Integrated Multimodal Interaction Using Texture Representations☆

Auston Sterling, Ming C. Lin

University of North Carolina at Chapel Hill

Abstract

In this paper, we explore texture mapping as a unified representation for enabling realistic multimodal interaction with finely-detailed surfaces. We show how both normal maps and relief maps can be adopted as unified representations to handle collisions with *textured* rigid body objects, synthesize complex sound effects from long lasting collisions and perform rendering of haptic textures. The resulting multimodal display system allows a user to see, hear, and feel complex interactions with textured surfaces. By using texture representations for seamlessly integrated multimodal interaction instead of complex triangular meshes otherwise required, this work is able to achieve up to 25 times performance speedup and reduce up to six orders of magnitude in memory storage. We further validate the results through user studies to demonstrate the effectiveness of texture representations for integrated multimodal display.

1. Introduction

In computer graphics, texture mapping has been one of the most widely used techniques to improve the visual fidelity of objects while significantly accelerating the rendering performance. There are several popular texture representations, such as displacement maps [1], bump mapping with normal maps [2, 3], parallax maps [4, 5], relief maps [6, 7], etc., and they are used mostly as "imposters" for rendering static scenes. These textures are usually mapped onto objects' surfaces represented with simplified geometry. The fine details of the objects are visually encoded in these texture representations. By replacing the geometric detail with a texture equivalent, the resulting rendered image can be made to appear much more complex than its underlying polygonal geometry would otherwise convey. These representations also come with a significant increase in performance: texture maps can be used for real-time augmented and virtual reality (AR/VR) applications on low-end commodity devices.

Sensory conflict occurs when there is a mismatch between information perceived through multiple senses and can cause a break in immersion in a virtual environment. When textures are used to represent complex objects with simpler geometry, sensory conflict becomes a particular concern. In an immersive virtual environment, a user may see a rough surface of varying heights and slopes represented by its texture equivalent mapped to a flat surface. In the real world, objects behave very differently when bouncing, sliding, or rolling on bumpy or rough surfaces than they do on flat surfaces. With visually complex detail and different, contrasting physical behavior due to the simple flat surface, sensory conflict can easily occur—breaking the sense of immersion in the virtual environment. In order to capture such behaviors, the geometry used in a physics simulator would usually require a fine triangle mesh with sufficient surface detail, but in most cases a sufficiently fine mesh is unavailable or would require prohibitive amounts of memory to store.

Since the given texture maps contain information about the fine detail of the mapped surface, it is possible to use that information to recreate the behavior of the fine mesh. Haptic display and sound rendering of textured surfaces have both been independently explored [8, 9], but texture representations of detail have not been previously used for visual simulation of dynamic behavior due to collisions and contacts between rigid bodies. For example, the system for sound rendering of contacts with textured surfaces [9] displays a pen sliding smoothly across highly bumpy surfaces. While the generated sound from this interaction is dynamic and realistic, the smooth *visual* movement of the pen noticeably does not match the texture implied by the sound. In order to minimize sensory conflict, it is critical to present a unified and seamlessly integrated multimodal display to users, ensuring that the sensory feedback is consistent across the senses of sight, hearing, and touch for both coarse and fine levels of detail.

Motivated by the need to address the sensory conflict due to the use of textures in a multimodal virtual environment, we previously examined the use of normal mapping as a unified representation of fine detail for sight, hearing, and touch [10]. In this paper, we explore both normal maps and relief maps for integrated multimodal display. The main results of this work include:

- A new effective method for visual simulation of physical behaviors for rigid objects textured with normal maps;
- A seamlessly integrated multisensory interaction system using normal maps;

☆http://gamma.cs.unc.edu/MultiDispTexture

- An extended system using relief maps;
- Evaluation and analysis of texture-based multimodal display and their effects on task performance; and
- Evaluation of perceptual differences between normal and relief map representations.

The rest of the paper is organized as follows. We first discuss why we have selected normal and relief maps as our texture representations for multimodal display. We then describe how each mode of interaction can specifically use normal maps to improve perception of complex geometry (Sec. 3). We highlight the behavior of virtual objects as they interact with a large textured surface, and describe a new method to improve visual perception of the simulated physical behaviors of colliding objects with a textured surface using normal maps. We also demonstrate how to use the same normal maps in haptic display and sound rendering of textured surfaces. We describe how the additional depth information in relief maps can be used to improve each mode of interaction (Sec. 4).

We have implemented a prototype multimodal display system using normal and relief maps and performed both qualitative and quantitative evaluations of its effectiveness on perceptual quality of the VR experience and objective measures on task completion (Sec. 5). A user study suggests that normal maps can serve as an effective, unified texture representation for seamlessly integrated multi-sensory display and the resulting system generally improves task completion rates with greater ease over use of a single modality alone. A second user study suggest that relief maps are also an effective representation of fine detail, with an improvement in sensory cohesiveness over normal maps.

2. Previous Work

Normal maps and relief maps are used throughout this paper as representations of fine detail of the surface of objects. Normal maps were originally introduced for the purposes of bump mapping, where they would perturb lighting calculations to make the details more visibly noticeable [2]. Relief mapping uses both depths and normals for more complex shading [6, 7]. Numerous other texture mapping techniques exist as well. Displacement mapping, parallax mapping, and a number of more recent techniques use height maps to simulate parallax and occlusion [1, 4, 5]. A recent survey goes into more detail about many of these techniques [11]. Mapping any of these textures to progressive meshes can preserve texture-level detail as the level-of-detail (LOD) of the mesh shifts [3].

Height maps mapped to object surfaces have been used to modify the behavior of simple collisions in rigid-body simulations [12]. We are not aware of similar work done using normal maps aside from our own.

In haptic rendering, a 3D object's geometries and textures can be felt by applying forces based on point-contacts with the object [13, 14]. Complex objects can also be simplified, with finer detail placed in a displacement map and referenced to produce accurate force *and torque* feedback on a probing object [8]. The mapping of both normal and displacement maps to simplified geometry for the purposes of haptic feedback has also been explored [15]. Dynamic deformation textures, a variant of displacement maps, can be mapped to create detailed objects with a rigid center layer and deformable outer layer. The technique has been extended to allow for 6-degree-of-freedom (DOF) haptic interaction with these deformable objects [16]. A common approach to force display of textures is to apply lateral force depending on the gradient of a height map such that the user of the haptic interface feels more resistance when moving "uphill" and less resistance when moving "downhill" [17, 18]. Our approach to haptic rendering of textures applies force feedback to simulate the presence of planes which reproduce this effect, and similarly we use a simplified model for interaction with dynamic rigid-body objects.

Modal analysis and synthesis are commonly used techniques for synthesizing realistic sound [19]. Modal synthesis has been integrated with rigid-body physics simulators in order to produce contact sounds that synchronize with collision events. To handle objects with arbitrary geometry, they can be decomposed with finite element methods [20]. Further speed optimizations can be made based on psychoacoustics, such as mode compression and truncation [21]. We synthesize transient impact sounds by directly using this technique.

Sounds created by long-lasting contacts between objects require some additional effort. Fractal noise is a common way of representing the small impacts generated during rolling and scraping [22]. We perform sound synthesis for lasting sounds by using the framework for synthesizing contact sounds between textured objects [9]. This work introduced a multi-level model for lasting contact sounds combining fractal noise with impulses collected from the normal maps on the surfaces of the objects. This application of normal maps to sound generation without similar application to rigid-body dynamics causes noticeable sensory conflict between the produced audio and visible physical behavior.

3. Overview and Texture Map Representation

Our system uses three main components to create a virtual scene where a user can experience through multiple modalities of interaction. A rigid body physics simulator controls the movement of objects. The only form of user input is through a haptic device, which also provides force feedback to stimulate the sense of touch. Finally, modal sound synthesis is used to dynamically generate the auditory component of the system. In this section, we briefly cover the details of texture mapping, discuss haptic illusions and justify the use of texture representations, then describe each of these components using normal maps as the representation of detail. The relief map representation is covered in greater detail in Section 4.

