SPARSE REPRESENTATION OF DENSE MOTION VECTOR FIELDS FOR LOSSLESS COMPRESSION OF 4-D MEDICAL CT DATA

Andreas Weinlich[1,2], Peter Amon[2], Andreas Hutter[2], and André Kaup[1]

[1]Chair of Multimedia Communications and Signal Processing, University of Erlangen-Nuremberg, Erlangen, Germany
{weinlich, kaup}@lnt.de
[2]Siemens Corporate Technology, Imaging and Visualization, Munich, Germany
{p.amon, andreas.hutter}@siemens.com

ABSTRACT

We present a new method for data-adaptive compression of dense vector fields in dynamic medical volume data. Conventional block-based motion compensation used for temporal prediction in video compression cannot conveniently cope with deformable motion typically found in medical image sequences encoded over time. Based on an approximation of physiologic tissue motion between two succeeding slices in time direction computed by optical flow methods, we find the most significant motion vectors with respect to their prediction capability for a second 2-D slice out of the first one. By coding the components of these vectors, we are able to reconstruct a high quality dense motion vector field at the decoder using only minimal side-information. We show that our approach can achieve a smoother prediction than block-based motion compensation for such data, reducing storage demands in spatially predictive lossless compression. We also show that such a predictive approach can yield better compression ratios than JPEG 2000 intra coding.

1. INTRODUCTION

Currently, in clinical environments diagnostic medical image data is usually stored in an uncompressed or lossless manner due to physicians demands and legal restrictions. A well established container format for such medical data is defined in the DICOM standard [3]. Other than uncompressed RAW images, a DICOM dataset can also contain images in lossless compression formats like TIFF or JPEG 2000. However, such 2-D image compression methods were not intended to make use of temporal correlations in dynamic datasets like 3-D + t reconstructions of the beating heart in cardiac computed tomography (CT, see Fig. 2) or magnetic resonance imaging. The time direction of a 3-D + t volume data set can also be considered as fourth dimension (Fig. 1). Thus, various other compression methods have been analyzed in the literature in order to exploit spatial and temporal redundancies, e. g., [6, 7]. Compared with redundancy reduction in depth (z-) direction like in JPEG 2000 3-D, the redundancy reduction over time allows to watch the motion at one axial slice without loading all other volume slices, even if for random access typically the whole uncompressed data set is stored in working memory anyway. A common approach is to use techniques from video coding, which involve block-based motion estimation / compensation (Block Matching, BM) for the precise prediction of consecutive slices in time direction. Yet, conventional BM assumes mostly translational motion like in real life movies, which is not always suitable for medical image and volume data. In dynamic medical data we observe mostly deformations of contiguous tissue caused by muscle contractions like heart beats, breathing, or swallowing. From a physiological point of view, the represented tissue is contiguous, so it is reasonable to assume similar motion in local neighborhoods. In order to account for the smoothness of this motion, we estimate a distinct motion vector for each voxel of a 2-D slice with respect to the previous slice in time direction (see Fig. 1), i. e., we obtain a dense motion vector field. To this end, well-known techniques from optical flow computations can be applied [2]. In structured regions, this leads to better approximations of tissue movements and thus better predictions. Furthermore, such a smooth motion approach has other advantages, e. g., it is not restricted to block based compression schemes, but also can be better combined with spatial prediction like in JPEG-LS or wavelet compression in spatial and temporal directions. Since in homogeneous regions the impact of inaccurate motion vectors on the residual error after motion compensation is rather low, we do not have to store these vectors for a good prediction. It is sufficient to transmit only reliable motion vectors, while interpolating the remaining ones at the decoder.

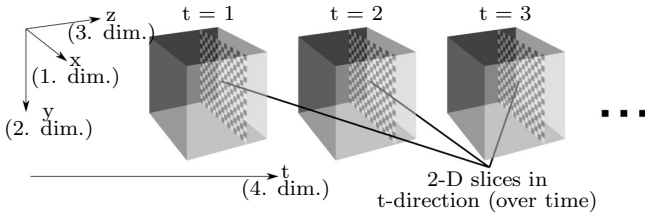

Fig. 1: Four dimensions (x, y, z, t) of a 4-D data set. Checkerboards show the slice sequence over time between which the motion is estimated.

