Proceedings of the 2006 IEEE
International Conference on Mechatronics and Automation
June 25 - 28, 2006, Luoyang, China

Alpha Mapping in Scene Simulation of Launch Vehicle

Jun Sun, Limei Xu
School of Mechatronic Engineering
University of Electronic Science and Technology of China
Chengdu, Sichuan Province 610054, China
jeremy_sun@163.com

Liangping Zhu
Department of Experimentation Technology
×× Experimentation and Training Base
Xichang, Sichuan Province 615000, China

Abstract - **The display of third dimension graphics is the most crucial part in scene simulation, and alpha mapping plays an important role in the synthesization of third dimension graphics. Alpha mapping is a texture mapping technique with alpha channel for simulating transparent effects caused by patterned irregularities on otherwise locally smooth surfaces. In this paper, the current mathematic theory of alpha channel and alpha estimation in image matting, which were used in alpha mapping, was introduced. Based on the Object-Oriented Graphics Rendering Engine (OGRE), alpha mapping was applied in the scene simulation system of launch vehicle, and achieved excellent result both in rendering time and quality. It is shown that the great improvement of real-time performance could be obtained based on the third dimension of the diversified objects and scene.**

Index Terms* - *scene simulation, alpha mapping, alpha channel, alpha estimation

I. INTRODUCTION

Since 1970's, Catmull [1] adopted texture mapping to create texture detail on the surface of objects for the first time. Texture mapping has been widely used to enhance realism of computer-generated images. The possibility to remap a 2-dimensional colour map over the projection of a 3-dimensional polygon is crucial in order to produce rich colour details in interactive time. Based on the represent form, texture can be compartmentalized to color texture, bump texture, and procedural texture [2]. Alpha mapping is one of color texture mappings.

Alpha mapping is a texture mapping technique with alpha channel for simulating transparent effects. Catmull and Smith invented the notion of the *integral alpha* in the 1970s at New York Tech [3]. This is the notion that opacity of an image is as fundamental as its color and should therefore be included as part of the image, not as a secondary accompaniment. The notion of alpha was invented as a fundamental component of an image. The resulting full-color pixel was called an "*RGBA*" pixel.

Red, Green, and Blue obviously are the three color channels of a full-color image, and Alpha is the transparency channel. The alpha channel typically contains as many bits as a color channel. So, for example, an 8-bit alpha channel can represent 256 levels of transparency, from 0 (completely transparent) to 255 (completely opaque).

In 1984, Porter and Duff [4] introduced the digital analog of the matte-the *alpha channel*, and showed how synthetic images with alpha could be useful in creating complex digital images. The use of an alpha channel to form arbitrary compositions of images is well-known in computer graphics. An alpha channel gives shape and transparency to a color image.

Alpha mapping can be very useful to simulate many interesting effect, through utilizing the alpha channel of the texture. One is the decaling, such as only a part of a flower image can be mapped onto a teapot, but not all the image. Another application is the simulation of cutout effect, i.e. the partial transparent effect on a given object can be rendered by alpha mapping. For example, a texture of tree can be mapped onto a polygon to simulate the tree. Alpha mapping simulates a surface's partial transparent appearance without the complexity and expense of modeling the patterns as true geometric shape to the surface.

Benefits of alpha mapping include: A high level of visual complexity can be provided in a scene, without adding more geometry. Content authoring can be simplified, because surface detail can be encoded in textures as opposed to requiring designing highly detail 3-dimensional models. Different alpha maps can be applied to different instances of the same model to give each instance a distinct surface appearance.

In the scene simulation system of launch vehicle, Alpha mapping is used in the simulation of many orifices on the body of rocket. This effect is achieved without requiring excessive geometric tessellation of the surface. The reason to avoid representing this kind of surface detail using geometry is that, representing the model with sufficient geometric detail to capture the transparent effect of the surface would be too large and cumbersome for interactive rendering.

In this paper, based on the correlative mathematic theory of alpha channel and alpha estimation in image matting, the algorithms of alpha estimation are adopted in alpha mapping. And the transparent effect of alpha mapping in scene simulation is realized in the 3-dimensional rendering engine, OGRE (Object-Oriented Graphics Rendering Engine). It is a scene-oriented, flexible 3-dimensional engine written in C++. The class library abstracts all the details of using the underlying system libraries like Direct3D and OpenGL and provides an interface based on world objects and other intuitive classes.