3.1. Normal and Relief Maps

Normal maps are usually stored as RGB images, with the color values encoding vectors normal to the details of the surface they are mapped to. Refer to Figure 1 for an example. It is

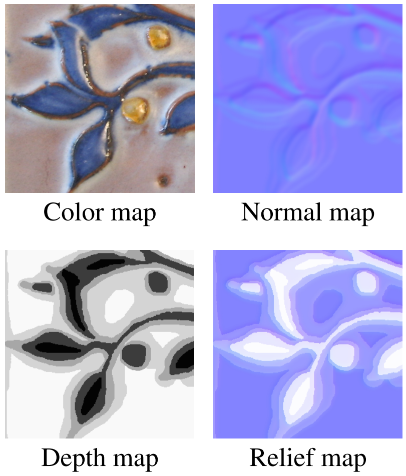

Figure 1: Texture map example. RGB values encode normal vectors in each texel. In relief maps, the alpha value encodes depth information.

common practice to create normal maps directly corresponding to a color map, such that the color map can be referenced at a location to get a base color and the normal map can be referenced at the same location for the corresponding normal vector.

Relief mapping is a technique for rendering textured surfaces using additional depth information. It is usually implemented on GPUs and can be briefly described as computing intersections with the height-field defined by the depth values using rays from the camera to each pixel [7]. Ray casting lets relief-mapped surfaces properly handle self-occlusion, and extra ray casts from a light source enable self-shadowing. Since rays are cast from the camera, proper perspective is maintained as the camera looks at the textured surface from different angles. Our surfaces are rendered using relief mapping, so we refer to their textures as "relief maps", though the same texture could be used for parallax occlusion mapping or for displacements on GPU-tessellated surfaces.

Our relief maps contain their depth information in the alpha channel of the image. In the alpha channel, a value of zero (black, entirely transparent) means the texel is at its highest, exactly along the geometry of the mapped object. Larger values (tending towards white/visible) indicate that the texel is recessed inside the object. Much like sculpted relief artwork, relief maps can only cut into the surface; they cannot raise a texel outside the object's geometry. The maximum depth as a percentage of mapped object dimensions can be set individually for each relief map.

Depending on the resolution of the texture image and the surface area of the object it is mapped to, a normal or relief map can provide very fine detail about the object's surface. As we describe in this paper, this detail—while still an approximation of a more complex surface—is sufficient to replicate other phenomena requiring knowledge of fine detail.

3.2. *Design Consideration*

Next we discuss various consideration in choosing texture maps as our representation of fine detail, beginning with a discussion on haptic perception.

3.2.1. *Haptic Illusions*

Perceptual illusions, including visual, haptic and auditory, have been explored in virtual reality for immersing users in computer generated environments through multi-sensory display. For example, bump mapping can be regarded as a *visual* illusion where a user who is expecting to see depth in a bump-mapped surface may interpret the shading as depth. Haptic illusions can be roughly defined as when a haptic stimulus is applied under specific conditions that change the perception of that stimulus. A classic example is the size-weight illusion in which a participant lifts two boxes of equal weight and unequal sizes and perceives the smaller box to be heavier. There are many types of haptic illusions, which have been well documented and catalogued [23].

There are some real-world examples of haptic illusions which are relevant for simulating slope and depth. In the "curved plate" illusion, a flat edge rolled over a fingertip at about 1 Hz produces the sensation that the edge is curved. As described earlier, previous work on simulating haptic textures also relies on haptic illusions: applying only lateral forces to a haptic probe can create the sensation of a vertical height difference.

In these illusions, the changing direction of normal force creates the illusion of curvature. That is, *the normal vector is an important haptic cue for curvature*. Texture maps with normal vectors provide exactly that information, and therefore should be able to simulate the curvature of a more complicated surface through haptic illusions. This observation forms the hypothesis of our exploration of texture representations.

3.2.2. *Choice of Representation*

On top of providing an important haptic cue, normal vectors have additional advantages over alternative options. Using very high-resolution geometry would automatically produce many of the desired effects, but the performance requirements for *interactive* 3D applications significantly reduces their viability in our early deliberation. This is especially important to consider in AR and VR applications, where real-time performance must be maintained while possibly operating on a low-end mobile phone or head mounted display.

Other texture map information may also be considered, such as height (or displacement) maps. For sound, Ren et al. [9] used normal maps because the absolute height does not affect the resulting sound; it is the change in normal which causes a single impulse to produce meso-level sound. With regard to force display of textured surfaces, the Sandpaper system [18] has been a popular and efficient method for applying tangential forces to simulate slope based on a height map. Using normal vectors we can instead scale a sampled normal vector to produce the same normal and tangential forces. Rigid body collision response also depends entirely on normal vectors.

Since each component of the system depends directly on the normals, a normal map representation emerges as the natural choice. An added convenience is that normal maps are widely supported (including mobile games) and frequently included alongside color maps. Although normal maps contain the most important cues for multimodal interaction, we would like to evaluate how much benefit is gained from combining normals with depth information. Relief mapping uses both for visual rendering and has become more common alongside GPUs, so relief maps provide a useful starting point for considering depth in multimodal interaction with textures. The application needs, the performance requirement, and the wide availability and support on commodity systems all contribute to our adoption of normal maps and relief maps as the mapping techniques in this work.

3.3. Rigid Body Dynamics

In order to simulate the movement of objects in the virtual scene, we use a rigid body dynamics simulator. These simulators are designed to run in real time and produce movements of rigid objects that visually appear believable.

Rigid body dynamics has two major steps: collision detection and collision response. Collision detection determines the point of collision between two interpenetrating objects as well as the directions in which to apply force to most quickly separate them. Modifying the normals of an object, as we do with normal maps, does not affect whether or not a collision occurs. This is a significant limitation of a normal map representation without any height or displacement information.

There are numerous algorithms for collision resolution, which determines how to update positions and/or velocities to separate the penetrating objects. In impulse-based approaches, collisions are resolved by applying an impulse in the form of an instantaneous change in each objects' velocity. Considering a single object's velocity vector $\mathbf{v}$, $\Delta\mathbf{v}$ is chosen to be large enough so that the objects separate in the subsequent timesteps. The change in velocity on an object with mass m is computed by applying a force f over a short time Δt in the direction of the geometric normal $\mathbf{n_g}$ of the other colliding object:

$$\Delta\mathbf{v} = \frac{f\Delta t}{m}\mathbf{n_g} \tag{1}$$

This process is highly dependent on the normal vectors of each object, and other collision resolution approaches have this same dependency.

3.3.1. Modifying Collision Behavior with Normal Maps

We focus on simulating collisions between small dynamic objects and large textured surfaces whose details would have a large effect on the dynamic object. To get an intuitive understanding of the behavior we seek to replicate, imagine a marble rolling on a brick-and-mortar floor. When the marble rolls to the edge of a brick, the expected behavior would be for it to fall into the mortar between bricks and possibly end up stuck at the bottom.

The level of detail needed to accurately recreate these dynamics with a conventional rigid body physics engine is too fine to be interactively represented with a geometric mesh, especially with large scenes in real-time applications. A normal map contains the appropriate level of detail and is able to represent the flat brick tops and rounded mortar indentations.

In order to change the behavior of collisions to respect fine detail, our solution is to modify the contact point and contact normal reported by the collision detection step. This is an extra step in resolving collisions, and does not require any changes to the collision detection or resolution algorithms themselves.

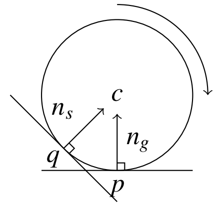

Figure 2: Contact point modification on a rolling ball: given the contact point $\mathbf{p}$ and sampled normal $\mathbf{n_s}$, we want to simulate the collision at point $\mathbf{q}$.

The contact normal usually comes from the geometry of the colliding objects, but the normal map provides the same information with higher resolution, so our new approach uses the normal map's vectors instead. Given the collision point on the flat surface, we can query the surface normal at that point and instruct the physics engine to use this perturbed normal instead of the one it would receive from the geometry. One side effect of using the single collision point to find the perturbed normal is that it treats the object as an infinitely small probe.

