The cross-section of the pit at the centre of each scene element.

2.1 Stimulus Design

In order to apply and evaluate our decomposition approach to developing an augmented training simulator for the haptic search task, we created a virtual environment implementation of the search stimulus. Since the virtual environment is used as the "real-world" reference implementation of the haptic search task, it should be rendered to feel as much like a rigid surface as possible. For this experiment, we implemented the task on a 6-DOF magnetic levitation haptic device (MLHD) [4]. The performance characteristics of this device allow for a realistic reference implementation of the haptic search task, while the *simulators* used to investigate our approach use artificially degraded capabilities to emulate the common design constraint that affordable/feasible hardware cannot simulate a real surgical procedure with complete fidelity.

The environment for the haptic search task consists of a 3-D workspace with smooth flat walls around four sides of a square floor whose height is varied to create the target and distractor stimuli. The environment is rendered at different scales on the haptic and graphical devices; the units used below to describe the geometry correspond to 0.06 mm in the haptic rendering and to 1 mm in the graphical rendering. The square floor is defined to be 240 units by 240 units.

The floor is a height field that is uniformly zero everywhere outside a target or distractor (a *scene element*) — an example stimulus height field is shown in Fig. 1. Each scene element consists of a groove surrounding a flat circular patch of roughly textured surface. At the centre of the patch, there is a small untextured pit. See Fig. 2 for a visualization of the cross-sectional geometry of the scene elements. The textured patch has a radius of 25 units, and the surrounding groove is 10 units wide (its maximum depth of 2 units is at a radius of 30 units). The groove's cross-section is smooth, with its bottom being a segment of a circle (of radius 2 units), and each lip being segments of circles (of radius 3 units). The pit at the centre is similarly smooth, with a total radius of 5 units and a maximum depth of 1 unit at the centre, where the radial cross-section is a segment of a circle (of radius 2 units); likewise, the cross-section of the lip is a segment of a circle of radius 3 units.

The only difference between targets and distractors is the texture of the patch; the texture is manifested as set of hemispherical bumps scattered over the

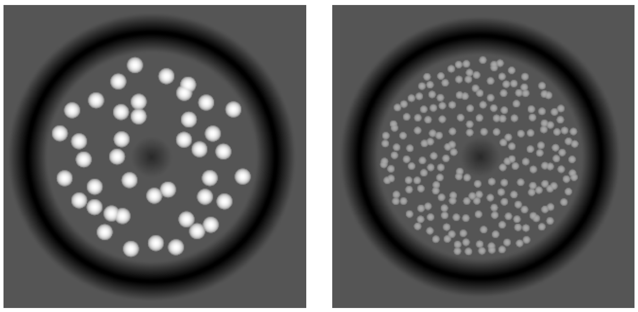
TABLE 1
Parameters for the functions that generate the (a) target textures and (b) the distractor textures. (See example textures generated with these parameters in Fig. 3a and Fig. 3b, respectively.)

Target	
Bump Radius	2 units
Grid Spacing	7 units
Jitter Square	5.6 units

(a)

Distractor	
Bump Radius	1 units
Grid Spacing	3 units
Jitter Square	2.4 units

(b)



(a) Target (b) Distractor

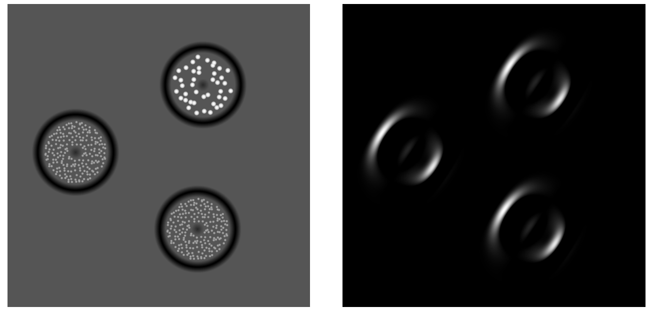
Fig. 3. Example (a) target and (b) distractor texture patches as height maps.

patch's surface. The bump pattern is generated by arranging the bumps on a grid and then pseudo-randomly jittering their positions within a square centred on their original positions. The target and distractor bump patterns were generated according to the parameters given in Table 1; the generated bump-patterns (shown in Fig. 3) were fixed and used as the target and distractor textures for all the targets and distractors in any instance of the haptic search task.

The full stimulus for one episode of the search task consists of two distractors and one target — see Fig. 4a for an example of the stimulus haptic environment. In a visual search task, scene elements are often (though not always) scattered about the visual field, rather than according to some fixed pattern. In those experiments, however, the viewer's peripheral vision is being used to locate scene elements in order to make saccades to inspect them foveally. In the case of haptic search, there is no source of peripheral information regarding the location of scene elements, so we provide a fixed structure for their placement. The scene elements are equally spaced around a circle of radius $\frac{200}{3}$ units; the only variation between episodes is the orientation of the triangle described by the three scene elements.

