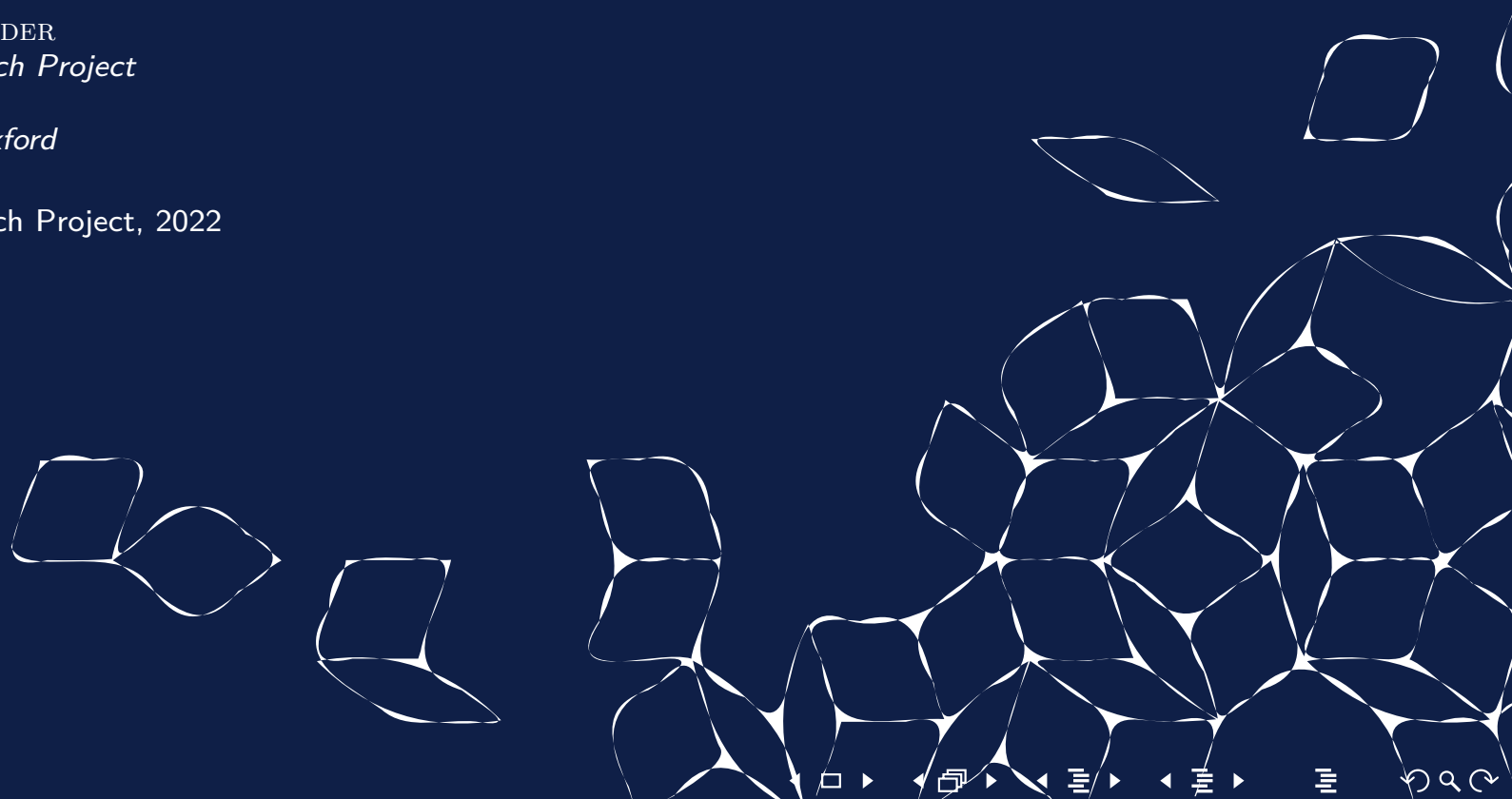
Fractal Image Compression Applied to 3D Grayscale MRI Images

T. Bretschneider
Summer Research Project
Balliol College
University of Oxford

Summer Research Project, 2022



Beginnings of Fractal Image Compression

Where does it come from.

- Barnsley Fern;

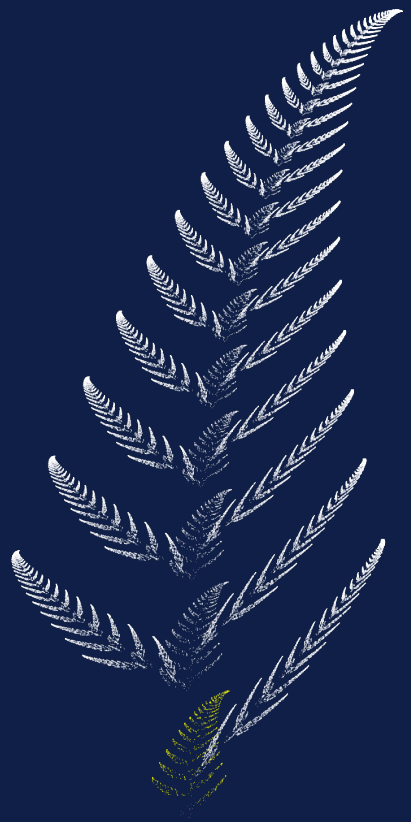


Figure: "Barnsely Fern" by Xerostomus CC 4.0 BY SA, recolorised

Beginnings of Fractal Image Compression

Where does it come from.

- IFS (Iterated Function System)[1];

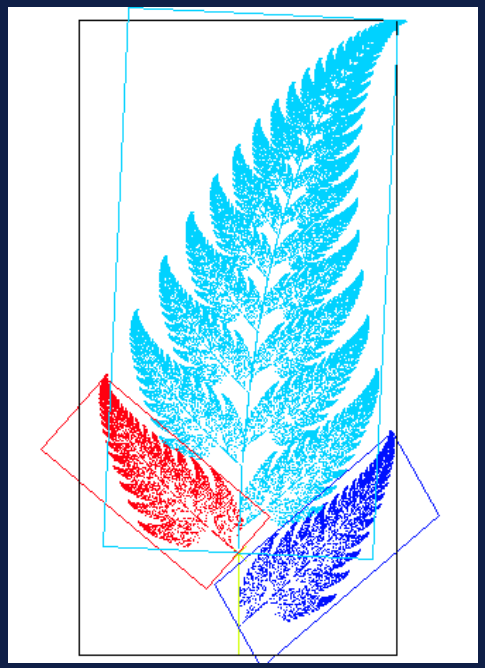


Figure: "Barnsely Fern Annotated" by António Miguel de Campos Public Domain

Beginnings of Fractal Image Compression

Where does it come from.

- MRCM (Multiple Reduction Copying Machine);

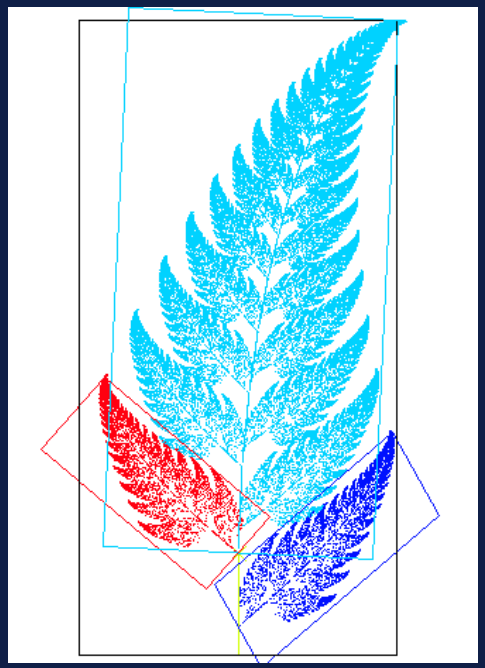


Figure: "Barnsely Fern Annotated" by António Miguel de Campos Public Domain

IFS

- An IFS [2] is a finite set of contraction mappings on a complete metric space.

$$\{f_i : X \to X \mid i = 1, 2, \ldots N\}, \quad N \in \mathbb{N} \tag{1}$$

IFS

- An IFS [2] is a finite set of contraction mappings on a complete metric space.

$$\{f_i : X \to X \mid i = 1, 2, \ldots N\}, \quad N \in \mathbb{N} \tag{1}$$

- Fixed-point theorem [3]

Which metric space

- Hausdorff metric [4],

$$d_H(X, Y) = \max\left\{\sup_{x\in X} d(x, Y), \sup_{y\in Y} d(X, y)\right\}, \qquad (2)$$

where $d(x, Y)$ and $d(X, y)$ are the distance from the points x, y to the subsets X, Y, given by $d(a, B) = \inf_{b\in B} d(a, b)$.