3.3.2. Rolling Objects and Collision Point Modification

There is a significant issue with this technique when simulating rolling objects. Refer to Figure 2 for an example. Two planes are shown, the horizontal one being the plane of the coarse geometry and the other being the plane simulated by the perturbed normal. Note that the contact points with the rolling ball differ when the plane changes. The vector $\mathbf{n_s}$ shows the direction of the force we would ideally like to apply. If we were to apply that force at the original contact point $\mathbf{p}$, the angular velocity of the sphere would change and the ball would begin to roll backwards. In practice, this often results in the sphere rolling in place when it comes across a more extreme surface normal. Instead, we use the sphere radius r, the perturbed surface normal $\mathbf{n_s}$, and the sphere center $\mathbf{c}$ to produce the modified contact point $\mathbf{q}$:

$$\mathbf{q} = \mathbf{c} - (r\mathbf{n}) \tag{2}$$

This modification applies the force directly towards the center of mass and causes no change in angular velocity, but is less accurate for large spheres and extreme normal perturbations.

This contact point modification is important for perceptually believable rolling effects. Shapes other than spheres do not have the guarantee that the contact point will be in the direction of the $\mathbf{c} - \mathbf{n}$ vector, so this does not apply in the general case. Generally, we can simply modify the normal without changing the contact point. In the case of relief maps, the true collision points and contact normals can be determined, so this correction is unnecessary.

3.4. Haptic Interface

We have designed our system to use a PHANToM Desktop haptic device [24]. This device can measure 6-DOF motion: three translational and three rotational, but only display 3-DOF

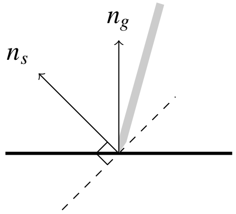

Figure 3: Haptic force is applied in the direction of the sampled normal $\mathbf{n_s}$ instead of the geometric normal $\mathbf{n_g}$.

forces (i.e. no torques). We have chosen to represent the PHANToM as a pen inside the virtual environment, which matches the scale and shape of the grip. While we could use forces determined by the rigid-body physics engine to apply feedback, the physics update rate (about 60 Hz) is much lower than the required thousands of Hz needed to stably simulate a hard surface.

We simulate the textured surface by projecting the tip of the PHANToM Desktop grip onto the surface in the direction of the coarse geometry's normal. The fine surface normal is queried and interpolated from nearby normal map vectors. The PHANToM simulates the presence of a plane with that normal and the projected surface point. Given the normal vector sampled from the normal map $\mathbf{n_s}$ and pen tip position projected onto the surface $\mathbf{p}$, the equation modeling this plane is:

$$(\mathbf{n_s} \cdot (x, y, z)) - (\mathbf{n_s} \cdot \mathbf{p}) = 0 \tag{3}$$

The PHANToM now needs to apply the proper feedback force to prevent the pen's tip from penetrating into the plane. This is accomplished using a penalty force, simulating a damped spring pulling the point back to the surface. Using the modified normal vector, the simulated plane serves as a local first order approximation of the surface. Note that while the slopes of the planes produced by the PHANToM can vary significantly based on the normal map, at the position of the pen the plane will coincide with the surface. This is illustrated in Figure 3, where the simulated plane intersects the geometric plane at the collision point. This creates an illusion of feeling a textured surface while keeping the pen in contact with the flat underlying surface geometry.

With this technique, stability can be concern in some cases. Most noticeably, in steep and narrow V-shaped valleys, a user pushing down on the surface might cause the tip of the pen to oscillate between the valley sides. Users sliding the pen rapidly across bumpy surfaces may also feel forces that are stronger and more abrupt than they would expect. We have mainly mitigated these concerns by smoothing the normal maps and scaling down the penalty forces. A side effect is that the surfaces end up feeling slightly smoother and softer, though we have found this an acceptable tradeoff for improved stability.

We use a simplified model to interact with dynamic objects. The PHANToM's corresponding pen appearance in the environment is added as an object in the rigid-body physics simulator. Whenever this pen comes in contact with a dynamic object, the physics simulator computes the forces on the objects needed to separate them. We can directly apply a scaled version of this force to the haptic device. This ignores torque as our 3-DOF PHANToM can only apply translational forces. This approach is fast, simple, and lets the user push and interact with objects around the environment.

3.5. *Sound Synthesis*

Sound is created due to a pressure wave propagating through a medium such as air or water. These waves are often produced by the vibrations of objects when they are struck, and human ears can convert these waves into electrical signals for the brain to process and interpret as sound. One of the most popular physically-based approaches to modeling the creation of sound is modal sound synthesis, which analyzes how objects vibrate when struck at different locations to synthesize contact sounds.

3.5.1. *Modal Analysis and Synthesis Background*

In order to perform modal analysis, we represent the objects using a discretized representation such as a spring-mass system or a tetrahedral mesh. The dynamics of the object can be represented with the system of differential equations:

$$\mathbf{M}\ddot{\mathbf{r}} + \mathbf{C}\dot{\mathbf{r}} + \mathbf{K}\mathbf{r} = \mathbf{f} \tag{4}$$

$\mathbf{r}$ is a vector of displacements from the given starting positions, which are assumed to be at rest. $\mathbf{f}$ is the vector of external forces applied to the system. $\mathbf{M}$ and $\mathbf{K}$ are the mass and stiffness matrices, respectively, which describe the distribution of mass and connectivity of the object. For the damping matrix $\mathbf{C}$, we use Rayleigh damping which expresses $\mathbf{C}$ as a linear combination of $\mathbf{M}$ and $\mathbf{K}$.

This system of equations can be decoupled to produce a bank of modes of vibration. The equation for each mode is a standard damped oscillator, which vibrates at a certain frequency and decays exponentially over time. Almost all of the complex calculations are dependent only of the properties of the objects and therefore can be precomputed and stored.

Sound synthesis at runtime is done in two steps. When an object is struck, the modes of vibration are excited depending on the strike's location and direction. Once the vibrations begin, the modes are sampled and updated at around $44,100$ Hz to create perceptually realistic sound. For more details on modal analysis and synthesis, refer to the work of O'Brien et al. for a FEM approach using tetrahedral meshes [20] or the work of Raghuvanshi and Lin for a spring-mass approach [21].

3.5.2. *Textures and Lasting Sounds*

Modal synthesis works well for generating sound that varies for each object, material, and impulse. However, for long-lasting collisions such as scraping, sliding, and rolling, the sound primarily comes from the fine details of the surface which are not captured in the geometry of the input mesh when using texture maps. We adopt the method by Ren et al. [9], which uses three levels of detail to represent objects, with normal maps providing the intermediate level of detail.

At the macro level, the object is represented with the provided triangle mesh. The first frame in which a collision is detected, it is considered transient and impulses are applied according to conventional modal synthesis. If the collision persists for multiple frames, we instead use the lower levels described below.

Even surfaces that look completely flat produce rolling, sliding, and scraping sounds during long-lasting collisions. The micro level of detail contains the very fine details that produce these sounds and are usually consistent throughout the material. Sound at this level is modeled as fractal noise. Playback speed is controlled by the relative velocity of the objects, and the amplitude is proportional to the magnitude of the normal force.

The meso level of detail describes detail too small to be efficiently integrated into the triangle mesh, but large enough to be distinguishable from fractal noise and possibly varying across the surface. Normal maps contain this level of detail, namely the variation in the surface normals. This sound is produced by following the path of the collision point over time. Any time the normal vector changes, the momentum of the rolling or sliding object must change in order to follow the path of that new normal. This change produces an impulse which can be used alongside the others for modal synthesis. This can be mathematically formulated as follows.

