Image based PTM synthesis for Realistic Rendering of Low Resolution 3D models

Pradeep Rajiv*
Center For Visual Information Technology
International Institute of Information Technology
Hyderabad, India
pradeepr@research.iiit.ac.in

Anoop M Namboodiri
Center For Visual Information Technology
International Institute of Information Technology
Hyderabad, India
anoop@iiit.ac.in

ABSTRACT

Capturing the shape and texture of large structures such as monuments and statues at very high resolution is extremely expensive, both in terms of time as well as storage space. In many cases the finer details are generated by surface properties of the material, and the appearance is statistically uniform. In this paper, we present an approach to add surface details to a coarse 3D model of an object based on two additional information: a set of images of the object and a high resolution model of the material that the object is made of. The material model that we employ is the Polynomial Texture Map (**PTM**), which captures the appearance of a surface under various illumination conditions. We use the observed images of the object as constraints to synthesize texture samples for each triangle of the object under any given illumination.

The primary challenge is to synthesize a polynomial model of the texture, where the constraints arise in the image domain. We use the knowledge of object illumination to map the texture models into image space and compute the optimal patch. The texture transfer then happens as a complete 3D texturemodel. We also consider the problems of pose, scale, reflectance and smoothness of surface while carrying out the texture transfer. We synthesize the texture of an object at a per-triangle basis while carrying out operations such as normalization and blending to take care of discontinuities at the edges.

Keywords

Texture, PTM, 3D Modelling

1. INTRODUCTION

Realistic rendering of real world objects is an important area of computer graphics. It is used in a variety of applications, the most prominent of them being movies, games and archival of historical artifacts. Real world objects are characterized by their shape/geometry as well as the surface properties. To faithfully model a real world object, both the shape and surface properties of the object have to be correctly captured and then rendered through the graphics pipeline.

However, when dealing with objects of large scale, it is difficult to capture these details at a fine level as the capture devices have limited resolution while working at large scales. One could handle this problem using a very high resolution shape models of the parts of the model and fitting them together [6]. However, this approach has a number of shortcomings: i) A shape model that can capture the surface details would be extremely large, ii) Assembling a single model from that of a large number of parts is often labor intensive, and iii) The shape and texture by itself is often unable to capture some of the surface properties of the object such as sub-surface scattering or translucency. On the other hand, capturing high resolution images is often easy due to the availability of low cost and high resolution digital cameras. The goal of this work is to facilitate realistic renderings of large-scale models by utilizing a coarse geometric model of the object augmented with surface details that match the appearance of the object in its images. In this work, we explore the possibility of using the appearance captured in the images as well as prior knowledge of surface properties to add realistic details to a coarse 3D mesh model of the object.

1.1 Image Based Modeling of 3D Objects

Images are the most abundant source of visual and structural information of the real world. They are capable of capturing a high level of object properties effectively. Hence, image based modeling techniques [15, 14, 2] have emerged as an effective approach for realistic rendering of **3D** objects, where multi-view geometry is utilized in directly synthesizing an unseen view of an object from nearby views without explicit surface reconstruction. Multi view modeling methods on the other hand use a set of images of the object, register them and recover the 3D locations of points within the image. A standard mesh model is derived from the point cloud, which is then texture mapped using the images that were used to derive the shape. Both approaches combine the pictorial details obtained from the individual photographs captured, to the shape information of the object inferred from the collection. While the first approach often leads to realistic rendering of unseen views, it lacks the flexibility of 3D model based visualization.

We notice that the traditional object models capture the shape information in the meshes, while the reflectance and

*Corresponding author

ICVGIP '10, December 12-15, 2010, Chennai, India

surface properties are relegated to the texture. 3D texture models such as Polynomial Texture Maps (PTMs) [12] capture the surface properties far more faithfully, including the effects of small scale height variations on the surface. Hence the combination of 3D texture models along with a coarse model of the object shape is capable of realistic rendering of objects. However, the acquisition of 3D textures such as PTMs is close to impossible in uncontrolled environments. In this work we look at synthesizing 3D textures on coarse object models that retains the appearance characteristics of the captured images.

Many texture synthesis algorithms [13, 7, 5, 17, 10, 4] have been developed to generate large samples of texture from scanned photographs. These methods are effective and make the texture mapping process more efficient and robust by facilitating the generation of textures of any required size. These have been later extended to texture **3D** models and arbitrary manifold surfaces[16, 11, 18]. However, unlike traditional texture synthesis, where the goal is to generate a new texture patch that retains the characteristics of the surface, we need to synthesize a 3D texture that agrees with a given image of the object in appearance. Before we look into the process of synthesis, we will explore some of the basic aspects of 3D texture.