Which metric space

- Hausdorff metric [4],

$$d_H(X, Y) = \max\left\{\sup_{x\in X} d(x, Y), \sup_{y\in Y} d(X, y)\right\}, \qquad (2)$$

where $d(x, Y)$ and $d(X, y)$ are the distance from the points x, y to the subsets X, Y, given by $d(a, B) = \inf_{b\in B} d(a, b)$.

- Root mean square

$$d_{rms}(f, g) = \sqrt{\int_{I^2} (f(x, y) - g(x, y))^2 \mathrm{d}x\mathrm{d}y} \qquad (3)$$

Fixed-point theorem I

Theorem (Fixed-point theorem)

Let X be a complete metric space and $f : X \to X$ be a contraction. Then there exists a unique fixed point x_f st $\forall x \in X$ we have

$$\lim_{n \to \infty} f^n(x) = x_f = f(x_f) \tag{4}$$

Fixed-point theorem II

Proof
Take $x \in X$.
Now as X is complete we know that all cauchy sequences converge. So we plan to show that $f^n(x)$ is cauchy.
So take $n > m$ and consider with the metric d,

$$d(f^m(x), f^n(x)) < sd(f^{m-1}(x), f^{n-1}(x)) < s^m d(x, f^{n-m}(x)). \quad (5)$$

Fixed-point theorem III

Now by the triangle inequality,

$$
\begin{aligned}
d(x, f^n(x)) &\leq d(x, f^{n-1}(x)) + d(f^{n-1}(x), f^n(x)) && (6)\\
&\leq d(x, f(x)) + d(f(x), f(f(x))) + \cdots + d(f^{n-1}(x), f^n(x)) && (7)\\
&< d(x, f(x)) + \cdots + s^{n-1} d(x, f(x)) && (8)\\
&= d(x, f(x))(1 + s + \cdots + s^{n-1}) && (9)\\
&= \frac{1 - s^n}{1 - s} d(x, f(x)) && (10)\\
d(x, f^n(x)) &\leq \frac{1}{1 - s} d(x, f(x)) && (11)
\end{aligned}
$$

Fixed-point theorem IV

So we have that in 5

$$d(f^m(x), f^n(x)) < \frac{s^m}{1-s} d(x, f(x)). \tag{12}$$

so that as $f^n(x)$ is cauchy. So by the completeness there is a unqiue point $x_f \in X$ to which this sequence converges. Lastly due to the contractivity f is also continuous so
$f(x_f) = f(\lim_{n\to\infty} f^n(x)) = \lim_{n\to\infty} f^{n+1}(x) = x_f$.
□

PIFS

- The Partitioned IFS [5] can be seen as

$$F(W) = \bigcup_{i \in N} f_i(\mathbb{1}_{(D_i \times \mathbb{R})} W), \tag{13}$$

where our f_i come from,

$$\{f_i : (D_i \times \mathbb{R}) \to (R_i \times \mathbb{R}) \mid i = 1, 2, \ldots, N\}, \quad N \in \mathbb{N} \tag{14}$$

Choices for f_i

Now usually we let our f_i be affine transformations of $(D_i \times \mathbb{R})$, where the third dimension contains the image. Which is changed by a contrast multiple (s_i) and a brightness correction (σ_i).

$$f_i \begin{pmatrix} x \\ y \\ u(x,y) \end{pmatrix} = \begin{pmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{pmatrix} \begin{pmatrix} x \\ y \\ u(x,y) \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \\ \sigma_i \end{pmatrix} \tag{15}$$

provided certain conditions this will be contracting.

A simple encoding scheme

We quickly explain one of the simplest methods as in [6]. We shall discuss a simple 256x256 image with 256 brightness levels. The plan is to find some PIFS such that $F(u) \approx u$. Even if it is close we will see from the following theorem that F^n will end up being close to u.

Collage Theorem

Theorem (Collage Theorem)

Given the same hypothesis as the contraction mapping theorem, we also get,

$$d(x, x_f) \leq \frac{1}{1-s} d(x, f(x)) \tag{16}$$

Proof.
Taking the limit in eqn. 11. □

Encoding

- Split image into ranges

Encoding

- Split image into ranges
- Get possible domains

Encoding

- Split image into ranges
- Get possible domains
- Iterate over ranges finding best map and domain

Encoding

- Split image into ranges
- Get possible domains
- Iterate over ranges finding best map and domain
- Done!

Best domain and range

Let the range be denoted by a vector with the image values $\mathbf{r}_i$ and a downsampled domain similarly by $\mathbf{d}$, then we get that for this section,

$$d(\mathbf{r_i}, F(\mathbf{r_i}) = ||\mathbf{r_i} - s_i\mathbf{d} - \sigma_i\mathbf{e}||^2 \tag{17}$$

So minimising this is simply a standard least squares problem and we can simply obtain optimum values for s_i and σ_i.

Recap

The encoded image;

- PIFS

Recap

The encoded image;

- PIFS
- Fixed-point theorem

Recap

The encoded image;

- PIFS
- Fixed-point theorem

The encoding;

Recap

The encoded image;

- PIFS
- Fixed-point theorem

The encoding;

- Collage theorem

Improvements

Possible improvements are categorised as follows [7],

- How the image is partitioned into range blocks.
- Composition of the pool of domain blocks.
- Class of transformations applied to the domain blocks.
- The type of search used in locating domain blocks.

Partitionings

Changes to the partitioning of the ranges an be used for adaptive methods, such as quadtree [8], HV[9], or triangular partitions and irregular partitions as in [10] as first approached by [11], and independently in [12].

Domain Pool

Domain pool reduction [13].

Non-affine transformations

Allowing a quadratic factor [14].

Matching range and domain block

Biggest increase in speed can be found here.

Matching range and domain block

Biggest increase in speed can be found here.

- Classification of domains.

Matching range and domain block

Biggest increase in speed can be found here.

- Classification of domains.
- Nearest-neighbour methods.

Nearest Neighbour Methods

Let M denote a block intensity transform. Here a block of pixel values is represented as a vector $\mathbf{p}$ and then transformed as follows,

$$M(\mathbf{p}) = s\mathbf{p} + o\mathbf{e}. \tag{18}$$

Here s is a contrast term, o is a brightness shift and $\mathbf{e}$ is a unit length vector with equal components.

Nearest Neighbour Methods

Let M denote a block intensity transform. Here a block of pixel values is represented as a vector $\mathbf{p}$ and then transformed as follows,

$$M(\mathbf{p}) = s\mathbf{p} + o\mathbf{e}. \tag{18}$$

Here s is a contrast term, o is a brightness shift and $\mathbf{e}$ is a unit length vector with equal components. We get the following error squared $E(\mathbf{u}, \mathbf{p})$ in the map,

$$E(\mathbf{u}, \mathbf{p}) = \min_{s,o \in \mathbb{R}} ||\mathbf{u} - M(\mathbf{p})||^2 = \min_{s,o \in \mathbb{R}} ||\mathbf{u} - s\mathbf{p} - o\mathbf{e}||^2 \tag{19}$$

Now minimising this with respect to s, o is just a simple least squares problem however we can do better.

More Nearest Neighbours

Let us define ϕ to denote the normalised projection operator onto the orthogonal complement of $\mathbf{e}$, so,

$$\phi(\mathbf{x}) = \frac{\mathbf{x} - \langle \mathbf{x}, \mathbf{e} \rangle}{||\mathbf{x} - \langle \mathbf{x}, \mathbf{e} \rangle||}. \tag{20}$$

Hence using this we can write $s\mathbf{p} + o\mathbf{e} = \tilde{s}\phi(\mathbf{p}) + \tilde{o}\mathbf{e}$, where we have,

$$\tilde{s} = s\langle \mathbf{p}, \phi(\mathbf{p}) \rangle \in \mathbb{R} \tag{21}$$

$$\tilde{o} = o + s\langle \mathbf{p}, \mathbf{e} \rangle \in \mathbb{R}. \tag{22}$$

More Nearest Neighbours

Let us define ϕ to denote the normalised projection operator onto the orthogonal complement of $\mathbf{e}$, so,

$$\phi(\mathbf{x}) = \frac{\mathbf{x} - \langle \mathbf{x}, \mathbf{e} \rangle}{||\mathbf{x} - \langle \mathbf{x}, \mathbf{e} \rangle||}. \tag{20}$$

Hence using this we can write $s\mathbf{p} + o\mathbf{e} = \tilde{s}\phi(\mathbf{p}) + \tilde{o}\mathbf{e}$, where we have,

$$\tilde{s} = s\langle \mathbf{p}, \phi(\mathbf{p}) \rangle \in \mathbb{R} \tag{21}$$

$$\tilde{o} = o + s\langle \mathbf{p}, \mathbf{e} \rangle \in \mathbb{R}. \tag{22}$$

So we are trying to approximate $\mathbf{u}$ by $\tilde{s}\phi(\mathbf{p}) + \tilde{o}e$. Now we can project $\mathbf{u}$ onto the hyper-plane described by $\phi(\mathbf{p})$ and e, and we see that this will be the best $M(\mathbf{p})$ that we can find.

