

# Transition Contour Synthesis with Dynamic Patch Transitions

**John Ferraris**

Department of Engineering &  
Computing  
Bournemouth University  
jferraris@bournemouth.ac.uk

**Feng Tian**

Department of Engineering &  
Computing  
Bournemouth University  
ftian@bournemouth.ac.uk

**Christos Gatzidis**

Department of Engineering &  
Computing  
Bournemouth University  
cgatzidis@bournemouth.ac.uk

## ABSTRACT

In this paper we present a novel approach for modulating the shape of transitions between terrain materials to produce detailed and varied contours where blend resolution is limited. Whereas texture splatting and blend mapping add detail to transitions at the texel level, our approach addresses the broader shape of the transition by introducing intermittency and irregularity. Our results have proven that enriched detail of the blend contour can be achieved with a performance competitive to existing approaches without additional texture, geometry resources or asset preprocessing. We achieve this by compositing blend masks on-the-fly with the subdivision of texture space into different sized patches to produce irregular contours from minimal artistic input. Our approach is of particular importance for applications where GPU resources or artistic input is limited or impractical.

## Author Keywords

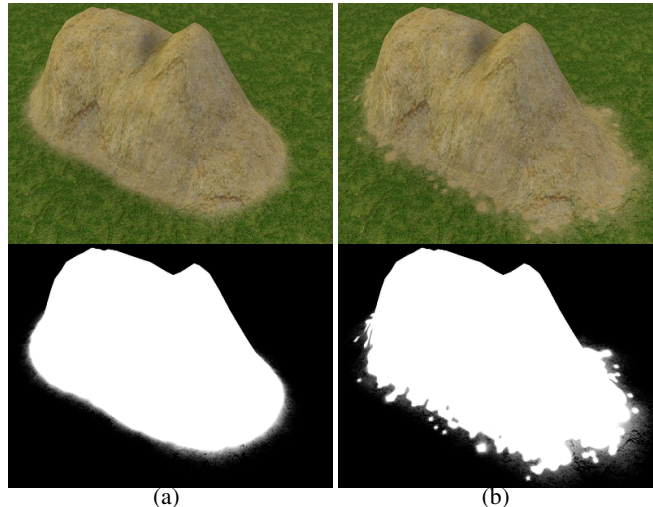
Texture mapping; terrains; real-time graphics

## ACM Classification Keywords

I.3.7 Computer Graphics: Three-Dimensional Graphics and Realism

## INTRODUCTION

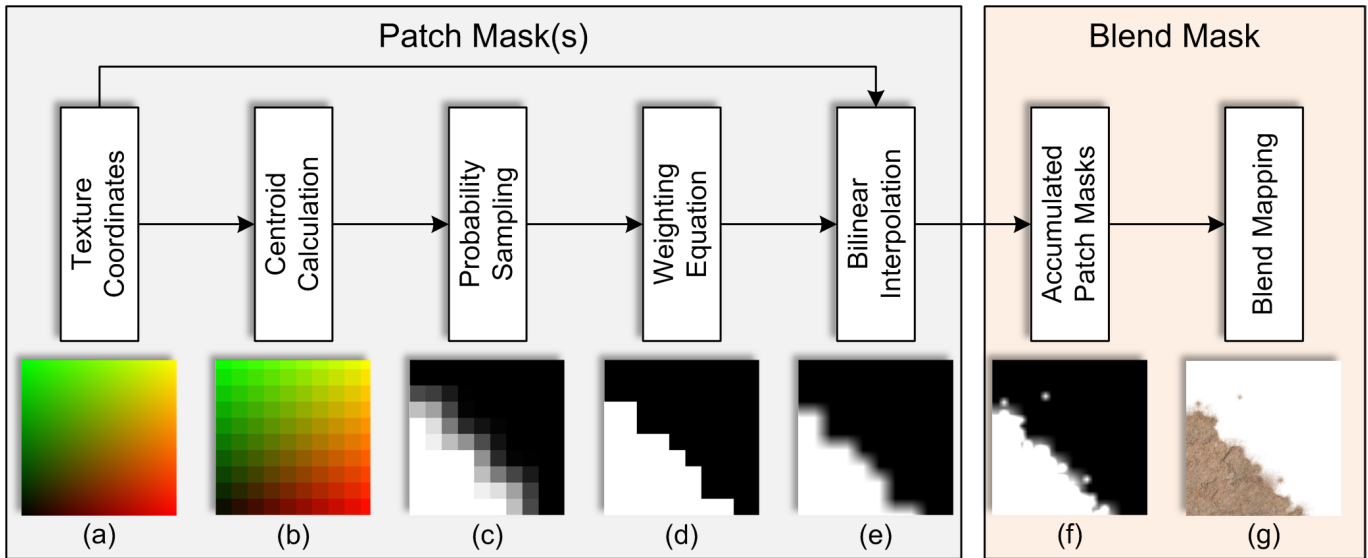
Cutting edge, real-time 3D graphics has always been about the trade-off between processing power and realism [17]. Terrain rendering in particular presents a number of challenges due to the vastness and irregularity of outdoor environments. As such, much research has been conducted into maximizing the detail of a scene from the finite resources of GPU hardware. Texture mapping especially remains essential for implying a degree of tessellation far greater than the underlying mesh. Advanced techniques such as normal [15, 18], relief [13, 16] and parallax mapping [9, 19] take this even further, conveying a sense of spaciousness and depth beyond that of conventional texture mapping.



**Figure 1** A blend between a rock material and grass underlay using slope information for the blend weight with Blend Maps (a) and Dynamic Patch Transitions (b). The bottom row demonstrates the blend masks generated by each algorithm.

In this paper we propose a novel approach for transition contour synthesis with dynamic patch transitions (DPT). DPT builds on the previous work of Feature-Based Probabilistic Blending (FBPB) [6], expanding the concept of probabilistic blending to present a general purpose algorithm for material transitions with textures which contain non-salient details. The work presented here is an extension of a short paper recently published in CGI 2013 [7], with this paper detailing our approach, addressing minification aliasing and contrasting the performance and results of our algorithm to Blend Maps [8].

A typical approach to increase the quality of transitions is to modulate the low frequency blend weight with a material's high frequency alpha map [8]. Whereas this technique adds texel-level accuracy to transitions (Figure 1(a)), DPT further increases the resolution of the blend by addressing the broader shape and contour (Figure 1(b)), generating stochastic, detailed material transitions of near-endless variations ranging from sporadic, intermittent transitions across flat plains to sharp, sudden transitions where abruptness is required.