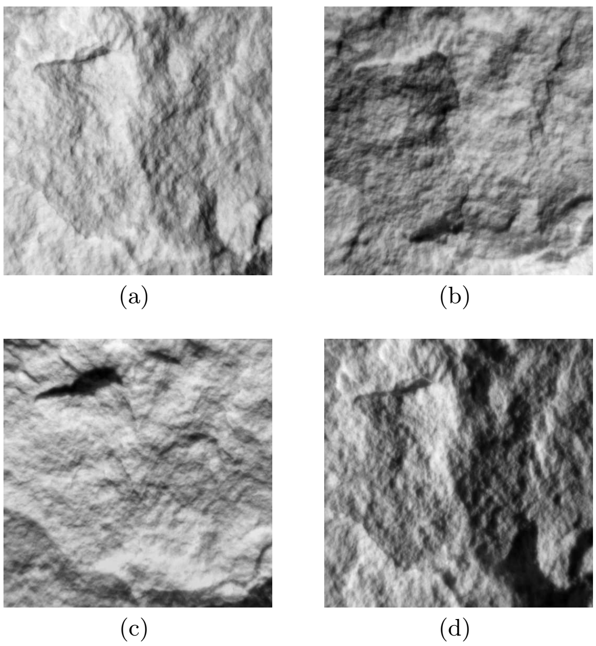


Figure 1: Images of a rough plaster surface obtained under varying light conditions. Note the change in surface appearance in each of (a), (b), (c) and (d)

1.2 Reflectance Properties of Natural Materials

The visual characteristics of natural surfaces arise from the variation of two properties across its surface: i) the variation in normals, and ii) reflectance. These cause effects such as shadows, self occlusions, inter reflections, and specularity, which affect the visual appearance of the surface. As a result, a surface looks considerably different under different lighting and viewing conditions. These effects are observed in all natural surface reliefs that are abundant in real world.

Simple **2D** texture models ignore these two properties of the natural textures. Hence they cannot model these variations in visual appearance caused under varying illumination/viewing conditions. Moreover, in **2D** texture mapping, the images that are texture mapped on to the **3D** models inherently contain the lighting conditions under which they have been captured. Hence the texture model so obtained looks good in lighting conditions similar to that of the available images, but very poor when viewed under different lighting conditions. Hence **2D** texture mapping of **3D** objects is insufficient and falls short of accomplishing the task of realistic modeling and rendering real world objects.

The characterization of surface reflectance properties is essential to achieve realistic rendering. The reflectance properties of a surface affects its appearance under different lighting and viewing conditions. This led to the study of relation between surface luminance and illumination/viewing conditions of natural material surfaces, and such models are often referred to as **3D** textures. This led to further investigation into the problem of representation, recognition, synthesis of natural materials and their rendering under arbitrary viewing/lighting conditions [9]. Image based re-lighting techniques[12, 3, 1] have been used to model the surface reflectance properties of the natural materials. In these techniques multiple images of the Object/surface are captured under different lighting/view point conditions and then the variations in visual appearance modeled as Reflectance Texture Maps.

1.3 Reflectance Texture Maps

Reflectance Texture Maps have emerged as the compact representation of **3D** textures. They model the spatial variation in surface luminance as a function of viewing and illumination conditions. **BTF**(Bi-Directional Texture Functions) and **UTF**(Uni-Directional Texture Functions) are the two reflectance texture functions usually used to model the surface reflectance properties of natural materials.

Bi-Directional Texture Functions(**BTF**) [3] measures the ratio of radiance L emitted at solid angle (ϕ_1, θ_1) to that incident at a different solid angle (ϕ_2, θ_2) at a spacial location (u, v). **BTF** effectively captures all the complex phenomena like shadows, specularities, self-occlusions, etc. However, the storage requirement of BTF is high and the generation is complex.

Uni-Directional Texture Functions(**UTF**) are less exhaustive but tractable representation of reflectance properties and confine themselves to the modeling of surface reflectance properties in relation to the lighting conditions. They do not consider the view point in surface intensity calculation. Hence they cannot model view point phenomenon such as specularities. But they are easy to capture and do not require any camera calibration. They require only a movable light source and a stationary camera. **UTF** implicitly models surface normal information. Hence surface normals can be retrieved and then used to artificially introduce view point phenomena at the time of rendering.

Polynomial Texture Maps(PTM)[12] belonging to the class of **UTF** are a compact representation of reflectance textures. They model the surface luminance variations as a Bi-quadratic polynomial function at each pixel of the texture.

Techniques to synthesize 2D textures on arbitrary shaped objects[16, 11, 18] have also been extended to synthesis reflectance texture maps on the same. In these techniques, the synthesis starts from an arbitrary patch and then it grows on till the complete mesh-model is covered. The only constraint imposed on the synthesis process is that a patch to be synthesized agrees with the already synthesized neighboring patches. This constraint makes sure that no visible seams appear on the textured model. Using these approaches, PTMs have been efficiently synthesized over 3D models and rendered[8].

The above mentioned 3D texturing algorithms when coupled with image based modeling techniques provide an effective platform for realistic modeling of real world objects. Pictorial information of the object can be obtained in a small set of images and later used to constrain the texture synthesis procedure, which is otherwise unconstrained (except for inter-patch consistency). This hybrid approach provides an effective way to synthesize the texture model of the object based on its appearance in the real world images. Hence it can be employed to generate the 3D texture models of the real world objects.

