Global Journals LaTeX JournalKaleidoscope™
Artificial Intelligence formulated this projection for compatibility purposes from the original article published at Global Journals. However, this technology is currently in beta. *Therefore, kindly ignore odd layouts, missed formulae, text, tables, or figures.*

Implementing 3D Warping Method In Wavelet Domain

Dr. Indrit Enesi[1], Elma Zanaj[2] and Betim AiAo[3]

[1] Polytechnic University of Tirana

Received: 27 October 2011 Accepted: 26 November 2011 Published: 7 December 2011

Abstract

A wide class of operations on images can be performed directly in the wavelet domain by operating on coefficients of the wavelet transforms of the images and other matrices defined by these operations. Operating in the wavelet domain enables one to perform these operations progressively in a coarse-to-fine fashion, operate on different resolutions, manipulate features at different scales, and localize the operation in both the spatial and the frequency domains. Performing such operations in the wavelet domain and then reconstructing the result is also often more efficient than performing the same operation in the standard direct fashion. Performing 3D warping in the wavelet domain is in many cases faster than their direct computation. In this paper we demonstrate our approach both on still and sequences of images.

Index terms— 3D warping, wavelet, multiresolution, planar, cylindrical, spherical, temporal coherence.

1 Implementing 3D Warping Method In Wavelet Domain

Indrit Enesi ? , Elma Zanaj ? , Betim Çiço ? Abstract -A wide class of operations on images can be performed directly in the wavelet domain by operating on coefficients of the wavelet transforms of the images and other matrices defined by these operations. Operating in the wavelet domain enables one to perform these operations progressively in a coarse-to-fine fashion, operate on different resolutions, manipulate features at different scales, and localize the operation in both the spatial and the frequency domains.

Performing such operations in the wavelet domain and then reconstructing the result is also often more efficient than performing the same operation in the standard direct fashion. Performing 3D warping in the wavelet domain is in many cases faster than their direct computation. In this paper we demonstrate our approach both on still and sequences of images.

Keywords : 3D warping, wavelet, multiresolution, planar, cylindrical, spherical, temporal coherence.

I.

2 IMAGE-BASED RENDERING AND 3D WARPING

any image-based rendering algorithms use prerendered or pre-acquired reference images of a 3D scene in order to synthesize novel views of the scene. The central computational component of such algorithms is 3D image warping, which performs the mapping of pixels in the reference images to their coordinates in the target image. In this paper we present wavelet warping -a new class of forward 3D warping algorithms for image based rendering. We rewrite the 3D warping equations as a point wise quotient of linear combinations of matrices. Rather than computing these linear combinations in a standard manner, we first precompute the wavelet transforms of the participating matrices. Next, we perform the linear combinations using only the unique non-zero wavelet transform coefficients. Applying the inverse wavelet transform to the resulting coefficients yields the desired linear combinations. We describe in detail wavelet warping algorithms for three common types of 3D image warps: planar-to-planar, cylindrical-to-planar, and spherical-toplanar. Current viewers allow the user to interactively change the viewing direction [1]. By using depth information, a 3D warper enables users to change the viewing position (center of projection), in addition to the viewing direction ??2]. A fast 3D warper enables users to view a scene interactively. We will show that the wavelet Author ? ? ? : Polytechnic University of Tirana, Albania. E-mails : ienesi @fti.edu.al, ezanaj@fti.edu.al, bcico@fti.edu.al warping algorithm is at least as fast as the most efficient warping algorithm known to date for planar and cylindrical warps, and is nearly twice as fast in the spherical case. Perhaps more importantly, our wavelet warping algorithms support progressive multiresolution rendering. Considering an object whose image-based model consists of one or more high-resolution reference images, the high resolution may be necessary for a close-up view of the object, but for most views of a 3D scene containing the object a much lower resolution suffices. Our approach makes it possible to perform the warp at the appropriate coarser resolution, without unnecessarily warping every pixel in the reference images. Multi-resolution warping can also be achieved within a standard warping framework by using an overcomplete pyramid-based image representation (e.g. a Laplacian pyramid), but at a cost of increasing the size of the representation [3]. Multiresolution wavelet warping has the advantage that the computation is progressive: a low resolution result can be progressively refined without redundant computations. We present a new algorithm for warping an entire sequence of images with depth to a novel view. This algorithm is also based on wavelet warping, and it utilizes the temporal coherence typically present in image sequences or panoramic movies to achieve considerable speedups over frameby-frame warping.

3 II. CHOICE OF WAVELET TRANSFORM

There are two main requirements that a wavelet transform should satisfy in order to be suitable for our framework [4]:

1. The transform should be sparse.

4 Reconstruction (inverse wavelet transform) should

be fast to compute.