2.2 Stimulus Interaction

To realize a repeatable haptic search task, we implemented a virtual environment that allows the subject



(a) Example stimulus (b) Processed Height Field

Fig. 4. (a) A height map of a stimulus environment presented to the user. (b) The result of convolving the height field from (a) (excluding the texture) with the Gabor filter shown in Fig. 5. This image is one slice of a 32×5×1024×1024 lookup table for the scan augmentation.

to interact with the stimulus through the MLHD. This device has a relatively small workspace ($\sim$11 mm radius), so the task environment is scaled to fit entirely inside the device's usable workspace; this works out to a scale of 24 units per mm. Interaction with the environment is simulated as the interaction between a virtual probe-point attached to the device handle and a constraint surface consisting of the environment's walls and floor. The device's output force is determined by a (PD) controller; the controller's target position is updated by a 1 kHz servo loop that computes the closest position to the actual device position that satisfies the constraint-surface. The magnetic levitation device has negligable static friction, and no simulated friction or other damping (beyond the inherent damping introduced by the inertia of the device handle) is applied. In addition to the PD forces, a feed-forward force is applied to reduce the effect of gravity on the device handle. Although the MLHD is capable of 6-DOF movement and force-output, the rotational axes are locked (using high-gain PD forces) to constrain the handle's motion to 3-DOF. The proportional gains for the positional axes are set to 5250 $\frac{\mathrm{N}}{\mathrm{m}}$ (i.e., 126 N per unit); the derivative gains are 20 $\frac{\mathrm{N}}{\mathrm{m \cdot s}}$ (0.48 N per unit per s).

Each episode of the search task begins with the device handle held in place (by high PD gains) in the centre of the workspace, 20 units above the surface. Once the episode begins, the spring force is released, and the subject is free to explore the environment. The goal of the task is for the subject to locate the centre of the target scene element and hold the probe-point there for 0.5 s.

2.2.1 *Visual Stimulus*

In addition to the haptic feedback described above, the subject is presented with a few visual cues: the outline of the boundaries of the workspace, and the position of the virtual probe-point. This sparse visual rendering of the environment is displayed on a vertical screen in front of the subject, at a much larger scale than the real ($\sim$11 mm) haptic workspace. This allows the subject to use visual cues to construct a spatial representation of the location of haptic features (which are not displayed) as they are felt.

3 Augmented Simulator Design

Having defined the stimulus for our haptic search task, we created a basic training simulator that is simply the same rendering algorithm as the real task but with artificially degraded stiffness (proportional/derivative gains of 1050 $\frac{\mathrm{N}}{\mathrm{m}}$/4 $\frac{\mathrm{N}}{\mathrm{m \cdot s}}$ respectively); this corresponds to the general design condition in which the rendering hardware cannot trivially reproduce a real interaction with high fidelity. We then applied the general decomposition approach described in Section 1.2 to develop an augmented "surgical" simulator for this specific haptic search task. In our approach to automatic simulation augmentation, a complete task is decomposed into subtasks for which different augmentation is applied in accordance with the perceptual features involved in executing the subtask.

3.1 Task Decomposition

For most real-world tasks, domain knowledge will be critical to successfully decompose the task into appropriate subtasks (e.g., identifying the sequence of low-level actions used to perform a surgical procedure). However, as discussed in Section 1.2, the decomposition can also be assisted by focussing on interactive procedures that are known to be performed in a somewhat atomic manner. For our haptic search task, we were able to use this assistance to create a subtask decomposition with limited domain knowledge.

The first subtask that the subject must execute is to locate a scene element; we call this the *scan* subtask. In this subtask, the subject typically scans the surface with large scale, high-speed motions, until he or she detects the high-temporal-frequency force discontinuity event that signals that the probe-point has encountered (the rising slope of) a groove around a scene element.

The second subtask is assessing the shape (and thus the extent of the texture patch) of the scene element. In the *shape assessment* subtask, the subject traces part or all of the groove around the scene element to generate a spatial representation of where the texture patch (and its centre) lies.

The other subtask the subject performs is the *identification* subtask; having located a scene element, the subject must explore it (with a scrubbing interactive procedure) to gauge the roughness of the surface in order to identify the scene element as a distractor or target.

Although there is only a small set of subtasks in this decomposition, a single execution of the overall task can include multiple instances of each subtask in different orders. For example, while the scan subtask is by necessity the first subtask performed, it may be performed again if the user decides to examine a second scene element after performing the shape assessment and identification subtasks on the first scene element. Likewise, after performing the identification subtask and deciding that the scene element is the target, the subject will likely perform the shape assessment subtask again (or for the first time if the subject went immediately from scanning to identification) to strengthen the spatial awareness of the shape of the scene element in order to locate its centre.

3.2 Subtask Augmentation

Having identified the different subtasks that make up our haptic search task, we need to assign augmentations for each subtask.