**Figure 2** The grey shaded area shows the steps performed to synthesize each patch mask (in this case illustrating the largest patch size used for the DPT blend in Figure 1) whilst the light brown shaded area shows the final steps of summing the patch masks (in this case the two different patch sizes in the DPT blend in Figure 1) and applying the blend map to synthesize the final blend mask.

The generation of the transitions themselves is entirely automated, reducing the workload of the artist and making it suitable for procedural applications. Unlike mapping large, offline texture sets using virtual texturing, DPT does not require the storage or bandwidth necessary to stream such textures into memory in real-time. As DPT does not need any additional assets, it is particularly useful for environments where video memory is limited, such as hand-held devices and very large terrains. By combining patches of different sizes and parameters, complex transitions suitable for a wide range of terrain materials can be produced.

## RELATED WORK

### Texture Splatting

Texture splatting [2] remains a popular technique for seamless terrain texturing due to its efficiency and simplicity [3], allowing artists to ‘paint’ terrains in an intuitive manner using interactive tools [4, 5] or off-the-shelf image editing software. The quality and control of the blend however is limited by the resolution of the alpha blend masks, producing noticeable translucency artefacts (especially for salient material features). As such, efforts have been made to reduce these artefacts for salient information [8] or eliminate them entirely [6] by identifying salient details (‘features’) in real-time and blending them with full opacity in a stochastic manner. DPT instead increases the apparent resolution of the blend by introducing detailed variations along the contour of transitions, whilst at the same time keeping the performance and additional memory overhead consistent and competitive with conventional techniques.

Procedurally generating these blend masks in real-time [1] can greatly reduce the texture memory overhead. However, the resolution of the blend masks becomes coupled with the resolution of the terrain geometry. To circumvent this, noise

is added to increase fidelity and intermittency at material transitions. Whilst this approach allows for vast terrains to be textured automatically and efficiently, areas such as flatlands which contain little or no height or slope information will still require conventional blend masks if multiple materials are desired. DPT instead decouples the detail of the blend from the terrain geometry itself, allowing for highly varied transitions at any point on the mesh.

### Virtual Texturing

The use of virtual textures [14, 21, 22] for terrain texturing eliminates the need for tiled textures entirely, streaming instead extremely large and high resolution texture data into memory on an on-demand basis. For large terrains, these generated virtual textures consume large volumes of disk space, with compressed texture sets measuring in the tens of gigabytes [10]. The task of manually generating textures for such terrains is a highly skilled and time consuming process as every transition between materials must be painted by hand. Whilst DPT is an extension of texture splatting, it can be adapted for virtual texturing to help automate the task of generating detailed transitions between material textures by producing a wide variety of transitions that artists can tweak and adapt by hand if required for further control over the final result.

### Texture Synthesis

A significant limitation of conventional texture tiling is its repetitive nature. While texture synthesis techniques offer exciting possibilities for generating terrain materials, only peripheral research within the field directly addresses the challenges of mapping multiple, overlapping materials to a terrain mesh. One such approach is to identify the features of a given terrain and synthesize textures accordingly [23]. Although such an approach reduces the time required for an artist to texture a terrain, the results suffer from the same translucency

artifacts from salient, feature-agnostic linear blending. To address this issue, such material transitions may be synthesized offline [11] or in real-time [12], producing high fidelity transitions between different material types. However, as each possible variation of material transitions must be generated offline and stored in memory, the fidelity comes at a cost of flexibility and memory overhead. DPT instead dynamically generates transition variations independent of the material texture itself, allowing for easy and rapid experimentation with different materials.

### DYNAMIC PATCH TRANSITIONS (DPT)

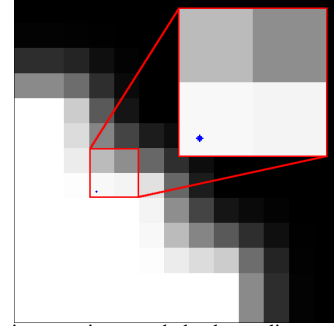
DPT aims to synthesize a high frequency blend mask for each material from the low frequency blend weights of texture splatting [2] in real-time to produce richer and more detailed transition between material textures. The outline of the approach is illustrated in Figure 2. The texture space (a) is divided up into arbitrary-sized patches. The texture coordinates of the patch’s centroid are assigned to each point within a given patch (b) whereupon each point samples the low frequency blend weight texture using these new texture coordinates. This sampled blend value, common to all points within a patch, is interpreted as the probability of a given patch appearing (c). These patches are then drawn or discarded in a probabilistic manner [6] to produce a binary patch mask (d). The patch mask is smoothed at a given point by bi-linearly interpolating the mask value of the patch upon which the point lies with that of the three adjacent patches. The point’s position relative to the area enclosed by the four patches in the range  $[0, 1]$  is used as the interpolation amount between the four patch mask values (e).

This process can be repeated to synthesize multiple patch masks using different patch sizes and parameters to increase the detail and complexity of the transition. The resulting smoothed patch masks are accumulated and clamped within  $[0, 1]$  range (f) before optionally being further modulated by the material’s high frequency blend map texture [8] to produce the final blend mask (g). The details of the approach are given in the subsequent sections below.

The patch dimensions are specified by the artist as the two-dimensional vector  $\vec{D}$  in the range  $(0, 1]$ , which determines the percentage of a material’s texture space a given patch spans and the vector  $\vec{c}$  denotes the centroid of the patch upon which a given fragment lies. Figure 3 illustrates a point on a patch along with the three adjacent patches. To bilinearly interpolate the patch mask at the blue point, the mask values of the three adjacent patches are evaluated along with the mask value of the patch upon which the point resides. For convention, we denote the patch upon which a given point lies as the bottom left (*bl*) of the set of four patches, the bottom right as (*br*), the top right as (*tr*) and the top left as (*tl*).

### Probability Sampling

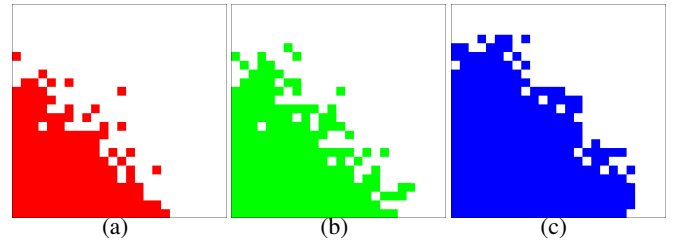
The probability for each of the four patches is obtained by transforming the centroids  $\vec{c}_{bl}$ ,  $\vec{c}_{br}$ ,  $\vec{c}_{tr}$  and  $\vec{c}_{tl}$  from the material’s texture space into the blend weight texture’s space and then sampling the blend weight with these newly transformed



**Figure 3** At any point on a given patch the three adjacent patches are included to form a cluster of four patches.

vectors. The random value is obtained for each patch following this same process, albeit sampling a noise function or texture instead. The four probability and random values are stored in the four component vectors  $\vec{p}$  and  $\vec{r}$  respectively.