We want to address this problem of constrained 3D texturing of mesh-models to make them more realistic and near to their real world objects. Our goal is to capture a small set of images of the object under known lighting conditions, and use these to systematically synthesize a PTM model of the object from a sample PTM of the object's material. These sparse set of images decide the sub-samples of the texture sample PTM_{in} that are stitched across the mesh model so that the resultant texture model would behave more similar to its real world counterpart.

This work draws from the ideas of Efros *et. al* [4] for effective Texture transfer and the other that of Yacov Hel-Or *et. al* [8] for synthesis of 3D texture models for 3D objects. We present a method to effectively 3D texture the standard mesh model of a real world object from a sample PTM of its material using a small set of images captured of the object under different known lighting conditions as constraints, so as to make the model appear realistic and similar to the original. The PTM model so generated can be efficiently rendered under arbitrary lighting conditions.

2. IMAGE BASED PTM SYNTHESIS

The goal is to generate a 3D texture for a coarse mesh model of a real world object, using a sample PTM of its material and a small set (atleast one) of images of the object captured under different known lighting conditions. The images provide information about the visual appearance of the object and make the synthesis process more effective in selection of subsamples that go into the making of the texture model. The resultant model not only appears realistic but more similar to the original. The PTM model so generated can be efficiently rendered under arbitrary lighting conditions.

2.1 Polynomial Texture Maps

Polynomial Texture Map(PTM)[12] is a concise model of surface reflectance properties which can be used to introduce photorealism in the texture mapping process. They properly model the real world phenomena such as self shadowing, inter-reflections, and sub-surface scattering, which cause variations in the visual appearance of an object surface under different lighting conditions. In order to synthesize a PTM, a set of images $\{I_k\}$ of the object surface are obtained under different light conditions $\{(lu_k, lv_k)\}$. These images amply capture the variations in the visual appearance of the surface with changing light conditions and are used to build the reflectance model of the surface. The behavior at each texel is modeled independently with a reflectance function that encodes its behavior with respect to the lighting conditions.Hence the PTM is parameterized on spatial location (u, v) and lighting position (ϕ, θ). Hence the Number of degrees of freedom is **4**.

The chromaticity of a pixel fairly remains constant under different lighting conditions and it is only the luminance that varies. Hence only the luminance $L(u, v)$ is modeled using PTM. The variations in luminance L at each pixel (u, v) is approximated using a bi-quadratic polynomial given by:

$$\begin{aligned} L(u, v; lu, lv) &= a_o(u, v)l_u^2 + a_1(u, v)l_v^2 + a_2(u, v)l_u l_v + \\ &\quad a_3(u, v)l_u + a_4(u, v)l_v + a_5(u, v), \end{aligned} \tag{1}$$

where L is the luminance at pixel (u, v) and (lu, lv) the projection of a unit vector in the direction of the light source on the texture co-ordinate system. The luminance $L(u, v)$ so obtained is modulated with the normalized color value $(R_n(u, v), G_n(u, v), B_n(u, v))$ of the pixel to get the actual color.

$$\begin{aligned} R(u, v) &= L(u, v)R_n(u, v) \\ G(u, v) &= L(u, v)G_n(u, v) \\ B(u, v) &= L(u, v)B_n(u, v) \end{aligned} \tag{2}$$

The above representation is called **LRGB PTM** and it takes advantage of the redundancy in surface color. At each texel (u, v) of the texture map, the coefficients $(ao, \ldots, a5)(u, v)$ of the corresponding bi-quadratic polynomial along with the normalized color value $(R_n(u, v), G_n(u, v), B_n(u, v))$ are stored. The luminance coefficients $(ao, ..., a5)(u, v)$ of each texel are estimated, so as to fit the corresponding pixel data in the images. Polynomial Texture Maps consisting of these surface luminance coefficients at each texel (u, v) approximately model the behavior of the surface under different lighting conditions. The visual appearance of the object surface under arbitrary lighting conditions can be generated from the bi-quadratic approximation of L for each texel, given the light vector (lu, lv).

Patch based 2D texture synthesis techniques can be extended to synthesize large patches of PTM from a given input sample and can also be extended to synthesize PTMs on 3D models [8].

2.2 Constrained PTM Synthesis

Given a PTM sample and a triangular mesh model of an object, small patches extracted from the sample can be seamlessly stitched across the mesh model and a PTM model of the object synthesized. The PTM model so synthesized behaves like a real world object in terms of its visual appearance under different lighting conditions. These patch based texture synthesis algorithms when coupled with image based modeling techniques provide an effective approach to synthesize the 3D texture models of real world objects. A set of images of the object captured varying light and viewpoints decide the set of texture patches that are to be stitched across the mesh model. Hence the texture model so obtained not only looks realistic but also similar to its real world counter part.

