FILOMAT (Niš) 9:2 (1995), 251–259
Filomat '94, Niš, October 22-24, 1994
Geometry. Computer Sciences

# "EXACT" DISPLAY OF OBJECTS WITH REAL VALUED POSITIONS AND DIMENSIONS

**Siniša N. Hristov, Miomir S. Stanković and Vesna I. Veličković**

ABSTRACT. *In this paper we consider the correct method for the "exact" display of objects with arbitrary forms, having positions and dimensions expressed as arbitrary real numbers. We also consider advantages of such an approach over the usual methods which do not produce "exact" picture, or can "exactly" display only some forms of objects which must have integer positions and dimensions. We also consider some difficulties that might arise in an implementation.*

## 1. Introduction

This paper deals with methods for generation of an image from an internal description of a scene.

An **image** is a two-dimensional array of numbers, held in computer memory, from which the **actual picture** on the screen is produced by some suitable hardware. A single element of this array is known as a **pixel**, short for "picture element".

A **scene** is some internal description of the **desired picture**. We leave the particular form of the description undefined, but assume that it describes every detail of the desired picture with the complete precision.

We emphasize the distinction between the desired picture, represented by the scene description, and the actual one, represented by the image and presented on the screen.

1991 *Mathematics Subject Classification.* 68U05; 68U10.
*Key words and phrases.* computer graphics, signal processing, filters, aliasing.

## 2. Usual "nonexact" methods

Image generation methods used in most computer graphics packages fall into the following three categories:

1. turn-on fully all pixels that have their centers covered by the object;
2. turn-on fully all pixels that have at least half of their area covered by the object;
3. set the intensity of a pixel in proportion with that part of its area which is covered by the object.

It is known that each of these methods suffers from one or more of the following imperfections:

1. object edges appear "ragged";
2. all dimensions must be expressed as integer multiples of the pixel size;
3. objects are not displayed accurately enough – there is significant distortion of object's shape and position.

Some graphics package implementors have recognized the first disadvantage as a serious one and have provided an option to use some form of "anti-aliasing", so that objects appear to have more "smooth" edges. As the "anti-aliasing" is usually performed by some semi-empirical procedure, the resulting picture may appear "smooth", but it is nevertheless inaccurate. And inaccurate picture, having either "ragged" or "smooth" edges, has, as we shall see, serious practical deficiences.

Let us note that most computer graphics applications involve presentation of some scene which is generally defined in a continuous two-dimensional space. There are some applications, circuit board design, for example, which place objects on a predefined grid, but when comes to the image generation, the grid does not help much. Therefore, we shall restrict our discussion to continuos space only.

Some basic graphics packages allow only integer values of object coordinates and dimensions. They force the programmer to write explicit conversions from the continuous space, be it rounding or whatever. In this way, the programmer has full control over the actual picture, which she uses to create some clever arrangements of objects, disguising aforementioned imperfections as much as he can, [1]. Besides placing enormous burden on the programmer, such an approach suffers from a phenomenon common to all "singular" designs: small changes in input data can completely invalidate all she has achieved.

Numerous graphics packages allow specification of real values for coordinates and dimensions. But, to specify is one thing, and to display is quite

another. Some "less sophisticated" graphics packages simply round the real values to the nearest integers. The scene is effectively converted into a "similar" one, from which the image is generated. Many have noticed that the rounding errors introduced in this process are by no means insignificant. Other, "more sophisticated" graphics packages attempt to draw approximations of objects without rounding coordinates first. Some clever algorithms are employed to determine pixel values, and in specific cases acceptable results are produced, [1]. However, it is our impression that all such methods rely too much on clever tricks, without having solid theoretical background, and can, therefore, produce acceptable results only in limited cases.

Contrary to the popular belief, we find that an error of "a pixel or two" is by no means negligible, at least given the resolution of today's equipment. Here is a short list of most common consequences of such "small" errors.

