Sensaura

Implementing ZoomFX™

3D objects in 3D space using DS3D

by Mike Percy

Since the early development of stereo in the 1930s, it has been possible to create aural *soundscapes* and assign spatial positions to the individually recorded sounds within the reproduced sound field. For many years, this was restricted to the space lying between (and bounded by) the loudspeakers. Recently, however, with the advent of efficient transaural crosstalk cancellation systems [1-4], it has become possible to recreate a full, 360 degree spherical sound field around the listener and to reproduce sound in three dimensions. Nevertheless, it is common practice to synthesise sound sources in such three-dimensional systems as *point sources*. Although this is reasonable for some types of sound emitter (such as an insect, for example), many real-world sound emitters, such as heavy trucks and railroad trains, are not point sources and are better represented by line-, area- or volume-type sound sources.

In order to confer more realism on synthesised soundscapes, Sensaura ZoomFX has been developed to overcome the restraints of point-source sounds[5]. For the first time, sounds are now able to have SIZE!

For example, it is now possible to create, realistically, the sound of a passing train, where the sounds appear to come from the whole length of the train, rather than an arbitrary point. Large objects, when close to the listener, can fill the soundscape in exactly the same way as they fill the field of view. When they depart the listener, they dwindle to a point source ... just like real-life. Using ZoomFX, audio designers can now position three-dimensional *objects* in their three-dimensional *spaces*.

1 How does ZoomFX work?

ZoomFX integrates seamlessly with DirectSound3D via a property set and extends the capabilities of DS3D to include new features that were previously unavailable to audio designers. These include the ability to assign a *size* to sound-emitting objects, in addition to its spatial position in three-dimensional space and the associated environmental acoustic effects.

Sensaura™
3D POSITIONAL AUDIO

www.sensaura.com

The API has been designed to be familiar to developers of Direct3D and DirectSound3D, in order to allow ease of use. This is described more fully in a subsequent section.

ZoomFX is an open-standard API (Applications Programming Interface), which means that any hardware manufacturer is free to implement it. This has been done to ensure uniformity throughout the audio community and to eliminate the need for conflicting proprietary standards. The API gives developers of hardware and software alike the ability to define an audio object's size and shape in one universally recognised form.

Sensaura's own implementation of the ZoomFX API creates wide-area sound emitters, using our proprietary *distributed sources* methodology. In essence, the original sound-source, which is to be *sized*, is processed via an algorithm that produces several complementary variants of it, such that the sum of these variants is equal to the original signal. Next, these variants (which are individually perceivable) are distributed throughout the area or volume that represents the sound source itself, thus creating a composite sound source. The brain interprets these similar, but distinct, signals by believing that they derive from a large area or volume source, in much the same way as the eye can interpret a series of similar images as one moving picture.

One important outcome of creating realistically large virtual sound-sources is that *they scale with distance*. They appear to increase in size as they draw near, and diminish as they depart from the listener, according to the angle subtended at the listener's head.

For more information about this, please refer to our other Technical White Paper[5]: *ZoomFX for 3D-sound*.

2 The ZoomFX Model

ZoomFX sources are described by a bounding box (**ZOOMFX_BOX**) and an orientation (**ZOOMFX_ORIENTATION**). The box works in much the same way as the bounding box for Direct3DRM. The orientation also is very similar to Direct3DRM, with a *front* and a *top* vector. The ZoomFX hardware takes this information, along with all the other DirectSound3D and property-set information, and creates a 'dimensioned' 3D sound-source.

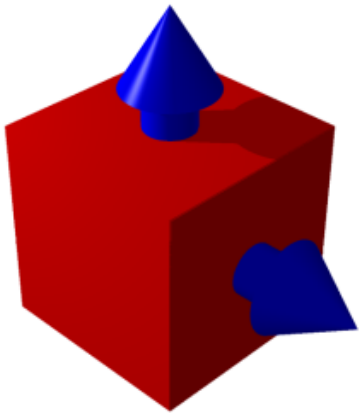

Figure 1: The ZoomFX object model

3 The ZoomFX API

The bounding box, used to describe the *size* of the ZoomFX sound source, is defined using the following, simple structure:

```
typedef struct
{
        D3DVECTOR vMin;
        D3DVECTOR vMax;
} ZOOMFX_BOX, *LPZOOMFX_BOX;

#define   ZOOMFXBUFFER_BOX_DEFAULT \
                { { 0.0f, 0.0f, 0.0f }, { 0.0f, 0.0f, 0.0f } }
```

Similarly, the *orientation* is described by the following, simple structure:

```
typedef struct
{
        D3DVECTOR vFront;
        D3DVECTOR vTop;
} ZOOMFX_ORIENTATION, *LPZOOMFX_ORIENTATION;

#define   ZOOMFXBUFFER_ORIENTATION_DEFAULT \
                { { 0.0f, 0.0f, 1.0f }, { 0.0f, 1.0f, 0.0f } }
```

These two structures combine to describe each ZoomFX buffer as shown in Figure 2.