More Nearest Neighbour

Now $P(\mathbf{u})$ is the best approximation and is given by,

$$P(\mathbf{u}) = \langle \mathbf{u}, \phi(\mathbf{p}) \rangle \phi(\mathbf{p}) + \langle \mathbf{u}, \mathbf{e} \rangle \mathbf{e}. \tag{23}$$

So the error $E(\mathbf{u}, \mathbf{p})$ becomes,

$$\begin{aligned}
E(\mathbf{u}, \mathbf{p}) &= ||\mathbf{u} - P(\mathbf{u})||^2 && (24)\\
&= ||\mathbf{u} - \langle \mathbf{u}, \phi(\mathbf{p}) \rangle \phi(\mathbf{p}) - \langle \mathbf{u}, \mathbf{e} \rangle \mathbf{e}||^2 && (25)\\
&= ||\langle \mathbf{u}, \phi(\mathbf{u}) \rangle \phi(\mathbf{u}) - \langle \mathbf{u}, \phi(\mathbf{p}) \rangle \phi(\mathbf{p})||^2 && (26)\\
&= ||\langle \mathbf{u}, \phi(\mathbf{u}) \rangle \phi(\mathbf{u}) - \langle \mathbf{u}, \phi(\mathbf{u}) \langle \phi(\mathbf{u}), \phi(\mathbf{p}) \rangle \phi(\mathbf{p})||^2 && (27)\\
&= \langle \mathbf{u}, \phi(\mathbf{u}) \rangle^2 ||\phi(\mathbf{u}) - \langle \phi(\mathbf{u}), \phi(\mathbf{p}) \rangle \phi(\mathbf{p})||^2 && (28)\\
&= \langle \mathbf{u}, \phi(\mathbf{u}) \rangle^2 \left(1 - \langle \phi(\mathbf{u}), \phi(\mathbf{p}) \rangle^2\right) && (29)
\end{aligned}$$

Now we consider $1 - \langle \phi(\mathbf{u}), \phi(\mathbf{p}) \rangle^2$.

Theorem

As before if we have multiple possible $\mathbf{p}_i$, $i = 1, 2, \ldots, N$, *then we will be able to minimise* $E(\mathbf{u}, \mathbf{p}_i)$ *over* $1, 2, \ldots, N$ *by finding the* $\mathbf{p}_i$ *such that either* $d(\phi(\mathbf{u}), \phi(\mathbf{p}_i))$ *or* $d(\phi(\mathbf{u}), -\phi(\mathbf{p}_i))$ *is smallest.*

More Nearest Neighbor

Proof.

Let $D : \mathbb{R}^n \times \mathbb{R}^n \to [0, \sqrt{2}]$ defined as
$D(\mathbf{u}, \mathbf{p}) = \min\{d(\phi(\mathbf{u}), \phi(\mathbf{p})), d(\phi(\mathbf{u}), -\phi(\mathbf{p}))\}$.

More Nearest Neighbor

Proof.

Let $D : \mathbb{R}^n \times \mathbb{R}^n \to [0, \sqrt{2}]$ defined as
$D(\mathbf{u}, \mathbf{p}) = \min\{d(\phi(\mathbf{u}), \phi(\mathbf{p})), d(\phi(\mathbf{u}), -\phi(\mathbf{p}))\}$.
Now $d(\phi(\mathbf{u}), \pm\phi(\mathbf{p})) = \sqrt{2(1 \mp \langle \phi(\mathbf{u}), \phi(\mathbf{p}) \rangle)}$.

Proof.

Let $D : \mathbb{R}^n \times \mathbb{R}^n \to [0, \sqrt{2}]$ defined as $D(\mathbf{u}, \mathbf{p}) = \min\{d(\phi(\mathbf{u}), \phi(\mathbf{p})), d(\phi(\mathbf{u}), -\phi(\mathbf{p}))\}$.

Now $d(\phi(\mathbf{u}), \pm\phi(\mathbf{p})) = \sqrt{2(1 \mp \langle \phi(\mathbf{u}), \phi(\mathbf{p}) \rangle)}$. So we then get that $D(\mathbf{u}, \mathbf{p}) = \sqrt{2(1 - |\langle \phi(\mathbf{u}), \phi(\mathbf{p}) \rangle|)}$.

Proof.

Let $D : \mathbb{R}^n \times \mathbb{R}^n \to [0, \sqrt{2}]$ defined as $D(\mathbf{u}, \mathbf{p}) = \min\{d(\phi(\mathbf{u}), \phi(\mathbf{p})), d(\phi(\mathbf{u}), -\phi(\mathbf{p}))\}$.

Now $d(\phi(\mathbf{u}), \pm\phi(\mathbf{p})) = \sqrt{2(1 \mp \langle \phi(\mathbf{u}), \phi(\mathbf{p}) \rangle)}$. So we then get that $D(\mathbf{u}, \mathbf{p}) = \sqrt{2(1 - |\langle \phi(\mathbf{u}), \phi(\mathbf{p}) \rangle|)}$. Hence we have that,

$$\langle \phi(\mathbf{u}), \phi(\mathbf{p}) \rangle^2 = \left(1 - \frac{D^2}{2}\right)^2 \tag{30}$$

Proof.

Let $D : \mathbb{R}^n \times \mathbb{R}^n \to [0, \sqrt{2}]$ defined as $D(\mathbf{u}, \mathbf{p}) = \min\{d(\phi(\mathbf{u}), \phi(\mathbf{p})), d(\phi(\mathbf{u}), -\phi(\mathbf{p}))\}$.

Now $d(\phi(\mathbf{u}), \pm\phi(\mathbf{p})) = \sqrt{2(1 \mp \langle \phi(\mathbf{u}), \phi(\mathbf{p}) \rangle)}$. So we then get that $D(\mathbf{u}, \mathbf{p}) = \sqrt{2(1 - |\langle \phi(\mathbf{u}), \phi(\mathbf{p}) \rangle|)}$. Hence we have that,

$$\langle \phi(\mathbf{u}), \phi(\mathbf{p}) \rangle^2 = \left(1 - \frac{D^2}{2}\right)^2 \tag{30}$$

Hence we have that, $E(\mathbf{u}, \mathbf{p}_i) = \langle \mathbf{u}, \phi(\mathbf{u}) \rangle^2 g(D(\mathbf{u}, \mathbf{p}_i))$, where $g(D) = D^2(1 - \frac{D^2}{4})$.

Now g is monotonically increasing for $0 \leq D \leq \sqrt{2}$ so for $i = 1, 2, \ldots, N$, $E(\mathbf{u}, \mathbf{p}_i)$ is equivalently minimised when $D(\mathbf{u}, \mathbf{p}_i)$ is minimised.