3.2.1 *Scan Augmentation*

In the **scan** subtask, the pertinent perceptual features of the interaction are the force discontinuities experienced when the probe-point passes over areas of changing height. However, since the scanning interactive procedure is performed with relatively high frequency motions (reducing the spatial accuracy of the proprioceptive system), the temporal coherence of these events is of greater significance than the precise spatial alignment of the force discontinuity with the surface feature. Since the simulator has low stiffness haptic feedback, these high-temporal-frequency force discontinuities are lost. These perceptual features can be restored through the use of open-loop augmentation generated by automated techniques similar to those used in computer vision.

A common technique for processing images to extract or highlight pertinent features is to convolve the image with a filter (either in the spatial domain or in the frequency domain). An example of this is edge detection by convolution with gradient-approximating kernels. Since we want to identify places in the environment where force discontinuities are experienced during scanning, our problem is similar to that of edge detection. Rather than edge detection in the 2-dimensional (x,y) space though, we are performing edge detection in the 4-dimensional (x,y,v_x,v_y) space of the interaction between the height of the surface at (x, y) and the velocity of the probe-point.

We can think of the height map of the surface as an image whose edges we want to find, where for a particular probe-point velocity we are only interested in edges of a certain orientation and spatial frequency (i.e., at higher speeds, we want to detect edges

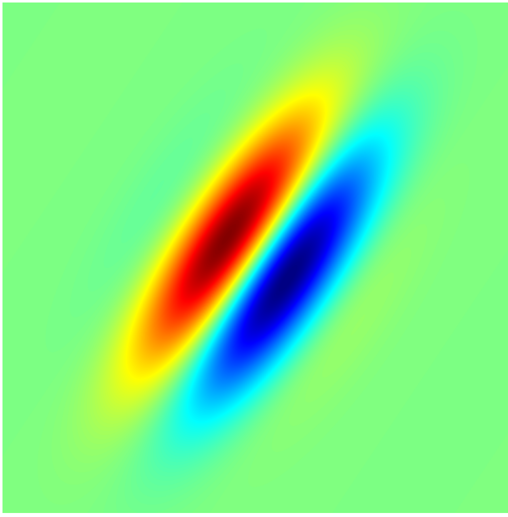


Fig. 5. One example of the 2-dimensional anti-symmetric Gabor filter used to pre-process the height maps of the stimuli. This filter (shown in false-colour) corresponds to $\sigma = 2^4$ pixels, $\theta = 0.589$ radians, and $\gamma = 0.5$.

with lower spatial frequency). The 2-dimensional anti-symmetric Gabor filter is a widely used convolution kernel for oriented edge detection at configurable spatial resolution:

$$g(x, y, \lambda, \theta, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi \frac{x'}{\lambda} + \frac{\pi}{2}\right) \quad (1)$$

$$x' = x \cos\theta + y \sin\theta \quad (2)$$

$$y' = -x \sin\theta + y \cos\theta \quad (3)$$

Here λ is the wavelength of the cosine factor, θ is the orientation of the filter (direction perpendicular to the parallel stripes), γ is the aspect ratio of the filter, and σ is the standard deviation of the Gaussian envelope that (together with λ) determines the spatial resolution of the filter. See Fig. 5 for an example of the type of filter used.

By pre-computing the convolution results of the surface's height map with Gabor filters of various orientations and spatial resolutions, we can create a 4-dimensional lookup table that indicates which surface locations (at a given probe-point velocity) should trigger a haptic pulse to signal an edge-crossing.

For the augmented training simulator, we pre-computed the convolution of each stimulus with Gabor filters at 32 different (equally spaced) orientations and 5 different scales ($\sigma = 2^0, 2^1, \ldots, 2^4$, $\lambda = 4\sigma$, where σ is in units of pixels, and the height map of the stimulus is represented as a 1024x1024 image — see Fig. 4b). The aspect ratio of the filter (γ) was uniformly 0.5. During the scan subtask, the probe-point location and velocity are used as indices into a lookup table formed by all 160 pre-processed images for the current stimulus; if the lookup value exceeds a threshold, an open-loop fixed-width force pulse is initiated (upwards).

3.2.2 Shape Assessment Augmentation

In the **shape assessment** subtask, the subject follows the groove around a scene element to determine the spatial extent of the element (and the location of its centre). This is an example of an interactive procedure that uses the environment to constrain and guide the exploratory motion (as the walls of the groove create an anisotropic resistance to motion that channels the probe-point tip longitudinally along the groove). Since this interactive procedure leverages the curvature of the surface (which produces the constraints on motion), we augment the simulation for this subtask by applying local force-fields based on surface curvature.

The motion constraints imposed by curved surfaces channel motion towards points (or paths) that are local minima of surface curvature (i.e., points of maximum concavity). By constructing force fields that attract the probe-point toward these loci of minimal curvature, the guidance used by the shape-exploration procedure can be replicated in the low-stiffness simulator.

Provided the force field is sufficiently smooth, it will not introduce the instability that results when a simulation's stiffness exceeds the capabilities of the rendering device. For example, a notch-shaped force field (where the magnitude of the force increases linearly with distance from the locus) is likely to create instability near the locus, but a cosine function (having zero slope at the locus) will be more stable. (Note that the smoothness requirement could be quantified by applying a rigorous analysis analagous to that used for force fields arising from surface textures [20].)