Given an object with mass m moving with tangent-space velocity vector $\mathbf{v_t}$ along a face of the coarse geometry with normal vector $\mathbf{n_g}$ whose nearest normal map texel provides a sampled normal $\mathbf{n_s}$, the component of the momentum orthogonal to the face $\mathbf{p_n}$ is:

$$\mathbf{p_n} = m\left(-\frac{\mathbf{v_t} \cdot \mathbf{n_s}}{\mathbf{n_g} \cdot \mathbf{n_s}}\right)\mathbf{n_g} \tag{5}$$

This momentum is calculated every time an object's contact point slides or rolls to a new texel, and the difference is applied as an impulse to the object. More extreme normals or a higher velocity will result in higher momentum and larger impulses. Whenever objects are in collision for multiple frames, both the micro-level fractal noise and the meso-level normal map impulses are applied, and the combined sound produces the long-lasting rolling, sliding, or scraping sound.

4. Relief Map Representation

As an extension to the modalities described above which rely solely on the surface's normal vectors, we have also explored how a relief map's depth information can be incorporated to improve each component. In this section, we explain these differences.

4.1. Modifying Collision Behavior with Relief Maps

When discussing rigid body physics with a normal map, we mentioned that collision *detection* remained unchanged while collision *resolution* required modification. With relief maps' depth information, collision *detection* now requires additional steps, as now objects may penetrate inside the geometry of a surface as long as they stay outside the recessed relief surface. Again focusing on the situation where a small object collides with a large textured surface, the problem is collision detection between an object and a height map. We adopt a similar approach described by Otaduy et al. for computing directional penetration depth between two textured objects [8].

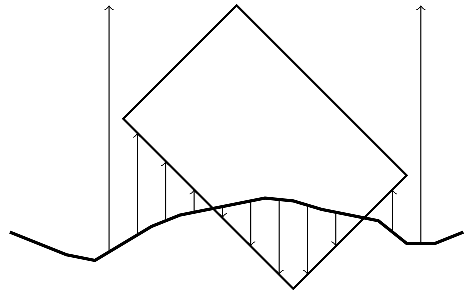

Figure 4: A rectangle colliding with a 1D relief map. Wherever arrows point downwards, the distance is negative and there is a collision.

In general, the penetration depth between two colliding objects is the shortest distance one of the objects would have to move in order to separate themselves. The *directional* penetration depth is the penetration depth where the objects can only move along one specified axis. Computing the general penetration depth between finely-detailed objects can be prohibitively slow for interactive applications. Directional penetration depth can be used in place of general penetration depth, sacrificing accuracy for speed, which is more appropriate for our goals.

The GPU-based method proposed by Otaduy et al. for computing directional penetration depth is to represent each colliding object as a height map perpendicular to the specified direction. These height maps are aligned with one another so that the distance between the objects at some point is the difference in height between two matching height map texels. Wherever the distance between objects is negative, there is a collision. The most negative distance value can then be reported as the directional penetration depth.

In our case, the large plane textured with a relief map is already a height map perpendicular to the normal vector of the plane. In order to adopt a similar technique on any CPU (and GPU), we need to convert the colliding object into a height map of its own. We primarily accomplish this by projecting the object onto the plane and rasterizing the result with the same resolution as the relief map. The depth information from that process can then be used as the object's height map. The difference between the relief map's depth and the object's height map is the distance between them, and one or more collision points can be found by searching for negative distances. The collision points and the normal vectors sampled from the relief map at the same locations can then be passed to the collision resolution solver.

A simple example is illustrated in Figure 4, where a rectangular object is colliding with a 1D relief map. Each arrow points from a relief map texel to the corresponding texel of the rasterized object height map, where upwards arrows are positive distance values and downwards arrows are negative. The most negative distance values would be reported as collision points. Since the points are found through a sampling process, there is naturally a tradeoff between speed and accuracy: each sample takes time to compute but contributes to finding a more accurate collision point.

4.2. Haptic Interface with Relief Maps

For haptic interaction through the PHANToM, as with rigid body physics, the change is in collision detection and not resolution. The tip of the pen is projected down in the direction of the surface normal, but collision is only reported if the pen's tip is below the relief map depth value. If there is a collision, the simulated plane is created in exactly the same way as described in the normal map section. With depth information, the pen can follow the actual contours of the surface.

4.3. Sound Synthesis with Relief Maps

With normal maps, it is necessary to track the change in the sampled normal vector to estimate the impulses felt by a rolling, sliding, or scraping object for the purposes of sound synthesis. In the case of a relief map with depth information, we can compute significantly more accurate collision information, and with that comes significantly more accurate impulse information. With the relief map collision detection described previously, we can directly take the impulses reported by the physics engine and apply them to the bank of modes of vibration to synthesize sound.

Since the physics engine properly takes into account the normal and depth information from the relief map, the resulting impulses already account for the texture detail. Adding in the same fractal noise to account for surface variations too small to be captured by either texture representation produces realistic long-lasting contact sounds.

5. Implementation and Results

We have described each component of our multimodal system using texture maps. We implemented this prototype system in C++, using NVIDIA's PhysX as the rigid body physics simulator, OGRE3D as the rendering engine, VRPN to communicate with the PHANToM [25], and STK for playing synthesized sound [26].

In our previous work, we discretized our objects using spring-mass systems to perform modal analysis for sound synthesis [10]. For this paper, we instead use a finite element method representation using tetrahedral meshes. The difference between the representations is primarily that the spring-mass model represents objects as hollow shells with a given shell thickness, while using tetrahedral meshes properly represents the full volume of objects. With either representation, the equation in Section 3.5.1 is used, but matrices are constructed differently. This provides an improvement in accuracy over spring-mass discretizations and only negatively impacts the runtime during the precomputation step. All scenarios we created contained at least one textured surface acting as the ground of the environment, and only its normal map was used to modify collision response, haptic display, or sound rendering.

5.1. Performance Analysis

The sound synthesis module generates samples at 44100Hz, the physics engine updates at 60Hz, and while the PHANToM hardware itself updates at around 1000Hz, the surface normal is sampled to create a new plane once per frame. On a computer with an Intel Xeon E5620 processor and 24GB RAM, the program consistently averages more than 100 frames per second. This update rate is sufficient for real-time interaction, with multi-rate updates [8, 9].

A natural comparison is between our texture-based method and methods using meshes containing the same level of detail. Most of our texture maps are around 512×512, so recreating the same amount of detail in a similarly fine mesh would require more than $512^2 = 262114$ vertices and nearly twice as many triangles. As a slightly more realistic alternative, we also compare to a relatively coarse 256×256 mesh with more than $256^2 = 65536$ vertices. For a discussion of LOD representations and the challenges in simplifying meshes for multimodal systems, refer to Section 5.4.2.

Table 1 presents memory and timing information when comparing our method to methods using the equivalent geometry meshes instead. The coarse mesh used for modal analysis is greatly reduced in size compared to the finer meshes. We generated these finely-detailed meshes for the sake of comparison, but in practice, neither mesh would be available to a game developer and they would have to make do with the constraints considered in our method.

Modal analysis for audio generation on the finer meshes requires significantly more memory than is available on modern machines, so a simplified mesh is required. The listed "Runtime Memory" is the runtime requirement for modal sound synthesis and primarily consists of the matrix mapping impulses to modal response. The listed memory requirements are based on a spring-mass discretization for normal maps and the FEM-based discretization for relief maps.

Our method is faster than using fine meshes in each mode of interaction. Haptic rendering time using our method took merely 60 μs per frame. The listed "Visual Time" is the time taken to render the surface, either as a flat texture mapped plane, or as a color-mapped mesh without normal mapping. The PHANToM's API integrated with VRPN does not support triangular meshes, and we could not test performance of collision detection and haptic rendering manually, though the time needed to compute collision with an arbitrary triangular mesh would have been significantly longer (at least by one to two orders of magnitude based on prior work, such as H-COLLIDE).

The main sound rendering loop runs at around 44 kHz regardless of the chosen representation of detail. The only difference comes from the source of sound-generating impulses: our method for normal maps collects impulses from a path along the normal map while a relief map or mesh-based approach collects impulses reported by the physics engine. Applying impulses to the modal synthesis system is very fast relative to the timed modes of interaction.