To achieve faster reconstruction we choose transforms with smaller support size, and therefore fewer vanishing moments. Thisrules out the 9-7 transform which is considerably slower that the other transforms ??5]. In particular, this transform requires floating point arithmetic, whereas the other transforms can be implemented using only integer additions and shifts. The S+P and TS transforms are similar. They are both special cases of the same transform, which is factored into the S transform followed by an additional lifting step, but with different prediction coefficients ??6]. For our purposes it is sufficient to experiment with the more efficient TS transform. In order to assess the speed and the sparsity of the remaining three wavelet bases (S, TS, and I(2, 2)) we gathered the relevant statistics over a database of 300 photography images representing landscapes, buildings, people, products, etc.. Each image was transformed from RGB to YIQ color space and processed at full (640x384) and at half (320x192) resolutions. The results are summarized in Tab. 1 and 2 and plots in Fig. 2. Our experiments indicate that all three transforms provide roughly the same sparsity of wavelet domain representation for natural images. We note that the percentage of remaining coefficients is typically higher when operating on the half-resolution versions of the image. Decreasing the resolution results in smaller smooth regions in the images, and applying a transform with few vanishing moments yields fewer near-zero coefficients over these regions. In terms of speed, the S and I(2, 2) transforms are the fastest (the S transform is slightly faster), while the TS transform is slower by a factor of roughly 2 [7]. Consequently, the S transform was chosen for wavelet domain image blending and for wavelet domain convolution.

Tab. 1 : A comparison between the S, TS, and I(2,2) transforms For each transform and each image resolution we list the mean reconstruction time in milliseconds and the mean percentage of remaining (non-zero) coefficients, and the standard deviation corresponding to each mean.

Tab. 2 : The average number of distinct non-zero values in a wavelet-transformed image for each of the Y, I,Q channels. So far we have only considered lossless wavelet domain representation of images (only coefficients that become identically zero as a result of the wavelet transform are eliminated from the representation). Lossy representations obtained by zeroing out small wavelet coefficients yield a drastic reduction in the number of remaining coefficient in return for a modest increase in RMS error, as demonstrated by the plots in Fig. 2. Such representations can be acceptable if numerical accuracy is not critical. When choosing a wavelet transform for a lossy wavelet domain representation one additional requirement must be taken into account, the graceful degradation in visual quality of the image. In this respect we found the slower biorthogonal TS transform to be superior to the S transform. More specifically, the lossy TS transform tends to produce smoother and more visually accurate results compared to the lossy S transform, which introduces blocky artifacts.

5 III.

6 INTEGER WAVELET WARPING

In order to perform 3D warping in the wavelet domain, we express the warping equations as elementwise divisions between linear combinations of four matrices ??8]. Let Fi denote the matrix of all the values fi(x, y), and let U and V denote the matrices containing all of the warped u and v target coordinates. Using these matrices we rewrite equation (2) as: (1) where:

(2)

In the planar-to-planar warp, for example, the linear combination coefficients mij are the pij's from equation (3), and the matrices are defined as follows:

(3) Thus, the matrix A is simply a linear ramp, increasing from left to right; all of its rows are the same vector [0, 1, ?, n _1]. Similarly, the matrix B is a linear ramp, and all of its columns are the same vector. The matrix C is constant. The wavelet transform of these matrices is extremely sparse, and the efficiency of our wavelet warping algorithm stems from this sparse representation [9]. In the cylindrical-to-planar case the matrices are slightly more complicated:

7 IMPLEMENTAION OF WAVELET TRANSFORM

There are two requirements that a suitable wavelet transform should satisfy: (i) the transforms T(A), T(B), T(C), and T(D) should be sparse; (ii) the reconstructions (inverse wavelet transforms) should be fast to compute. Based on the experiments reported before, we chose a slightly modified version of the second-order interpolating wavelet transform, I(2, 2). The modification consists in omitting the update phase of the lifting scheme. The resulting transform requires 83 n2 operations to decompose an n _n matrix using the 2D nonstandard wavelet transform. The wavelet coefficients of this transform measure the extent to which the original signal fails to be linear. In the case of a planar warp, the matrices A and B are simply linear ramps and matrix C is constant (eq. 7)). Consequently, the transforms T(A) and T(B) consist of two non-zero coefficients each, and T(C) consists of a single non-zero coefficient. Note that this is lossless compression of the three matrices, they can be reconstructed exactly from these sparse transforms. In the case of a cylindrical warp (eq. (**??**)) the transforms T(A) and T(B) have fewer than 19 n2 nonzero coefficients each, while T(C) has two non-zero coefficients. In the case of a spherical warp (eq. (**??**)) the transforms T(A), T(B) and T(C) have fewer than 19 n2 non-zero coefficients each. Once again, the compression of the matrices is lossless. As for the disparities matrix D, the number of non-zero coefficients depends, of course, on the reference image. In our experiments, roughly one third of T(D) coefficients were non-zero. Although the number of non-zero coefficients can be decreased further by lossy wavelet compression, it is not beneficial to do so. As we shall see in the next section, the computational bottleneck of wavelet warping lies in the reconstruction stage. A slight reduction in the number of coefficients does not significantly improve performance, while a more drastic truncation causes errors in the mapping, resulting in visible artifacts.

V.