Besides other motion concepts like mesh-based coding [8], there have been also attempts to encode a dense motion vector field for consumer video, most notably by Han et. al. [4] and in some previous work referenced by him. The drawback of dense motion vector field approaches for ordinary video is that they cannot naturally deal with occlusion, so the advantages compared to an elaborate BM scheme like in H.264/AVC [5] are rather small. If rotational, deformable, or zooming motion ever occurs in such video, this motion is mostly slow with respect to the frame rate, so the residual error after BM is rather moderate. Another advantage of BM is that it may predict image distortions like correlated noise or reconstruction artifacts, e. g., low frequency intensity gradients up to a certain level, in that it searches only for smallest differences between blocks. This is why dense motion vector approaches usually perform only well in lossy compression schemes with high compression ratios.

2. MOTION VECTOR AND CONFIDENCE ESTIMATION

For high efficient coding of CT data it is crucial to determine an accurate prediction for each voxel intensity in order to minimize the variance and thus the entropy of the residual error. In our approach, this goal is accomplished using an individual prediction for each voxel, which is obtained from

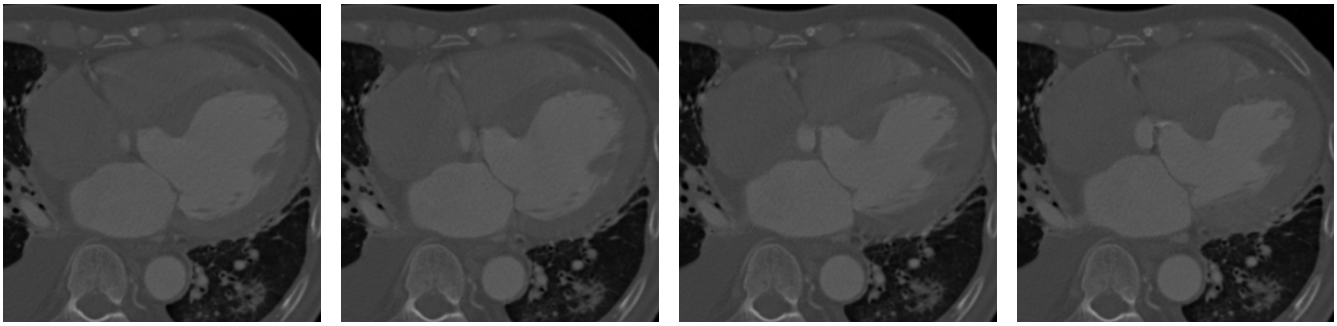

Fig. 2: Slices over time from a dynamic cardiac CT dataset. The two center ones are used in the following illustrations.[3]

the axial 2-D slice at the same z-position in the previous time step. For the estimation of a dense motion vector field, we employ a correlation (or matching) based hierarchical optical flow method similar to the one in [1], but somewhat simplified for our purpose. Compared to differential optical flow methods, it minimizes the residual error between the frames and can inherently handle large displacements, so it is much better suited for this application.

The algorithm seeks to minimize a weighted linear combination of the sum of squared intensity differences (SSD) of a neighborhood around a moving voxel and the disparity of adjacent motion vectors. First, both previous and current frame are repeatedly scaled down in image size by a factor of two until we reach a size where the magnitude of movements amounts only one voxel. After motion estimation, the vector field is hierarchically scaled up by a factor of two in resolution as well as in vector length by using bilinear interpolation until the original image size of the images has been reached. In each stage the estimation gets refined by an iterative algorithm that compares a 5×5 neighborhood N of each position $\mathbf{x}$ in the current frame i_{u} with candidate positions $\mathbf{v}$ in the previous frame $i_{\mathrm{u\text{-}1}}$. The candidate positions are within a search range of one voxel (eight-neighborhood) around the position where the previous vector estimation $\mathbf{v}_{t-1}$ is pointing to:

$$\mathbf{v}_t(\mathbf{x}) = \underset{\mathbf{v} \in N_{3\times3}(\mathbf{v}_{t-1}(\mathbf{x}))}{\operatorname{argmin}} \quad \sum_{\mathbf{r}\in N_{5\times5}(\mathbf{x})} \left(i_{\mathrm{u}}(\mathbf{r}) - i_{\mathrm{u\text{-}1}}(\mathbf{r}+\mathbf{v})\right)^2 + \lambda \left| \mathbf{v} - \tfrac{1}{8} \sum_{\mathbf{l}\in N_{3\times3}(\mathbf{x})\setminus\{\mathbf{x}\}} \mathbf{v}_{t-1}(\mathbf{l}) \right|$$