	Mesh Size	Offline Memory	Runtime Memory	Physics Time	Visual Time	Haptic Time
Normal Map	10KB	2.7 MB	270 KB	175 μs	486 μs	60 μs
Relief Map	110KB	1 GB	18 MB	2.2 ms	900 μs	60 μs
Coarse Mesh	4.5 MB	288 GB*	450 MB*	3.0 ms	2.1 ms	–**
Fine Mesh	19 MB	4500 GB*	1700 MB*	4.9 ms	7.0 ms	–**

Table 1: Memory and timing results for our (texture-based) methods compared to a similarly detailed coarse mesh (66,500 vertices) and fine mesh (264,200 vertices). Entries marked with * are extrapolated values, since the memory requirements are too high to run on modern machines. Haptic time (**) was not measurable for triangle meshes due to an API limitation. Normal maps are able to achieve up to **25 times** of runtime speedup and up to **6 orders of magnitude** in memory saving.

5.2. *Normal Map Texture Identification User Study*

In order to evaluate the effectiveness of this multimodal system, we conducted a user study consisting of a series of tasks followed by a questionnaire. One objective of this user study was to determine the overall effectiveness of our system. For this study, only the normal map representation was used. A subject is interacting with the normal map through sight, touch, and sound. If each of these components are well designed and implemented, a subject should be able to identify the material by multimodal interaction. The other goal is to see how well the use of multiple senses helps to create a cohesive recognition of the material being probed. Even if subjects find the haptic display alone is enough to understand the texture of the material being probed, does adding sound cues speed up their process of identifying textures or instead cause sensory conflict?

5.2.1. *Set-up*

Twelve participants volunteered to take part in this study experiment. Each subject was trained on how to use the PHANToM and was given some time to get used to the system by playing in a test scene (see Figure 7, top row). The subject then completed a series of six trials. In each trial, a material for the surface was chosen at random, and all aspects of it *except* its visual appearance were applied. That is, the subject would be able to feel the surface's texture with the PHANToM, hear the sound generated from ball and PHANToM pen contacts, and see the rolling ball respond to ridges and valleys on the surface. The subject was able to cycle through each material's visual appearance (in the form of a texture) by pressing the button on the PHANToM's grip. Their task was to select the material's unknown visual appearance based on the multimodal cues received.

The first three trials provided all three cues—sound, ball, and pen—but in each of the remaining three trials only two of the three cues would be available. The subject would be informed before the trial began if any cues were missing. The subjects were recommended to use all available cues to make their decision, but were otherwise unguided as to how to distinguish the materials. After the trials were completed, a short questionnaire was provided for subjective evaluation and feedback.

This study utilizes sensory conflict to guide the subjects to correctly identify the visual appearance. If the multimodal cues present the sounds, haptic texture, and physical response of a metal surface with regular grooves, but the subject has currently selected the visual appearance of a flat, smooth wooden surface, they should recognize the sensory conflict and reject the wooden surface as the answer. Once the subject has selected the correct visual appearance (grooved metal in this example), they should feel relatively little sensory conflict and from that realize they have found the answer.

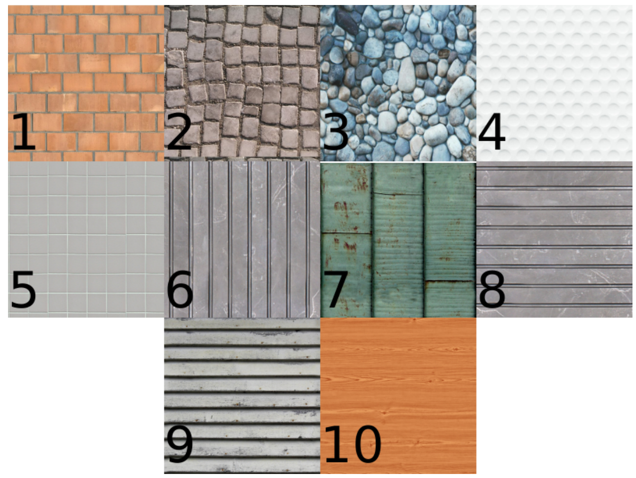

Figure 5: The available materials for the texture identification user study. 1–3 sounded like bricks, 4–5 sounded like porcelain, 6–8 sounded like metal, and 9–10 sounded like wood.

	ID rate	Time (s)	Ease (1–10)
All modes	78%	38 ± 18	7.9 ± 1.3
No sound	81%	46 ± 45	4.9 ± 2.2
No haptics	54%	41 ± 23	3.6 ± 1.8
No physics	72%	47 ± 58	6.4 ± 2.6

Table 2: Results comparing effectiveness when limiting the available modes of interaction in the texture identification user study. "Ease" is evaluated by the subjects where 1 is difficult and 10 is easy. When using all modes of interaction, subjects were generally able to identify the material more frequently than when only using two modes and reported that they found identification to be easiest when all modalities of interaction were engaged.

Figure 5 shows the materials chosen for the user study. The subjects were allowed to look at each of these textures before the trials began, but were not able to feel or hear them. Some of these were specifically chosen to be challenging to distinguish.

5.2.2. *Experimental Results*

In Table 2, we compare the results when varying which modes of interaction are available to subjects. The ID rate is the percentage of trials in which the subject was able to correctly identify the material, and the mean time only takes into

	Guesses (%)									
ID	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	**50**	0	33	0	0	17	0	0	10	0
2	0	**80**	0	20	0	0	0	0	0	0
3	0	0	**100**	0	0	0	0	0	0	0
4	0	0	0	**83**	17	0	0	0	0	0
5	0	13	25	0	**50**	0	12	0	0	0
6	0	0	17	0	0	**83**	0	0	0	0
7	8	0	8	0	0	8	**60**	8	8	0
8	0	0	0	0	0	0	0	**75**	25	0
9	0	0	17	0	0	0	0	16	**67**	0
10	0	0	0	0	0	0	0	0	12	**88**

Table 4: Confusion matrix showing the guesses made by subjects in the texture identification study. For all materials, a significant majority of subjects were able to identify the right materials.

account time for correct guesses. The "ease" was provided by the subjects on the questionnaire, where they were asked to rate on a scale from 1–10 how easy they found it was to identify the material for each combination of modes of interaction. Higher "ease" scores mean the subject found it easier to identify the material.

In all cases, the identification rate was higher than 50%, and usually much higher than that. The loss of haptic feedback caused the largest drop in ID rate and ease. The loss of sound actually improved material identification—although the difference is not statistically significant—but subjects still found identification to be much more perceptually challenging.

Two more noteworthy results were gathered from a subjective questionnaire, with results shown in Table 3. Subjects were asked how frequently they used each of the modes in identifying the material. The subjects were also asked how well each mode of interaction represented how they would expect the materials to sound or feel. These results could help explain the low identification rate when haptics are disabled: most subjects both relied heavily on tactile senses and found it be the most accurate mode. The subjects considered the sound and physics somewhat less accurate but still occasionally useful for determining the materials.

More detailed results from the study are presented in Table 4. An entry in row i and column j is the percentage of times the subject was presented material i and guessed that it was material j. The higher percentages along the diagonal demonstrate the high correct identification rate. Also note that in most categories there is no close second-place guess. The largest exception is that 33% of the time material 1 (brick grid) was mistakenly identified as material 3 (pebbles), likely due to similarity in both material sounds and patterns.

5.2.3. Analysis

Our analysis is largely based on comparing the results from interactions with different sets of modalities using a t-test to analyze the difference between the modalities. In addition to the p value for statistical significance, we also use Cohen's effect size d, defined as the difference between the means of two samples divided by their pooled standard deviation [27]. Effect size is an important factor to consider alongside statistical significance, explaining not just if there is a difference, but explaining (in units of standard deviations) how large that difference actually is.

Due to the relatively low sample size in the study of each material, many of the possible direct comparisons would not be statistically significant. Therefore, for this study the reported statistics are based on combined data from all study materials; we do not compare the result on each material to one another.