1. A uniform set of objects from the continuous space appears as non-uniform, and vice versa.
2. Parts of objects or entire objects disappear.
3. Shape of a small object appears very distorted.
4. The original proportions of object positions and dimensions are not retained.
5. A small change in object's position or size can sometimes cause significant effect on the picture, while in some other case a much bigger change produces no effect.

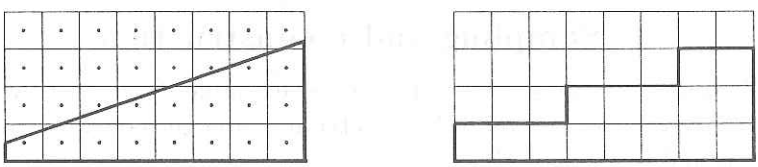

Figure 1. "Ragged" appearance of an object edge.

Figure 2. A uniform set of objects appears as non-uniform.

Increasing the image resolution makes such errors somewhat less noticeable, but still visible. The right way to fight this problem is certainly not to increase the image resolution. This is very expensive and quite limited by

Figure 3. A non-uniform set of objects appears as uniform.

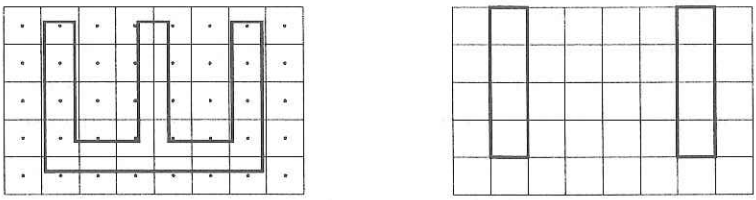

Figure 4. Some parts of the object disappear.
The shape appears very distorted.

Figure 5. The relative proportions are not retained.

the state of technology, and does not even touch the heart of the problem which is simply the improper sampling.

## 3. Sampling and reconstruction

The Shannon sampling theorem (see, for example [3]) says that a signal can be properly reconstructed if its spectrum is non-zero only at frequencies less than a half of the sampling rate.

If there is a signal component not satisfying this condition, it will be sampled, but the samples will look exactly as if they came from a component at some frequency less than a half of the sampling rate. In this case the reconstructed signal will have a component not originally present. This phenomenon is known as "aliasing".

As it was mentioned above, the scene is defined in a continuous space, and therefore shall be regarded as a continuous signal. The "image generation" process is, in fact, sampling. The scene may or may not satisfy the sampling theorem condition. The picture is produced from the image by reconstruction, or interpolation, which is performed by the graphics hardware.

Now we can define the correct method for image generation.

If the scene satisfies the sampling condition, everything is well – the aforementioned procedure **will** produce the exact picture. But, if the scene does not satisfy the sampling condition, the exact picture **cannot** be produced. Instead, a "correct" replacement shall be provided.

The question of the "correct" replacement is more philosophical and aesthetical one, rather than technical. We choose the following line of thought: if a component of the scene can be displayed exactly, then do so, and if it cannot be displayed exectly, then suppress it, rather than displaying it as something else that did not exist in the scene.

In other words, we do not attempt to sample the scene that does not satisfy the sampling condition. Instead, we transform that scene into a "similar" one satisfying the sampling condition. This is done by filtering the scene with well chosen low-pass filter. Please note that the scene is a continuous signal – we **cannot** apply a digital filter for this purpose because we do not have a digital signal.

To further substantiate our choice of the "correct" replacement, we note that the components that we have suppressed carry the structure too fine to be displayed by the hardware and/or noticed by the viewer. Therefore, we hope that the absence of those components will not do much harm neither. Had we done otherwise, those components would "alias" to lower frequencies, translating to much larger structure which will be displayed by the hardware and noticed by the viewer.

## 4. Practical advantages of exact display

Practical advantages of exact display of objects follow from the fact that dimensions and positions do not have to be unnaturally restricted to integer multiples of the pixel size.