We study the problem of generating a PTM model of a real world object by constraining the synthesis procedure using the information obtained from a set of images of the object captured under known lighting conditions. Our work essentially builds on the work by Efros *et. al* [4] for Texture Transfer and the work by Yacov Hel-or *et. al* [8] for PTM modeling of 3D objects. We suggest an approach to generate the reflectance texture model of a real world object from a PTM sample of the object material and a set of images of the object. We extended the patch based PTM synthesis algorithm to also include the image based information in influencing the selection of the texture patches so as to make the resultant texture model more similar to the object. The PTM model so obtained can be used to generate the views of the object under arbitrary lighting conditions.

In the next section, we detail our method for PTM modeling of planar rectangular surfaces and then follow it up with another section for the same on real world objects.

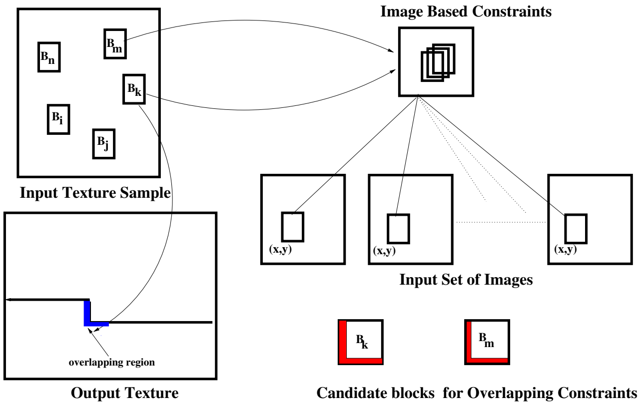


Figure 2: Blocks from input sample are checked for image based and overlap constraints and the best ones are transferred to the output.

2.3 Image constrained PTM synthesis for Planar Rectangular Surfaces

In this section we describe how the patch based 2D texture synthesis algorithm[10, 4] and PTM synthesis algorithm by Yacov Hel-or *et. al* [8] is extended to synthesize the texture model of a planar rectangular surface from a sample PTM of the surface material and a set of images of the surface taken under various known lighting conditions.

A Polynomial Texture Map PTM_{out} of the same dimensions as the planar surface being modeled is generated as the output of this procedure. The algorithm uses patches taken from input sample PTM_{in} as the building blocks to synthesize the output texture PTM_{out}. At each step k, a candidate block B_k is taken from the input sample PTM_{in} and stitched into the output texture PTM_{out}. The selected blocks are stitched into PTM_{out} with an overlap W_e between neighboring blocks and then blended in the overlapping region. The texture map so obtained can be used to generate novel views of the surface under arbitrary lighting conditions.

The selection strategy of candidate block B_k that is stitched into the output texture PTM_{out} at every step is the core of our algorithm. The output texture PTM_{out} is traversed in a raster scan fashion from left to right starting at lower left corner and moving upwards. At step k, a candidate block B_k is selected to go into the next position (x, y). The selection of the patch B_k is governed by two constraints namely:

1 Image based constraints

2 Overlapping constraints

The set of images $\{I_n\}$ that are captured under varying light positions (l_{un}, l_{vn}) decide the candidate patches that go into the making of the output texture. The texture block B_k that is selected to go into the position (x, y) should agree with the set of image blocks $\{b(I_n, x, y)\}$ that are covers the position (x, y) in each of the images $\{I_n\}$. Let the PTM evaluation function be denoted by $f(P, (l_u, l_v))$, where P is a PTM patch and (l_u, l_v) projection of unit light vector on to the texture co-ordinate system. The PTM patch B_k, when evaluated with the light vector (l_{un}, l_{vn}) corresponding to the image I_n, should result in an image patch $f(B_k, (l_{un}, l_{vn}))$ that matches the image block $b(I_n, x, y)$. Hence each image I_n of the set $\{I_n\}$ imposes constraint on the selection of the patch B_k. These together constitute the image based constraints involved in the synthesis. Each texture block B from the input sample PTM_{in} is ranked according to a scoring measure S which is given as follows:

$$S(B) \;=\; \sum_{n=1}^{N} \|f(B, (l_{un}, l_{vn})) - b(I_n, x, y)\|_2 \qquad (3)$$

The blocks from the input sample PTM_{in} are ranked according to the scoring mechanism S and a fraction of the top candidates selected.

At every step k of the synthesis procedure, the patch B_k that is selected to go into the output texture PTM_{out} should also agree with the patches $\{B_0, B_1, ..., B_{k-1}\}$ that have so far been pasted in the previous steps. The candidate block B_k that is currently being pasted should agree with its neighboring patches in the overlapping region. This constitutes the overlapping constraint and is a must for seamless stitching of input blocks.

The set of candidates that are selected based on image based constraints in step(1) are again ranked based on their overlapping measure. $L2$ norm is calculated over the difference of luminance coefficients in the overlapping region between PTM_{out} that has been synthesized so far and each B of the candidate blocks picked by step(1). The norm is calculated with the coefficients of both PTM_{out} and block B in the overlapping region transformed to a orthogonal space so that the distance between functions is same as that between function coefficients. The block B_k with minimal error measure is introduced into the output PTM_{out}.