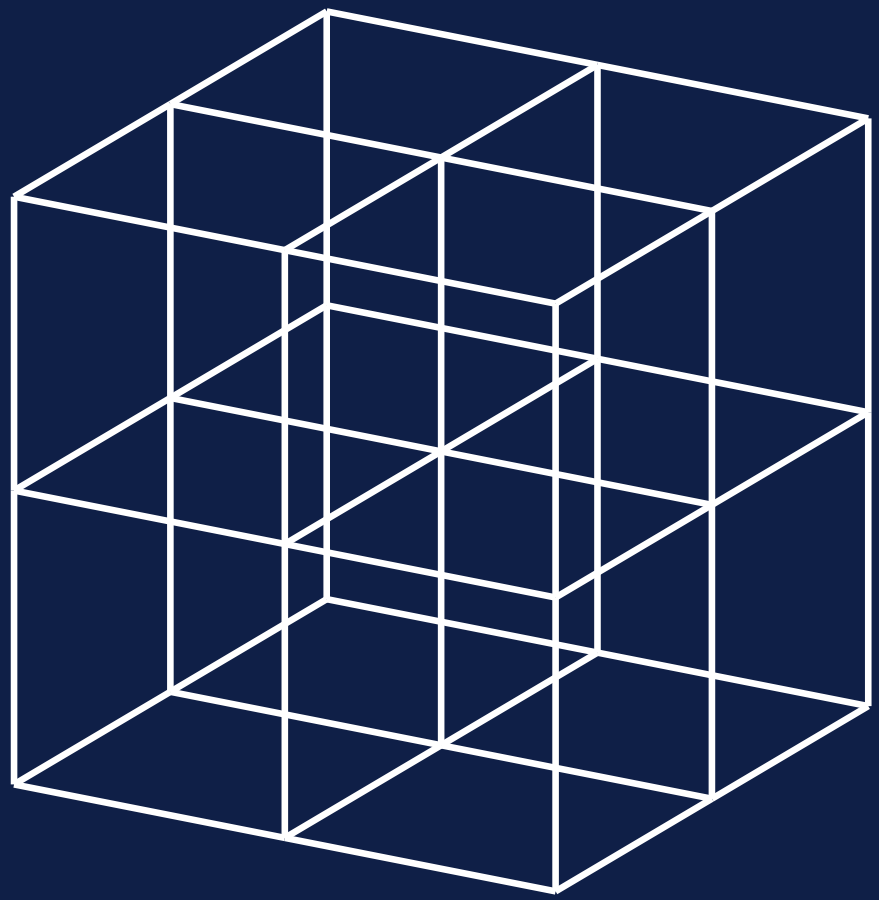
□

Final Nearest Neighbor

This has now become a nearest neighbor problem as we have to find the nearest neighbor in $\{\pm\phi(\mathbf{p}_i)|i \in 1, 2, \ldots, N\}$ to the point $\phi(\mathbf{u})$. This can be done very quickly, an algorithm for logarithmic time once various KDTrees have been set up is in [15]. We used the NearestNeighbors.jl [16] code to implement this in our Julia [17] code.

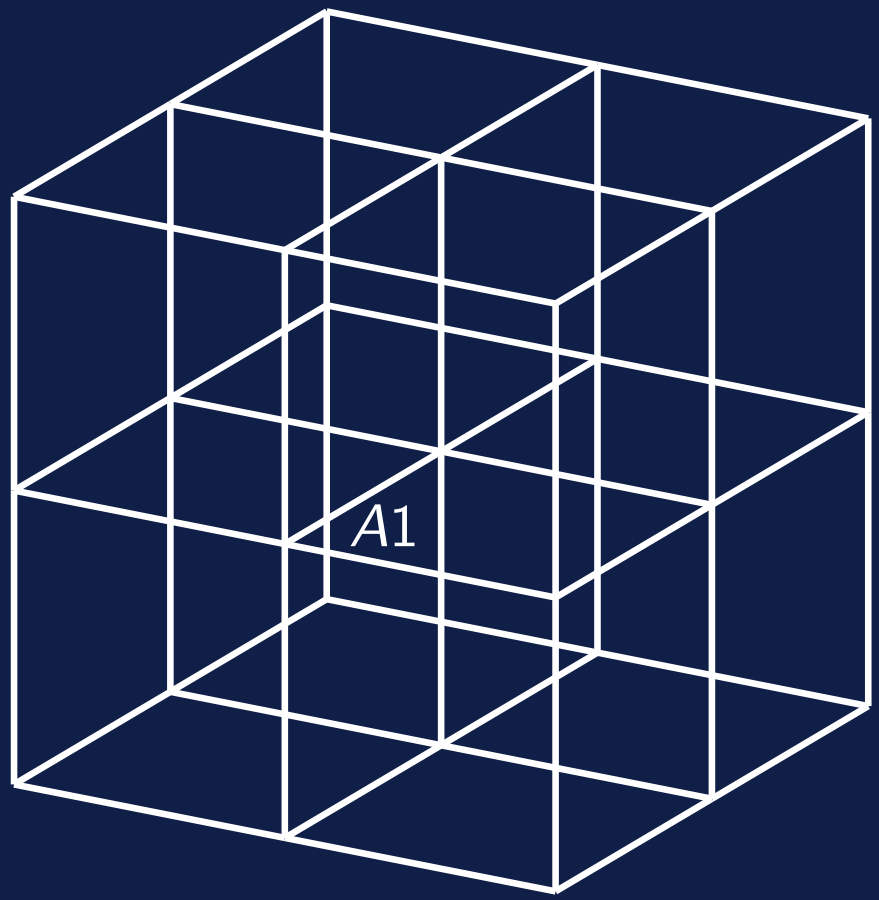
In our case we used feature keys instead of the actual pixel values in the tree. We downsampled the domains to 2x2x2, and then used those vectors. Since this then no longer produces an exact match we then used approximate algorithms to find the 10 nearest neighbors and then evaluated these all separately to find the best one of these.

Classification



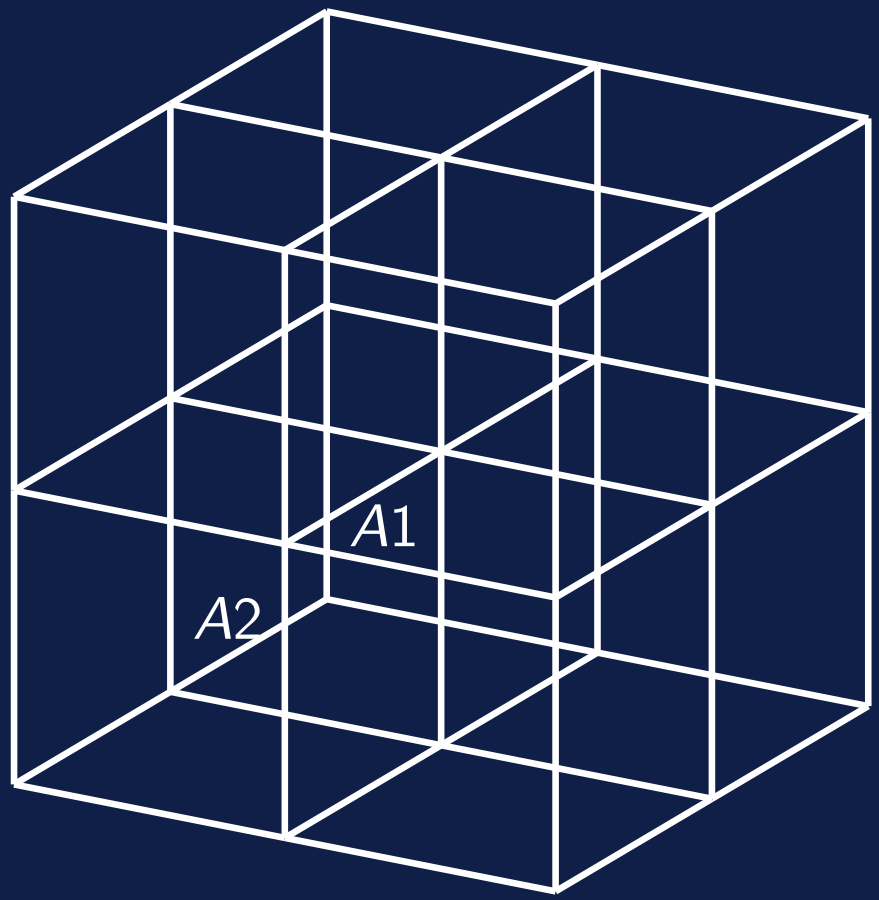
Similar to [18].