An object may have arbitrary size and can be placed anywhere on the screen with the resolution determined by the precision of the floating point numbers used. The uniformity of a set of objects is preserved, as is the non-uniformity. The proportions are retained. A small change in object position or size produces the corresponding small effect on the pucture. In animation, objects do not jump irregularly from pixel to pixel, instead they move in uniform steps. The object shape is not distorted, although very small objects can be smeared or even completely invisible.

The programmer does not have to care about screen resolution and round off errors. And she gets exactly the picture she specified, unlike some modern graphics packages which do allow such a freedom of expression, but distort the picture and leave no possibility for the user to control the picture quality.

The quality of the computer graphics equipment is usually specified by the resolution in the sense of the size of the array holding the image. This is in contrast to the quality specification method used all other visual and optical devices, where the resolution denotes the size of the smallest object that can be reliably reproduced. The resolution of the graphics equipment with exact display of objects can be also given as the size of the smallest object that can be reliably reproduced, regardless of its allignment relative to the pixel grid.

## 5. How to produce exact pictures systematically?

A new graphics package must be written in order to enable application programmers to use exact pictures within their programs regularly. Although at the moment we do not have the complete proposition for such a package, we can state some basic requirements.

The scene shall be defined in a continuous space and shall be represented in the computer as a set of objects (not to be confused with the so-called "object oriented programming").

There shall be a predefined repertoire of parametrized primitive objects and the user will generate required number of instances and supply actual values for parameters, e.g. size, position, color, etc.

There shall be a systematic way of building complex objects from more primitive ones. Complex objects constructed in this way could also be parametrized, and any number of instances could be generated, with possibility to include them in still more complex objects.

Notions of a point and a line shall be defined in the mathematical sense, i.e. having no area. Therefore, they will not itself be objects, but will be used to build primitive objects.

Some set of predefined primitive objects shall be provided. It is important to select them very carefully, as it must be possible to draw them very efficiently, and, at the same time, to effectively use them in building complex objects and constructing typical scenes.

The package shall include basic geometric transforms, such as translation, rotation, scaling, etc. It shall be possible to apply those transforms uniformly to any kind of object, and to define complex transforms in terms of simpler ones.

Finally, when the complete scene is defined, a drawing procedure will be invoked to produce the image array by filtering the scene and sampling the filter output. Filtering must be performed analyticaly because a numerical approximation will involve sampling and result in aliasing. As long as all

objects in the scene are disjoint, filtering can be performed object by object, with all outputs summed – the filtering is a linear process.

Note that the filter output must be known only at sampling points. Therefore, filtering and sampling can be combined into a conceptually simple procedure of centering filter's impulse response at the sampling point and computing the convolution integral. The result is recorded as the pixel value.

## 6. Desirable filter properties

The most critical thing in implementation of the proposed approach is certainly the choice of the low-pass filter, which has crucial impact both on the quality of the picture and on implementation efficiency. We'll present now our preliminary view of desirable properties of such a filter.

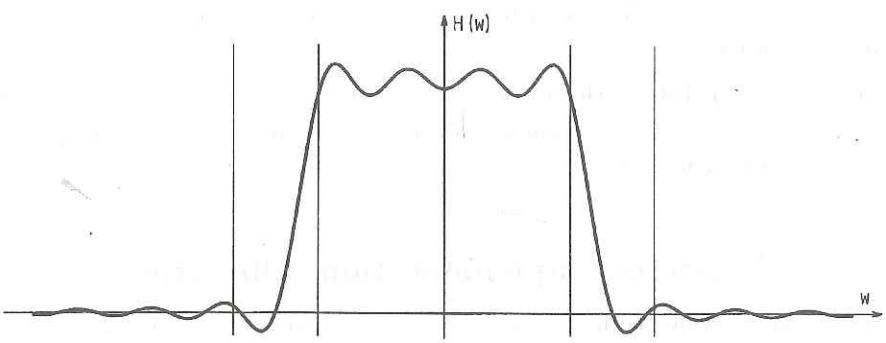


Figure 6. A filter shape in the frequency domain.