Between identification rates, there was no statistically significant change when removing a mode ($p > .05$), but the removal of haptics came close with $p = .066$. The subjective subject-reported values of ease and accuracy were generally more significant. Subjects reported that they found material identification to be more difficult when either sound or haptics were removed in comparison to having all modes available ($p < .05$), but did not find identification more difficult when the physics modification was removed ($p > .05$). Cohen's effect size values (d) of 1.66 for the removal of sound and 2.79 for the removal of haptics suggest a very large change in perceptual difficulty when removing these modes. Subjects also reported that they found the haptics to be more accurate than physics or sound ($p < .05$), but did not find a significant difference in accuracy between physics and sound ($p > .05$). Cohen's effect size values of 1.02 comparing haptics to physics and 1.36 comparing haptics to sound suggest a large difference in the perception of how accurate these modes are.

Overall, these results demonstrate that each mode of interaction is effectively enabled through use of normal maps. Combining multiple modes increases accuracy, which suggests that the subjects are receiving cohesive, non-conflicting information across their senses. This was a deliberately challenging study, using materials which sounded similar and had similar geometric features and patterns. Furthermore, the task asked subjects to carefully consider properties of materials not often noticed. Not many people take the time to consider the difference in frequency distributions between the sounds of porcelain and metal, but that distinction could have been important for these tasks. Within such a context, a 78% rate for identifying the correct material out of ten options appears rather promising, and significantly better than random selection.

5.3. Normal and Relief Comparison User Study

We now move on to discuss a second, separate user study. In order to evaluate the effectiveness of the relief map representation, we conducted another user study where subjects compared normal mapped surfaces to relief mapped surfaces. Since the previous study found most of the benefit in the subjects' perception of the surface, this study was largely designed to test the perceptual aspects of these representations.

5.3.1. Set-up

Twenty-two subjects volunteered to participate in this study, primarily students with computer literacy in the age between 20 to 30. The subjects were allowed to interact with six textured surfaces, where, for each subject, three textures were randomly

	Always	Frequently	Occasionally	Rarely	Never	Reported accuracy (1–10)
Haptics	88%	0%	6%	0%	6%	9.3 ± 0.9
Sound	34%	22%	22%	11%	11%	7.6 ± 1.4
Physics	29%	6%	47%	6%	12%	7.3 ± 2.6

Table 3: Texture identification study: Results from question asking how often subjects used each mode of interaction and question asking how well each mode represented the materials (10 is very accurate).

Figure 6: The available materials for the normal and relief map comparison user study. Material 2 and 5 sounded like stone; 3 sounded like ceramic tile; 4 sounded like metal; 1 and 6 sounded like wood.

selected to use the normal map representation and the remaining three used the relief map representation. Much like in the previous user study, subjects controlled the PHANToM, which corresponded to a virtual pen that could strike the surface or a rolling ball. Through this interaction the subjects would feel the surface, watch the ball roll across the surface, and hear sound synthesized from the surface. Subjects were given as much time as needed to interact with the textured surfaces, and were able to switch between textures at will. Feedback was obtained through a questionnaire in which subjects evaluated each texture, rating the perceived realism of the visual appearance, how well each mode of interaction matched what they would expect from the visual appearance, and the overall quality of interaction.

Figure 6 shows the relief map versions of each surface chosen for the user study. These were selected to provide a range of complexity, depth, and materials. The subjects were allowed to spend as much time as needed to properly evaluate each surface.

The subjects were not informed that some surfaces would have relief maps and some would have normal maps, nor were they specifically told to consider the depth of the surface. Furthermore, no subject ever saw both the normal and relief versions of the same surface, always one or the other. With the subjects largely going into the study unaware of the multiple representations, we pose the following questions:

- With this scenario, do the subjects find the relief maps more accurate and realistic? If not, do they instead significantly prefer the normal maps, or are the two representations indistinguishable?
- Do subjects interacting with a relief mapped surface rate it more highly than the subjects interacting with its normal map equivalent?
- How much, if any, does depth information help with reduction of sensory conflict?

5.3.2. *Experimental Results*

A general way to look at the results is to, for each question, compare all responses (across all surface materials) to use of normal maps vs. use of relief maps. This way can provide a general idea of which texture representation was preferred for each mode of interaction. When subjects were asked how realistic the surfaces appeared, how much the ball physics matched their expectations, and how much the synthesized sound matched their expectations, there was no significant difference between normal maps and relief maps ($p >> .05$). Cohen's effect size for each of these was no greater than 0.11, further indicating little distinction between the texture representations.

When subjects were asked how well the haptics matched their expectations, there was weak evidence showing that subjects preferred the relief maps ($p \approx .053$), and Cohen's effect size of .34 indicates some moderate preference of relief maps. However, when subjects reported their overall perceived quality of interaction, they significantly favored relief maps over normal maps ($p < .05$), with Cohen's effect size of .36 further suggesting a moderate preference of relief maps.

In Table 5, we show the results from comparing the two versions of each texture to one another. For each of the six surfaces, the ratings from the subjects who were given the normal map version are compared to the ratings from the subjects who were given the relief map version, and the table presents the p values and effect sizes for each category the subjects were questioned about. See the beginning of Section 5.2.3 for a brief description of effect size. Notice that the results vary largely from surface to surface.

Recall that, out of the six surfaces each subject experienced, three at random were chosen to be normal maps and the other three were relief maps. Comparing each subject's average normal map rating to that same subject's average relief map rating, we found that each subject tended to prefer their three relief maps over their three normal maps ($p < .05$).

5.3.3. *Analysis*

We can now revisit our originally posed questions, which each involve different means of analyzing the data:

Accuracy and realism of relief maps. In order to assess the overall quality of interaction with relief maps, we can consider

		Surface					
		1	2	3	4	5	6
Visuals	p	.03	.61	.96	.14	.21	.66
	d	.84	-.21	.03	.65	-.57	.18
Physics	p	.80	.64	.38	.83	.08	.56
	d	-.1	.20	-.4	.09	-.78	.25
Sound	p	.31	.84	.47	.27	.07	.14
	d	-.45	-.09	-.34	.49	-.83	.65
Haptics	p	.03	.70	.77	.03	.002	.002
	d	.9	.16	.16	1.03	-1.42	1.44
Overall	p	.2	.68	.92	.08	.14	.02
	d	.52	.18	.05	.80	-.65	1.02

Table 5: For each of the six surfaces, subjects interacted with either the normal or relief map version of that surface's texture. This table contains results of t-tests for each surface and each modality determining whether there are significant differences between the subjects' responses for each texture representation. A small p indicates a statistically significant difference. A positive d value indicates that subjects prefer the relief map version; negative indicates a preference for the normal map.

the data in aggregate, regardless of surface or user. Based on the subjects' ratings of the surfaces' overall quality across all surfaces, on average subjects preferred relief maps over normal maps. We also know that, despite not being informed of the multiple representations, subjects significantly preferred their three randomly selected relief maps over their three normal maps. This neglects the subjects' opinions on individual modes of interaction, but that will be discussed later in the context of sensory conflict. When considered as a whole, relief maps were considered to be of somewhat better overall quality.

Comparisons between normal and relief map versions of the same surface. In order to see how subjects compared different versions of the same surface, we now focus on the data in Table 5, which groups ratings by surface. When broken up in this way, we now see that results varied greatly from surface to surface. For most surfaces and most modes of interaction, the differences in ratings were not statistically significant, and the effect sizes ranged from medium preference of the normal map to medium preference of the relief map. Certain textures therefore may be more suitable for representation as relief maps than others. For example, subjects often commented that haptics and ball physics were unrealistic near vertical edges in a relief map (likely due to limitations of directional penetration depth). Surface five contained many prominent near-vertical edges, and subjects strongly preferred the normal map version. Even though there is an average preference for relief maps across all surfaces, this and other situational reasons for preferring a particular representation mean that the choice of representation may need to be considered on a case-by-case basis.

