CONTENT DEPENDENT IMAGE SAMPLING USING MATHEMATICAL MORPHOLOGY

Application to texture mapping

ETIENNE DECENCIERE FERRANDIERE,
BEATRIZ MARCOTEGUI and FERNAND MEYER
Centre de Morphologie Mathematique
Ecole Nationale Superieure des Mines de Paris
35, rue Saint Honore, 77305 Fontainebleau, France
Tel: +33 1 64 69 48 09, fax: +33 1 64 69 47 07
Email: decenciere@cmm.ensmp.fr

Abstract. Interactivity has become one of the main objectives of multimedia applications. Texture mapping techniques are essential to allow its development, but they all have one point in common: they are not content dependent. In this paper the development of content dependent texture mapping techniques is studied. A content dependent sampling process, based on a reference image which indicates the importance given to each pixel of the original image, is proposed. Two methods of building the reference image by means of morphological tools are described. This content dependent sampling method is used to build a mipmap, a classical structure used in texture mapping.

Key words: image sampling, content dependent sampling, texture mapping

1. Introduction

Multimedia interactive applications are rapidly developing. This enhanced interactivity and new freedom needs to be supported by texture rendering tools. Different texture mapping techniques exist [7], but, as long as we know, they all have one point in common: they are not content dependent. This means that images with different content will be treated the same way, and that inside one image all pixels are also processed without taking into account their "meaning". For example imagine a 3D scene featuring a notice-board hanging from a textured wall. The wall and the text will be rendered exactly in the same way. However, it could be interesting to better preserve the details of the text than the details of the wall texture, for instance.

In this article we focus on the downsampling problem. Note however that content based upsampling algorithms have also been recently developed in the same context by Albiol and Serra [1].

In classical computer graphics, the treatment of a texture pixel depends only on its position. In the present work, we will analyze the content of the texture image and downsample it depending on the importance associated to each pixel. To this end, a content dependent sampling method based on a reference image which indicates the importance given to each pixel of the original image is presented. Two methods for building the reference image that make use of mathematical morphology are proposed. Moreover, we will apply these downsampling techniques to texture mapping using mipmaps.

In order to test the resulting texture rendering algorithms in real time conditions, and to compare them with existing techniques, we have chosen MESA, a freeware implementation of OpenGL (Open Graphics Library), a software interface to graphics hardware. OpenGL is one of the leading products in its field. It is based on an open specification and many implementations are available. It is widely used by graphics software developers.

2. Content dependent sampling

We refer the reader interested by a review on texture mapping techniques to an article by Heckbert [7]. The techniques described in this review do not take into account the contents of the image when downsampling.

Mathematical morphology has already been applied to image sampling [5, 6, 8, 10]. In these works the aim was to simplify the image before sampling, in order to keep only those details that could be represented at the lower resolution level. In that sense these non-linear approaches already were content dependent, but in our case we want to develop tools that will enable us to preserve some details that are considered important, even if they are theoretically too small (the theoretical limit being given by the Shannon sampling theorem), while avoiding aliasing. Of course, this will only be possible if there is enough place in the image. In that sense, our work is inspired from [3, 4].

2.1. A general framework for image sampling

The simplest downsampling method is point sampling. It can be analyzed in the following way: the original image I is partitioned into regular 2×2 square blocks, and within each block the first pixel (in the video scanning order) is kept in order to build the sampled image J. We will denote $B(x, y)$ the block composed of pixels $\{(2x, 2y), (2x+1, 2y), (2x, 2y+1), (2x+1, 2y+1)\}$. This procedure is illustrated by figure 1.