II. ALPHA ESTIMATE IN IMAGE MATTING

Image matting aims to extract objects from one image and to synthesize them into another one. It is used widely in computer graphics (CG), film and video industry and can be

1-4244-0466-5/06/$20.00 ©2006 IEEE

classified as *blue screen matting* [5, 6] and *natural image matting* [7, 8, 9, 10, 11] according to background images.

A. Blue screen matting

Blue screen matting approaches try to simplify the problem by photographing foreground objects against a constant-colored background, and extracting foreground and alpha treating each frame in isolation. Smith and Blinn [5] analyzed the matting problem in great detail. In their paper, the Vlahos formula for α_0, abstracted from the claims of his earliest electronic patent and converted to their notation, is

$$\alpha_0 = 1 - \alpha_1 (B_f - \alpha_2 G_f) \quad (1)$$

clamped at its extremes to 0 and 1, where the α_i are tuning adjustment constants (typically made available as user controls), and B_f and G_f are the blue and green channels of the input image, respectively.

They showed that a unique solution can be found in the general case if the foreground object is filmed against two backgrounds that differ in every pixel. However, even knowing the background color is still insufficient to fully constrain the problem thus some simple constraints are made, which require expert level parameter tuning and can fail on fairly simple foregrounds.

Mishima [6] improved this technique by introducing statistical methods based on representative foreground and background samples. The algorithm started with two identical polyhedral (triangular mesh) approximations of a sphere in *rgb* space centered at the average value of the background samples.

B. Natural image matting

Natural image matting is a more general problem than blue screen matting. It extracts foreground and alpha from relatively arbitrary photographs or video streams. The most successful systems include Knockout [7], the approach proposed by Ruzon and Tomasi [8], Hillman [9], Bayesian matting [10] and Poisson matting [11]. All these systems start by having the user segment the image into three regions: definitely foreground, definitely background and unknown, which is often referred to as a *trimap*. The algorithms then estimate foreground, background, and α for all pixels in the unknown region. Here we briefly describe their alpha estimating algorithms.

1)Knockout approach

For an observed color *C*, when its foreground color component *F* and background color component *B* are computed, Knockout approach decomposes the final α into three alpha components α_r, α_g, α_b along three axes in RGB space using the following formula:

$$\alpha = \frac{f(C) - f(B)}{f(F) - f(B)} \quad (2)$$

where $f(\cdot)$ projects a RGB color onto one of the *r-, g-, b-* axes. Fig.1 illustrates alphas on *r-* and *g-* axes. These three alpha components are computed separately by projection onto the three axes, and the final α is taken as a weighted mean over all the projections, where the weights are proportional to the denominator in (2) for each axis. This approach decomposes α into three components along the axes in RGB space.

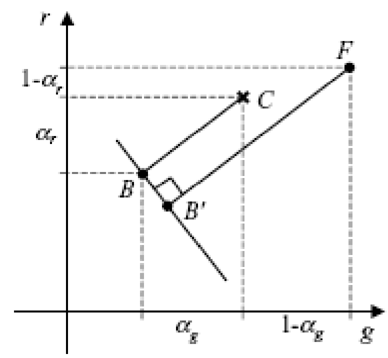


Fig.1 Matte parameters are computed in the Knockout approach

(2)Ruzon-Tomasi approach

Ruzon and Tomasi [8] partition the unknown boundary region into sub-regions, In which the encompassed fore-ground and background pixels are then treated as samples from distributions *P(F)* and *P(B)*, respectively, in color space. In the end, through splitting the pixels into coherent clusters, every foreground cluster is paired with every background cluster. Fig.2 shows a single pairing for a foreground and background distribution.