The probability value of the patches is modulated by an artist-defined constant  $P$ . By raising the components of vector  $\vec{p}$  to the power of  $P$ , the proliferation of patches can be sustained (when  $P$  is less than 1.0) or dampened (when  $P$  is greater than 1.0). For this see Figure 4. This constant is set for each patch mask allowing for artistic control over the size and shape of the transition. The constant  $R$  modulates  $\vec{r}$  in an identical manner by raising  $\vec{r}$  to the power of  $P$  in order to control the density of the patch mask.



**Figure 4** Patch modulation when  $P$  is 0.25 (a), 1.0 (b) and 3.75 (c).

### Weighting Equation

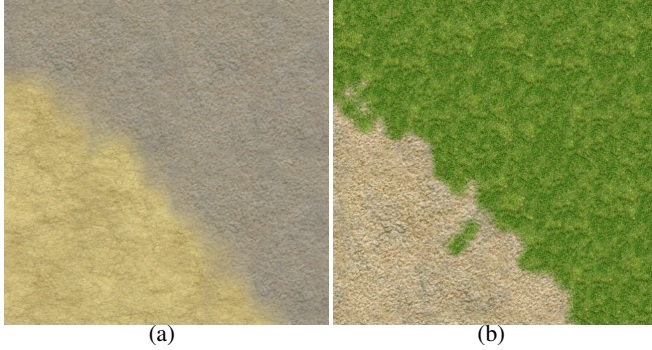
To determine whether a patch will be drawn or be discarded altogether, the random value is checked against the probability of that patch appearing. Should the value lie equal to or under the probability, all points within that patch will contribute the value 1.0 to the binary patch mask, otherwise they will be discarded with no contribution. A simplified version of the weighting equation from FBPB (below) is used to obtain the mask values for the four patches given the probability and random value for each patch. The mask values of each of the four patches are stored in the four component vector  $\vec{m}$ .

$$\vec{m} = \text{sgn}(\lfloor \frac{1}{\vec{r}} \vec{p} \rfloor) \quad (1)$$

For any given point on a patch, the position  $\vec{f}$  within the unit square enclosed by the four patch cluster (Figure 3) is calculated by taking the fractional components of the component-wise product of the vectors  $\vec{uv}$  and  $\vec{D}$ . This new normalized

position is then used to bilinearly interpolate the four patch mask values to obtain a smoothed scalar weight value  $w$  at that point  $uv$  [20].

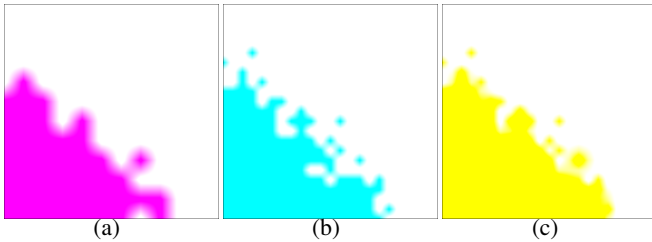
The patch masks can be further shaped by modulating  $w$  with a constant scalar  $W$  that is set for each mask. The weight is raised to the power of  $W$  such that values less than 1.0 sharpen the mask contour whilst values greater soften it. Different patch masks can have different modulation amounts, allowing for a broad range of transition shapes. Figure 5 below illustrates this modulation, with the sand material on the left having a smoother, broader contour of the lower modulation amount, whilst the mud material has a sharper, more distinct contour from the higher amount.



**Figure 5** Different materials with different weight modulation, from the smoother contour of the sand (a) to the sharper contour of the mud (b).

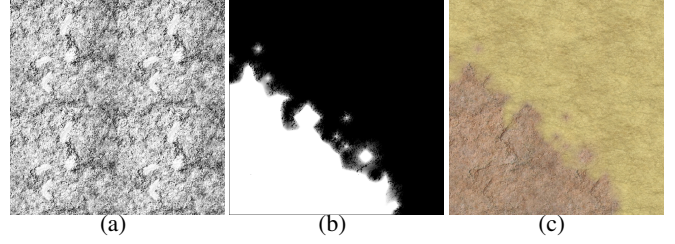
### Blend Mask Composition

The steps in the grey shaded area of Figure 2 can be performed multiple times inside the fragment shader using different patch sizes and parameters to produce different patch masks for more complex and detailed blends. These patch masks are accumulated to compose the final blend mask. In Figure 6, the first patch mask was synthesized using a patch size of 0.9 (a) and the second mask using a size of 0.045 (b), thus the steps in the grey shaded area of Figure 2 are performed twice. These two masks are accumulated to yield a final blend mask weight of  $b$  at any given point (c).



**Figure 6** The smoothed masks of different patch sizes and parameters (a,b) are combined together to produce the final blend mask (c).

To add further details to the transition (Figure 7), the material’s high frequency blend map (a) can be used to modulate the lower frequency blend mask  $b$  to produce the final blend weight (b). This weight is then used as the interpolation amount between the material and the underlying texture (c).



**Figure 7** The material’s blend map (a) can further modulate the blend mask (b) to produce the final blend (c).

The steps to compose the final blend mask (the light brown shaded area in Figure 2) are illustrated in the pseudo-code below, where  $j$  is the number of patch masks to accumulate and PatchMask is a function that synthesizes the patch mask from the artist supplied size and modulation constants for each mask (indexed by the variable  $i$ ). For materials containing high frequency details, the blend mask is then reduced to  $[0, 1]$  range before the material’s blend map is applied with the function BlendMap [8].

---

### Algorithm 1 Overview of Blend Mask Composition

---

```

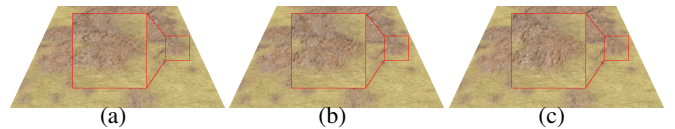
 $b \leftarrow 0.0$ 
for  $i = 0 \rightarrow j$  do
     $b \leftarrow b + \text{PatchMask}(i)$ 
end for
 $b \leftarrow \min(b, 1.0)$ 
 $b \leftarrow \text{BlendMap}(b, \vec{\text{albedo}}_a)$ 

```

---

### Parallax Mapping

Materials using displacement effects such as parallax mapping can benefit from interpreting the smoothed patch masks as height information. This is particularly useful for material textures that contain little in the way of meaningful height data, such as the textures used throughout this paper. In Figure 8, the rock material has been blended three times, first without parallax mapping (a), then with parallax mapping using the texture’s height channel (b) and, finally, using the smoothed patch masks as height maps (c). As the rock texture has little in the way of distinct height details, the magnified element of (a) and (b) looks very similar, with the only difference being the noisy peppering of parallax offsetting from the texture’s height map in the latter. On the other hand, the lump of rock synthesized with the DPT blending process has a recognisable shape and depth to it.