Desirable filter properties in the frequency domain are:

1. Relative intensity of components with different frequencies must not be considerably distorted, that is, the variation of the $|H(\omega)|$ in the passband shall be from 0.5% to 2%.
2. Components not satisfying the sampling theorem condition should be sufficiently supressed, that is, peaks of $|H(\omega)|$ in the stopband shall be from 0.2% to 1%.
3. In order to use as wide frequency band as possible, which means as much scene details as possible, the transition band shall be as narrow as possible, that is

$$\frac{\text{upper limit of the passband}}{\text{lower limit of the stopband}} = \quad \text{from 0.3 to 0.9.}$$

Desirable filter properties in the spatial domain are:

1. To enable computationally efficient filtering, the filter's impulse response shall be non-zero only over a finite interval, and that interval shall be as narrow as possible. This allows us to consider only a relatively small number of neighbouring objects while computing the value of the filtered scene at a given point.
2. To prevent apperance of a fine structure which did not exist in the scene, the filter's step response should be monotonic, or at least the amplutude of oscillations should not exceed 0.05% to 2%. Also, the filter's impulse response should not have negative values.
3. The transition from one light intensity to another should be as fast as possible, that is, the filter's rise time should be as short as possible.
4. The filter's delay should not depends on frequency or, even better, the delay should be zero. This will be satisfied if the filter's impulse response is an even function.
5. If the scene is rotated, the displayed picture shall appear rotated, but otherwise unchanged. This will be satisfied if the filter's impulse response is rotationally symmetric.

## 7. Expected implementation difficulties

A relatively complex calculation must be performed in order to obtain the value of a single pixel. The same procedure must be performed about a milion times to complete the image. We expect that achieving a reasonable drawing speed will be the major problem in the implementation.

A rough estimate shows that commonly available processors such as 486, 68040 and T805 permit only experimentation with the proposed approach. For practical applications processing must be faster for at least one order of magnitude. As we have to work with continuous signals and relatively complex data structures and algorithms, currently available digital signal processors do not seem to be particularly useful – they are very diffucult to use for anything outside their intended application area.

Our attention is directed towards the T9000, if it becomes regularly available. It seems that a single T9000 might satisfy basic application requirements. Even more important is its capability for parallel operation, which far exeeds capabilities of all other commercial processors. The single PPC 604 also seems to have enough power for basic applications, but with much less hope for efficient parallel operation.

Another serious difficulty arises from the possibility that a single filter may not be conveniently applicable to all necessary types of objects.

## 8. Conclusion

We have described some initial results of our work is this area. Currently we are investigating various classes of continuous finite-response filters in order to select viable candidates for an experimental implementation of a small graphics package exploiting the principles set in this paper.

## References

[1] Foley J.D., van Dam A., Feiner S.K., Hughes J.F., *Computer Graphics - Principles and practice*, Addison-Wesley, 1990.

[2] Corner, J.B., *Return of the Jaggy*, IEEE Computer Graphics **9 No 2** (March 1989), 82 – 89.

[3] Jerri A.J., *The Shannon sampling theorem – its various extensions and application: A tutorial review*, Proc. IEEE **65 No 11** (Nov. 1977), 1565 – 1596.

[4] Castleman, K.R., *Digital Image Processing*, Prentice-Hall, Inc, New Jersey, 1979.

[5] Max N.L., *Antialiasing scan-line data*, IEEE Computer Graphics and Aplications **10 No 1** (January 1990), 18 – 30.

[6] INMOS Limited, *The T9000 Transputer Instruction Set Manual* (1993), Bristol.

[7] INMOS Limited, *The T9000 Transputer Hardware Reference Manual* (1993), Bristol.

Božidara Adžije 19/I-11, 18000 Niš.
*E-mail address*: sike@unitop.elfak.ni.ac.yu

Faculty of occupational safety, Čarnojevića 10, 18000 Niš.

Filozofski fakultet, Ćirila i Metodija 2, 18000 Niš.
*E-mail address*: vesna@archimed.filfak.ni.ac.yu