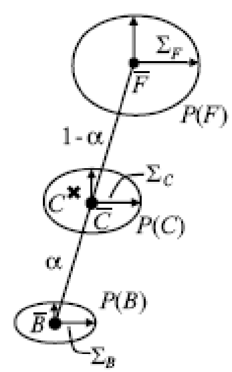


Fig.2 Matte parameters are computed in the Ruzon-Tomasi approach

After building this network of paired clusters, Ruzon and Tomasi treat the observed color *C* as coming from an intermediate distribution *P(C)*, somewhere between the foreground and background distributions. As depicted in Fig. 2. The optimal alpha is the one that yields an intermediate distribution for which the observed color has maximum probability. As a post-process, the *F* and *B* colors are perturbed to force them to be endpoints of a line segment passing through the observed color and satisfying the compositing equation.

(3)Hillman approach

Hillman use principal component analysis (PCA) to represent color samples with oriented line segments [9]. PCA is used to find the major orientation of the foreground and background color samples. Then, the collected color samples are projected onto the major axis found by PCA, essentially discarding the variances on the minor axes. Extreme points of the projected color samples are found to form the line segments $P_{0f}P_{1f}$ and $P_{0b}P_{1b}$, as depicted in Fig.3, for the foreground and background, respectively.

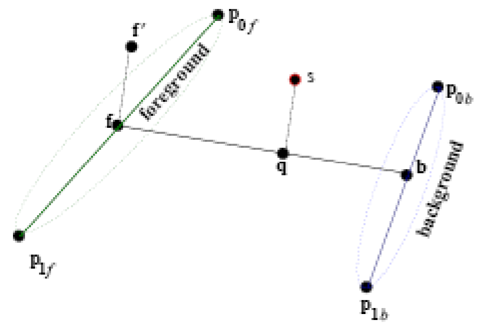


Fig.3 Matte parameters are computed in the Hillman approach

Fig.3 illustrates the matte and color estimation procedure. Recombining *f* and *b* (the estimated clean foreground and background colors) using the calculated value for alpha, produces a point *q* which is the closest point on the line to the original pixel *s*. By adding the vector $\overrightarrow{qs}$ to the estimated foreground point *f*, a better estimate *f'* of the clean foreground color can be generated. Finally, the foreground and background colors are estimated by perturbing *f'* and *b'* in a similar way as in the Ruzon-Tomasi algorithm.

(4)Bayesian approach

Chuang's method [10] uses a continuously sliding window for neighborhood definitions, marches inward from the foreground and background regions, and utilizes nearby computed *F*, *B*, and α values (in addition to these values from "known" regions) in constructing oriented Gaussian distributions. Further, the approach formulates the problem of computing matte parameters in a well-defined Bayesian framework and solves it using the maximum a posteriori (MAP) technique. Fig.4 illustrates the distributions over which Chuang solve for the optimal *F*, *B*, and α parameters.

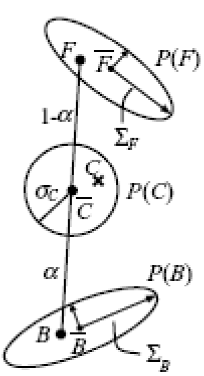


Fig.4 Matte parameters are computed in the Bayesian approach

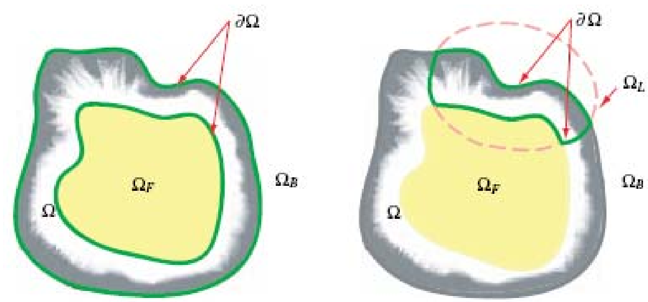
As depicted in Fig.4, Bayesian approach assumes that *F* and *B* are constant, yielding a quadratic equation in α. The solution to this equation is to project the observed color C onto the line segment *FB* in color space:

$$\alpha = \frac{(C-B)\bullet(F-B)}{\|F-B\|^2} \tag{3}$$

where the numerator contains a dot product between two color difference vectors.

(5)Poisson matting

Poisson matting [11] operates directly on the gradient of the matte. This reduces the error caused by mis-classification of color samples in a complex scene. Poisson matting consists of two steps. First, an approximate gradient field of matte is computed from the input image. Second, the matte is obtained from its gradient field by solving Poisson equations.