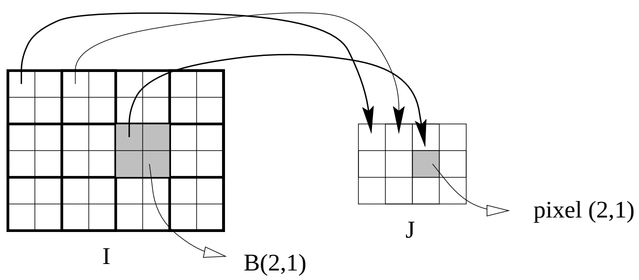

Fig. 1. Point sampling

Let R be a grey level image the same size as I, whose pixel values are defined by:

$$R(x,y) = 1 \quad \text{if } x \text{ and } y \text{ are even,} \tag{1}$$
$$R(x,y) = 0 \quad \text{otherwise.} \tag{2}$$

Then the point sampling operator can be expressed as follows:

$$J(x,y) = \sum_{P \in B(x,y)} R(P)I(P). \tag{3}$$

But why should we choose the first pixel in each block? The only reason is that its coordinates are even. The question then is whether it would not be better to keep the most interesting pixels among the four, or even better to combine the four pixels in some way. This leads us to the definition of a general sampling operator based on a reference image.

Definition: Reference sampling Let Ω be an operator which takes as input one color image I of size $n \times m$ (the original image) and one grey level image R (the **reference image**) of same size and which gives back an image J of half their size ($\frac{n}{2} \times \frac{m}{2}$):

$$J = \Omega(I, R). \tag{4}$$

Ω is a reference sampling operator if and only if:

$$J(x,y) = (\Omega(I,R))(x,y) = \begin{cases} \frac{\sum_{P \in B(x,y)} R(P)I(P)}{\sum_{P \in B(x,y)} R(P)} & \text{if } \sum_{P \in B(x,y)} R(P) \neq 0 \\ \frac{1}{4} \sum_{P \in B(x,y)} I(P) & \text{otherwise} \end{cases} \tag{5}$$

This means that in order to compute the value of the sampled image pixel $J(x,y)$ we compute a convolution of the pixels of I belonging to $B(x,y)$ using as weights the respective values of R.

Existing downsampling methods can be described via reference sampling. For example, a classical downsampling method based on convolution can be written as:

$$I_{n+1}(x,y) = \frac{1}{4} \sum_{P \in B(x,y)} I_n(P) \tag{6}$$

Using the above definition, this can be expressed as a reference sampling where the reference image is constant. The fact that this sampling method is not content dependent clearly appears here: the values of the reference image do not depend on the pixel values of the original image.

Precisely, the main interest of reference sampling is that it allows the construction of content dependent downsampling methods. The value in the reference image of a pixel P indicates its importance: the higher its value, the better it will be represented after downsampling.

The next step is the construction of the reference image R. Here is where morphological tools come into play. They will allow us to identify interesting pixels.

2.2. From the morphological gradient to the morphological Laplacian

From now on, for the sake of clarity, we will only consider grey level images. The generalization to color images is easy thanks to the fact that we will only use morphological operators to compute the reference image, and not to process the texture image itself. When applying these operators to a color image, we will in fact apply them to its luminance.

Interesting points in an image (maxima and minima, crest points) belong to high gradient regions. This is why we first tested the morphological gradient as reference image. However the gradient proved to be insatisfactory for this application for one main reason: not all high gradient points are interesting (for example slanted planes or noise produce high gradients).

Therefore we looked for an operator with a higher discrimination factor, and we found the morphological Laplacian, defined as:

$$L_S(I) = \delta_S(I) + \epsilon_S(I) - 2I, \tag{7}$$

where δ_S and ϵ_S respectively denote the morphological dilation and erosion, and S denotes the used structuring element.

Contrary to the gradient, the Laplacian may have negative values, which are as meaningful as positive ones. In order to build the reference image we take the absolute value of the Laplacian. Hence the sampling operator can be written:

$$J = \Omega(I, |L_S(I)|), \tag{8}$$

where $|.|$ denotes the absolute value operator. The second column of figure 3 shows two downsampling steps using this method. The original image is given by figure 2. In smooth areas the result is practically the same as the classical sampling method based on convolution (constant reference image), but in high contrasted areas contrast is preserved without introducing any aliasing.

2.3. Using the morphological tophat to detect details

Now we are going to detect details in a more explicit way using the tophat transform [9].

However, we cannot directly use the result of the tophat as a reference image; by doing so we would favor noise. We have to filter the tophat images in order to extract only the meaningful details. To this aim we use a hysteresis threshold followed by an area filter in order to obtain a binary image indicating interesting details. Finally we take the intersection between this binary image and the original tophat image in order to recover the grey levels of the details.