The complete algorithm is outlined as follows:

Alpha-blending is usually employed to blend the texture coefficients in the overlapping regions. The texture coefficients are transformed to an orthogonal space before blending is employed. After blending, the coefficients are transformed back by applying an inverse transformation.

The above approach generates a better estimated PTM model of the given surface and it can be used to calculate

Algorithm 1 The Constrained PTM Synthesis Algorithm

1: Traverse PTM_{out} in a raster-scan order block by block starting at lower-left corner.
2: At every new position (x, y), select a small set s of candidate blocks from PTM_{in} using the image based constraints.
3: Pick the best block B_k among the set s which best fits the overlapping constraint.
4: Paste the block B_k at the location (x, y) in the output texture PTM_{out} and blend it in the overlapping region.

the appearance of the given surface under arbitrary lighting conditions.

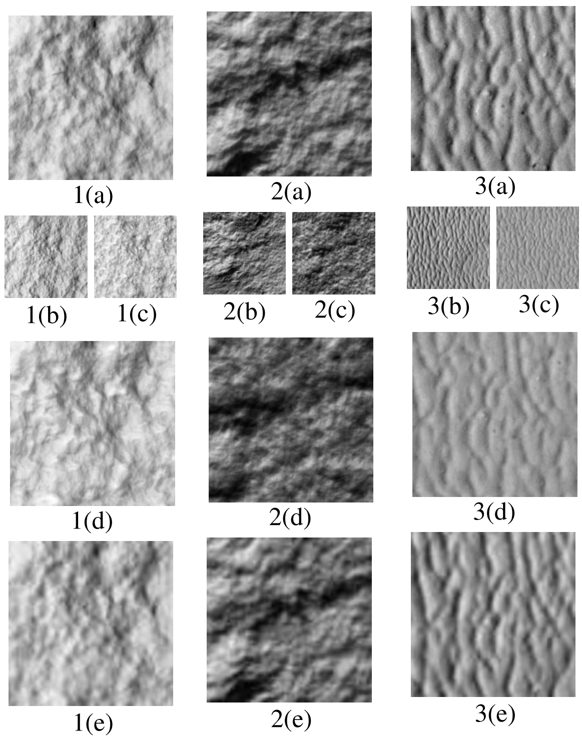


Figure 3: Demonstration of Our 2D Synthesis Algorithm for Natural Material Surfaces:. Top row, 1(a)-3(a), shows high resolution texture samples of three different rough plaster surfaces. Images 1(b), 2(b), 3(b) are low resolution samples of surfaces above and 1(c), 2(c), 3(c) shows the results of synthesis using images in (b) as constraint. Row 1(d)-3(d) are the textures synthesized at high resolution using images in (b) and 1(e)-3(e) shows the scaled versions of (b) for comparison.

In Figure 3, we demonstrated the results of our 2D synthesis algorithm on three different surfaces. PTMs of variants of plaster surface were created from their high resolution images. Images 1(a), 2(a) and 3(a) show the high resolution texture samples used in the creation of their corresponding sample textures. From a sparse set of low resolution images of an object and its sample PTM, the texture model of the object was constructed and its views shown at the resolution at which the object was captured as well as at a higher resolution. The material information present at high resolution affirms our synthesis procedure.Images 1(b), 2(b) and 3(b) are the object views captured at low resolution and 1(c), 2(c) and 3(c) represent the corresponding low resolution views of their texture models generated using our method. Images 1(d),2(d) ad 3(d) are the high resolution views of 1(c),2(c) and 3(c) respectively and 1(e),2(e),3(e) were obtained by scaling images 1(b), 2(b) and 3(b) respectively. Comparing the results with the scaled versions, we note that while the synthesized images has minor variations with the original high resolution images (in a), the finer details visible in the synthesized patches are missing in the plain scaled version.

3. IMAGE CONSTRAINED PTM SYNTHESIS FOR REAL WORLD OBJECTS

Patch-based 2D Texture Synthesis algorithms use square patches as the quilting blocks because of their simplicity to work with. The same cannot be said about synthesis for real world objects which are 3D in nature. The 3D objects are usually represented with standard triangular mesh models.

Triangle is the basic primitive for rendering 3D models. Hence it is more appropriate to consider triangle as the quilting block and texture map triangles rather than the usual square patches. But the triangles of the mesh model are of different sizes and shapes unlike the square patches used in the previous section which are all of uniform size. It becomes only difficult that all the triangles having various texture orientations.

Considering all the above mentioned issues, we devised an approach to synthesize image based PTM models of real world objects. Given a set of images $\{I_n\}$ of the object captured under known light and camera positions $\{(l_{un}, l_{vn}), C_n\}$ and a texture sample PTM_{in} of the object material, we synthesize a texture model of the object by pasting triangular subsamples taken from PTM_{in} all across the triangular mesh model of the object. Like the earlier approach for planar surfaces, this approach also considers the image-based constraints and overlapping constraints in selection of triangular patches for the texture model.

We outline the basic steps of our synthesis algorithm followed by a detailed description of each of them.