Classification



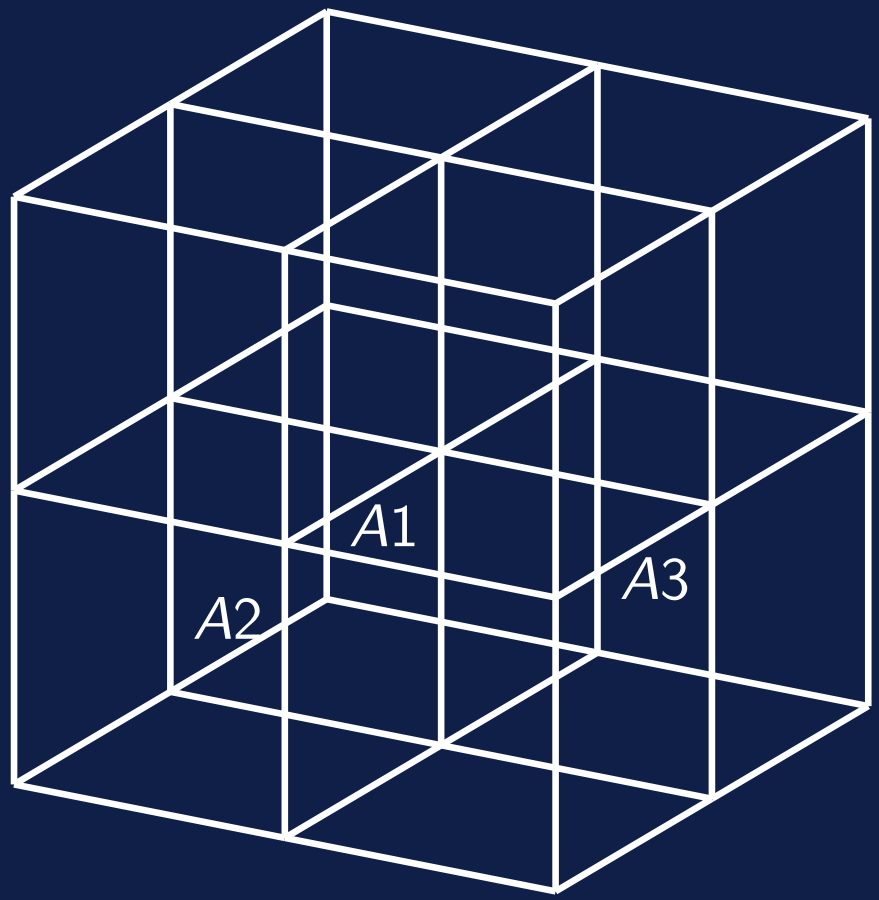
Similar to [18].

Classification



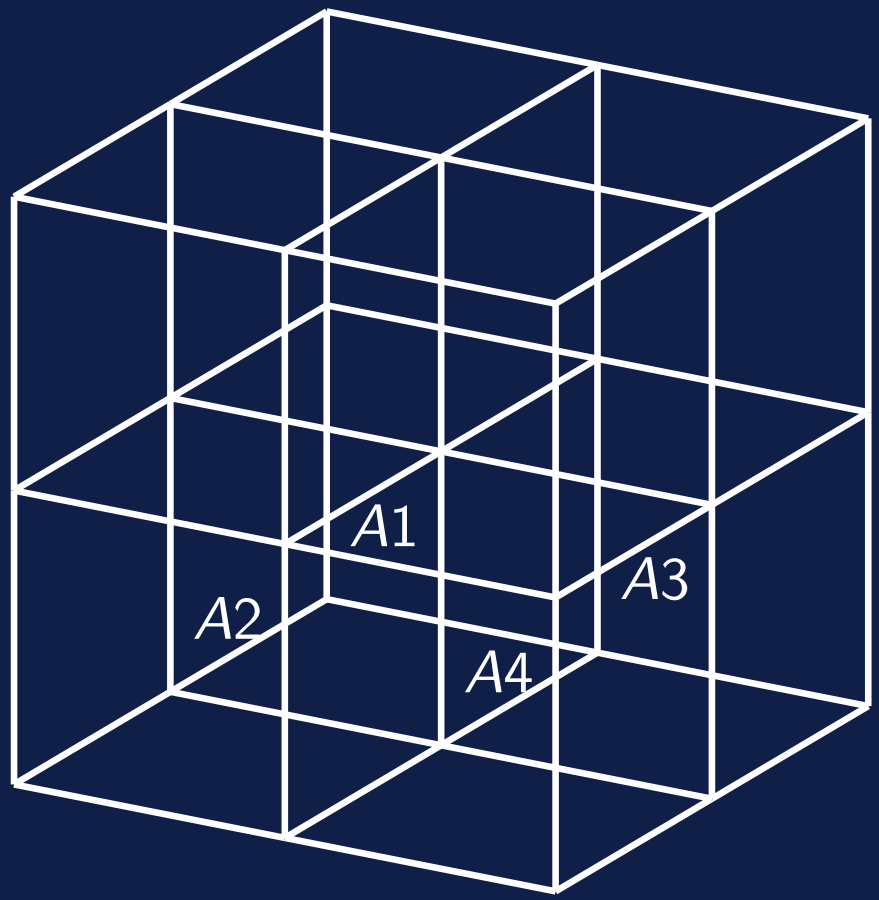
Similar to [18].

Classification



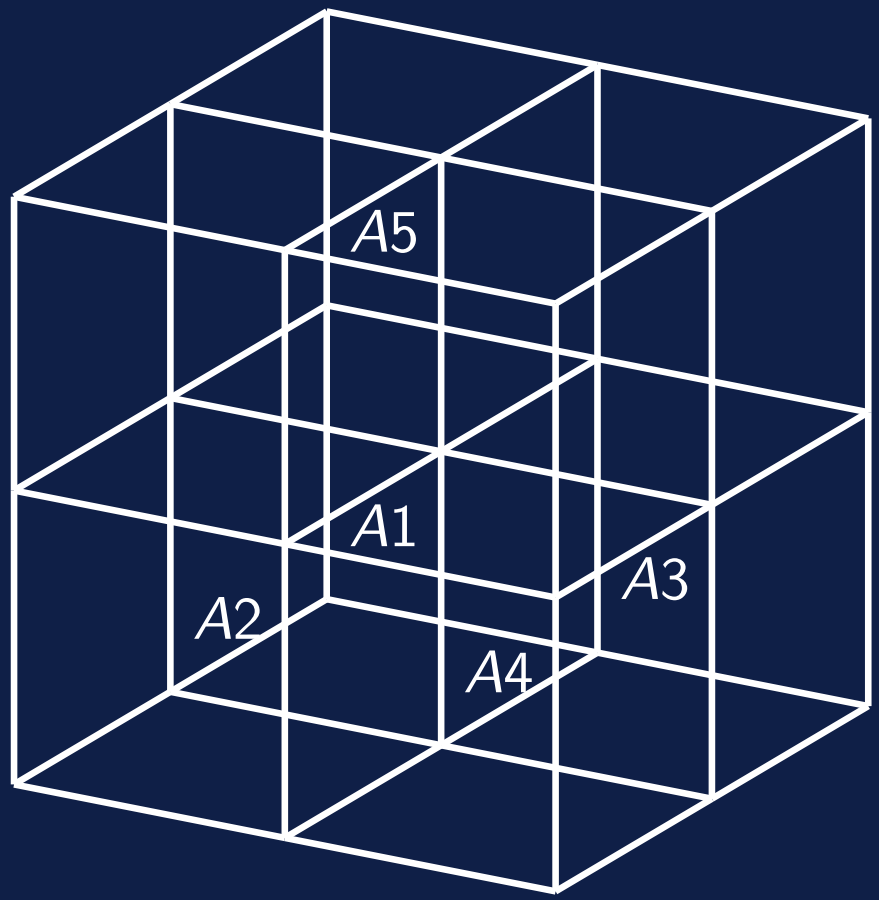
Similar to [18].

Classification



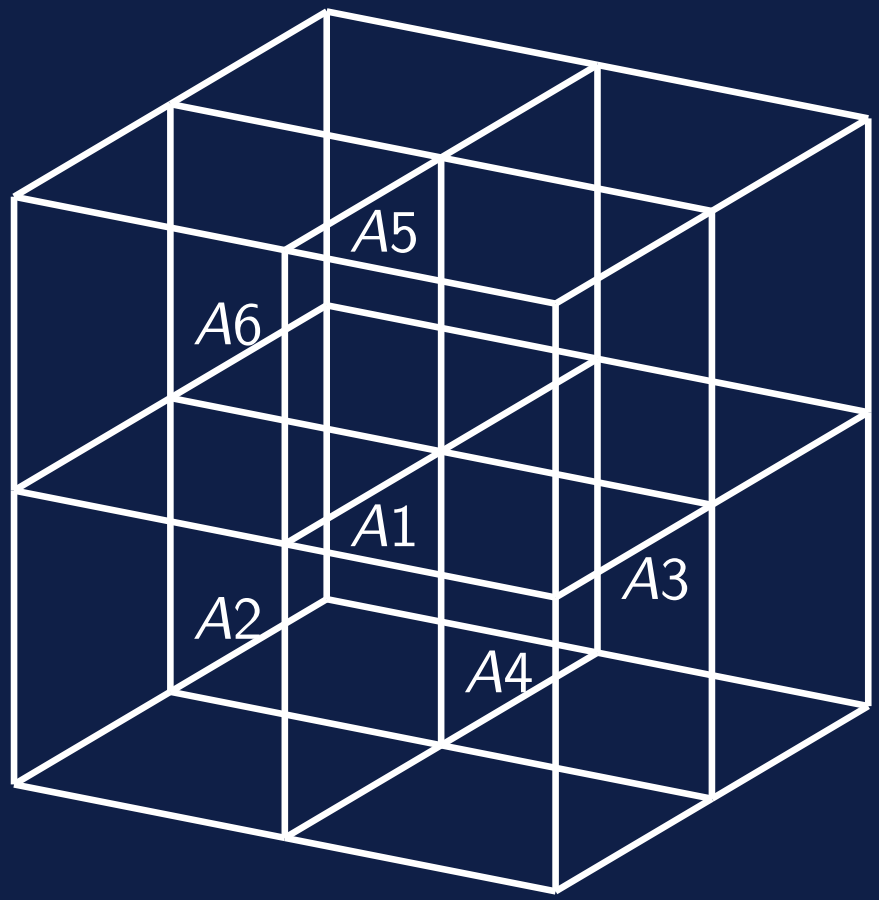
Similar to [18].