**Figure 8** A transition between rock and sand (the red square is under magnification) using no parallax mapping (a), parallax mapping using a height texture (b) and, finally, using the height and blend mask (c)

The parallax mapping technique is applied after the final blend mask has been synthesized so that the albedo and normal maps are properly offset by the height data. As seen in the



listing below, the heightfield  $h$  accumulates the patch masks in a similar manner to the blend mask, albeit each patch mask is being modulated by the constant  $H$  (set by the artist for each patch mask in the range  $[0, 1]$ ), allowing masks to contribute different amounts to the heightfield. In this particular code listing, Parallax is a function that samples the material’s height field (typically stored in the normal texture) and modulates it with the synthesized height field before performing the parallax operations (omitted for clarity).

---

**Algorithm 2** Parallax Mapping with DPT

---

```

b ← 0
h ← 0
for  $i = 0 \rightarrow j$  do
     $k \leftarrow \text{PatchMask}(i)$ 
     $b \leftarrow b + k$ 
     $h \leftarrow h + kH$ 
end for
 $b \leftarrow \min(b, 1.0)$ 
 $h \leftarrow \min(h, 1.0)$ 
 $\vec{px} \leftarrow \text{Parallax}(u\vec{v}, h, E\vec{y})$ 
 $\vec{albedo} \leftarrow \text{TextureSampler}(\vec{px}, \text{Albedo})$ 
 $\vec{normal} \leftarrow \text{TextureSampler}(\vec{px}, \text{Normal})$ 
b ← BlendMap( $b, \vec{albedo}_a$ )

```

---

**Minification Correction**

As the DPT process alters the shape and content of a texture after the texture has been sampled, texture mipmapping no longer eliminates the aliasing artifacts from undersampling errors so, instead, this undesirable effect must be corrected manually. For materials containing particularly small patch sizes, these artifacts manifest themselves as “swimming pixels” as the camera moves around when the screen-space patch size approaches one pixel. An example of this is shown in Figure 9. Here, the camera has moved backwards slightly between the two images which has caused some of the high frequency details of smaller patches to distort or be removed completely. Whilst it is difficult to spot these artifacts in a still image, this effect is pronounced when viewing animated images of camera movement.



**Figure 9** Camera movement can cause small patches to exhibit undersampling errors.

To eliminate these aliasing artifacts, a transition band is used to derive a distance value  $d$  in the range  $[0, 1]$  which is used to modulate the translucency of a given patch, as described in Equation 2 below. Here, the value  $z$  represents the distance of the fragment from the viewer,  $Offset$  is the distance at which the band starts and  $Size$  is the distance that the band spans.

$$d = \max\left(\frac{z - Offset}{Size}, 0.0\right) \quad (2)$$

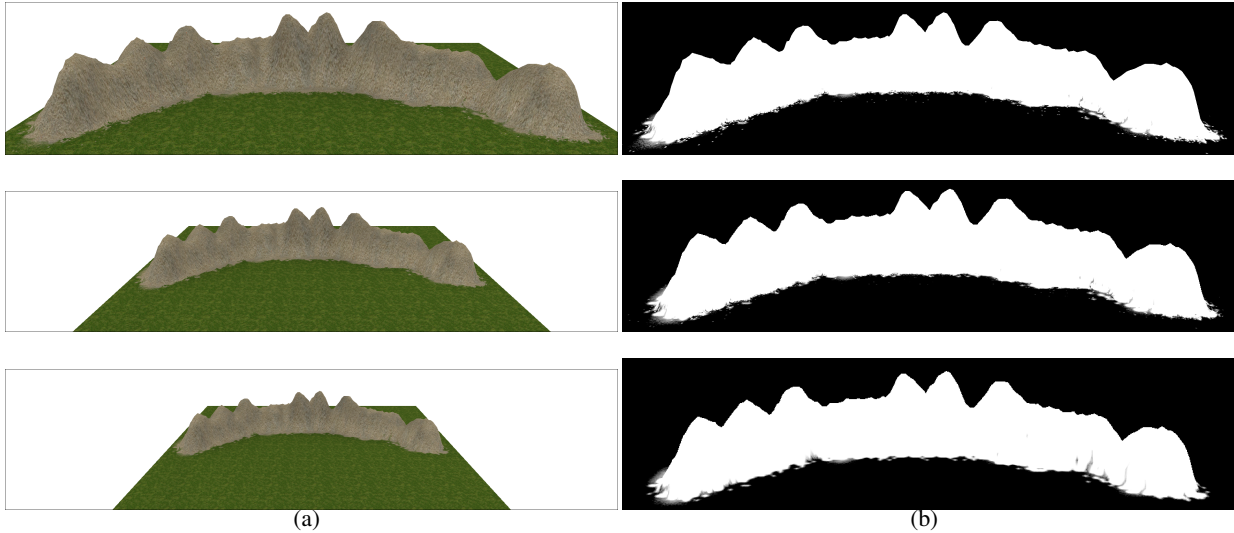
The value  $d$  is then used to modulate a given patch mask value  $w$ , causing it to fade over the duration transition band before being removed completely from the composition of the final blend weight. This can be seen in Figure 10 where the high frequency pebble details are displayed when the terrain is viewed from closer distances but gradually faded over distance to be removed entirely at further distances. The size and offset of transition bands will be entirely dependent on the size of the patch and the size of texture space relative to object/world space (in this particular example, an offset of 75 units was used with a band size of 150 units).

For larger patches, we cannot resort to transitioning them all into translucency as we will be left with none of the original material. However, as the undersampling errors for such patches occur at greater distances we can simply transition the final blend mask  $b$  between DPT and a simple linear blend using the technique described above. As the details of DPT are difficult or impossible to pick out over large distances this also has the advantage of reducing processing overhead by falling back to the simpler linear blending process when such details offer negligible or no benefit to the scene. In Figure 11, we can see a terrain feature viewed from increasing distances (100, 500 and 1000 units respectively). Here, the details of DPT (a) become increasingly difficult to spot compared to the linear blend (b) such that in the final image the differences are near impossible to identify at run-time.

