Lossless Compression of Brain Volumes

James Chen, Tiffany Jou, Jason Su
Stanford University

Abstract-- We implemented a lossless compression method tailored to brain volumes. By making use of the particular characteristics of these images, we sought to improve compression ratios compared to other general encoders. We exploit global and local symmetries of sub-bands after wavelet decomposition to achieve intra-band prediction. We also implement a shift-based inter-slice DPCM prediction scheme that exploits the correlation between slices of the volume [3]. After entropy-coding the final residuals of these stages using EBCOT, we compare our results against encoders including gzip, JPEG-LS, and JPEG-2000. In addition, the relative compression of brain volumes between normal subjects and multiple sclerosis patients was examined. Results showed that in general, our encoder performs on par with JPEG-LS and JPEG-2000 and compression ratios were lower in patients due to asymmetries in their brain structure.

I. INTRODUCTION

A. Motivation

Magnetic Resonance Imaging (MRI) has become an indispensable diagnostic tool in medicine. MRI scanners are in almost constant use throughout the day, continually producing high-resolution image volumes that demand significant disk space. Radiologists cannot afford the risk of overlooking crucial information by using lossy versions of these images, so every image is archived in its original form. The sheer amount of data compels the development of an effective, lossless compression scheme.

In this report, we specifically look at three-dimensional MRI brain volumes. Various symmetrical features of the brain can be exploited for compression. A compression algorithm tailored to exploit such symmetries should offer compression gains when used in conjunction with established lossless compression techniques.

B. Goals

Our goal is to develop an encoder tailored to MRI brain volumes. We hope to optimize the compression ratio of these image volumes by making use of brain symmetry and inter-slice prediction. We wish to achieve a higher compression ratio with these characteristics when evaluated against other encoders and various combinations of encoders such as gzip, JPEG-LS +gzip, and JPEG-2000+gzip.

C. Dataset

Our coder was evaluated with image data from a multiple sclerosis study with 26 normal subjects and 26 patients. Two types of MRI scans were available for each subject, a high resolution (256x256x124, 1mm³ isotropic voxels, 15.5MB) “MPRAGE” and a lower slice resolution (256x256x48, 1x1x3 mm³ voxels, 6MB) “FLAIR”.

Fig.1 Axial slices of MPRAGE with normal on left and patient on right. Enlarged asymmetric ventricles appear in the patient.
exploits the global and local symmetries prediction method using symmetry detection block. Then the resulting slices of sub-band wavelet transform decomposition is iteratively applied 4 times on the LL band. The advantages of our new intra-band prediction method are summarized in Table 1.

The resulting residuals from this stage are then encoded in the inter-slice DPCM prediction stage which takes advantage of the correlation between slices. Finally, the residuals from this stage are entropy-coded using EBCOT, and the parameters are encoded using a variable-length coder.

B. 2D Integer Wavelet Transform

A 2D wavelet transform is applied to each slice. An integer-to-integer transform is used with a lifting scheme and the bi-orthogonal Cohen-Daubechies-Feauveau 5/3 wavelets, as recommended by Sanchez et al. [3]. Better separation of energy could possibly be achieved with orthogonal wavelets, which should be desirable since sub-bands are treated independently. This is an aspect of the design that remains to be explored. The wavelet decomposition is iteratively applied 4 times on the LL sub-band to produce a total of 13 sub-bands.

Multiple sclerosis patients typically display asymmetries between the left and right halves of the brain, such as deformations in the structure of the brain and focal lesions. Since the coder is designed to exploit symmetry, we expected patients to be more difficult to compress.

II. PROPOSED METHOD

A. Block Diagram/Overview of System

The block diagram of our system is shown below [3]. First, we process the original 3D brain images by taking a 2D wavelet transform slice by slice through the image volume. Then the resulting slices of sub-bands are processed on a 16x16 block-by-block basis, where we implement an intra-band prediction method using symmetry detection. This method exploits the global and local symmetries within the sub-bands.