In order to simplify matters computationally (and to match the local effect of curvature-induced constraints), we want force-fields that have bounded extent. We choose a single-cycle cosine function:

$$\mathbf{f}_{shape} = \begin{cases} f_{max}\left(1 - \cos(2\pi \frac{d}{d_{max}})\right) \frac{1}{r_{curv}} \hat{\mathbf{n}} & \text{if } d \le d_{max} \\ 0 & \text{if } d > d_{max} \end{cases} \quad (4)$$

where f_{max} is a parameter controlling the overall scale of the augmentation force (we used f_{max} = 10 N), d is the distance to the nearest local minimum of curvature, d_{max} is the distance threshold imposed to make the force-fields local in extent (we used d_{max} = 5 units), $\hat{\mathbf{n}}$ is a unit vector towards the attracting point, and r_{curv} is the radius of curvature (along the principal direction in which the attracting point is a local minimum of curvature).

3.2.3 Identification Augmentation

In the **identification** subtask, the subject uses the lateral motion interactive procedure to assess the roughness of the surface. Klatzky and Lederman [21] found that when perceiving roughness through a probe (as opposed to perceiving roughness from direct skin contact), humans are able to achieve some measure of speed constancy in their perception of the vibratory phenomena induced by surface roughness (i.e., roughness is judged not by vibratory frequency alone, but by speed-normalized vibratory frequency).

Since the subject's perception of the surface roughness is affected by the speed of the subject-controlled motion, it is insufficient to simply augment the identification subtask by applying open-loop vibration at a fixed frequency. Instead, we wish to produce vibratory effects that mimic those of high-stiffness texture interaction, independent of speed. To achieve this, we can work in the speed-independent space of the original texture.

At the level of stiffness used in our simulator, the small bumps on the constraint surface that make up the texture are barely perceptible when the device is handled with typical user-compliance levels. However, we can use the same set of texture-feature positions to generate open-loop augmentation forces. When the probe-point is in contact with the constraint surface we check whether it is inside the footprint of one of the texture bumps; if so, an open-loop vertical pulse is generated. Although this augmentation only generates vertical forces, the sequence of pulses generated as the probe-point is moved over the texture corresponds spatio-temporally to the vibratory motion experienced when interacting with the actual bumps.

3.3 Subtask Identification

In our approach to automatic augmentation of interactive simulation, we call for the use of domain knowledge to assist in identifying the subtasks performed in the course of completing a high-level task, and the use of pre-selected augmentations that help develop skill at the interactive procedure used in each subtask. In this decomposition scheme, the remaining problem is to identify during the interaction which subtask is being performed, and which augmentation(s) should thus be active.

Here we once again leverage the coupling between subtask and interactive procedure; since different procedures are used to accomplish different subtasks, we can identify the subtask that the subject is attempting to perform by identifying the interactive procedure being used. In our case, we can distinguish between the scanning, tracing, and scrubbing procedures used respectively in the scan, shape assessment, and identification subtasks, based solely on position in the environment and velocity thresholds.

Since scanning is a relatively high-speed motion, the scan augmentation is only activated if the probe-point speed is at least 468.75 units/s — chosen because at the simulator's 1 kHz this speed corresponds to the spatial frequency of the finest Gabor filter used to preprocess the height map.

TABLE 2
The simulations used by the subjects in each of the blocks of the experiment.

Block	Group 1	Group 2
Baseline	Full Stiffness	Full Stiffness
Training	Low Stiffness	Augmented Low Stiffness
Evaluation	Full Stiffness	Full Stiffness

Since the shape assessment procedure is executed using finer-controlled (slower) actions than the scan procedure, the shape assessment augmentation (the local force fields around local minima of curvature) is activated when the probe-point speed is below the scanning augmentation threshold. Of course, since the force fields are local in extent and are located around the curvature minima, the shape assessment augmentation is only truly active when the subject is exploring pertinent surface geometry.

The identification subtask is characterized by the use of lateral motion to investigate an area of surface texture. The identification augmentation is thus only activated when the probe-point is in contact with a textured surface, and only when the probe-point is "moving laterally." In the context of a discrete time-step rendering loop this lateral motion criterion is deemed to be satisfied when the probe-point's tangential speed is at least 400 units/s (a threshold selected by simple experimental tuning).

4 User Study

To test the effectiveness of the augmented simulator design generated by our approach, we conducted a user study comparing the augmented simulator against the basic unaugmented simulator. 24 subjects were included in the experiment. 23 subjects used the haptic device with their right hands; 1 subject used the haptic device with his left hand. 3 of the subjects did not complete the experiment and are excluded from the results.

A subject's participation consisted of three blocks of trials taking place in two sessions on different days (see Table 2). In the first session, each subject was familiarized with the capabilities of the MLHD and the task to be performed; the subject then performed a baseline block of 60 episodes of the search task (in 3 sets of 20 trials, punctuated by rest-breaks).