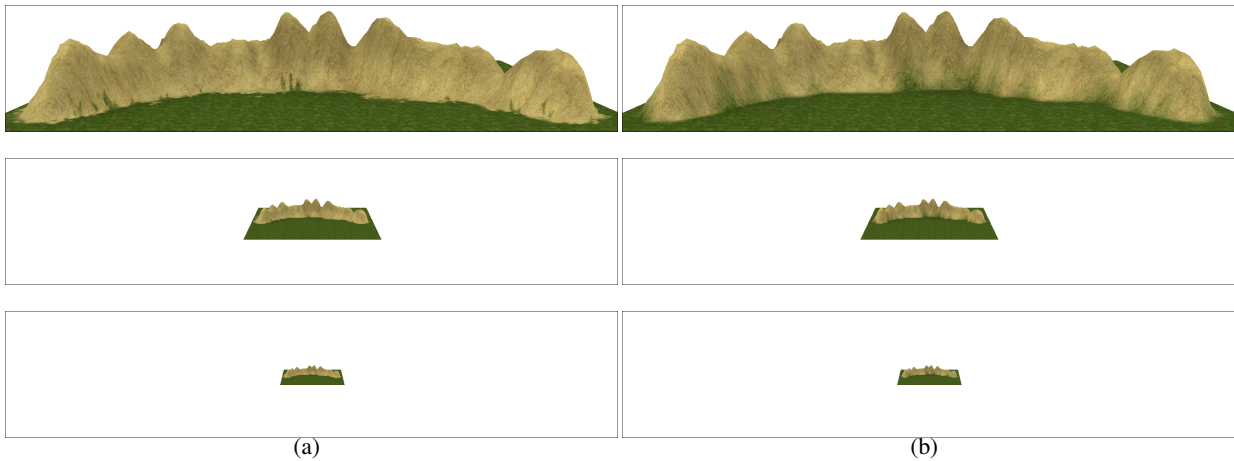
**USING DPT**

DPT is a texture space effect so the size and parameter range will be entirely dependent on the size of texture space and the texture information contained within. However, in our use of DPT we have developed a work flow that should translate to other materials regardless of the actual texture and parameter settings used. Consider Figure 12 which shows the example low resolution weight texture (a) along with a grass underlay texture that span three units of texture space along the x-axis and one unit along the y-axis.

DPT can work with any number of patches although we have found that two is sufficient for most cases. The first patch is used to lay down the body of the transition so we will use a relatively large size of 5 units and a  $P$  value of 6.5 to cover a broad area of the transition (Figure 13(a)). As the material used is sand, we would expect mostly mid and low frequency details so we will soften the patch with a  $W$  value of 0.5 (Figure 13(b)).



**Figure 10** A terrain as viewed from a distance of (from top to bottom) 80, 135 and 170 units from origin (a) along with the associated blend weights (b).



**Figure 11** As the distance from the viewer increases, the details of DPT (a) become increasingly difficult to pick out over a simpler linear blend (b).

With the first patch providing the bulk of the transition, we will use the second patch to add subtle detail and shape to the transition contour. We achieve this by using a far smaller patch size of 12.5 units but with a slightly higher  $P$  value of 7.1 to extend the range of this patch slightly beyond the first patch (Figure 14(a)). We add more definition to this patch layer by increasing its  $W$  value to 3 to sharpen the patch shapes. This combination of sharp, mid frequency detail from this patch with the softer, low frequency detail of the first patch results in a transition that has far more contour detail and variation from the low resolution weight texture that can be obtained from standard texture splatting.

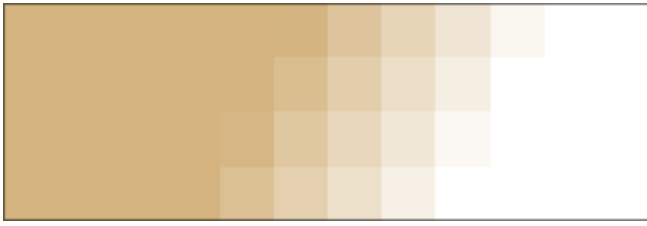
This method of using smaller patches to modulate the contour shape of larger patches is sufficient for most material types, although for materials composed of high frequency detail one patch may be sufficient (Figure 15(a)). However, excessive  $W$  values can produce geometric artifacts as a byproduct of the bilinear interpolation stage of DPT where adjacent patches cause thin spines (15(b)). This is because the result-

ing patch shape from high  $W$  values is too small compared to the unmodulated patch shape. In such cases, one can either instead use smaller patch sizes, reduce the strength of  $W$  modulation or ensure that the proliferation of such patches does not greatly exceed that of any other larger patches.

## RESULTS & ANALYSIS

The Figures in this (and the following) section demonstrate a single terrain material being blended over an underlay material using DPT and Blend Maps. For DPT, two patch masks were accumulated to synthesize the final blend mask. In Figures 16 to 22 a low resolution texture was used as the blend weight source (where a grid of 4x4 weight texels spanned one unit of the material's texture space) whereas Figure 1 instead used the slope information of the underlying geometry.

In all Figures except Figure 20 it can be seen that whilst Blend Maps help to add high frequency detail to the transition, the contour of the transition remains unchanged. On the other hand, DPT utilizes Blend Maps for finer details but also adds

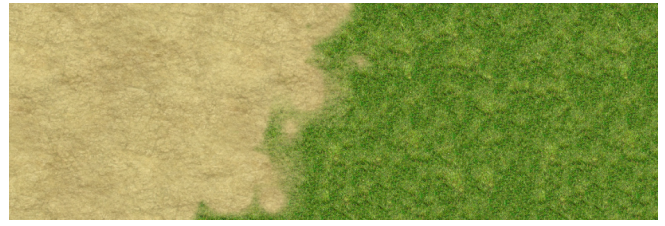


(a)



(b)

**Figure 12** An example weight (a) and underlay texture (b).



(a)



(b)

**Figure 14** A smaller size for the second patch (a) further sharpened with  $W$  modulation (b).



(a)



(b)

**Figure 13** A large initial patch size (a) further softened with  $W$  modulation (b).

stochastic detail and definition to the broader shape of the transition. The flexibility of DPT by using different patch and modulation settings allows for a wide range of materials to be represented. In Figure 16, DPT adds subtle definition and variation to the sand's transition, helping break up the uniformity of the blend. The sharper contour of the DPT transition in Figure 17 is achieved by using higher  $W$  values for each patch mask with the different patch masks adding detail and irregularity to the transition. In contrast, Blend Maps produce the same uniform and monotonous blend contour shape regardless of material.