After applying this filtering step to the white tophat and the black tophat of the original image, we obtain two reference images. The first R_w corresponds to the light details in the original image, and the second R_b to the dark details. The final reference image is built by taking the supremum of both images: $R = R_w \bigvee R_b$.

The resulting downsamplig operator has been applied to our test image in figure 3 (third column). Note that preserved details are proportionally larger than in the original image, which might not be convenient for all applications, but in many cases this feature is be interesting.

2.4. Comparison and comments

We have presented two methods for computing reference images. We have applied them to one test image, showing text and rich textures. Figure 3 summarizes the results and allows to compare them with the classical downsampling method based on convolution.

Fig. 2. Test 2: Original image (256×256).

Note that the three sampling methods perform similarly in smooth regions. Differences appear in non-smooth areas like texture rich regions and near borders. Despite the fact that the representation of these images in this document cannot allow detailed viewing, it can be appreciated that images sampled with the Laplacian method are slightly less blurred than the images obtained with the classical convolution method. As noted previously, this has been achieved without introducing any visible aliasing. The improvement brought by the tophat method is more visible. More details are preserved from one downsampling level to the next. For example text in images downsampled with the tophat method is easier to read.

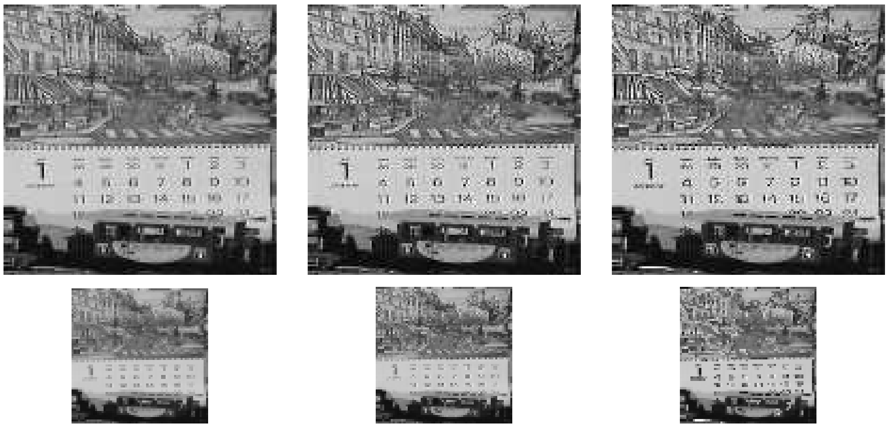

Fig. 3. Comparison of sampling results; the first line corresponds to the first downsampling step (image sizes are 128×128), and the second line to the second downsampling (image sizes are 64×64). First column: constant reference image. Second column: Laplacian sampling. Third column: Tophat sampling

Before concluding, we explain how to apply these content dependent sampling methods to texture mapping.

3. Application to texture mapping

3.1. Mipmapping

A classical class of texture mapping methods is based on the pre-calculation of a set of smaller versions of the original texture image I_0. If I_0 is of size $n \times m$, then a pyramid is built with images I_i of size $\frac{n}{2^i} \times \frac{m}{2^i}$. This pyramid is called a MIP map (MIP stands for the Latin phrase *Multum In Parvo* which means many things in a small place). These structures were first applied to texture mapping in [2]. They can be built and used in different ways.

3.2. Construction

A possible way of building a mipmap is recursively. Let I_i be the i-th level of the mipmap. I_i is thus an image of size $\frac{n}{2^i} \times \frac{m}{2^i}$. With a reference sampling operator Ω we can recursively compute the pyramid using the rule:

$$I_{i+1} = \Omega(I_i, R_i). \tag{9}$$

However if we proceed this way, in some cases image details could survive along several mipmap levels, which could be annoying for some applications. We propose here an alternate mipmap construction method which uses content dependent operators but that ensures that small details will be kept only in one mipmap level.