Reduction of sensory conflict. In order to assess sensory conflict, we now see if the results indicate that the experience as a whole was more appealing than each separate modality would indicate. Preferences were mixed when subjects were told to rate a specific mode of interaction, but they rated the overall quality of relief maps to be significantly higher than normal maps. This suggests that when interacting with multiple modes of interaction simultaneously, relief maps appear to produce more cohesive multimodal interaction than normal maps. Normal vectors already provided most of the cues for depth and curvature, so adding depth information in the form of a relief map only had a small effect on any one mode of interaction. It is only when all modes are considered together that the combined effect is significantly larger. While the overall quality of interaction with reliefs maps may only be moderately better on average and dependent on traits of the surface itself, this reduction in sensory conflict provides its own, possibly subconscious, advantages.

5.4. Discussion

5.4.1. Applications

We demonstrate several possibilities on the potential use of normal and relief maps as unified representations for accelerating multimodal interaction in the supplementary video. Given the prevalence of texture mapping in numerous interactive 3D graphics applications (e.g. games and virtual environment systems), our techniques enable the users to interact with textured objects that have extremely simple underlying geometry (such as flat surfaces) so that they would be able to observe *consistent* dynamic behaviors of moving textured objects, hear the resulting sounds from collisions between them, and feel the object contacts, as shown in Figure 7 (top row). The example of the simplified pinball game in Figure 7 (bottom right), balls rolling down Lombard Street in San Francisco City in Figure 8, and letter blocks sliding down sloped surfaces with noise or obstacles in Figure 7 are a few additional examples, where texture maps can be incorporated into physics simulation with multimodal display to provide a more cohesive, immersive experience without sensory disparity. Please see the supplementary video for demonstration of these results.

5.4.2. Comparison with Level-of-Detail Representations

While we have shown comparisons between normal maps and high-resolution meshes as representations of fine detail, using multiple levels-of-detail when appropriate can also improve runtime performance [28, 29, 30]. These LOD meshes can also reduce the complexity of the geometry while trying to retain the most important features, as determined by perceptual metrics. Since human perception is limited, there may be no significant perceptual benefit in using meshes past a certain quality, in which case the simplified version could be used throughout for significant performance gain.

However, there would be a number of challenges to overcome in designing a multimodal LOD system. The metrics defining important visual features are known to be different than the metrics defining important haptic features [31]. It remains an open problem to create metrics for selecting important audio features for switching between LODs in a multimodal system. Furthermore, the haptic LOD meshes are different from

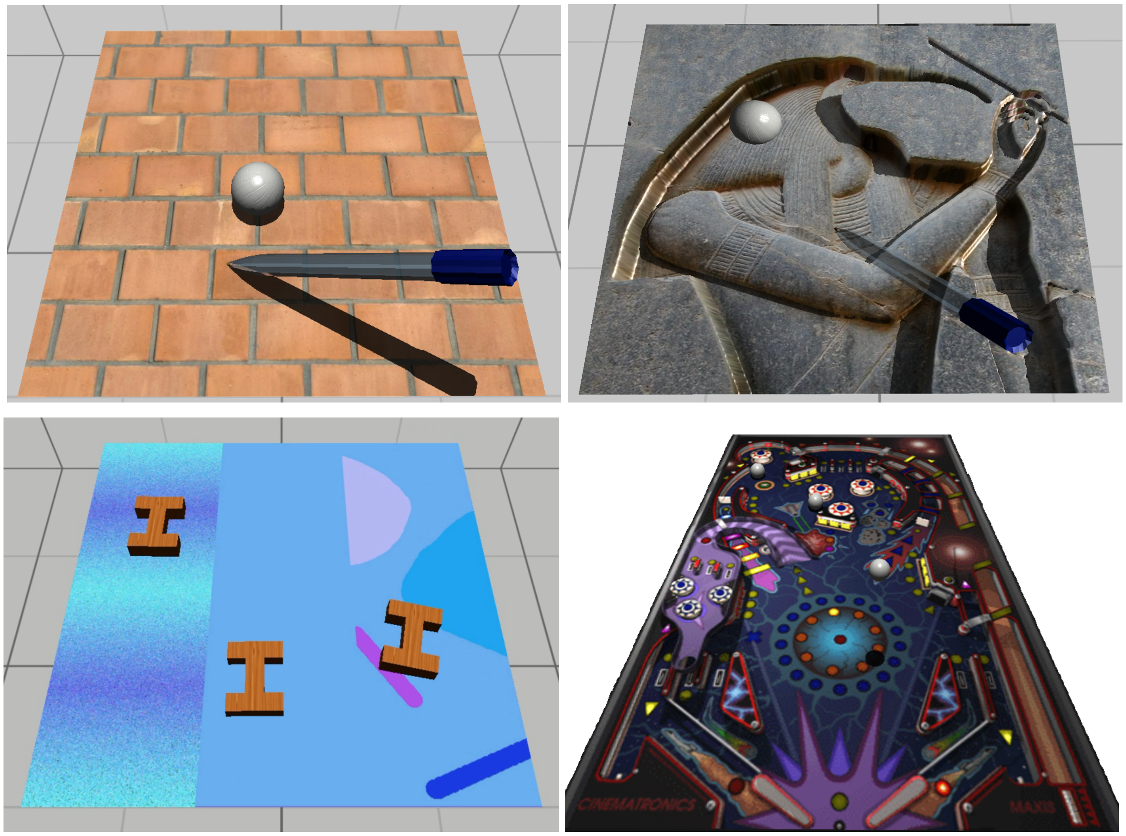

Figure 7: A selection of applications based on our system: a virtual environment with multimodal interaction with a normal map used in the texture identification user study (top left), multimodal interaction with a relief map used in the normal and relief map comparison user study (top right), letter blocks sliding down a normal-mapped surface (bottom left), and a pinball simulation on a normal-mapped flat plane (bottom right).

LOD meshes for visual rendering [31], leading to significantly higher memory requirements than texture-based representation in general.

6. Conclusion

In this paper, we presented an integrated system for multimodal interaction with textured surfaces. We demonstrated that normal maps and relief maps can be used as unified representations of fine surface detail for visual simulation of rigid body dynamics, haptic display and sound rendering. We showed that in a system which uses normal maps to present fine detail to subjects through multiple modes of interaction, subjects are able to combine this information to create a more cohesive mental model of the material they are interacting with. Our first user evaluation result further provides validation that our system succeeded in reducing sensory conflict in virtual environments when using texture maps. Our second user evaluation result demonstrates that relief maps, when chosen carefully, may produce a further reduction in sensory conflict.

We have now explored two different texture representations of fine detail, but some limitations should be addressed. Our current implementation and studies limited the texture-mapped surfaces to single flat planes and we assume our multimodal method would translate gracefully to more complex shapes, as techniques exist for *visually* rendering relief maps on arbitrary polygonal surfaces [7]. We have also only been detecting collisions between static relief-mapped surfaces and dynamic *non-relief-mapped* objects. A more generalized and versatile system could consider the texture of both colliding textured objects, even if both are dynamic, although performance may become more of a limitation. Vectorial textures may be used to help reducing the aliasing artifacts of relief maps in better rendering sharp edges. Additionally, our choice of haptic device has limited our results to 3-DOF force feedback, though it should be possible to compute torques with a slight extension of our method.

For future research, it may be possible to explore the integration of material perception [32, 33] for multimodal displays based on some of the principles described in this paper. Future work may also attempt to generalize this system by addressing the limitations described. We hope this work will lead to further interest in development of techniques on minimizing sensory conflicts when using texture representations for interactive 3D graphics applications, like AR and VR systems.

Acknowledgments

This research is supported in part by National Science Foundation and the UNC Arts and Sciences Foundation.

[1] Cook RL. Shade trees. In Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '84; New York, NY, USA: ACM. ISBN 0-89791-138-5; 1984, p. 223–31. URL: `http://doi.acm.org/10.1145/800031.808602`. doi:10.1145/800031.808602.

Figure 8: Lombard street color map with normal map (left) and mapped to a plane with rolling balls (right).

[2] Blinn JF. Simulation of wrinkled surfaces. SIGGRAPH Comput Graph 1978;12(3):286–92. URL: `http://doi.acm.org/10.1145/965139.507101`. doi:`10.1145/965139.507101`.