In the second session, each subject performed 40 training episodes (2 sets of 20) on one of two simulations of the search task (with subjects randomly assigned to a simulation group), followed by 60 episodes (3 sets of 20) of evaluation on the "real" search task.

The first simulation was simply a degraded version of the rendering of the real search task (lower PD

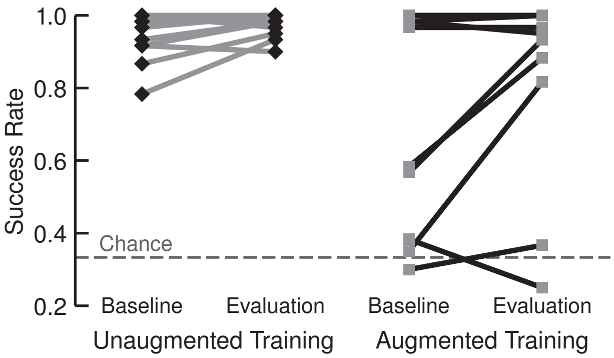


Fig. 6. Baseline success rate and evaluation success rate are plotted for each subject. On the left is the group of subjects that trained on the unaugmented simulation. On the right is the group of subjects that trained on the augmented simulation.

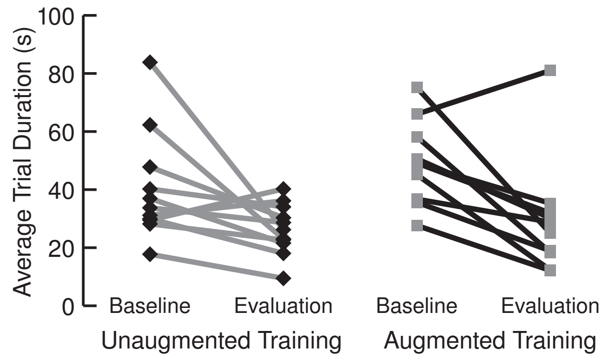


Fig. 7. Baseline average trial duration and evaluation average trial duration are plotted for each subject. On the left is the group of subjects that trained on the unaugmented simulation. On the right is the group of subjects that trained on the augmented simulation.

gains). The second simulation was an augmented version of the degraded simulation. The low gains were still used for the PD controller, but feed-forward augmentation was applied according to the subtask being performed.

4.1 Results and Analysis

The first part of our analysis of the user-study results looks at the aggregate performance of all the subjects in each of the two subject groups. As is the case with many real surgical procedures, our task had both an explicit measure of success (correct discrimination of target from distractor) and an implicit measure (time taken to perform the task). To quantitatively assess the effect of our proposed augmentation technique on the simulator's training effectiveness, we evaluated subject improvement on each of these two overall success metrics.

For each subject, we measured separately the rate of successful task execution (i.e., finishing the trial by selecting the target scene element rather than one of the distractors) before and after simulator training:

$$\text{success rate} = \frac{\text{successful executions}}{\text{total executions}} \tag{5}$$

$$0 \leq \text{success rate} \leq 1 \tag{6}$$

Fig. 6 shows the success rates of each subject, grouped by whether the subject was trained on the augmented or unaugmented training simulation. Coincidentally, all five of the subjects whose baseline success rates were closer to chance than to complete success were randomly assigned to the augmented training group. To gauge the difference in training effectiveness between the two simulators, we compared the success rate before and after simulator training to determine the subjects' absolute improvements (i.e., the slopes of the lines in Fig. 6):

$$\text{improvement} = \text{success rate}_{after} - \text{success rate}_{before} \tag{7}$$

$$-1 \leq \text{improvement} \leq 1 \tag{8}$$

We performed an ANOVA on the success-rate improvement with the type of training simulator as the single factor; although the group that trained on the augmented simulator had a higher average improvement (0.11 vs. 0.05), the inter-group difference in improvement at successful discrimination alone was not statistically significant ($F(1, 19) = 1.41, p = 0.25$).

A similar analysis was applied to the other overall-success metric. Fig. 7 shows the average trial duration of each subject before and after training on a simulator. An ANOVA on the average relative trial-duration decrease yielded $F(1, 19) = 0.03, p = 0.87$.

The results described above illustrate a common difficulty in assessing the effectiveness of training simulators for surgical procedures: the two measures of success tend to oppose each other, in that working slowly is likely to increase the correctness of the procedure performance at the expense of timely completion. While a training simulator would ideally generate improvement in both metrics, each individual subject's balancing of the correctness-speed tradeoff means that isolated analysis of each of the metrics may not capture the complete effect of the training simulator. In order to analyze overall improvement, we normalized the two metrics by scaling each subject's success rate improvement with respect to the standard deviation of the success rate improvements of all the subjects; we likewise normalized each subject's average relative trial-duration decrease. When visualized with respect to both metrics simultaneously (see Fig. 8), the inter-subject variation that was concealed by the correctness-speed tradeoff is more clearly revealed. When we compare the *summed* improvement of each subject on *both* of the two metrics, the difference between the two subject groups is more striking than for either metric alone. An ANOVA comparing the two groups of subjects in terms of their summed improvements yields: $F(1, 19) = 3.14, p =$