Let I_0 be the original texture image, Ω be the reference sampling operator, and g be the function that computes the reference image from a texture image. We want to build a mipmap starting from I_0: $(I_0, I_1, I_2, \ldots)$. Let K_n be a constant reference image the same size as I_n. Then the recursive procedure to build a mipmap which only preserves small details during one mipmap level is defined by:

$$I_0^{smooth} = I_0, \tag{10}$$

$$I_{n+1}^{smooth} = \Omega(I_n^{smooth}, K_n), \tag{11}$$

$$I_{n+1} = \Omega(I_n^{smooth}, g(I_n^{smooth}).) \tag{12}$$

This means that we build a parallel smooth mipmap structure $(I_0^{smooth}, I_1^{smooth}, I_2^{smooth}, \ldots)$ and that we apply a content dependent sampling operator to compute I_{n+1} from I_n^{smooth}.

3.3. Implementation

We have implemented the general reference downsampling method, as well as the two reference image construction methods (Laplacian method, tophat sampling) using XLIM3D, a mathematical morphology library developed at the Center of Mathematical Morphology.

The reference sampling procedure itself is very fast. The computation of the reference image may be slower, depending on the chosen method. For instance the Laplacian method is very fast, requiring only the computation of one dilation and one erosion, but the tophat method can take longer. It is not the computation of the tophat that takes most time, but the filtering step: the reconstruction used in the hysteresis threshold, as well as the area filtering, may require many iterations. However, thanks to the algorithms used in XLIM3D, which are based on hierarchical queues, we can now use these operators in real time. Finally, note that the construction of the mipmap is not as critical as its use during the rendering process.

The resulting mipmaps have been used within MESA, an implementation of OpenGL. The resulting texture mappings preserve the details better than the classical methods.

4. Conclusion

A general sampling method, called reference sampling, has been defined. It generalizes some classical downsampling strategies and moreover it allows the construction of content dependent downsampling operators.

Using the morphological Laplacian and the tophat operator two content dependent downsampling operators, which aim at keeping meaningful image details, have been proposed.

Finally, it has been explained how to build and use mipmaps based on these content dependent sampling operators. Their implementation under MESA gives interesting results.

A certain number of questions remain. What sampling method should we use? What values should we give to the parameters? The sampling method can be chosen according to the original texture image. If it is very smooth, then the Laplacian method should be used. Otherwise, the tophat method is more interesting. The parameters used by the tophat algorithm can also be adapted to the image: the higher their values, the less the details that will be kept.

These choices can be made on the spot, after analyzing the image, or be somehow included with the image. For example the composer of the 3D scene could choose the sampling technique for the textures he uses.

Finally, note that MPEG-4 provides a very interesting framework for the application of these new texture mapping techniques: each Video Object could be associated to one of these sampling techniques.

References

1. A. Albiol and J. Serra. Morphological image enlargements. *Journal of Visual Communication and Image Representation*, 8(4):367–383, December 1997.
2. E. Catmull. *A subdivison algorithm for computer display of curved surfaces.* PhD thesis, University of Utah, December 1974.
3. E. Decencière. Echantillonnage morphologique et réduction d'images. Technical report N-05/95/MM, Ecole des Mines de Paris, Paris, March 1995.
4. E. Decencière Ferrandière, B. Marcotegui, and F. Meyer. Application de la morphologie mathématique au mipmapping. In *CORESA'98*, June 1998.
5. D.A.F. Florêncio and R.W. Schafer. Critical morphological sampling and its applications to image coding. In *Mathematical Morphology and its Applications to Image Processing (Proceedings ISMM'94)*, pages 109–116, September 1994.
6. R.M. Haralick, X. Zhuang, C. Lin, and J.S.J. Lee. The digital morphological sampling theorem. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37:2067–2090, December 1989.
7. P.S. Heckbert. Survey of texture mapping. *IEEE Computer Graphics and Applications*, 6(11):56–67, November 1986.
8. H.J.A.M. Heijmans and A. Toet. Morphological sampling. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 54:384–400, November 1991.
9. F. Meyer. *Cytologie quantitative et morphologie mathématique.* PhD thesis, Ecole des Mines de Paris, Paris, 1979.
10. J. Serra. A sampling approach based on equicontinuity. In J. Serra and P. Soille, editors, *Mathematical Morphology and its Applications to Image Processing (Proceedings ISMM'94)*, pages 117–124, September 1994.