8 EMPIRICAL RESULTS

We have implemented our wavelet warping algorithm, as well as the standard warps: incremental planar-to-planar, LUT-based cylindrical-to-planar and spherical-to-planar, with the optimizations mentioned earlier. The algorithms were implemented in Java. All of the results reported in this paper were measured on a 3.0 GHz Pentium Dual Core processor. In all our comparisons we measured the entire warping time at full resolution, including reconstruction, clipping, and the divisions by the homogeneous coordinate. The averaged performance of the different warping algorithms (in frames per second) is summarized in Table **??** As predicted by our analysis, we found wavelet warping to be roughly as fast as the standard algorithm in the planar case and slightly faster (up to 25 percent) in the cylindrical case. Note that in the planar case the reference image has twice as many pixels as in the cylindrical case. This is the reason that the number of warps per second in the first row of the table is smaller almost by a factor of two. As expected, in the spherical case, wavelet warping outperforms the standard algorithm by a factor of roughly 1.8.

9 VI.

10 CONCLUSIONS

We have presented a simple way of computing 3D image warping in the wavelet domain. We have demonstrated both analytically and experimentally that performing these operations in the wavelet domain is in many cases faster than their direct computation. Furthermore, wavelet domain operations enable progressive and multi-resolution computations, as well as space and frequency locality. We have demonstrated our approach both on still images and on image sequences. To extend and improve our approach, we would develop an adaptive multiresolution scheme, which would allow operating upon different regions of an image at different resolutions. [1] [2] [3] [4]

1

Figure 1: Fig. 1 :

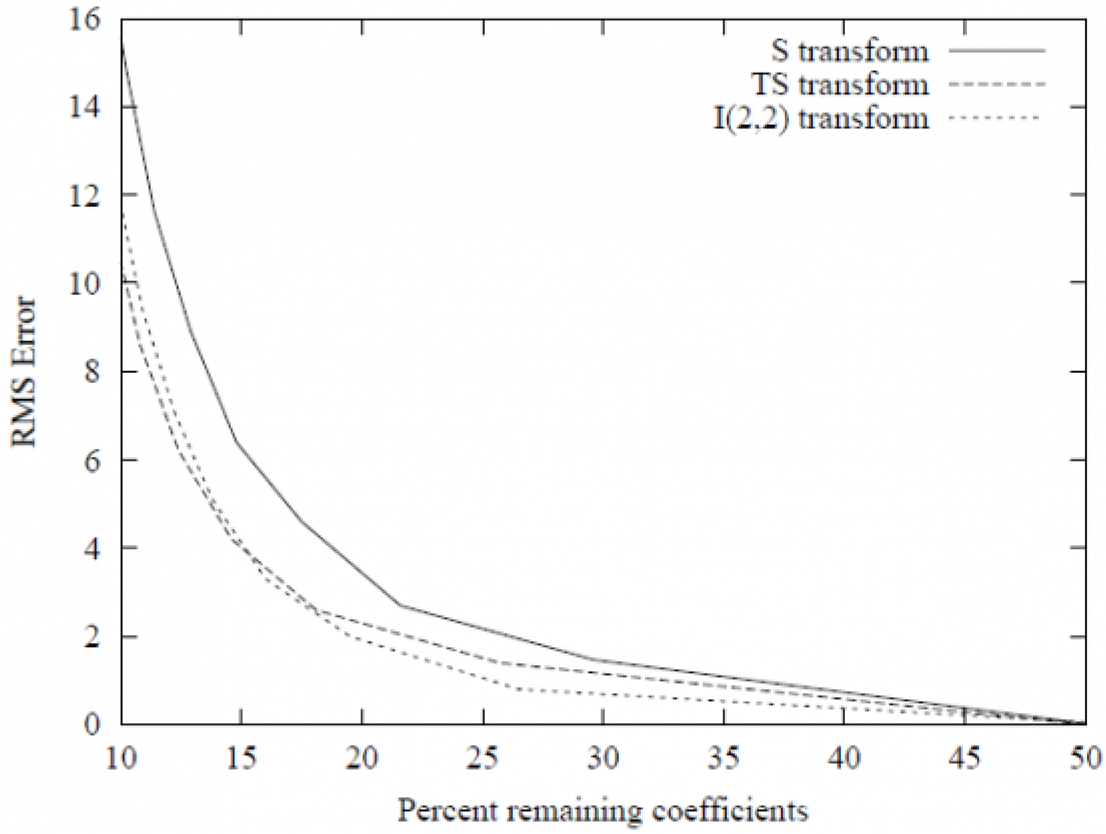

Figure 2:

$$\mathbf{U} = \frac{\mathbf{F}_1}{\mathbf{F}_3} \qquad \mathbf{V} = \frac{\mathbf{F}_2}{\mathbf{F}_3},$$

2

Figure 3: Fig. 2 :

Type of warp (number of pixels warped)	Standard warp	Wavelet warp
Planar (512 x 512)	6.5	7
Cylindrical (512 x 256)	12	15
Spherical (512 x 256)	7.7	14

Figure 4: .

[Shenchang ()] 'QuickTime VR -an imagebased approach to virtual environment navigation'. E C Shenchang . *Computer Graphics Proceedings*, (Annual) 2005.