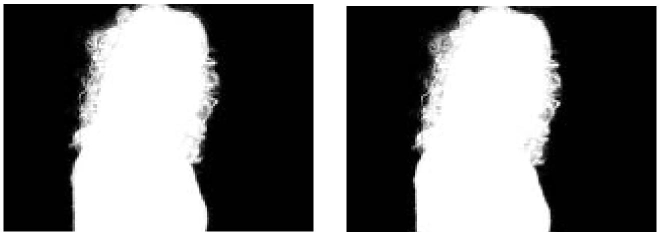
(a) (b)

Fig.5 Boundary condition for Poisson matting.

As shown in Fig.5, Ω_F, Ω_B and Ω are defined as "definitely foreground", "definitely background" and "unknown" regions respectively. Fig.5 (a) is Global Poisson matting, in which the trimap $\{\Omega_F, \Omega_B, \Omega\}$ is specified by the user. $\partial\Omega$ is the exterior boundary of unknown region Ω. Fig.5 (b) is Local Poisson matting in which the user selects a local region Ω_L interactively. $\partial\Omega$ is the exterior boundary of local unknown region $\Omega \cap \Omega_L$.

To sum up, among them, Ruzon-Tomasi, Hillman and Chuang use the same projecting algorithm in character to estimate alpha value. Knockout adopts a decomposing algorithm. In contrast, Poisson matting estimates the alpha value by solving a Poisson equation on the gradient field. These approaches can iteratively generate good mattes, which are applied in many fields. Some matting results are shown in Fig.6.

(a) Original (b) Knockout



(c) Bayesian (d) Poisson

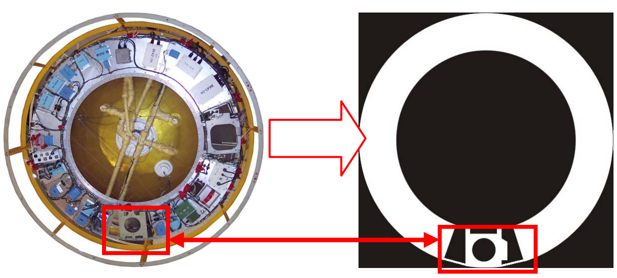
Fig.6 Natural image matting results

III. APPLYING ALPHA MAPPING

As described in section III, the image matte, namely the alpha channel, can be computed by the alpha estimate methods. And the results can be used in texture mapping to simulate all kinds of transparency effect, this is alpha mapping. In this section, based on the theory above, the applying of alpha mapping in OGRE is introduced. Through alpha mapping, many transparency effect of the scene, such as the ports of rocket body, the overflow and filling orifice of oxidant and fuel, can be realized.

A. Alpha estimation

There are five alpha estimation approaches in nature image matting introduced in section III. According these methods, at first, we segment the images into three regions: foreground, background and unknown. Then α for all pixels in the three region are estimated. The matting process of an image of the alveolate board on instrument cabin is shown in Fig.7.



(a) Original (b) Matting result

Fig.7 Alpha estimation of the alveolate board on instrument cabin

As shown in Fig.7, (a) is the original image of the alveolate board on instrument cabin, (b) is the matting result, in which, the α for all pixels are confirmed. The α for pixels in white region are set to 1, means completely opaque. And the α for pixels in black region are set to 0, means completely transparent. The final transparent effect is shown in Fig.11.

B. Material system of OGRE

After the alpha estimation, we can obtain the mattes of the textures, which are used in alpha mapping by the material system of OGRE. Fig.8 illustrates the framework of material system class in OGRE.