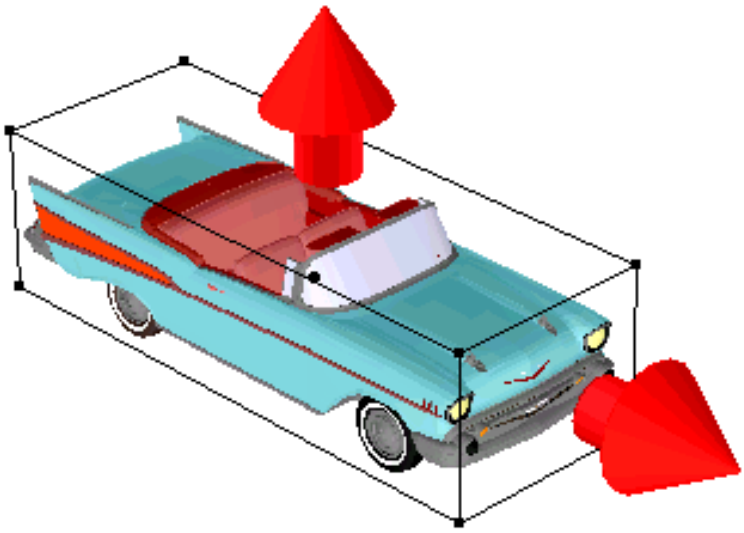

Figure 2: Example object showing ZOOMFX_BOX and ZOOMFX_ORIENTATION

The co-ordinate system used in all of these structures is the same as that used by the rest of DirectSound3D.

N.B. ZoomFX is *off* (i.e. no resources are used) if the **ZOOMFX_BOX** is set to (0,0,0),(0,0,0).

4 Recommended use

The ZoomFX API is intended to be used with sounds or ambient effects in a game where a point source does not accurately represent the sound emitter. For example, in a racing game, if you were driving past a stand of seats, the sound of the crowd cheering would be better represented by a wide area sound-source rather than by a simple point-source. Similarly, if you were standing close to, say, a military tank in a first- or third-person game, then the sound of the tank would be best represented by a large sound-source. (A tank is certainly not a point source!)

Obviously, it is not necessary to apply ZoomFX to all sounds. When rockets and bullets fly past, for example, they are quite well simulated by point source emitters.

To preserve system resources, the Sensaura implementation of ZoomFX will not apply any additional processing to sounds that are far enough away to subtend only a small angle at the listener.

Some ambient sound effects, such as crowd noise in a sports stadium, are well suited to a large-area or volume sound source. The developer must weigh up the advantages of using a 3D-buffer with ZoomFX to render such sounds against the advantages of using a pre-rendered 2D-buffer. Using a 3D buffer is more flexible, gives a better positional effect and supports headphones, 2- and 4-speaker systems, whereas pre-rendering uses no 3D resources, and works on non 3D-sound/ZoomFX capable systems.

Since the **ZOOMFX_BOX** property is likely to be similar to a graphical bounding box (indeed, it is identical to the **D3DRMBOX** structure) used for an in-game object, it is recommended that developers use the two interchangeably, getting and setting the box to the same values. Likewise the orientation is likely to be similar for both graphics and sound, and can be treated similarly, as is shown in the example overleaf.

At all times, of course, the developer should bear in mind how the scene will sound on a system that is not ZoomFX capable. The spatial point-position of the buffer remains important within the ZoomFX box, since it is from this position that a point source will be rendered when ZoomFX is unavailable.

```
#include <d3drm.h>
#include "zoomfx.h"

LPKSPROPERTYSET pKsPropertySet;
LPDIRECT3DRMFRAME3 pScene;
LPDIRECT3DRMFRAME3 pFrame;

bool SetZoomFX()
{
D3DRMBOX box;
ZOOMFX_ORIENTATION orientation;
ZOOMFX_BOX zbox;

if(FAILED(pFrame->GetOrientation(pScene, &(orientation.vFront),
&(orientation.vTop))))
return false;

if(FAILED(pFrame->GetBox(&box)))
return false;

zbox.vMin = box.min;
zbox.vMax = box.max;

if(pKsPropertySet->Set(DSPROPERTY_ZOOMFXBUFFER_ORIENTATION, NULL, 0,
(void*)&orientation, sizeof(ZOOMFX_ORIENTATION)) != S_OK)
return false;

if(pKsPropertySet->Set(DSPROPERTY_ZOOMFXBUFFER_BOX, NULL, 0,
(void*)&zbox, sizeof(ZOOMFX_BOX)) != S_OK)
return false;

return true;
}
```

5 References

1. Apparent sound source translator.
B S Atal and M R Schroeder
US Patent 3,236,949.

2. Models of hearing.
M R Schroeder
Proc. IEEE, Sep 1975, 63, (9),
pp. 1332-1350.

3. Transaural acoustic crosstalk cancellation.
A Sibbald
Sensaura White Paper (devpc009.pdf).

4. XTC crosstalk cancellation.
A Sibbald
Sensaura White Paper (devpc008.pdf).

5. ZoomFX for 3D-sound.
A Sibbald
Sensaura White Paper (devpc012.pdf).

For further information please contact:

Email: dev@sensaura.com
WWW: www.sensaura.com
Tel: +44 20 8848 6636

Sensaura Limited
Dawley Road, Hayes
Middlesex, UB3 1HH
United Kingdom

A Scipher Company

DEVPC/013/0002/1