Figures 18 and 19 demonstrate DPT's effectiveness with materials containing non-salient details. Whereas these details would be too ambiguous and indistinct for techniques such as FBPB [6], they contain enough contour information to exhibit the translucency artefacts demonstrated by Blend Maps tran-

sitions. DPT masks these artefacts by creating the illusion of contour preservation to produce more sporadic, intermittent transition rather than the faded, uniform transitions of Blend Maps.

Figure 20 illustrates the limitations of DPT. Here, the cobble features are too large and salient for DPT to handle correctly as DPT does not take into consideration the topography of the material. As such, the patch shapes bear no correlation to that of the cobble features resulting in a contour shape that is non-sensical and unrealistic. In comparison, Blend Maps handles this particular material with far more plausibility, preserving the feature shapes without disrupting the overall detail of the texture. Instead, such materials would be better handled by an algorithm such as FBPB which can take into consideration the size and shape of salient features and ensure that the integrity of such features is retained at the transitions.

In Figure 21, the transition uses FBPB for the salient cobble features and either DPT or Blend Maps for the grey non-salient mortar. Whereas the uniform contour of Blend Maps exaggerates the translucency artefacts of the non-salient details, DPT completely reshapes the contour into one that is sharper and more ragged. This helps break the uniformity of the transition giving it a more natural, more realistic look. When used in conjunction with one another, FBPB and DPT complement each other, with the former accommodating the more prominent, salient details of a material and the latter introducing variation to the shape and contour of the non-salience that cannot be easily identified by FBPB.

The uniformity of a blend's contour becomes apparent when geographical features protrude from a ground plane, such as in Figures 22 and 1. In Figure 22, Blend Maps exhibit strong banding artefacts around the hill, causing it to float artificially above the ground. DPT's stochastic and detailed blend contour conveys a greater sense of shape to the hill, allowing it to sit more realistically with the grass beneath. In Figure 1 the use of height/slope information rather than a texture as





(a)



(b)

**Figure 15** A single patch can be sufficient in some cases (a) although excessive  $W$  modulation can produce artifacts (b).

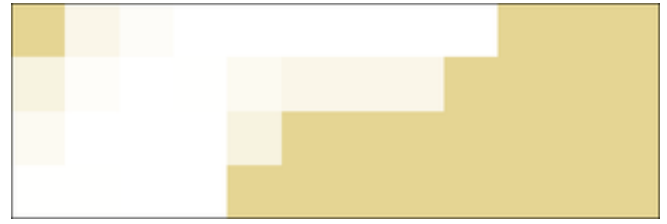
the blend source does help to break up the banding of Blend Maps but the uniformity of the contour shape still looks unnatural. The intermittency of the DPT blend obfuscates the uniformity of the slope information around the base of the hill, conveying far more detail and rooting it naturally to the grassy ground beneath.

## PERFORMANCE ANALYSIS

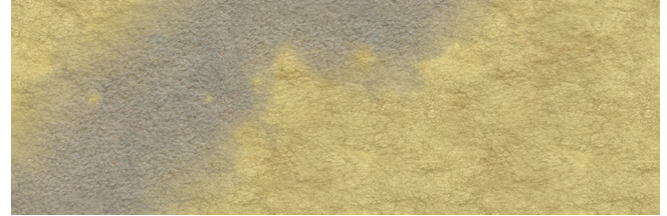
Our performance test case consisted of a multi-pass rendering approach where two terrain materials were blended over a base texture to cover a  $256 \times 256$  terrain mesh completely (with no culling or geometry level of detail optimizations for the mesh itself). The first pass was an early z pre-pass that mapped the base texture onto the mesh with a subsequent blending pass for each terrain material. The material and base textures were mipmapped and sampled with anisotropic filtering. The camera was positioned to fill the viewport (measuring  $1920$  by  $1080$  pixels) entirely with a distribution of near and far fragment data. The hardware upon which the tests were performed on was a Radeon HD6970M GPU in a 2.2 GHz CPU laptop with 8 GB of RAM running Microsoft Windows 7.

The two algorithms tested were Blend Maps and DPT blending. In particular, DPT was tested accumulating one and two patch masks for each terrain material. For each algorithm, three blends were performed: a straight blend, where only the blending process is executed, a normal mapped blend and a normal and parallax mapped blend. As the technique for eliminating minification distortion in DPT can be extended to switch over to linear blending for distance fragments, this basic optimization was used for both algorithms.

Referring to Table 1, it can be seen that DPT performs only slightly slower than Blend Maps in all cases which is to be expected considering the difference in complexity and detail of the transitions. As the normal and parallax mapping techniques were applied, the relative performance of DPT to



(a)



(b)



(c)

**Figure 16** A blend between a sand material and concrete underlay with a low resolution weight texture (a) using DPT (b) and Blend Maps (c).

Blend Maps remained the same, with the single patch mask blend increasing slightly as the overhead of normal and parallax mapping begins to become the bottleneck. After averaging the performance of DPT in relation to Blend Maps for each test case, DPT typically operates at 98% (one patch) or 95% (two patches) of the performance of Blend Maps. Nevertheless, DPT has a number of advantages over Blend Maps. The performance is competitive with Blend Maps without any extra video memory overhead or asset preprocessing. DPT is capable of producing transitions of far greater variation than Blend Maps. As the generation of these stochastic transitions is entirely automated at run-time, artists do not need to manually create transitions for each and every shape and variation for any given material combination.

**Table 1** Example entries for the meta-texture.

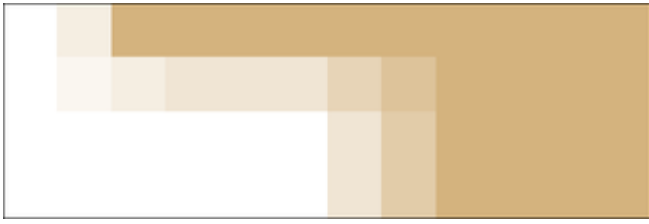
| Algorithm     | Straight |                  | Normal |                  | Parallax |                  |
|---------------|----------|------------------|--------|------------------|----------|------------------|
|               | FPS      | MPS <sup>1</sup> | FPS    | MPS <sup>1</sup> | FPS      | MPS <sup>1</sup> |
| Base Texture  | 650      | 1347.8           | -n/a-  | -n/a-            | -n/a-    | -n/a-            |
| Blend Maps    | 333      | 690.5            | 322    | 667.7            | 311      | 644.9            |
| DPT 1 Patch   | 324      | 671.8            | 315    | 653.2            | 305      | 632.4            |
| DPT 2 Patches | 316      | 655.3            | 305    | 632.4            | 297      | 615.9            |

<sup>1</sup> Pixel throughput (in megapixels per second)

## CONCLUSION & FUTURE WORKS

We have proposed a novel technique for creating irregular blend contours of near-endless variation in real-time that offer far greater resolution using low resolution blend weights than conventional techniques. We achieve this by breaking up