The first term minimizes intensity differences, whereas the second regularizing term minimizes the differences between the vector and its eight neighbors. The weighting parameter λ has to be chosen heuristically, but turned out to be not very critical as long as the SSD is weighted high enough. Around 5 to 10 iterations in each stage are usually enough to gain a good approximation of the motion. Our (single thread) C implementation is able to estimate a 512×512 voxel vector field with 10 iterations per stage in less than two seconds on a current 2.8 GHz CPU. A vector field estimation computed using this algorithm is shown in Fig. 3 (left).

After the vector field has been estimated, a confidence measure has to be determined for each of the vectors. Again we applied a slightly modified version compared to [1]. As if we would perform another iteration of motion refinement in the largest resolution stage, once again we determine the SSD values in a 3×3 search range around the estimated vectors. Out of the curvature of the resulting 3×3 SSD matrix S we are now able to compute two confidence values c_x in x- and c_y in y-direction. For this purpose we calculate

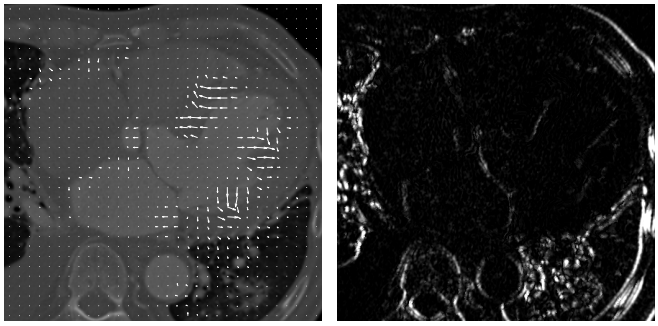

Fig. 3: Left: Current frame (to be predicted) with dense motion vector field, down sampled for illustration. Right: Confidence values for x-components of motion vectors, bright regions show high reliability.

the second derivative of the matrix in both directions:

$$c_x = \mathbf{w}^T S \mathbf{d}, \quad c_y = \mathbf{d}^T S \mathbf{w}, \quad \text{with } \mathbf{d} = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}, \ \mathbf{w} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

For example, if a pixel is located in a homogeneous region, all entries of S are similar, so the confidences of both estimated vector components will be low. If it is located at a vertical intensity border, the SSD increases when changing the search position in x-direction, so S has higher values in its left and right columns and thus c_x is large.

3. SPARSE VECTOR FIELD REPRESENTATION AND RECONSTRUCTION

In our algorithm we store motion vector components only at important positions (feature points). Due to the contiguous nature of tissue, other vectors can be extrapolated at the motion decoder without introducing a huge residual error. A block diagram of our vector field encoding algorithm is depicted in Fig. 4. After an initial feature point detection (step 1), the feature culling reduces this initial set of feature points with respect to the introduced error (step 2). Using a vector field reconstruction algorithm, an approximation of the original vector field can be restored out of the remaining features (step 3). In the remainder of this section these three steps will be described in more detail.

In step one, initial feature positions, at which either the x or the y component of a motion vector will be stored, are selected independently, applying a local maxima detection

[3] The dynamic cardiac CT data was kindly provided by Siemens Healthcare.