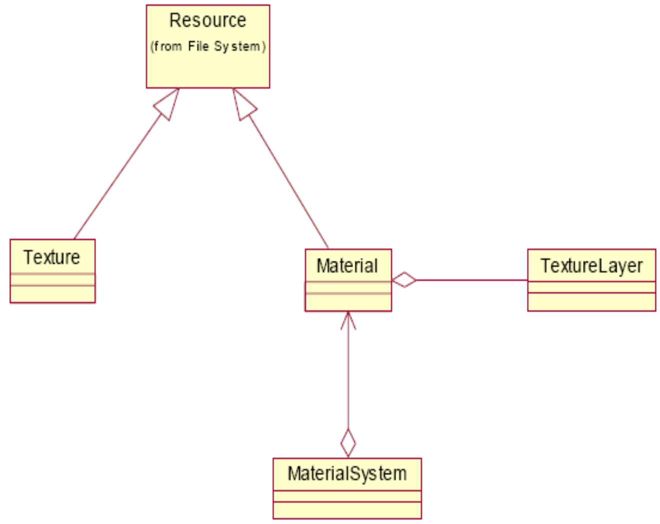


Fig.8 UML diagram of material system class in OGRE

Ogre's Material class, as a subclass of Resource (as shown in Fig.9), encapsulates all aspects of the visual appearance, of an object. It also includes other flags which might not be traditionally thought of as material properties such as culling modes and depth buffer settings, but these affect the appearance of the rendered object and are convenient to attach to the material since it keeps all the settings in one place. This is different to Direct3D which treats a material as just the color components (diffuse, specular) and not texture maps etc.

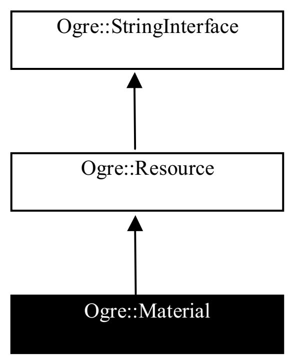


Fig.9 Inheritance diagram for Ogre::Material

Fig.9 shows the inheritance relation of Material class. Ogre comes configured with a number of default settings for a newly created material. These can be changed by retrieving the default material settings. Any changes we make to the Material returned from this method will apply to any materials created from this point onward.

Materials control the eventual surface rendering properties of geometry. The MaterialManager class manages the library of materials, dealing with programmatic registrations and lookups, as well as loading predefined Material settings from scripts. Fig.10 shows the inheritance relation of MaterialManager class.

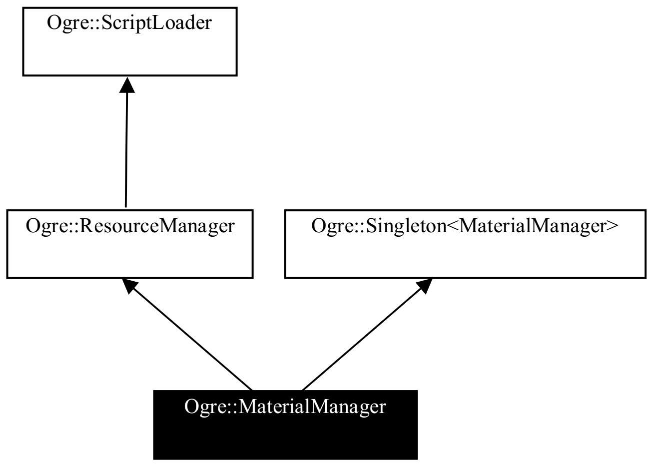


Fig.10 Inheritance diagram for Ogre::MaterialManager

As shown in Fig.10, it is a subclass of ResourceManager, any files loaded will be searched for in any path or archive added to the resource paths or archives. When loaded from a script, a Material is in an "unloaded" state and only stores the settings required. It does not at that stage load any textures. This is because the material settings may be loaded from bulk material script files, but only a subset will actually be required.

C. *Material scripts of OGRE*

In OGRE, the complex materials can be defined in a script which can be reused easily. We can store material definitions in text files which can then be loaded whenever required. Whilst we can also use this facility to replace existing material definitions which get loaded in from resource files.

Several materials may be defined in a single script. The script format is pseudo-C++. A material can be made up of many *techniques*. A technique is one way of achieving the effect we are looking for. OGRE supply more than one technique in order to provide fallback approaches where a card does not have the ability to render the preferred technique, or where we wish to define lower level of detail versions of the material in order to conserve rendering power when objects are more distant.

Each technique can be made up of many *passes*, which is a complete render of the object can be performed multiple times with different settings in order to produce composite effects. OGRE may also split the passes we have defined into many passes at runtime, if we define a pass which uses too many texture units for the card we are currently running on.