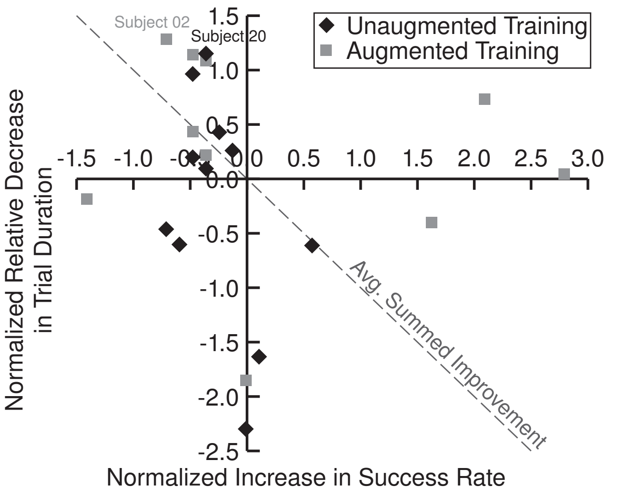


Fig. 8. The normalized improvement in both of the overall success metrics for each subject. The subjects that trained on the unaugmented simulation are plotted as black diamonds. The subjects that trained on the unaugmented simulation are plotted as grey squares.

0.09. The distribution of the datapoints in Fig. 8 suggests that subjects may have been making a binary choice: to focus their post-training efforts on either accurate target discrimination or on speedy trial completion. An ANOVA comparing the *maximum* of the two normalized improvements for the two groups yields: $F(1, 19) = 5.80, p = 0.026$. We can interpret the difference between the two groups by noting that the subjects who trained on the augmented simulator tended to show significantly greater than average improvement in at least one of the measures of success.

4.1.1 Learning

The analysis described above examined the coarse effects of the training simulators (average baseline performance vs. average evaluation performance); we can see other training effects by inspecting single-subject performance over the course of the entire experiment.

Although we labelled the three blocks of our experiment "Baseline," "Training," and "Evaluation," it would be unreasonable to expect that subject performance is constant within each block, or that learning only occurs during the training block. By looking at subject performance on a per-trial basis, we would expect to see performance generally improve as the subject becomes more familiar with the task and adept at performing it correctly and quickly. This type of trend is found in many subjects for the trial-duration metric, but the quantum nature of the correctness metric (correct target selection or incorrect distractor selection) leads to less of an observably continuous progression. The more continuous nature of the trial-duration progression allows us to contrast the effect of training on the unaugmented simulator vs. the augmented simulator.

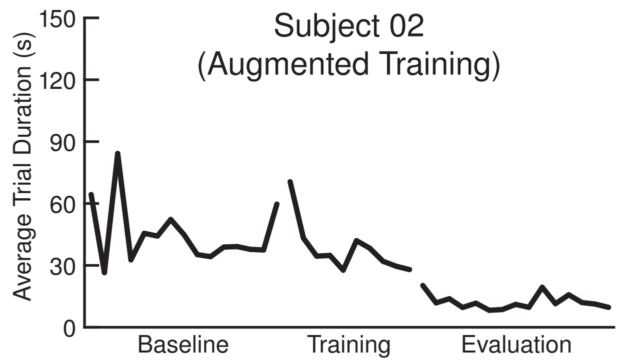


Fig. 9. The trial durations for a single subject who trained on the augmented simulator. Each plotted point is the average of 4 consecutive trials.

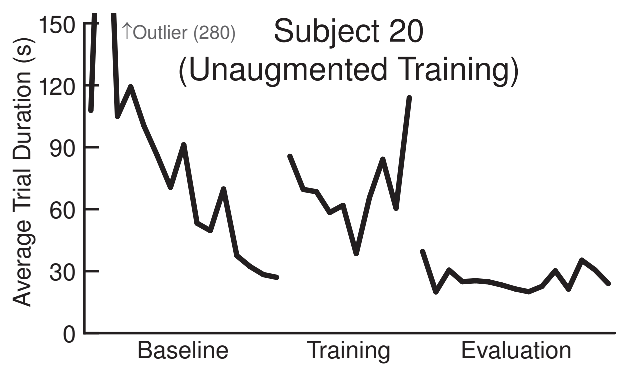


Fig. 10. The trial durations for a single subject who trained on the unaugmented simulator. Each plotted point is the average of 4 consecutive trials.

In the majority of the subjects who trained on the augmented simulator we found a similar trend, exemplified by the subject shown in Fig. 9: the trial duration trended generally downward over the course of the experiment, *including the training block*. While there is a temporary increase in the trial durations at the beginning of the training block (where the subject is first encountering the different feel of the simulator), the downward trend carries through to the evaluation block. Note that there is a considerable discontinutity in performance between the end of the baseline block and the beginning of the evaluation block; this jump verifies that learning was taking place over the course of the simulator training.