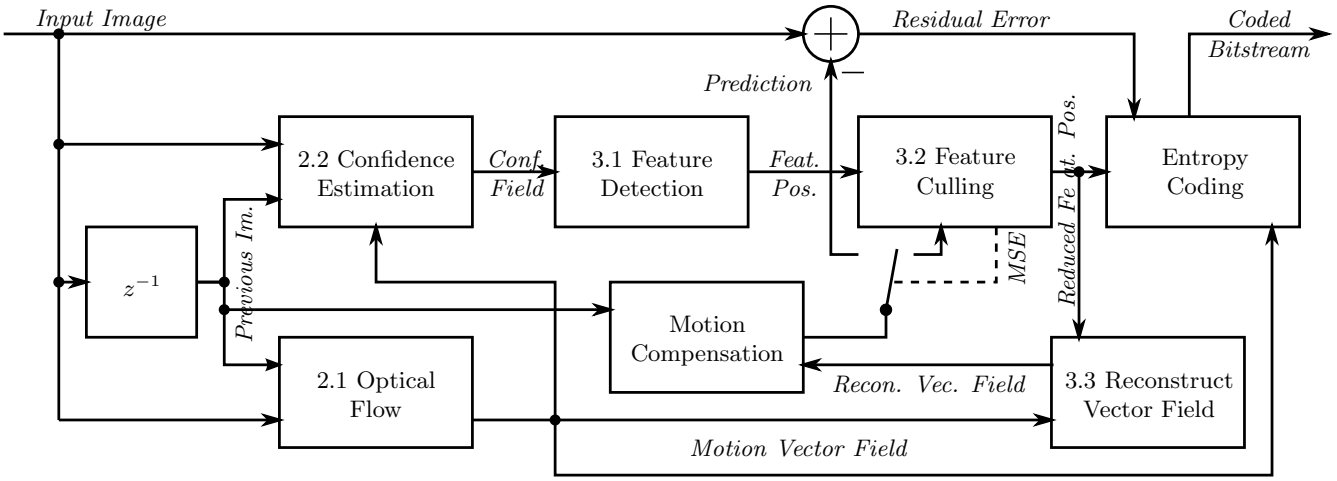

Fig. 4: Block diagram of the encoder (numbers indicate the sections with descriptions for the blocks).

algorithm to the confidence function (Fig. 3, right). In this context we define a feature as a triple $f = (m, n, c)$ with (m, n) being the *position* of an important motion vector and c being the important *component* of this vector. A voxel position is defined to be a local maximum, if it has the highest confidence value in a local neighborhood. The size of this neighborhood specifies the minimum distance of local maxima and thus the total number of initially selected feature points. Its choice represents a trade-off between complexity of the subsequent feature culling and potential loss of important vector information. Our experiments revealed, that neighborhoods in between 3×3 and 5×5 are much suited. Furthermore, in order to reduce the detection of maxima in noisy regions, only maxima with magnitudes above a certain threshold are accepted as feature point candidates. The particular value depends on the intensity range and on the present image noise. A choice of about 5% – 10% of the maximum confidence value yields good results for our dataset.

The feature culling in step two reduces the number of features. To do so, it checks for different sets of feature points, how the mean square error (MSE) of the prediction using the reconstructed vector field would change, if these feature points would be dropped ("culled"). The set of feature points introducing the lowest penalty is culled. This step incorporates a vector field reconstruction as described later, a motion compensation and a MSE calculation for each set. Note that an optimal culling would require to check every possible combination of feature points for a given number of desired features n, that is $\binom{N}{n}$ checks, where N is the number of preselected features. In order to decrease the complexity, we suggest a sub-optimal greedy culling strategy. This strategy checks independently for each feature point how the MSE changes with its removal. The removal, which minimizes the MSE is finally realized. In this manner, we successively reduce the number of feature points, until reaching a desired number or a maximum MSE. With this, the number of checks reduces to $0.5 \cdot N \cdot (N + 1)$, which results in a complexity of $O(N^3 \cdot Pixels)$ (including reconstructions), so at the moment the run time of this method is significantly higher than BM.

After feature culling, we got the final feature positions. For each feature its position and the important component c of the motion vector at this position has to be transmitted, while the other vector component gets extrapolated from more reliable positions. A set of 1170 features can be seen in Fig. 5, left. In general, the order in which the features are transmitted, is of no interest. Therefore, a simple run-level encoding in raster-scan order can be applied to the features, i. e., for each feature the raster-scan distance to the previous one is stored instead of its absolute position. Both the distances and the vector components can then be encoded using an arithmetic coder for example. Notice, that methods like differential coding of proximate features do not essentially reduce data rate. During feature culling, closely seated features with similar components have been reduced to a minimum already, so there is not much inter-feature redundancy left. Furthermore, when storing voxel accurate positions as we do, the least significant bits contain the bulk of information and so short distances not necessarily account for few position data. According to our experiments, even the shortest "travelling-salesman" route through the feature positions, which minimizes the distances between them, has no advantage.