The coder is designed to exploit any local symmetries by employing small blocks of 16x16, where we implement an intra-band prediction parameters. We partition each sub-band into two areas of equal size, and the parameters are encoded using a variable-length coder.
C. Symmetry Coding

In this stage of our system, we wish to exploit the inherent symmetries of the brain. This symmetry can be observed in the figure below. [3]

![Symmetry in Sub-bands](image)

Fig. 5 Symmetry in Sub-bands

The main motivation for this stage is that the similarities between regions of the brain can be used to minimize energy in residuals. We wish to process each sub-band of the wavelet transform output on a 16x16 block-by-block basis. For each block, we find the most similar reference block within a symmetry transformation. In other words, we use the Mean Squared Error (MSE) criterion to find a symmetry transformation of a reference block that would result in the most similar prediction of the current block. These reference blocks are the set of blocks that have already been scanned and processed, i.e. the blocks in the causal neighborhood of the current block. The symmetry transformations are chosen from the eight in the table below. [3]

<table>
<thead>
<tr>
<th>Geometric operation</th>
<th>Sample input block</th>
<th>Output block</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Vertical flip</td>
<td>▲</td>
<td>▼</td>
</tr>
<tr>
<td>2. Horizontal flip</td>
<td>▶</td>
<td>◀</td>
</tr>
<tr>
<td>3. Diagonal flip</td>
<td>▼</td>
<td>▲</td>
</tr>
<tr>
<td>4. Left rotation(90°)</td>
<td>◀</td>
<td>▶</td>
</tr>
<tr>
<td>5. Right rotation(90°)</td>
<td>▶</td>
<td>◀</td>
</tr>
<tr>
<td>6. Left rotation(90°) +  vertical flip</td>
<td>▼</td>
<td>▲</td>
</tr>
<tr>
<td>7. Right rotation(90°) +  vertical flip</td>
<td>▶</td>
<td>◀</td>
</tr>
<tr>
<td>8. No operation</td>
<td>▼</td>
<td>▲</td>
</tr>
</tbody>
</table>

In this process, the choice of scan order is important. Essentially, we want the group of reference blocks, or processed previous blocks, to be as similar to the current block as possible. In order to most efficiently exploit symmetries of the images, we need to first find the global symmetry of the sub-band so that we can appropriately change the scanning order of blocks within a sub-band. This is shown in the figure below, with two types of symmetry, horizontal and vertical. [3] The numbers represent the order in which the blocks are scanned.

![Global Symmetry Scan Order](image)

Fig. 6 Global Symmetry Scan Order

By processing similar blocks one after the other, we are able to make better predictions and produce residuals with less energy. Therefore, after the integer wavelet transform from the previous stage, the global symmetries of each sub-band are determined [3].

MSE is used as the criterion to determine whether there is horizontal or vertical symmetry. We did not set a default case for no symmetry. In the case of “no symmetry”, the scan order of the subsequent stage would reduce to a simple, arbitrary raster-scan. This scan order would not be advantageous at all, considering the inherent symmetrical characteristics of brain images. Therefore, exploiting even just a slight symmetry in a sub-band would be better than not considering any symmetry at all.

After the scan order of a particular sub-band is determined, this information is passed on to the subsequent “local symmetry” detection as described above. The resulting parameters—reference block and corresponding symmetry transformation—and residuals of every sub-band are passed on for further encoding.

D. Inter-slice Prediction

Residuals from the symmetry coding are fed into the inter-slice prediction. This stage starts by first chopping up the volume of residuals into chunks of 16 slices. Within each chunk of slices, the sub-bands are processed in raster-scan order with 16x16x16 voxel blocks. There are five prediction modes: no shift, left, right, up, and down shift by 1. On a slice-by-slice basis in the block, each of these shift operations is performed on the current slice and then subtracted it from the previous slice, producing a new set of inter-slice residuals. This process is efficiently formulated as a series of 3D convolutions. Thus, prediction is performed on a block by trying out each of the five
different modes and choosing the one that produces the smallest residual MSE.

E. Entropy Coding

The encoder uses a partial implementation of the embedded block coding with optimized truncation (EBCOT) algorithm first proposed by Taubman [1], and implemented in JPEG-2000 [2]. We decided to forego optimized truncation and instead focus on the lossless aspect of the encoder.