Each pass has a number of top-level attributes such as 'ambient' to set the amount & color of the ambient light reflected by the material. Within each pass, there can be zero or many texture units in use. These define the textures to be used and optionally some blending operations and texture effects. In the alpha mapping of OGRE, we use the image with alpha channel as texture unit.

In order to achieve the transparent effect, besides the texture with alpha channel, we need setting the kind of scene blending and depth buffer writing in the pass. Scene blending is about combining the output of this pass as a whole with the existing contents of the rendering target. This blending therefore allows object transparency and other special effects. Depth buffer writing sets the pass renders with depth-buffer writing on or off.

IV. Results

In the alpha mapping of scene simulation system on launch vehicle, we set the scene blending as "alpha_add", and set depth buffer writing as "off". Then the alpha value of the rendering output is used as a mask, and pixels are written without updating the depth buffer. Fig.11 illustrates the transparent effect of the alveolate board on instrument cabin. The zooming in of the region enclosed by red rectangle is the fixing hole of an instrument.

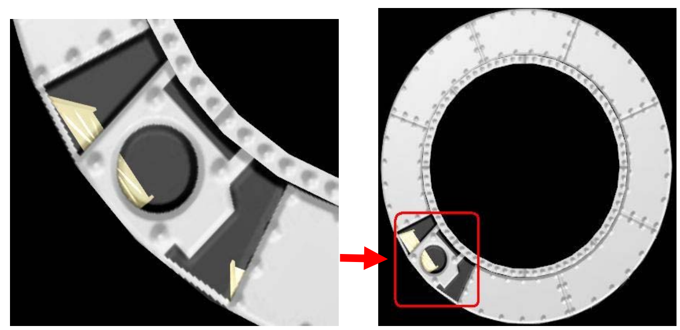


Fig.11 Transparent effect of the alveolate board on instrument cabin

Fig.12 illustrates the transparent and accidented effect of the sect between second and third segment on launch vehicle. The alpha mapping is combined with bump mapping in this example. In Fig.12, the upper and nether zooming in of the red rectangle are the ports on rocket body, the left one is the cable orifice, and the right on is the orifice of separating plug.

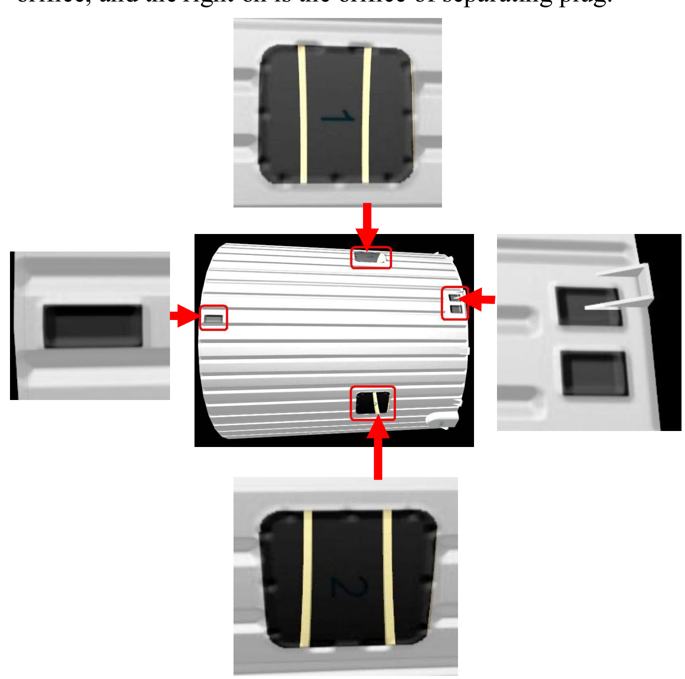


Fig.12 Combining alpha mapping with bump mapping

Fig.13 illustrates the transparent and decal effect of the low temperature store container in third segment. The same texture coordinates used for the alpha map are typically shared with decal map textures. In the pass of material scripts, we combine alpha mapping with a decal texture layer using multi-texture. In Fig.13, the zooming in of the red rectangle are the filling orifice of liquid oxygen and the overflow valve orifice.