In order to obtain a good approximation to the original vector field out of the available sparse motion information, there are some prerequisites on the reconstruction process in step three:

- At feature positions the available vector component must exactly be reproduced due to its high confidence.
- Vectors in the vicinity of feature positions should have similar vector components according to the connected tissue, while the influence of more distant motion information should be very low.
- Far away from feature positions the vectors should be short, as the nature of tissue attenuates local motion, so there is no global motion.
- Long motion vectors at feature positions should affect larger regions than short vectors, also by reason of connected tissue.

Still treating both vector components independently, we use a weighted nonlinear superposition of 2-D Gaussian functions for the extrapolation of the feature vector components c_k to get the reconstructed motion vector field $\mathbf{g} = (g_x(m, n), g_y(m, n))$:

$$g_x(m,n) = \left(\sum_{k=1}^{K_x} d_k^{-4}\right)^{-1} \cdot \sum_{k=1}^{K_x} \frac{c_{k,x}}{d_k^4} \exp\left(-\frac{d_k^2}{(\sigma c_{k,x})^2}\right),$$
$$d_k^2 = (m - m_{k,x})^2 + (n - n_{k,x})^2$$

Gaussian functions are suited for this problem, as they take high values near their maximum but have a fast and smooth decay. For each of the K_x features a Gaussian function is added with its maximum at the feature position $(m_{k,x}, n_{k,x})$,

the width proportional and the height equal to the vector component $c_{k,x}$. In order to preserve the property of exact interpolation at the center of each Gaussian and further reduce the impact on distant vectors if there are closer features, a d^{-4} weighting function is used for each Gaussian, where d is the distance to its center (the feature position). Finally, the vector components are normalized by the sum of all weighting functions. The second components of the vector field g_y are computed the same way. Note that the parameter σ should depend on the rigidity of the tissue, even if it can be chosen in a wide range (up to infinity) without much impact on the reconstruction result.

For a lossless image data compression, the reconstructed vector field can be used to obtain a prediction of the current slice: Each voxel gets predicted using the corresponding intensity value from the previous slice according to the motion vectors. After subtraction from the real data, only this residual error has to be transmitted. If only full-pel accuracy is used for the motion compensation, some difficultly codeable high-frequency noise might be introduced to the prediction and consequently to the residual error, especially in regions of high contrast. However, this can be solved with a simple oversampling of the vector field by a factor of two. In Fig. 5 on the right a vector field reconstruction with its corresponding compensated prediction can be seen.

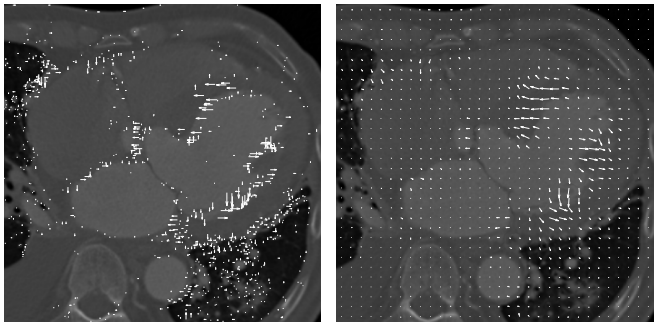

Fig. 5: Left: Previous frame with white arrows (features) pointing to best fitting positions. Only these feature vector components are transmitted. Right: Predicted frame using the dense vector field reconstruction out of the features on the left (white arrows, down sampled for illustration).

4. RESULTS

For our experiments we used a 3-D + t scan of a beating heart [6]. Our tests were conducted using the ten axial slices over time of one heart beat at a fixed position in z-direction (Fig. 2). They represent the change of a heart muscle in time, even if during the reconstruction several heart beats have been compensated for respiratory motion and then merged to one single beat. The individual slices have a resolution of 512×512 voxels and an intensity range of eight bits per voxel. Dynamic CT datasets in general contain deformable motion as well as a high amount of image noise due to limited X-ray exposure. Another characteristic is the low temporal sampling (frame) rate. With the maxima detection algorithm, 5540 features have initially been computed between the two frames in the middle of Fig. 2 (using 12-point neighborhoods and 2% of the maximum confidence as a minimum). At first, the feature culling reduces the residual error, as some of these features actually degrade the prediction capability of the reconstructed vector field but after several culls, the error increases. In terms of lossless compression, the culling should continue, until the combined motion and residual error information reaches its minimum. However, this residual error information depends on the applied encoding method.