In contrast, none of the subjects who trained on the unaugmented simulator displayed this characteristic performance jump between the end of the baseline block and the beginning of the evaluation block. A more typical trend for the unaugmented simulator subjects is shown in Fig. 10; trial durations are consistently longer throughout the training block, and then revert in the evaluation block to no better than at the end of the baseline block. When considering only the baseline and evaluation blocks, the continuity of performance before and after the training block is striking. It seems that training on the unaugmented simulator has little efficacy in speeding up the task performance — more significant improvement is seen over the course of the baseline block.

4.2 Conclusions

In our user-study, we examined two metrics of procedure success (neither of which was explicitly emphasized to the subjects), and found that while the subjects who trained on the unaugmented and augmented simulators had similar average improvements on each metric, when the two metrics were considered in combination, a clearer difference between the two groups emerged. Examining the task-completion duration across the entire experiment for individual subjects gave further qualitative evidence that the augmented simulator designed according to the proposed decomposition approach provided more effective training than the unaugmented simulator.

5 Implications for Surgical Simulator Design

The results of our user study indicate that our decomposition-based augmentation approach enhances the training effectiveness of a low-fidelity training simulation for the haptic search task, but what are the implications for more realistic surgical tasks? Although the haptic search task is not a surgical procedure, it was selected as the laboratory task because of properties that support the generalization of results to the broader class of surgical tasks: the haptic search task requires the subject to integrate a high-level understanding of the steps needed to successfully complete the procedure with a low-level sensorimotor strategy that can exert the necessary fine control over the tool's interaction with the environment.

Our previous work [1] examined the training effectiveness of an augmented simulator on the surgical procedure of bone-pin insertion, but used an augmentation strategy built around a single event (the sudden resistance change when transitioning from hard cortical bone to the spongey cancellous). In this work we have addressed the integrated view of how decomposition of a task into subtasks with different augmentations can improve training simulations for a wider variety of surgical procedures.

We can extract some specific guidelines for surgical-simulator design from our results. The first step of incorporating our proposed method of augmentation into a simulator design is the identification of the subtasks that make up the surgical procedure. The decomposition will likely leverage the expert domain knowledge of experienced surgeons, but it can also be guided by analyzing the task in terms of the sequence of interactive procedures used (such as the exploratory procedures identified by Klatzky and Lederman [7]). This approach to the task decomposition simplifies the next steps of designing a method to identify the relevant subtask at run-time and selecting augmentations for each subtask.

Structuring the decomposition so that different subtasks correspond to different interactive procedures makes it possible to identify the subtask being performed by analyzing the action being performed. The interactive procedure used may be temporally complex (e.g., the back-and-forth scraping in roughness examination), but identifying the current procedure may be simpler (particularly when the expected set of procedures are known). For instance, a simple velocity threshold can be effective for identifying the scraping procedure to indicate that the user is performing a roughness assessment subtask. This same process of identifying the current subtask by determining what interactive procedure is being performed could be used to assist in the original decomposition by automated analysis of recordings of expert executions of the surgical procedure. By comparing position/velocity/force profiles from instrumented recordings to a library of known interactive procedures, an automated process could propose candidate decompositions of the overall surgical procedure.

Since the goal of a surgical simulator is to improve performance of the surgical procedure, it is important to note that the augmentation chosen for a particular subtask should not necessarily aim to reproduce the real-world interaction experience, even for the targeted interactive procedure. Rather, the purpose of the augmentation is to reproduce or enhance the aspects of the interaction that are perceptually pertinent for successful performance of the subtask. For example, in the case of texture comparison for the purpose of surface identification, the application of vertical pulses at surface micro-geometry (rather than accurate horizontal forces corresponding to the surface orientation of the micro-geometry) reproduces the spatially-linked vibratory phenomenon encountered during the back and forth scraping interactive procedure used to assess roughness. However, since horizontal forces are not applied, it does not actually fully mimic the experience of the interaction. For interactive procedures that target the inspection of micro-geometry (such as slow dragging), we expect that this augmentation would be ineffective. As well as a library mapping position/velocity/force profiles to interactive procedures that we proposed above for use in automatically decomposing a surgical procedure based on expert recordings, it would be useful to build a library that maps from an interactive procedures to a low-level descriptions of the perceptual features that contribute to successful accomplishment of that interactive procedure's purpose.

Acknowledgments

This work was supported in part by the Canada Research Chairs Program, the Human Frontier Science Program, NSERC, Peter Wall Institute for Advanced Studies, CFI, and BCKDF.