(a)



(b)

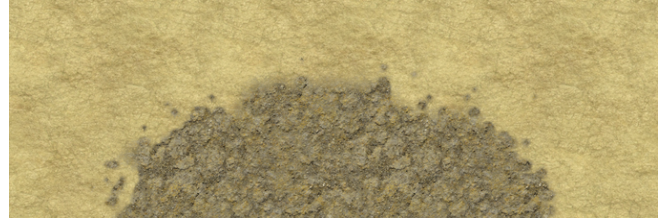


(c)

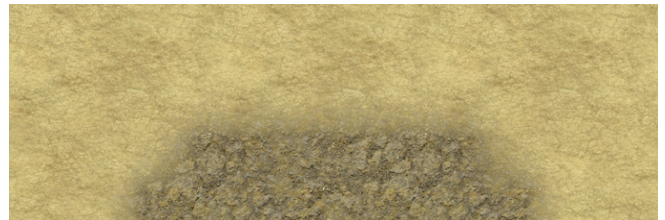
**Figure 17** A blend between a rock material and grass underlay with a low resolution weight texture (a) using DPT (b) and Blend Maps (c).



(a)



(b)



(c)

**Figure 19** A blend between a mud material and sand underlay with a low resolution weight texture (a) using DPT (b) and Blend Maps (c).



(a)

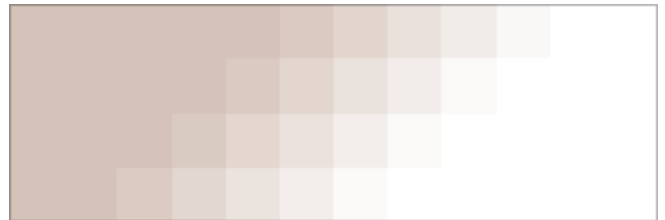


(b)

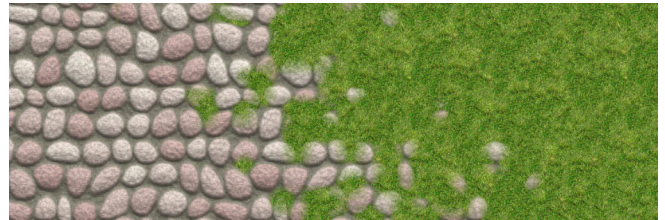


(c)

**Figure 18** A blend between a pebble material and grass underlay with a low resolution weight texture (a) using DPT (b) and Blend Maps (c).



(a)



(b)

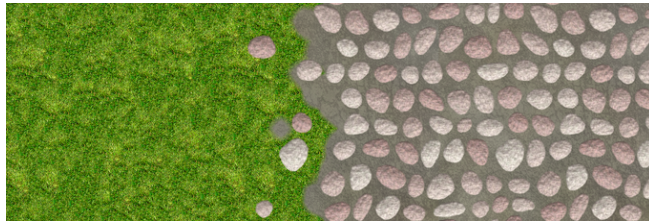


(c)

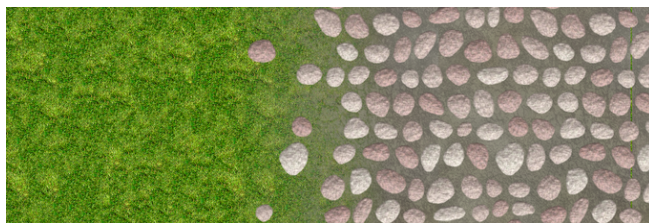
**Figure 20** A blend between a cobble material and grass underlay with a low resolution weight texture (a) using DPT (b) and Blend Maps (c).



(a)



(b)



(c)

**Figure 21** A FBPB blend between a cobble material and grass underlay with a low resolution weight texture (a) using DPT (b) and Blend Maps (c).

the contour of the blend to create detailed and stochastic transitions between materials. This process is entirely automated and offers consistent, competitive performance for no additional video memory overhead. Artists have great freedom to create a wide range of material transitions through patch mask accumulation and parameter modulation. As DPT does not require any pre-processing of assets, any non-salient textures can be used, allowing for instant previewing and appraisal of results.

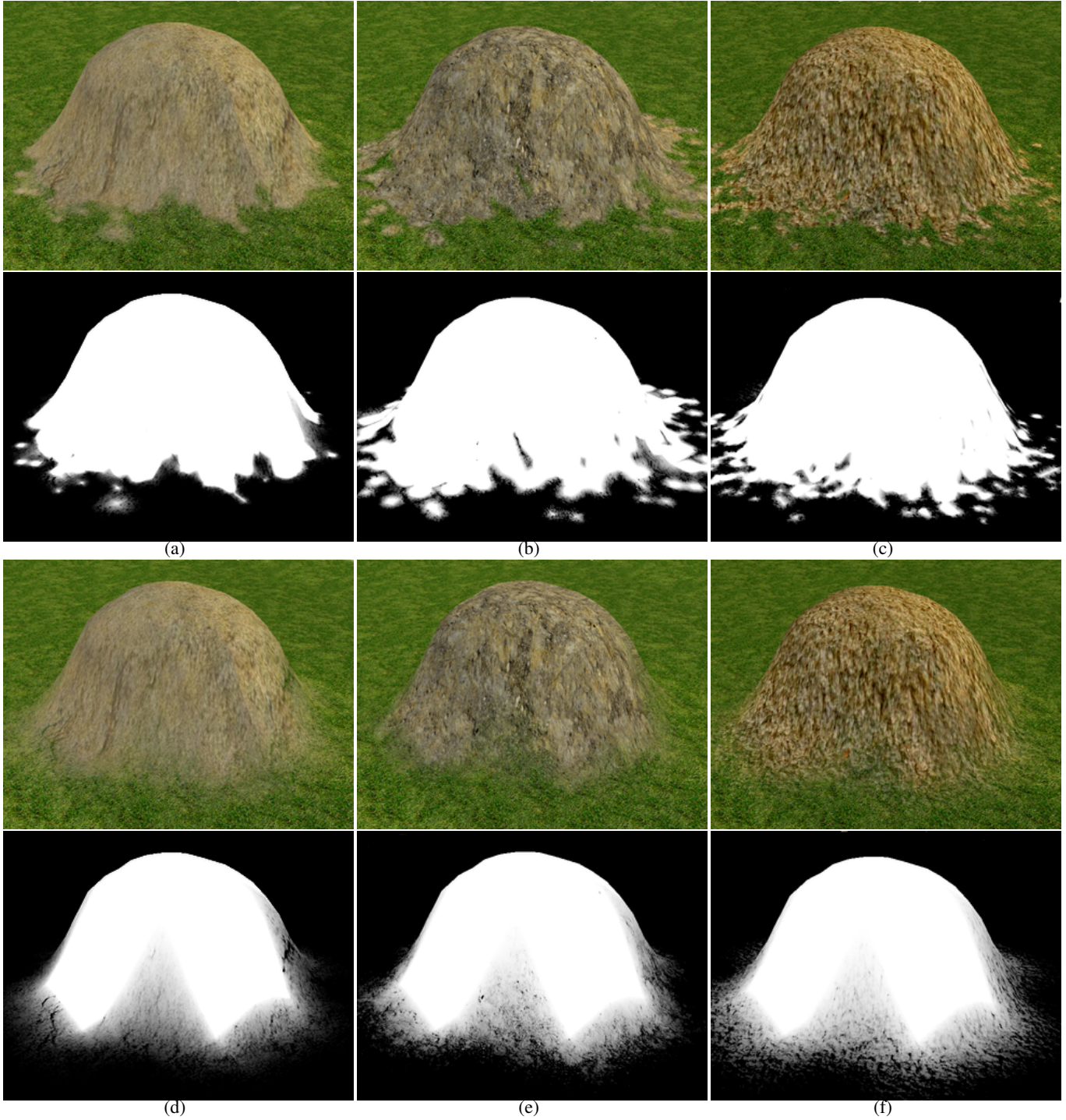
Future work will look into incorporating elements of FBPB's feature awareness to preserve salient contours of the materials themselves to further minimize artefacts caused by abrupt changes in blend weighting mid way through such contours. Furthermore, a user study will be performed as a priority to quantitatively show the improvement of DPT over existing techniques in terms of the resulting transitions and artist work flow.