The EBCOT algorithm encodes each 16x16 residual block independently and sequentially. The samples in each block are raster-scanned by bit-plane and an adaptive arithmetic coder using one of eighteen probability mass functions encodes each binary symbol. The probability mass functions are constantly updated as each new symbol is encoded. The 18 contexts are categorized into 4 primitives. The primitives are sign coding (SC), zero coding (ZC), run-length coding (RLC), and magnitude refinement (MR). The primitives and contexts are mostly determined by the significance and value of the sample and the sample’s neighbors in the most current bit-plane [1].

In JPEG-2000, the entropy coder takes wavelet coefficients as input. In our system, the entropy coder processes the residuals after symmetry coding and DPCM. Therefore, we expect different signal statistics. As such, Sanchez et al. proposed eight new contexts in the sign coding primitive [3]. Unfortunately, the new contexts performed worse with our dataset, so we present results using the original EBCOT coder, which worked very well and remained a robust entropy coder within our system.

The prediction parameters, which include reference block numbers, symmetry modes, and inter-slice prediction modes, are encoded separately from the residuals with an Exponential-Golomb coder of order 0. This works by using a run of zeros to indicate to the decoder how many digits to read in for the next value. For example, a 2 zeros would tell the decoder that the next value is a 3-digit binary number. This type of coder is a universal code, which means it is within a factor of the optimal code if the distribution of values is monotonically decreasing, which seems to more or less be the case with our parameters (see below).

III. ATTEMPTED IMPROVEMENTS

A. Image Domain vs. Transform Domain

Symmetry is more intuitive in the image domain, even though [3] implements it in the transform domain. In the image domain, a symmetry mode is almost always used (including no transform), whereas no prediction is used for 60% of the time in the transform domain. The median and range is displayed in the following plots.

![Distribution of Global Symmetry (Wavelet Domain)](Fig. 7)

![Distribution of Symmetry Modes (Wavelet Domain)](Fig. 8)
Similarly, for inter-slice prediction, the prediction modes suggested by Sanchez et al. also make sense in the image domain, but it was not clear to us if they would be effective in the wavelet transform domain since shifts there would correspond to something quite different in the image. After plotting the distribution of chosen modes, we observed that our suspicions were correct.

Inter-slice prediction is very rarely used when operating in the wavelet domain. It is used more often in the image domain, about 10% of the time.
With the promise that the image domain showed, we incorporated them into the full coder. However, the compression ratio decreased. This possibly indicates that the entropy coder is not equipped to handle the new residuals generated by a high amount of prediction. A possibility for future work is to understand this trend and make the prediction and entropy coding systems work well with each other.

C. Mode Choice Criterion

In our implementation, symmetry prediction stage and the inter-slice prediction stage both use MSE as the mode choice criterion. In addition, we had also explored the choice of using entropy as the criterion for the latter. For symmetry detection, we ended up not using entropy as a criterion, because training the residual data to acquire the corresponding PMFs would take a very long time. Also, since the residuals are cascaded down to inter-slice DPCM, entropy at this stage would not accurately estimate the bitrate after EBCOT coding. In addition, for symmetry detection, measuring the similarity between blocks is much more important than the compressing of the parameters. These compression ratio concerns, we decided, were more crucial in later stages. Therefore for the symmetry detection stage, we kept the criterion to be MSE and not entropy.

On the other hand, for inter-slice prediction we speculated that we should use entropy as the criterion for prediction. The final residuals of this predictor would be sent to the entropy encoder, so it made sense to try to minimize the entropy at this stage. However, this attempt resulted in worse compression ratios. We think that it may be due to the nature of the resultant residuals from the previous symmetry coding stage.

After the symmetry predictions, residuals are passed onto the inter-slice prediction stage. At this point, the range for the residuals would be small. Therefore, there would be less of an advantage to use entropy as the criterion for the current stage since EBCOT only encounters a smaller range of values that are more commonly used across all blocks, whereas entropy might allow some random large values that only occur once in one of the blocks. Therefore, since MSE still proved to be the better scheme, we reverted back to using MSE on both stages for our final implementation.

C. Entropy Coding

We attempted to improve the performance of the entropy coder by adjusting how often context probability mass functions are reset to 50/50. We tested three options: (1) all context PMFs are reset when encoding a new 16x16 block, (2) all context PMFs are reset after encoding a new sub-band within a slice, and (3) separate context PMFs are maintained for each sub-band frequency across all slices, i.e. there are more context PMFs in parallel, but they are never reset.