Classification



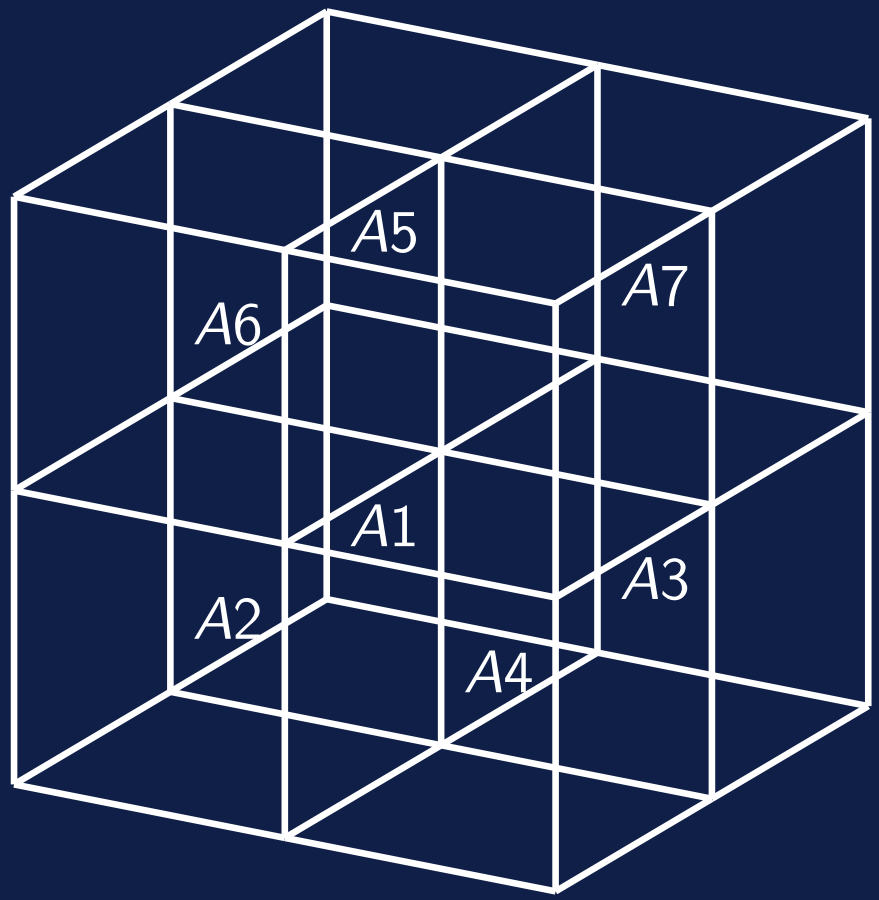
Similar to [18].

Classification



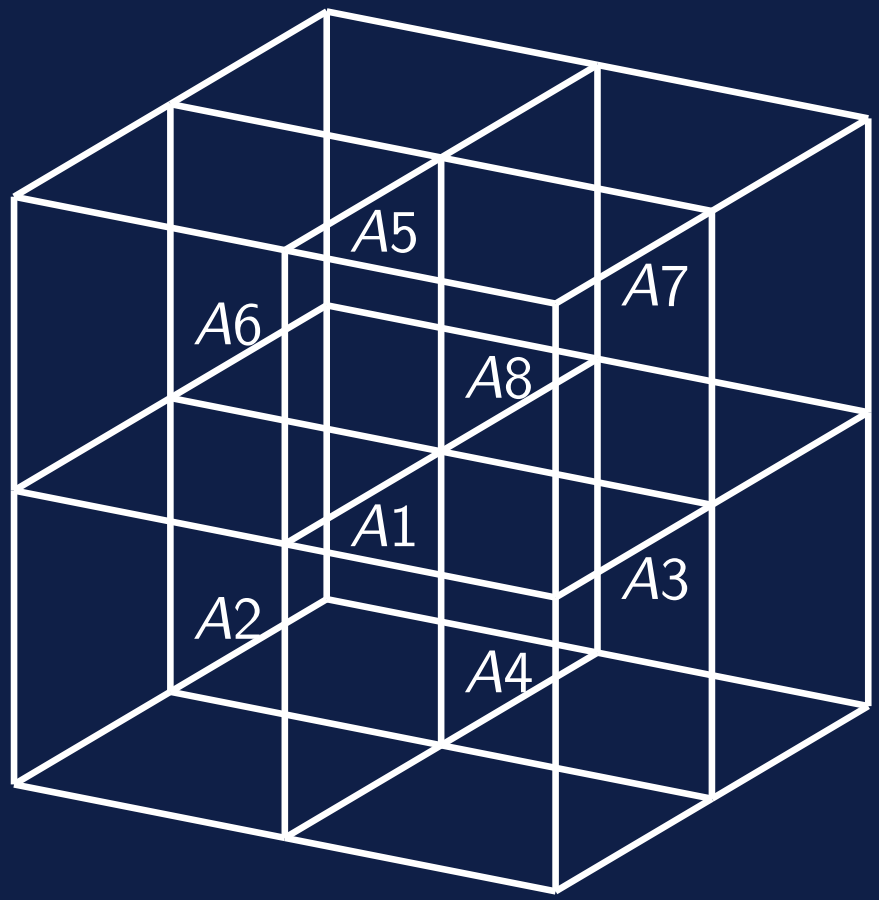
Similar to [18].

Classification



Similar to [18].

Classification



Similar to [18].

Classification continued

- 840 Classes
- More with variance...

Classes in MRIs

From openneuro.org [19].