Algorithm 2 PTM Synthesis of Real world Objects

1: Assign each triangle T of the mesh model, an image $I_k \in \{I_n\}$ in which it is best visible and calculate its mapping t in I_k.
2: Generate the normal view t_n from t, find its best matching triangular texture patch p in PTM_{in} and extract a rectangular patch B containing p.
3: Perform Alpha-blending across every edge of the mesh model, update the texture patches $\{B_i\}$ with blended values.
4: Extract the minimal bounding box b_i contained inside B_i of each triangular texture patch p_i, and pack all such b_i into a number of texture atlases.

3.0.1 Step1

The object is imaged multiples times from different known lighting directions and view-points $\{(l_{un}, l_{vn}), C_n\}$ to obtain a set of images $\{I_n\}$. Each triangle T of the mesh model is then mapped to an image $I_k \in \{I_n\}$ in which it is best visible. The images in which the triangle T is completely visible are picked and then an image I_k among them in which it is best visible is taken. The criteria for visibility is the angle made by the normal n of the triangle T with the directional vector of the camera C from its center. We rely on the assumption that each triangle T of the mesh model is completely visible in at least one image.The camera matrix M_k corresponding to the image I_k is calculated and then used to obtain the mapping t of triangle T in the image I_k.

3.0.2 Step2

After step 1, each T is mapped to its best-view t in an Image I_k. Now based on the angle θ between the normal n of T and the direction of the camera center C, the lengths of sides of the triangle in the normal view t_n are obtained using the following formula

$$l_{in} = l_i / \cos\theta, \tag{4}$$

where l_i is the length of ith side of t in the image I_k.

The geometry of t_n is determined by the sides $\{l_{in}\}$ and angles A, B, C of the original triangle T. For simplicity, the side connecting the first 2 vertices of t_n is made parallel to X-axis. The color information from t to t_n is transferred using a re-sampling algorithm.

Calculate the local light vector l_T with respect to a co-ordinate system placed at the centroid of T. The X-axis of this co-ordinate system aligns with the side connecting the first two vertices of T, Z-axis along the normal of T and Y-axis decided by the former two.

Evaluate the input sample PTM_{in} using l_T and search the resultant image for a set of patches $\{t\prime\}$ which best agree with t_n. This constitutes the image based constraints. Each $t\prime$ corresponds to a triangular texture patch $p\prime$ in the input texture sample PTM_{in}.

Now pick the best texture patch $p \in \{p\prime\}$ which best agrees with the texture patches of already processed neighboring triangles $\{T_j\}$ of T. This constitutes the overlapping constraint imposed on the synthesis. In order to impose overlapping constraints, at least one of the three neighbors of the current triangle T should have been already processed. Hence random processing of triangles of the mesh model might result in occasional weakening of the selection strategy and the quality of texture model so obtained.

To prevent this, the triangles of the mesh model are processed in a Breadth-First-Search(**BFS**) order. By doing so, overlapping constraints are imposed in the selection of texture patch p for every triangle T of the mesh model except for the first one.

For each triangle T_i, a minimal bounding box b_i surround its triangular texture patch p_i is identified and a bigger rectangular patch B_i containing b_i surrounded by a extra texel strip (5 to 10 texels) all around is extracted from the texture sample PTM_{in}. The extra strip of texels is used for blending with texture patches of neighboring triangles.

3.0.3 Step3

Each of the above mentioned boxes B_i include an extra strip of W_e on all the 4 sides of the corresponding minimal bounding box b_i. This strip is essentially used for alpha-blending across edges. The extra texel padding around the actual triangular texture is blended with the border information of neighboring triangle as shown in Figure 4. The alpha-blended information is written back to the set of boxes $\{B_i\}$

3.0.4 Step4

Minimal bounding boxes $\{b_i\}$ are extracted from $\{B_i\}$ by cutting off the extra strip of texels present around. These $\{b_i\}$ are then packed in to a number of atlas maps of desired dimensions W and H using any of the standard bin-packing algorithms. The texture mapping co-ordinates of all the triangles T_i are updated all along the procedure and the final mapping co-ordinates with respect to the PTM atlases $\{P_j\}$ are stored.

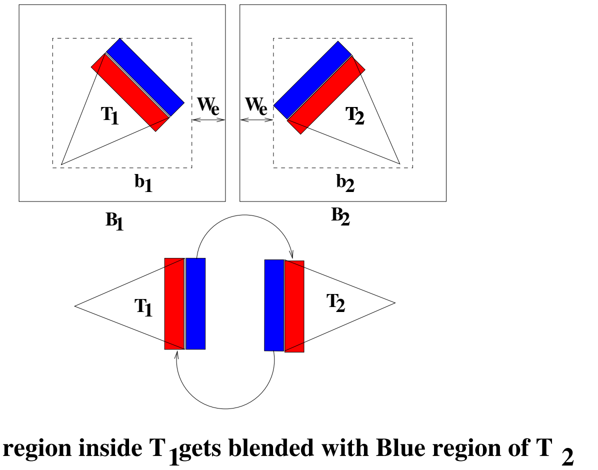


Figure 4: Blending of neighboring triangles. Red region of T1 blends with blue region of T2 and vice versa.