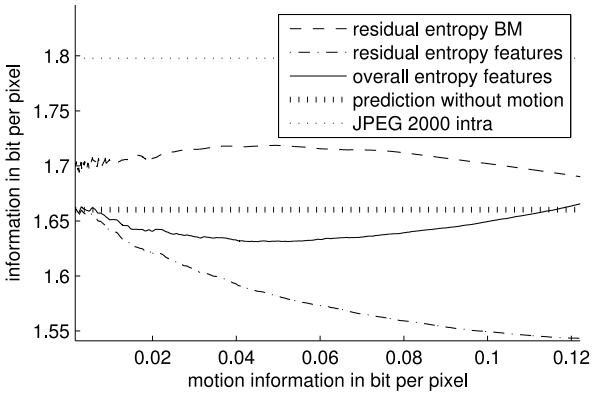

Fig. 6: Residual error information over vector information per pixel with BM (dashed line) and feature based approach (dash-dot line). The solid line shows the combined vector and residual information for the feature based approach. For reference: Information with prediction using zero motion vectors (wide dotted line) and JPEG 2000 intra coding (upper dotted line). Curves are averaged over ten slices.

As an objective measure of information, the entropy of an element is used, where elements are residual values, vector components or feature distances. With an arithmetic coder we could confirm our computed entropies.

One major advantage of our approach over BM consists in the smooth motion vector field and in consequence the absence of blocking artifacts. Therefore the subsequent exploitation of spatial redundancies is not restricted to pixels inside a block like in block-based discrete cosine transform schemes. Applicable methods for encoding the residual error encompass wavelet encoding like in JPEG 2000 or pixel-based prediction from a causal context like in the LOCO-I algorithm. In order to show that it is not convenient to combine BM with such approaches, we tested a simple additional voxel-based spatial prediction for residual error coding, where the context consists of the left and upper neighbors. Our BM implementation uses square blocks to searches for the best matching position in the previous frame within a search range of 24 voxels in terms of the SSD. It uses all possible integer block sizes in order to scale motion information.

Fig. 6 shows how the residual entropy after temporal and spatial prediction changes with growing vector information for both BM (dashed line) and our scheme (dash-dot line). The solid line shows combined motion and residual information with our scheme. Because of blocking artifacts, BM is worse than directly using the previous frame as predictor (thick dotted line), whereas in our method the residual information decreases when adding motion information. The minimum overall information (including motion) for this residual error coding method in our approach is reached at around 1–2 kilobyte of motion information per slice, depending on the motion (see solid line). Only for high amounts of motion information (small block sizes), BM can achieve better predictions as to its prediction of correlated noise structures but the overall information is remarkably larger then. As a reference, also the data rate of JPEG 2000 is shown (dotted line) when coding the slices individually. Since it cannot use information in previous time slices, its compression ratio is indeed worse than with the predictive schemes.

For the example in Fig. 3 the minimum was reached with the 1170 motion vector components in Fig. 5 (left). The according vector field reconstruction ($\sigma = 10$ voxels) as well

as the slice prediction are shown on the right. The residual error image of both methods (without spatial prediction) with equal motion information of 1128 bytes (in BM: block size 16×16) can bee seen in Fig. 7. We can notice here, that our approach does not produce blocking artifacts and leads to a lower residual error in regions with high intensity variations and non-rigid motion (upper left and lower right). On the other hand it cannot compensate correlated noise patterns and the low-frequency intensity gradients above the center.

In order to find out, whether the approach is able to compete with BM in terms of this residual error, we also compared the residual error of both methods. Fig. 8 shows that with few and medium motion information the results of our method are comparable to BM. For higher motion information the ability of BM to compensate for correlated noise patterns and moving low-frequency intensity gradients (from reconstruction) outperforms the feature-based method. Moreover, the performance of the feature-based method for a high amount of motion information is limited by the initial motion estimation.

