Appendix B

A Simple Image Compression Example using WAILI

```
//
//      Simple image compression example
//

#include <waili/Image.h>

int main(void)
{
    const char infile[] = "image.pgm";
    const char outfile[] = "result.pgm";
    double threshold = 20.0;
    Image image;

    // Read the image
    image.Import(infile);

    // Transform the image using the Cohen-Daubechies-Feauveau
    // (2, 2) biorthogonal wavelets
    Wavelet *wavelet = Wavelet::CreateCDF(2, 2);
    TransformDescriptor transform[] = {
        { TT_ColsRows, wavelet },
        { TT_ColsRows, wavelet },
        { TT_ColsRows, wavelet },
        { TT_ColsRows, wavelet },
        { TT_ColsRows, wavelet },
        { TT_ColsRows, wavelet },
        { TT_ColsRows, wavelet },
        { TT_ColsRows, wavelet }
```

```
    };
    image.Fwt(transform, sizeof(transform)/sizeof(*transform));

    // Zero all entries smaller than the threshold
    for (u_int ch = 0; ch < image.GetChannels(); ch++)
        image[ch]->Threshold(threshold);

    // Inverse wavelet transform
    image.IFwt();

    // Write the reconstructed image to a file
    image.Export(outfile);
    return(0);
}
```

After the extension of WAILI to support very large images that are divided in blocks, the above example had to be changed:

```
//
//      Simple image compression example
//

#include <waili/Image.h>

int main(void)
{
    const char infile[] = "image.pgm";
    const char outfile[] = "result.pgm";
    double threshold = 20.0;
    BlockMgrM mgr;

    // Read the image
    Image *image = new Image(&mgr, infile);

    // Transform the image using the Cohen-Daubechies-Feauveau
    // (2, 2) biorthogonal wavelets
    Wavelet *wavelet = Wavelet::CreateCDF(2, 2);
    u8 id = wavelet->GetID();
    delete wavelet;
    TransformDescriptor transform[] = {
        { TT_ColsRows, id },
        { TT_ColsRows, id },
        { TT_ColsRows, id },
        { TT_ColsRows, id },
        { TT_ColsRows, id },
        { TT_ColsRows, id },
        { TT_ColsRows, id },
```

```
        { TT_ColsRows, id }
    };
    image->Fwt(transform, sizeof(transform)/sizeof(*transform));

    // Zero all entries smaller than the threshold
    for (u_int ch = 0; ch < image->GetChannels(); ch++)
        (*image)[ch]->Threshold(threshold);

    // Inverse wavelet transform
    image->IFwt();

    // Write the reconstructed image to a file
    image->Export(outfile);
    return 0;
}
```

The main difference between the two examples is the addition of the BlockMgr object when creating a new image. The *BlockMgrM* class manages blocks that are stored in main memory. The addition of the *GetID()* function is related to other evolutions of the internals of WAILI.