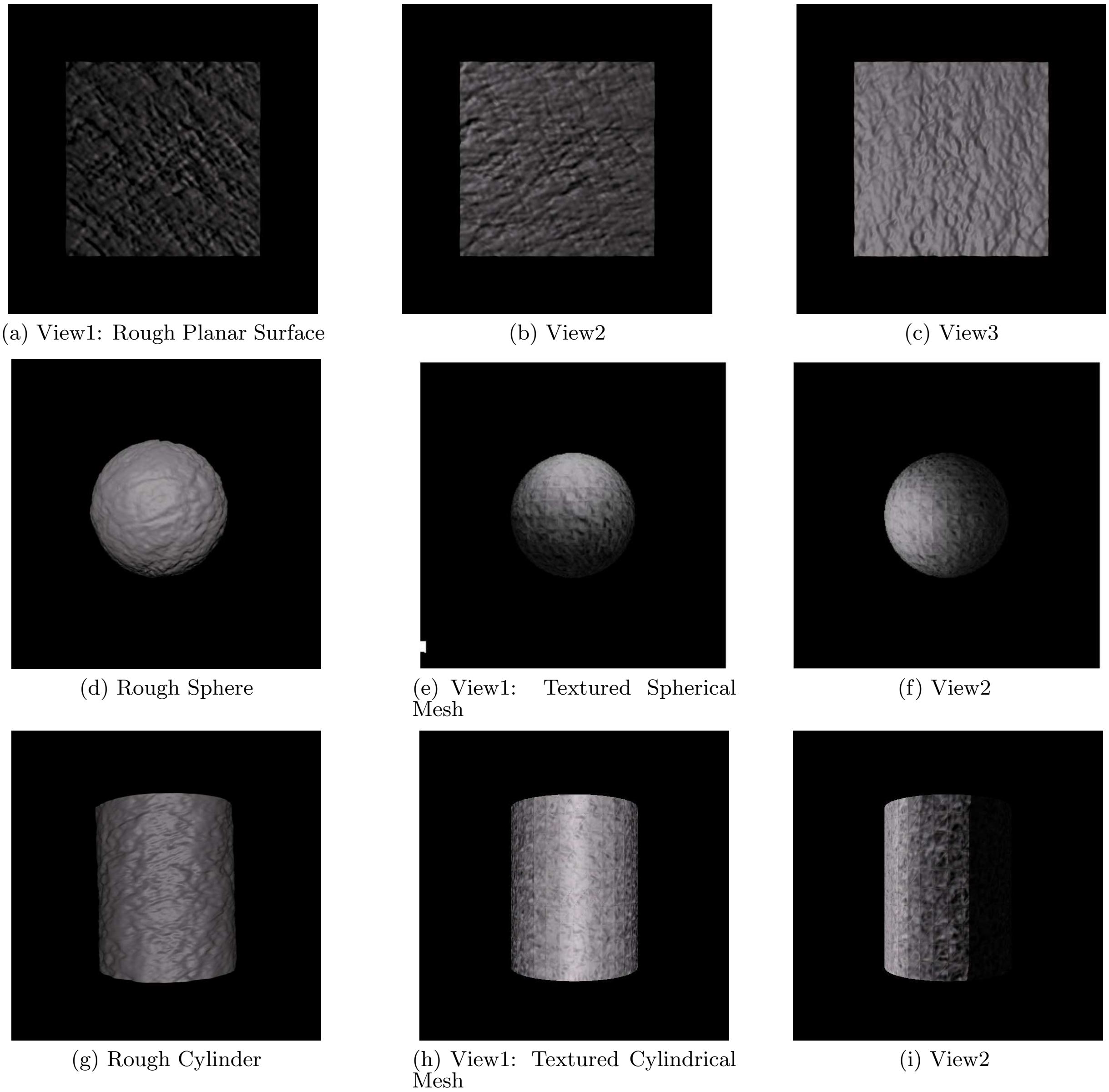
The above process of PTM synthesis for a real world object is an off-line process. Hence we limit ourselves to only the synthesis procedure and not the time complexity, techniques to speed it, etc.

3.1 Rendering of the PTM Model

The PTM model of the object obtained can be efficiently rendered at run time to generate novel views of the real world object under different lighting conditions. In this procedure, each triangle T is considered separately and the position of light with respect to it calculated. The unit vector (lu_T, lv_T) so obtained is used to evaluate its PTM patch p to generate an image patch. Hence we get image atlases corresponding to the set of PTM atlases. These image atlases are used as texture objects, loaded and the textured model rendered.

