METHOD AND SYSTEM FOR MULTI-LEVEL ITERATIVE FILTERING OF MULTI-DIMENSIONAL DATA STRUCTURES

RELATED APPLICATIONS

This application claims priority to U.S. Provisional Patent Application No. 60/088,196, filed on Jun. 5, 1998 and European Patent Application No. 98200258.6, filed on Feb. 2, 1998.

BACKGROUND OF THE INVENTION

1. Field of the Inventions

The present invention relates to a method of multi-level iterative filtering of data structures of two dimensions or more and to a filtering system for carrying out the method which may be included in an encoder and/or a decoder. The present invention is particularly suitable for the filtering of image data.

2. Description of the Related Technology

In image processing systems, memory represents a high cost in size, power and speed, especially in multi-pass processing (e.g. iterative processing on multi-resolution data). In low-cost VLSI implementation styles, only limited-sized memories can be put on-chip, since for example, 10 kB cover already a silicon area of 11 mm^2 in 0.5 μm MIETEC CMOS triple layer metal technology. Off-chip memories, on the other hand, also represent a considerable cost, because read/write operations to and from external memory engender a power consumption that is typically orders of magnitude higher than the power consumption emanating from arithmetic operations. Furthermore, accesses to external memories are slow, compared to on-chip memory accesses, causing an impediment to the overall speed of the system. Real-time, power-efficient systems should therefore minimize the on-chip memory size and off-chip memory accesses.

Texture mapping on 3D objects in a virtual reality scene requires different texture resolutions, depending on the viewing distance. Current Discrete Cosine Transform (DCT) coding of textures only supports two levels of scalability (base layer+enhancement layer). Extending the number of resolution levels in a DCT scheme to more than two can be achieved with the multi-level Laplace Pyramid representation, at the expense of a 33% increase in the number of pixels to be coded. On the other hand, the wavelet texture coding, based on the Discrete Wavelet Transform (DWT), achieves an unlimited number of resolution levels, while providing excellent compression performance and is therefore better suited for applications requiring a large range of spatial scalability. FIGS. **1**(*a*) and **2**(*a*) show the algorithmic flow graphs of the multi-level DCT and Wavelet codings, respectively. Both schemes use essentially the same approach: a first stage transforms the image into a multi-resolution representation by successive filtering operations, and a second stage for the actual coding: parent-children coding for DWT, 8×8 block-oriented transform (DCT)-coding for DCT. With reference to FIG. **1**(*a*), in multi-level DCT coding the input image **10** is filtered in the first filtering step **1** to form a high pass subimage **4** and a low pass subimage **11**. High pass subimage **4** is output to the interface memory (IM) **8**. The low pass subimage **11** is filtered in the second level filtering step **2** to form a high pass subimage **5** and a low pass subimage **12**. Each filtering step **1,2,3** outputs a high pass subimage **4,5,6** to the IM **8**. The low pass subimage **13** from the last filtering step (the highest level) is also output to the IM **8**. Parent-children trees are indicated at **7**. The stored subimages are compressed by DCT compression circuits **9** to form the transmitted compressed image.

With reference to FIG. **2**(*a*), in multi-level DWT coding input image **10** is filtered in the first step **31** to form four subimages **11, 34–36**. These subimages are referred to as LL (**11**), LH (**36**), HL (**35**) and HH (**34**). The LL subimage **11** contains the low frequency image information from both the vertical and the horizontal wavelet convolutions. The LH and HL subimages **36, 35** contain information from the vertical and horizontal wavelet convolutions whereby in each subimage each direction takes a different one, of the high frequency and low frequency image informations. The HH **34** transform contains the high frequency image information from both the vertical and horizontal wavelet convolutions. The LL subimage **11** is filtered in the second filtering step **32** to again form four LL, HH, HL and LH subimages **12**, **37**, **38**, **39** respectively. The LL image **13** from the last filtering step (in the last level) is stored in the IM **8**. The subimages **34–42** in the three levels are stored in the IM **8** before being compressed by the compression circuits **43, 44** for the HL, LH and HH subimages **34–42** and the LL subimage **13** respectively. Parent-children trees are shown at **7**.

Note that the DWT coding requires information throughout the levels of the multi-resolution representation, while the DCT coding codes the blocks in each level separately. However, the DCT decoding does require a parent-children tree approach for the decoder memory optimization: all the DCT blocks that after decoding correspond to one particular 8×8 block in the decompressed image are preferably processed in the decoder simultaneously and should therefore be transmitted to the decoder as one cluster. Thus, the DCT encoding does not require the parent-children trees, but a memory optimized decoding process may exploit the data-dependencies of a parent-children tree. As a consequence, the data processing in the DWT and DCT encoders is essentially similar as seen from the memory optimization point of view: a successive filtering stage for obtaining the multi-resolution representation is followed by a coding stage with a parent-children data-dependency graph used at least in the decoding. Differences between the DCT and the DWT can be summarized as follows:

1. The parent-children data-dependency in the DCT codec is larger than in the wavelet codec: in the latter, the parent represents only one pixel, while in,the former, the parent extends over an 8×8 block.
2. The DWT inherently uses the multi-resolution representation for the image coding, while in the scalable DCT coding, the multi-resolution representation is an awkward pre-processing step that does not prepare the actual coding stage, i.e. the inter-relation between the levels is not exploited.
3. The number of pixels increases with 33% in the multi-resolution representation of the DCT codec, compared to the original image size, while the multi-level wavelet transformed image has the same size as the input image.
4. The arithmetic complexity of the multi-level DWT is typically smaller than that of its DCT counterpart.

These reasons indicate that DCT coding is not optimal for scalable coding.

In many applications it would be desirable to be able to change the resolution of not only the whole but also a part of a transmitted image. For instance in medical diagnosis many parts of an X-ray image or photograph are irrelevant whereas certain areas maybe vitally important and require maximum resolution (preferably-loss-free) and size. Where