References

[1] T. Edmunds and D. K. Pai, "Perceptual rendering for learning haptic skills," in *HAPTICS '08*, 2008, pp. 225–230. 66, 76

[2] ——, "Perceptually augmented simulator design through decomposition," in *WHC '09*, 2009, pp. 505–510. 66

[3] *PHANTOM Premium 1.0*, SensAble Technologies, Inc., Woburn, MA, USA. 66

[4] *Maglev 200*, Butterfly Haptics, LLC, Pittsburgh, PA, USA. 66, 69

[5] K. J. Kuchenbecker, J. Fiene, and G. Niemeyer, "Event-based haptics and acceleration matching: Portraying and assessing the realism of contact," in *WHC '05*, 2005, pp. 381–387. 67, 68

[6] D. C. Wightman and G. Lintern, "Part-task training for tracking and manual control," *Human Factors*, vol. 27, no. 3, pp. 267–283, Jun. 1985. 67

[7] R. L. Klatzky and S. Lederman, "Intelligent exploration by the human hand," in *Dextrous Robot Hands*, S. T. Venkataraman and T. Iberall, Eds. Springer-Verlag New York, Inc., 1990, pp. 66–81. 67, 76

[8] S. J. Lederman and R. L. Klatzky, "Relative availability of surface and object properties during early haptic processing," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 23, no. 6, pp. 1680–1707, 1997. 67, 68

[9] G. Robles-De-La-Torre and V. Hayward, "Force can overcome object geometry in the perception of shape through active touch," *Nature*, vol. 412, pp. 445–448, 2001. 67

[10] D. K. Pai, K. van den Doel, D. L. James, J. Lang, J. E. Lloyd, J. L. Richmond, and S. H. Yau, "Scanning physical interaction behavior of 3D objects," in *SIGGRAPH '01*, 2001, pp. 87–96. 68

[11] K. A. Purdy, S. J. Lederman, and R. L. Klatzky, "Haptic processing of the location of a known property: Does knowing what you've touched tell you where it is?" *Canadian Journal of Experimental Psychology*, vol. 58, no. 1, pp. 32–45, 2004. 68

[12] K. E. Overvliet, J. B. J. Smeets, and E. Brenner, "Haptic search with finger movements: using more fingers does not necessarily reduce search times," *Experimental Brain Research*, vol. 182, no. 3, pp. 427–434, Sep. 2007. 68

[13] A. Lecreuse and D. M. Fragaszy, "Hand preferences for a haptic searching task by tufted capuchins (*Cebus apella*)," *International Journal of Primatology*, vol. 17, no. 4, pp. 613–632, Aug. 1996. 68

[14] E. Gobbetti, M. Tuveri, G. Zanetti, and A. Zorcolo, "Catheter insertion simulation with co-registered direct volume rendering and haptic feedback," in *MMVR 2000: Medicine Meets Virtual Reality*, 2000, pp. 96–98. 68

[15] B. J. Unger, A. Nicolaidis, P. J. Berkelman, A. Thompson, R. L. Klatzky, and R. L. Hollis, "Comparison of 3-D haptic peg-in-hole tasks in real and virtual environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001, pp. 1751–1756. 68

[16] S. E. Salcudean and T. D. Vlaar, "On the emulation of stiff walls and static friction with a magnetically levitated input/output device," *ASME Journal of Dynamic Systems, Measurement and Control*, vol. 119, no. 1, pp. 127–132, 1997. 68

[17] D. Constantinescu, S. E. Salcudean, and E. A. Croft, "Haptic rendering of rigid contacts using impulsive and penalty forces," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 309–323, Jun. 2005. 68

[18] J. D. Hwang, M. D. Williams, and G. Niemeyer, "Toward event-based haptics: Rendering contact using open-loop force pulses," in *HAPTICS '04*, 2004, pp. 24–31. 68

[19] C. A. Baker, D. F. Morris, and W. C. Steedman, "Target recognition on complex displays," *Human Factors*, vol. 2, pp. 51–61, 1960. 69

[20] G. Campion and V. Hayward, "On the synthesis of haptic textures," *IEEE Transactions on Robotics*, vol. 24, pp. 527–536, Jun. 2008. 72

[21] R. Klatzky and S. Lederman, "Perceiving texture through a probe," in *Touch in virtual environments*, M. L. McLaughlin, J. P. Hespanha, and G. S. Sukhatme, Eds. Prentice Hall PTR, 2002, ch. 10, pp. 182–195. 72

Timothy Edmunds is a Postdoctoral Research Fellow in the Department of Computer Science at the University of British Columbia, Vancouver. He received his Ph.D. from Rutgers University, NJ, and his B.Sc. degree from UBC. His current research focus is on the perceptual aspects of sensorimotor control.

Dinesh K. Pai is a Professor and Tier 1 Canada Research Chair in the Department of Computer Science at the University of British Columbia, where he directs the Sensorimotor Systems Laboratory. He received his Ph.D. from Cornell University, Ithaca, NY, and his B.Tech. degree from the Indian Institute of Technology, Madras. At UBC he is a member of the Institute of Applied Mathematics, the Graduate Program in Neuroscience, the ICORD International Collaboration on Repair Discoveries, and the Brain Research Centre. He has been a Professor at Rutgers University, and a visiting professor at Carnegie Mellon University's Robotics Institute, New York University's Center for Neural Science, the University of Siena, and the Collège de France, Paris. His current research is focused on sensorimotor computation in computer graphics, haptics, robotics, and neurobiology.