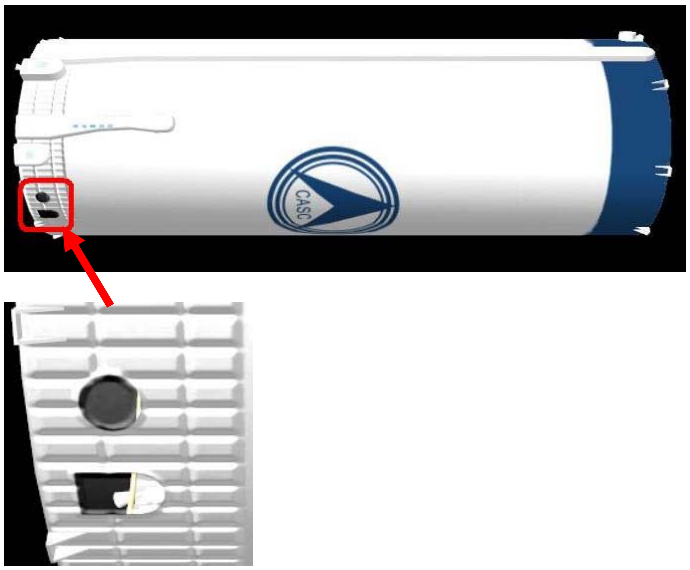


Fig.13 Combining alpha mapping with decal mapping

V. CONCLUSION

Several alpha estimation approaches in nature image matting were used in computing the α for pixels in an image, which was the texture unit of alpha mapping. After the alpha estimation, the mattes of the textures were used in the scene simulation system of launch vehicle by the material system of OGRE. Under the environment of OGRE engine, the transparent surfaces of the actual objects were simulated by alpha mapping approach, which allowed a low-cost real-time rendering of shape detail of an object surface without changing model geometry. The developed whole rocket model in the scene simulation system had only 13568 triangles. However, the model without alpha mapping method had more than 60000 triangles. On the other hand, an animation rate of more than 80 frames per second can be achieved by alpha mapping in the scene simulation system, which exceeds 60 frames per second requirement in a 3-dimensional application program. It shows that the alpha mapping improves the real-time performance significantly, ensures the third dimension of the diversified objects and scene, and plays an important role in the synthesization of third dimension graphics.

REFERENCES

[1] Catmull,E. “A subdivision algorithm for computer display of curved surfaces”, Ph.D. dissertation, University of Utah, 1974.

[2] Q.SH. Peng, H.J.Bao, X.G.Jing, “Arithmetic elements of computer third dimension graphics”, The Publishing Company of Science, 1999

[3] Alvy Ray Smith, “Alpha and the history of digital compositing,” Microsoft Tech Memo 7, 1995

[4] T. Porter and T. Duff, “Compositing digital images,” in *SIGGRAPH 1984*, pages 253-259, July 1984.

[5] A. R. Smith and J. F. “Blinn. Blue screen matting,” in *Proceedings of ACM SIGGRAPH 1996*, pages 259-268, Aug. 1996.

[6] Y.Mishima. “Soft edge chroma-key generation based upon hexoctahedral color space,” U.S. Patent 5,355,174, 1993.

[7] A. Berman, A. Dardourian, and P. Vlahos. “Method for removing from an image the background surrounding a selected object,” U.S. Patent 6,134,346, 2000.

[8] M. Ruzon and C. Tomasi. “Alpha estimation in natural images,” in *Proceedings of CVPR 2000*, pages 18-25, Jun. 2000.

[9] P. Hillman, J. Hannah and D. Renshaw. “Alpha channel estimation in high resolution images and image sequences,” in *Proceedings of CVPR 2001*, pages 1063-1068, Dec. 2001.

[10]Y. Y. Chuang, B. Curless, D. Salesin and R. Szeliski. “A Bayesian approach to digital matting,” in *Proceedings of CVPR 2001*, pages 264-271, Dec. 2001.

[11]Jian Sun, Jiaya Jia, Chi-Keung Tang and Heung-Yeung Shum. “Poisson Matting,” in *SIGGRAPH 2004*, Vol. 23, No. 3, pages 315-321. Jul. 2004.