[3] Cohen J, Olano M, Manocha D. Appearance-preserving simplification. In Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '98; New York, NY, USA: ACM. ISBN 0-89791-999-8; 1998, p. 115–22. URL: `http://doi.acm.org/10.1145/280814.280832`. doi:`10.1145/280814.280832`.

[4] Kaneko T, Takahei T, Inami M, Kawakami N, Yanagida Y, Maeda T, et al. Detailed shape representation with parallax mapping. In In Proceedings of the ICAT. 2001, p. 205–8.

[5] Tevs A, Ihrke I, Seidel HP. Maximum mipmaps for fast, accurate, and scalable dynamic height field rendering. In Proceedings of the 2008 Symposium on Interactive 3D Graphics and Games. I3D '08; New York, NY, USA: ACM. ISBN 978-1-59593-983-8; 2008, p. 183–90. URL: `http://doi.acm.org/10.1145/1342250.1342279`. doi:`10.1145/1342250.1342279`.

[6] Oliveira MM, Bishop G, McAllister D. Relief texture mapping. In Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '00; New York, NY, USA: ACM Press/Addison-Wesley Publishing Co. ISBN 1-58113-208-5; 2000, p. 359–68. URL: `http://dx.doi.org/10.1145/344779.344947`. doi:`10.1145/344779.344947`.

[7] Policarpo F, Oliveira MM, Comba JaLD. Real-time relief mapping on arbitrary polygonal surfaces. In Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games. I3D '05; New York, NY, USA: ACM. ISBN 1-59593-013-2; 2005, p. 155–62. URL: `http://doi.acm.org/10.1145/1053427.1053453`. doi:`10.1145/1053427.1053453`.

[8] Otaduy M, Jain N, Sud A, Lin M. Haptic display of interaction between textured models. In IEEE Visualization Conference. 2004, p. 297–304. doi:`10.1109/VISUAL.2004.37`.

[9] Ren Z, Yeh H, Lin M. Synthesizing contact sounds between textured models. In Virtual Reality Conference (VR), 2010 IEEE. 2010, p. 139–46. doi:`10.1109/VR.2010.5444799`.

[10] Sterling A, Lin MC. Integrated multimodal interaction using normal maps. In Proceedings of the 41st Graphics Interface Conference. GI '15; Toronto, Ont., Canada, Canada: Canadian Information Processing Society. ISBN 978-0-9947868-0-7; 2015, p. 33–40. URL: `http://dl.acm.org/citation.cfm?id=2788890.2788898`.

[11] Szirmay-Kalos L, Umenhoffer T. Displacement mapping on the GPU - State of the Art. Computer Graphics Forum 2008;27(1).

[12] Nykl S, Mourning C, Chelberg D. Interactive mesostructures. In Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games. I3D '13; New York, NY, USA: ACM. ISBN 978-1-4503-1956-0; 2013, p. 37–44. URL: `http://doi.acm.org/10.1145/2448196.2448202`. doi:`10.1145/2448196.2448202`.

[13] Ho CH, Basdogan C, Srinivasan M. A ray-based haptic rendering technique for displaying shape and texture of 3d objects in virtual environments. In ASME Winter Annual Meeting. 1997,.

[14] Ho CH, Basdogan C, Srinivasan MA. Efficient point-based rendering techniques for haptic display of virtual objects. Presence: Teleoper Virtual Environ 1999;8(5):477–91. URL: `http://dx.doi.org/10.1162/105474699566413`. doi:`10.1162/105474699566413`.

[15] Theoktisto V, Gonzlez MF, Navazo I. Hybrid rugosity mesostructures (hrms) for fast and accurate rendering of fine haptic detail. CLEI Electron J 2010;:–1.

[16] Galoppo N, Tekin S, Otaduy MA, Gross M, Lin MC. Interactive haptic rendering of high-resolution deformable objects. In Proceedings of the 2Nd International Conference on Virtual Reality. ICVR'07; Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-540-73334-8; 2007, p. 215–33. URL: `http://dl.acm.org/citation.cfm?id=1770090.1770116`.

[17] Minsky M, Ming Oy, Steele O, Brooks Jr. FP, Behensky M. Feeling and seeing: Issues in force display. SIGGRAPH Comput Graph 1990;24(2):235–41. URL: `http://doi.acm.org/10.1145/91394.91451`. doi:`10.1145/91394.91451`.

[18] Minsky MDRR. Computational haptics: The sandpaper system for synthesizing texture for a force-feedback display. Ph.D. thesis; Cambridge, MA, USA; 1995. Not available from Univ. Microfilms Int.

[19] van den Doel K, Pai DK. The sounds of physical shapes. Presence 1996;7:382–95.

[20] O'Brien JF, Shen C, Gatchalian CM. Synthesizing sounds from rigid-body simulations. In Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. SCA '02; New York, NY, USA: ACM. ISBN 1-58113-573-4; 2002, p. 175–81. URL: `http://doi.acm.org/10.1145/545261.545290`. doi:`10.1145/545261.545290`.

[21] Raghuvanshi N, Lin MC. Interactive sound synthesis for large scale environments. In Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games. I3D '06; New York, NY, USA: ACM. ISBN 1-59593-295-X; 2006, p. 101–8. URL: `http://doi.acm.org/10.1145/1111411.1111429`. doi:`10.1145/1111411.1111429`.

[22] Doel KVD, Kry PG, Pai DK. Foleyautomatic: Physically-based sound effects for interactive simulation and animation. In in Computer Graphics (ACM SIGGRAPH 01 Conference Proceedings. ACM Press; 2001, p. 537–44.

[23] Hayward V. A brief taxonomy of tactile illusions and demonstrations that can be done in a hardware store. Brain Research Bulletin 2008;75(6):742 –52. URL: http://www.sciencedirect.com/science/article/pii/S0361923008000178. doi:http://dx.doi.org/10.1016/j.brainresbull.2008.01.008; special Issue: Robotics and Neuroscience.

[24] Massie TH, Salisbury JK. The phantom haptic interface: A device for probing virtual objects. In Proceedings of the ASME winter annual meeting, symposium on haptic interfaces for virtual environment and teleoperator systems; vol. 55. Chicago, IL; 1994, p. 295–300.

[25] Taylor II RM, Hudson TC, Seeger A, Weber H, Juliano J, Helser AT. Vrpn: a device-independent, network-transparent vr peripheral system. In Proceedings of the ACM symposium on Virtual reality software and technology. ACM; 2001, p. 55–61.

[26] Cook PR, Scavone GP. The synthesis toolkit (stk). In In Proceedings of the International Computer Music Conference. 1999,.

[27] Nakagawa S, Cuthill IC. Effect size, confidence interval and statistical significance: a practical guide for biologists. Biological Reviews 2007;82(4):591–605. URL: http://dx.doi.org/10.1111/j.1469-185X.2007.00027.x. doi:10.1111/j.1469-185X.2007.00027.x.

[28] Otaduy MA, Lin MC. Sensation preserving simplification for haptic rendering. ACM Trans on Graphics (Proc of ACM SIGGRAPH) 2003;:543–53.

[29] Otaduy MA, Lin MC. CLODs: Dual hierarchies for multiresolution collision detection. Eurographics Symposium on Geometry Processing 2003;:94–101.

[30] Yoon S, Salomon B, Lin MC, Manocha D. Fast collision detection between massive models using dynamic simplification. Eurographics Symposium on Geometry Processing 2004;:136–46.

[31] Otaduy MA, Lin MC. Sensation preserving simplification for haptic rendering. In ACM SIGGRAPH 2005 Courses. SIGGRAPH '05; New York, NY, USA: ACM; 2005,URL: http://doi.acm.org/10.1145/1198555.1198607. doi:10.1145/1198555.1198607.

[32] Ren Z, Yeh H, Klatzky R, Lin MC. Auditory perception of geometry-invariant material properties. Proc of IEEE VR 2013;.

[33] Ren Z, Yeh H, Lin MC. Example-guided physically-based modal sound synthesis. ACM Trans on Graphics 2013;32(1):Article No. 1.