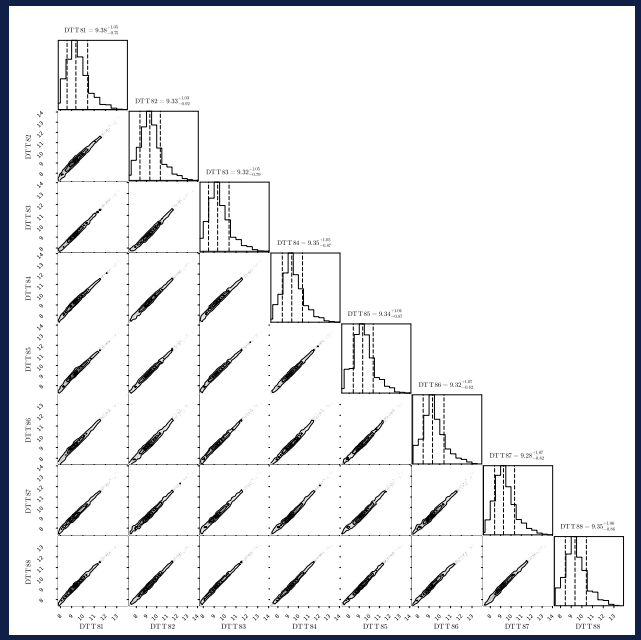


Figure: 8 Compared MRIs

Classes in MRIs

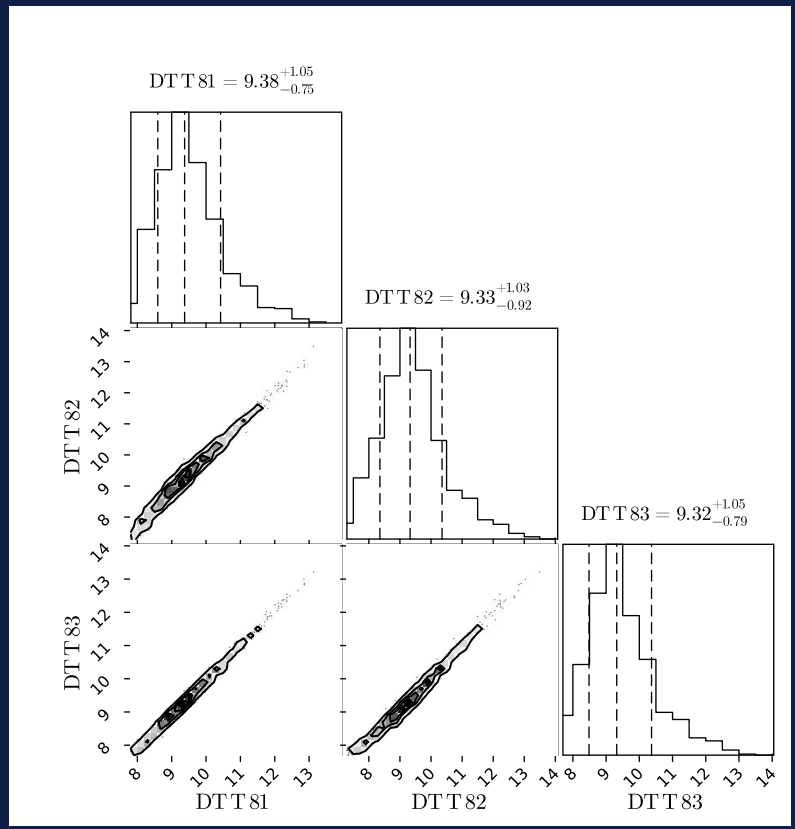
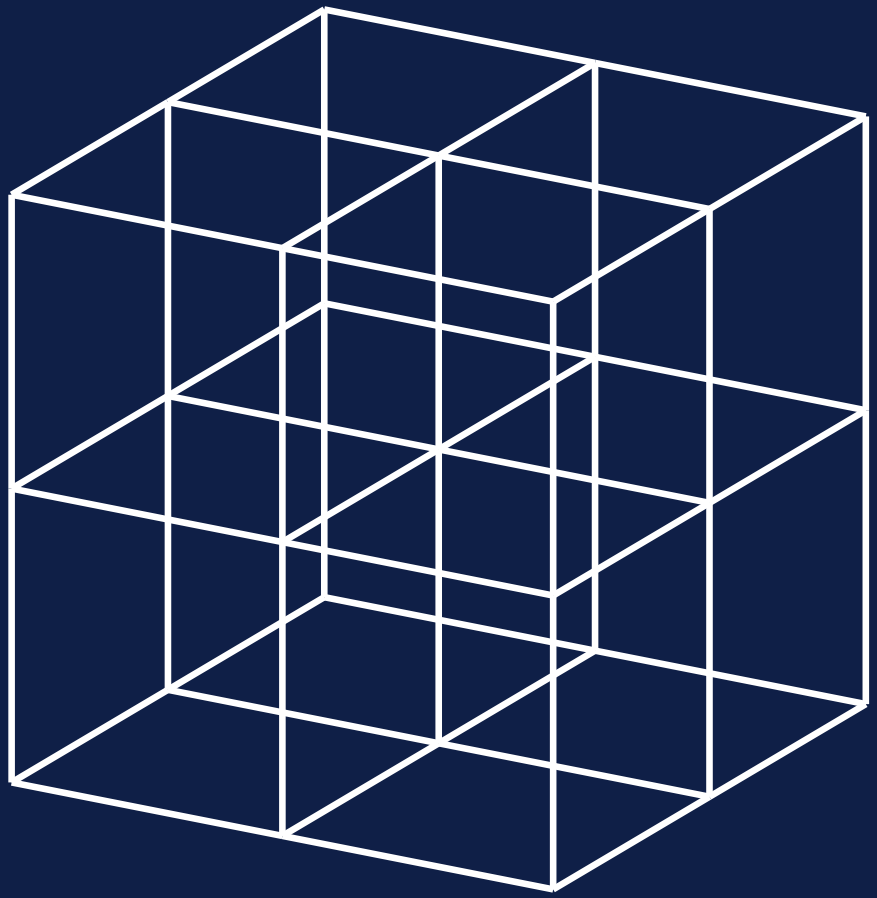