Resetting the PMFs imposes a penalty on the arithmetic coder as it adapts a uniform Bernoulli distribution to the signal’s statistics. However, if signal statistics (keeping in mind that these are residuals) vary significantly across sub-bands and slices, then resetting PMFs would decrease the cost of using the wrong probabilities.

Surprisingly, option (1) performed best in our tests, although it was only slightly better (3-4%) than the worst option, option (3). We opted to use option (1) in our final integrated system.

IV. Results

All 104 images in the dataset were compressed with our symmetry-based coder as well as general coders including gzip, JPEG-LS, and JPEG-2000. JPEG-LS and JPEG-2000 were applied for each slice in a brain volume then compressed with gzip to reduce the overhead from any header information stored in the slice images. The following box plots show the median and range of compression ratios with each of the compression methods.
Our symmetry-based coder performs on par with JPEG-LS and JPEG-2000. The generic gzip coder performs the worst because it is not specialized for images. Patients were a bit more difficult to compress than normals for all the coders. As expected, our coder has greater difficulty with the asymmetry in patient images. This is evident in the significant shortening of the upper range compared to the others.

For the normal FLAIR images, the symmetry-based coder performs slightly worse than JPEG-LS/2000 but still in
the same general range. There is, however, an impressive improvement for some particularly noisy patient images. The JPEG family struggles with these difficult images, doing even worse than gzip, but the symmetry-based coder handles them with aplomb. Again, patient images in general have lower compression ratios than normals.

V. DISCUSSION AND CONCLUSIONS

We briefly attempted using transformations of a standard brain as an initial stage of prediction. This proved to be too limited and rate expensive in application. MR images can have varying detail and tissue contrasts but the standard brain can only effectively be used to predict another brain image if it was scanned in exactly the same way. Also, a nonlinear transformation is required to make the standard brain closely resemble the image. This requires the costly transmission of a warp transformation image that would have to be floating point accurate. Perhaps these values could be quantized with acceptable prediction inaccuracy but the added challenge of dealing with patients that appear quite different from the standard brain, which is averaged from normals, deterred us from pursuing this concept much further.

We question the shift-based inter-slice DPCM prediction modes suggested by Sanchez since they do not seem appropriate in the wavelet domain. An alternative would be perhaps to treat a slice as a scaled version of the previous one. Additionally, we found it curious that the “right rotation + vertical flip” symmetry mode was never used. This is in fact, because it is exactly the same as a diagonal flip; it should be removed from Sanchez’s coder.

The approximate running time for our MATLAB code was about 1 hour for each high-resolution MPRAGE volume. This is quite substantial, especially compared to the few seconds it takes to encode with JPEG-LS or JPEG-2000. Some heuristics could be developed to make mode decision less computationally expensive. For example, the most likely modes for a given block could be trained from another dataset of brains, then the encoder would only have to find the best out of a limited subset of modes.

In general, we found patients to be more complex and difficult to compress than normals for all encoders. The symmetry-based coder we developed has greater difficulty achieving higher compression ratios in this context. However, since it is tailored for brains, it seems to be more robust to noise than general image coders for our dataset.

VI. FUTURE WORK

The top priority is to implement a decoder, which will verify the validity of our encoder. We would also like to test other global symmetry modes, including diagonal flips and rotations, and combinations of them.

We also believe the entropy coder can be tuned with new contexts to more effectively process the residuals. We do have to be cautious about over-optimizing with our dataset in case our system cannot effectively compress brain volumes acquired from other sources using other techniques and reconstruction methods.

After we are able to improve our lossless compression performance, we would like to implement optimized truncation in EBCOT. Optimized truncation is useful for viewing MRI files. The speed at which an image is transported across a network and displayed on a doctor’s screen is often network-bound. Optimized truncation will allow a distorted version of the image to be displayed quickly, then gradually enhanced to its lossless form as more data arrives.

REFERENCES


APPENDIX

A. Workload Breakdown

James Chen: EBCOT Entropy Coding, Attempted Improvements, Testing code, Presentation Slides, Report

Tiffany Jou: Symmetry Coding, Attempted Improvements, Presentation Slides, Report, Integrated report

Jason Su: 2D wavelet transform and inverse, Inter-slice DPCM and its decoder, Integrating and processing dataset, Presentation Slides, Report, Presentation