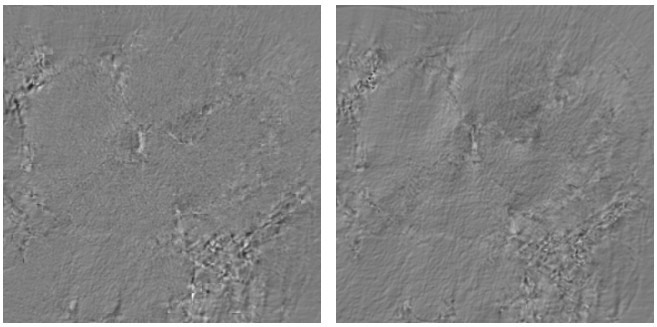

Fig. 7: Quadrupled residual error with BM (left, block size 16×16) and with our vector field reconstruction (right, 608 features).

5. CONCLUSION

We presented a new method for the compact representation of smooth motion vector fields for coding 4-D medical images. In a two-step approach the most important vectors are detected using a confidence function and then transmitted. An adapted reconstruction algorithm for the motion of deformable tissue can restore the dense vector field, resulting in an accurate prediction.

Both block matching and smooth vector field prediction in time direction can lead to better compressions than JPEG 2000 coding of individual slices. However, it was shown that even in lossless compression of noisy data an adapted approach for deformable tissue images is better suited for advanced residual error coding schemes than simple block matching. While the MSE of the predictions are comparable, a smooth motion vector field can be particularly useful when using wavelet or pixel context predictive coding.

In order to further improve our algorithm, we plan to apply a noise de-correlation between frames and extend the approach to three dimensions in order to incorporate motion in z-direction. For a comparison with other 3-D state of the art compression methods like JPEG 2000 3-D or H.264/AVC, we plan to evaluate other volumes and larger data sets. During our future research we will also look for an improved motion estimation scheme which can provide better vectors for feature culling.

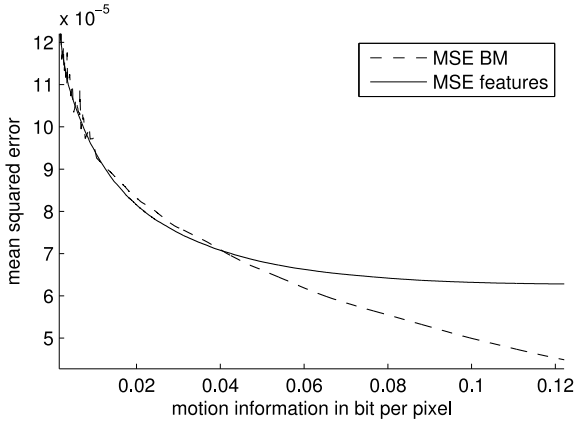

Fig. 8: MSE over vector information per pixel with BM (dashed line) and feature based approach (solid line). Intensity range of the images was normalized to one.

6. ACKNOWLEDGEMENT

This work was achieved with the help of the European Community's Seventh Framework Program through grant agreement ICT OPTIMIX n○ INFSO-ICT-214625.

REFERENCES

[1] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2(3):283–310, Jan. 1989.

[2] S. S. Beauchemin and J. L. Barron. The computation of optical flow. *ACM Computing Surveys*, 27(3):433–466, Sept. 1995.

[3] R. N. J. Graham, R. W. Perriss, and A. F. Scarsbrook. DICOM demystified: A review of digital file formats and their use in radiological practice. *Clinical radiology*, 60(11):1133–40, Nov. 2005.

[4] S.-C. Han and C. I. Podilchuk. Video compression with dense motion fields. *IEEE Transactions on Image Processing*, 10(11):1605–12, Jan. 2001.

[5] ITU and ISO. Advanced video coding for generic audiovisual services. *ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 AVC)*, 2010.

[6] U.-E. Martin and A. Kaup. Analysis of spatio-temporal prediction methods in 4D volumetric medical image datasets. In *Proc. IEEE International Conference on Multimedia and Expo*, pp. 525–528, Apr. 2008.

[7] P. Schelkens, A. Munteanu, J. Barbarien, M. Galca, X. Giro-Nieto, and J. Cornelis. Wavelet coding of volumetric medical datasets. *IEEE Transactions on Medical Imaging*, 22(3):441–458, Mar. 2003.

[8] R. Srikanth and A. G. Ramakrishnan. Contextual encoding in uniform and adaptive mesh-based lossless compression of MR images. *IEEE Transactions on Medical Imaging*, 24(9):1199–206, Sept. 2005.