4. EXPERIMENTAL RESULTS AND ANALYSIS

We demonstrate our 3D synthesis algorithm on a set of rough surface models created using displacement mapping. Synthetic 3D textures and object models are used so that the same texture model can be used to generate surface textures for the mesh model. We generated a height map and applied it individually on the plane surface, a smooth sphere and a cylinder using displacement mapping to create rough objects. A sample PTM is then created using a set of images of the rough planar surface. A small set of images of the rough sphere and cylinder were taken to provide the image based constraints. These images and the sample PTM were used to construct the PTM models of the rough sphere and the cylinder. In Figure 5, the images (a), (b) and (c) show



(a) View1: Rough Planar Surface

(b) View2

(c) View3

(d) Rough Sphere

(e) View1: Textured Spherical Mesh

(f) View2

(g) Rough Cylinder

(h) View1: Textured Cylindrical Mesh

(i) View2

Figure 5: Synthesis Results for 3D object:Images (a), (b) and (c) show three different views of a rough planar surface. Image(d) shows a rough sphere and (e),(f) show two views of the textured mesh model that was generated. Likewise images (g),(e) and (f) show the same for a cylinder of similar texture.

three views of a rough plane and these are used to construct the texture sample employed in synthesis. Images (d) and (g) show a rough sphere and a cylinder created using displacement mapping. Images (e) and (f) show two arbitrary views of the constructed PTM model of the rough sphere and images (h) and (i) show the same for the rough cylinder.

As we note, the synthesized model is able to capture the surface properties, as the lighting directions change, which would be impossible in the case of 2D textures. Moreover, as seen from synthesis of the planar object, the synthesized 3D texture generates images resemble the observed images of the original object.

However, there are primarily two issues that still remain to achieve photo-realistic rendering of 3D mesh models: i) The PTM model itself does not handle shadows and specularities in the texture well as it creates an overly smooth approximation of the transition from light to shadows with change in light direction, ii) Variations in appearance with lighting direction is accentuated at the triangle boundaries. Currently we are working on developing improved models of the PTM to handle the first issue, and with synthesis techniques that directly create smooth transitions over triangle boundaries.

5. CONCLUSION AND FUTURE WORK

We demonstrated an image based texture synthesis technique to effectively synthesize reflectance textures for material surfaces and objects. We developed the idea of transferring texture on to the mesh models of real world objects to realistically reproduce the natural visual appearance, perception and their interaction with the lighting environment.

Polynomial texture maps implicitly contain the surface normal information. We are presently working on artificially introducing viewpoint phenomenon such as specularities in the rendering of PTMs of 3D models of real world objects. We also want to investigate the idea of manipulation of luminance coefficients in influencing the visual perception of shape information of the object surface, so that the seams that occur at the join of adjacent patches can be effectively removed and smooth transitions introduced in place.

6. REFERENCES

[1] M. Ashikhmin and P. Shirley. Steerable illumination textures. *ACM Trans. Graph.*, 21(1):1–19, 2002.

[2] A. Baumberg. Blending images for texturing 3d models. In *Proc. Conf. on British Machine Vision Association*, pages 404–413, 2002.

[3] K. J. Dana, B. van Ginneken, S. K. Nayar, and J. J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Trans. Graph.*, 18(1):1–34, 1999.

[4] A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346, New York, NY, USA, Aug. 2001. ACM.

[5] A. Efros and T. Leung. Texture synthesis by non-parametric sampling. In *International Conference on Computer Vision*, pages 1033–1038, 1999.

[6] M. L. et al. The digital michelangelo project: 3d scanning of large statues. In *Proceedings of SIGGRAPH*, pages 131–144, New Orleans, LA, USA, 2000.

[7] D. J. Heeger and J. R. Bergen. Pyramid-based texture analysis/synthesis. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 229–238, New York, NY, USA, 1995. ACM.

[8] Y. Hel-Or, T. Malzbender, and D. Gelb. Synthesis and rendering of 3d textures. *3rd International Workshop on Texture Analysis and Synthesis*, 2003.

[9] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal On Computer Vision*, 43(1):29–44, 2001.

[10] L. Liang, C. Liu, Y. Xu, B. Guo, and H. yeung Shum. Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics*, 20:127–150, 2001.

[11] S. Magda and D. Kriegman. Fast texture synthesis on arbitrary meshes. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Sketches & Applications*, New York, NY, USA, 2003. ACM.

[12] T. Malzbender, D. Gelb, and H. Wolters. Polynomial texture maps. In *Computer Graphics, SIGGRAPH 2001 Proceedings*, pages 519–528, 2001.

[13] K. Popat and R. W. Picard. Novel cluster-based probability model for texture synthesis, classification, and compression. In *Visual Communications and Image Processing*, pages 756–768, 1993.

[14] C. Rocchini, P. Cignoni, C. Montani, and R. Scopigno. Multiple textures stitching and blending on 3d objects. In *Eurographics Rendering Workshop*, pages 119–130. Springer-Verlag, 1999.

[15] M. Soucy, G. Godin, R. Baribeau, F. Blais, and M. Rioux. Sensors and algorithms for the construction of digital 3-d colour models of real objects. pages II: 409–412, 1996.

[16] G. Turk. Texture synthesis on surfaces. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 347–354, Aug. 2001.

[17] L.-Y. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 479–488, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[18] L.-Y. Wei and M. Levoy. Texture synthesis over arbitrary manifold surfaces. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 355–360, New York, NY, USA, 2001. ACM.