## REFERENCES

1. Andersson, J. Terrain rendering in frostbite using procedural shader splatting. In *ACM SIGGRAPH 2007 courses*, SIGGRAPH '07, ACM (New York, NY, USA, 2007), 38–58.
2. Bloom, C. Terrain texture compositing by blending in the frame-buffer. <http://goo.gl/xDALP>, Nov. 2000.
3. Catanese, S. A., Ferrara, E., Fiumara, G., and Pagano, F. Rendering of 3d dynamic virtual environments. In *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, SIMUTools '11, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering) (ICST, Brussels, Belgium, Belgium, 2011), 351–358.
4. Crytek. Painting terrain. <http://goo.gl/7OXXKM>, 2011.
5. Epic Games. Setting up terrain in unreal. <http://goo.gl/H3AEv>, 2009.
6. Ferraris, J., Tian, F., and Gatzidis, C. Feature-based probabilistic texture blending with feature variations for terrains. *Computer Animation and Virtual Worlds* 23, 3-4 (2012), 435–445.
7. Ferraris, J., Tian, F., and Gatzidis, C. Automatic terrain texturing with dynamic patch transitions. In *CGI '13: Proceedings of the 2013 Computer Graphics International Conference*, ACM (2013), 471–474.
8. Hardy, A., and Mc Roberts, D. A. K. Blend maps: enhanced terrain texturing. In *Proceedings of the 2006 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*, SAICSIT '06, South African Institute for Computer Scientists and Information Technologists (Republic of South Africa, 2006), 61–70.
9. Kaneko, T., Takahei, T., Inami, M., Kawakami, N., Yanagida, Y., Maeda, T., and Tachi, S. Detailed shape representation with parallax mapping. In *Proceedings of the ICAT 2001* (2001), 205–208.
10. Kooima, R., Leigh, J., Johnson, A., Roberts, D., SubbaRao, M., and DeFanti, T. Planetary-scale terrain composition. *Visualization and Computer Graphics*, *IEEE Transactions on* 15, 5 (sept.-oct. 2009), 719–733.
11. Lai, Y.-Y., Tai, W.-K., Chang, C.-C., and Liu, C.-D. Synthesizing transition textures on succession patterns. In *Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, GRAPHITE '05, ACM (New York, NY, USA, 2005), 273–276.
12. Lefebvre, S., and Neyret, F. Pattern based procedural textures. In *Proceedings of the 2003 symposium on Interactive 3D graphics*, I3D '03, ACM (New York, NY, USA, 2003), 203–212.
13. McGuire, M., and Whitson, K. Indirection mapping for quasi-conformal relief mapping. In *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D '08)* (February 2008).
14. Mittring, M., and GmbH, C. Advanced virtual texture topics. In *ACM SIGGRAPH 2008 classes*, SIGGRAPH '08, ACM (New York, NY, USA, 2008), 23–51.
15. Peercy, M., Airey, J., and Cabral, B. Efficient bump mapping hardware. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co. (New York, NY, USA, 1997), 303–306.

16. Policarpo, F., and Oliveira, M. M. Relief mapping of non-height-field surface details. In *Proceedings of the 2006 symposium on Interactive 3D graphics and games*, I3D '06, ACM (New York, NY, USA, 2006), 55–62.
17. Rong-hua, Z. Real-time optimization technology and its application in terrain rendering. In *Image and Signal Processing (CISP), 2011 4th International Congress on*, vol. 3 (2011), 1349–1352.
18. Sloan, P.-P. Normal mapping for precomputed radiance transfer. In *Proceedings of the 2006 symposium on Interactive 3D graphics and games*, I3D '06, ACM (New York, NY, USA, 2006), 23–26.
19. Tatarchuk, N. Practical parallax occlusion mapping with approximate soft shadows for detailed surface rendering. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, ACM (New York, NY, USA, 2006), 81–112.
20. van Verth, J., and Bishop, L. *Essential Mathematics for Games & Interactive Applications: A Programmer's Guide*. Morgan Kaufmann series in interactive 3D technology. Elsevier, Morgan Kaufmann Publ., 2008.
21. van Waveren, J. id tech 5 challenges: From texture virtualization to massive parallelization. In *ACM Annual SIGGRAPH Conference 2009: Beyond Programmable Shading*, SIGGRAPH '09, ACM (New York, NY, USA, 2009).
22. Widmark, M. Terrain in battlefield 3: A modern, complete and scalable system. In *Game Developers Conference 2012*, GDC '12 (2012).
23. Zhang, H., Ouyang, D., Lin, H., and Guan, W. Texture synthesis based on terrain feature recognition. In *Proceedings of the 2008 International Conference on Computer Science and Software Engineering - Volume 02*, IEEE Computer Society (Washington, DC, USA, 2008), 1158–1161.





**Figure 22** A small hill blended with three different materials over a grass underlay using DPT (a, b & c) and Blend maps (d, e & f) along with the generated blend masks (bottom row).