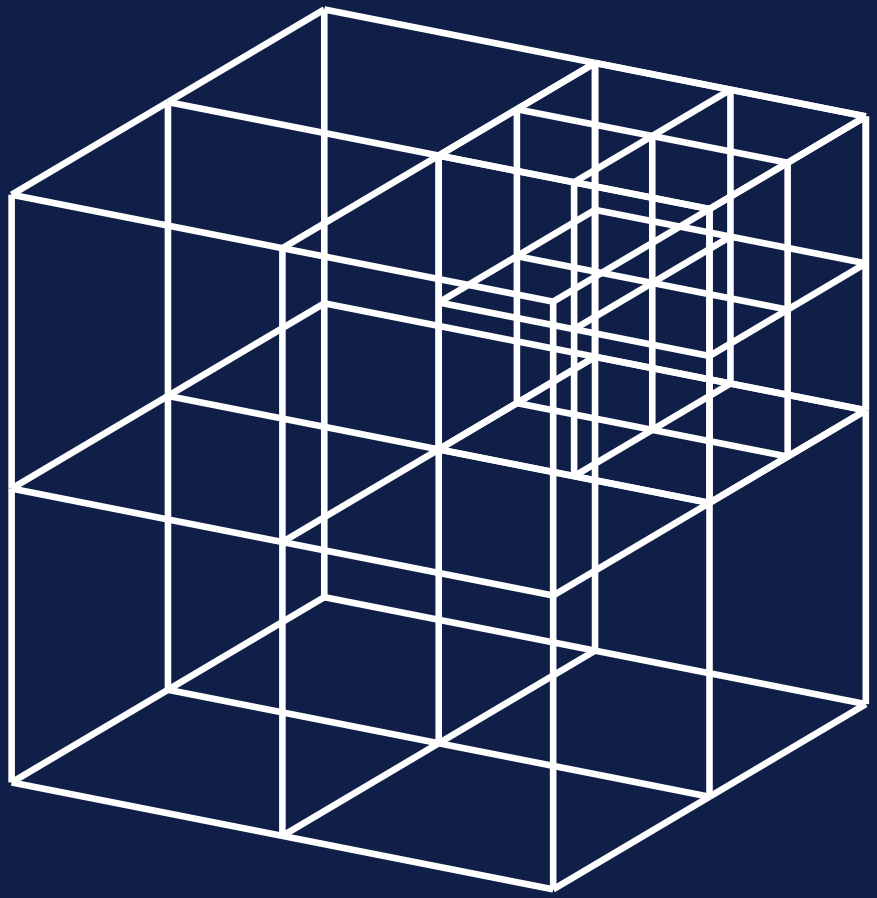
Figure: Just 3 compared

Octotree method

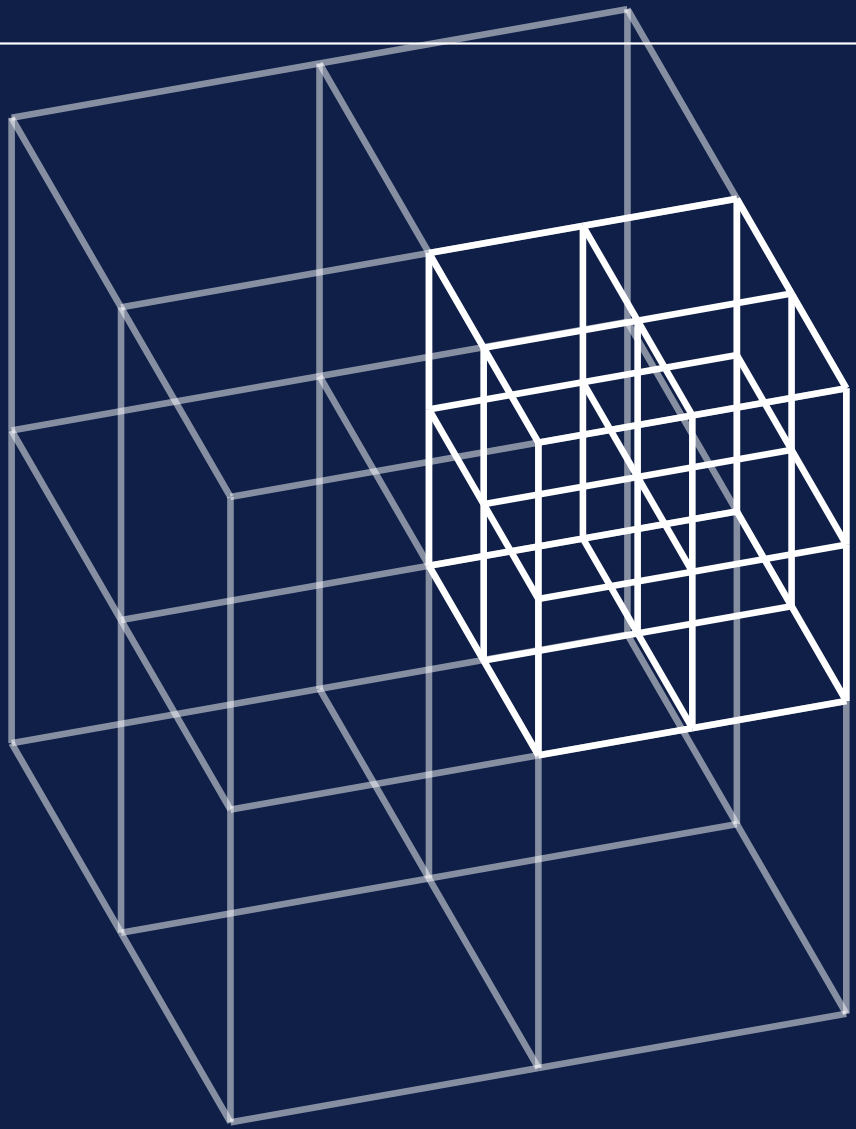


Octotree method

Octotree method



Octotree method



Computers

This was done on the maths department CPU compute machines. With between 36 and 48 cores, and 768 and 2046 GB or RAM with 4.8 to 9.8 TB or Scratch disk space.

- Vanisher
- Polaris
- Cyclops
- Wolverine

Octotree method

Tolerance	Compression Ratio (%)	Time (s)
1,000,000	8.63	530.9
500,000	16.93	532.4
250,000	28.10	606.5
100,000	40.56	674.5
64,000	44.72	700.7
25,000	52.09	815.0
8,000	63.13	869.6
1,000	99.49	1,134.8
200	102.61	1,136.5

Table: How the different tolerance affected program performance.

Types of blocks at different tolerance levels.

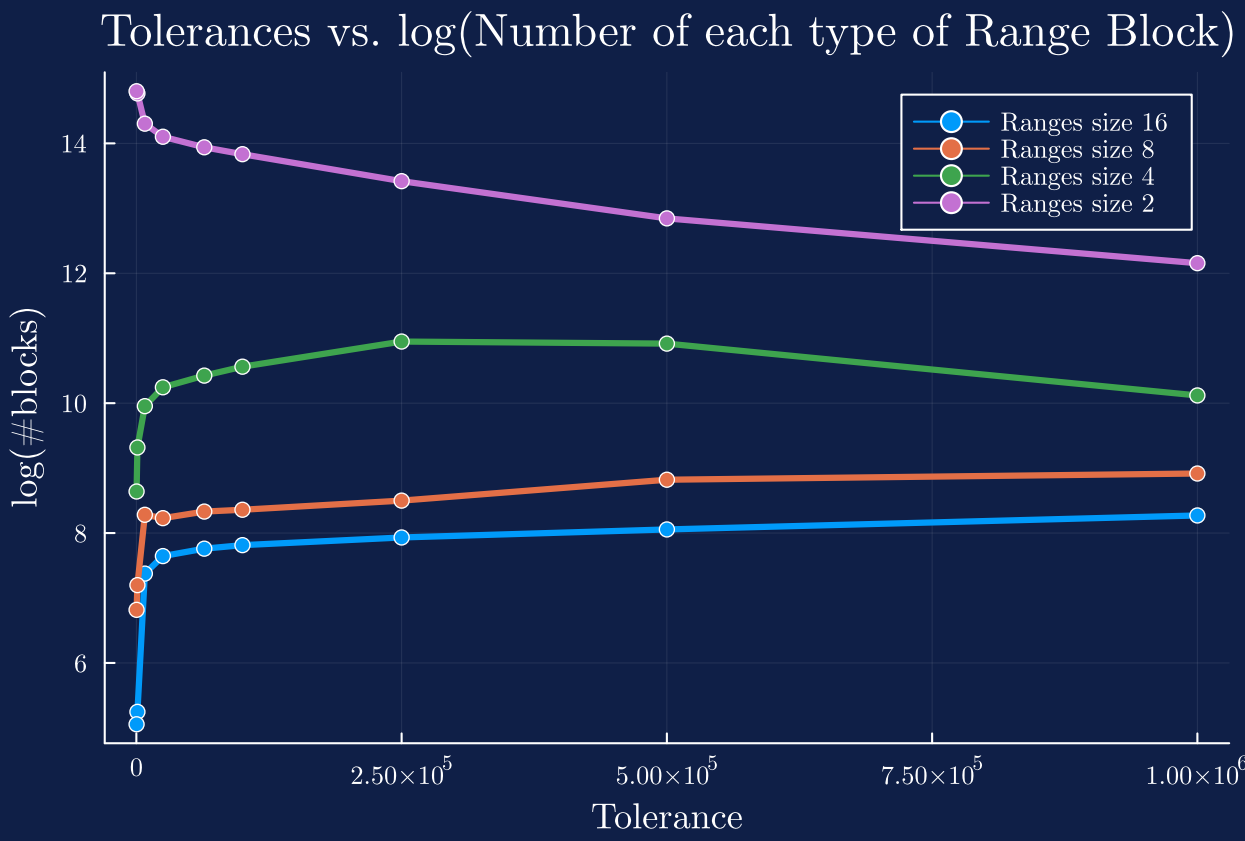


Figure: How quadtree partition changes with size.

Original MRI

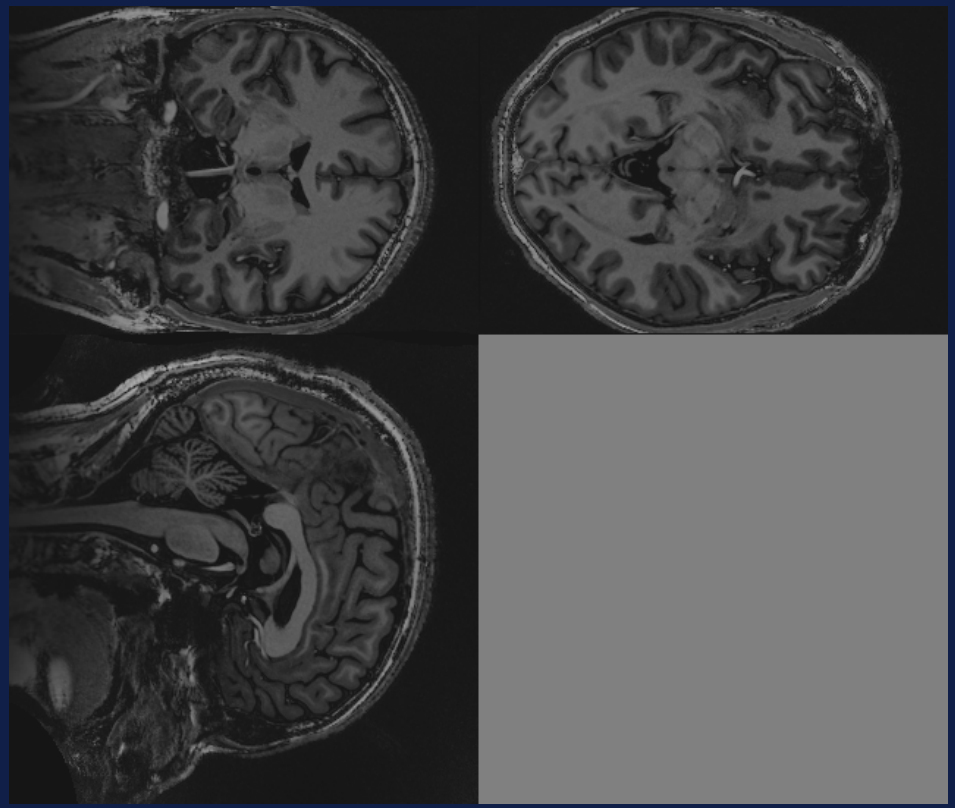


Figure: Original MRI

Original MRI Zoomed

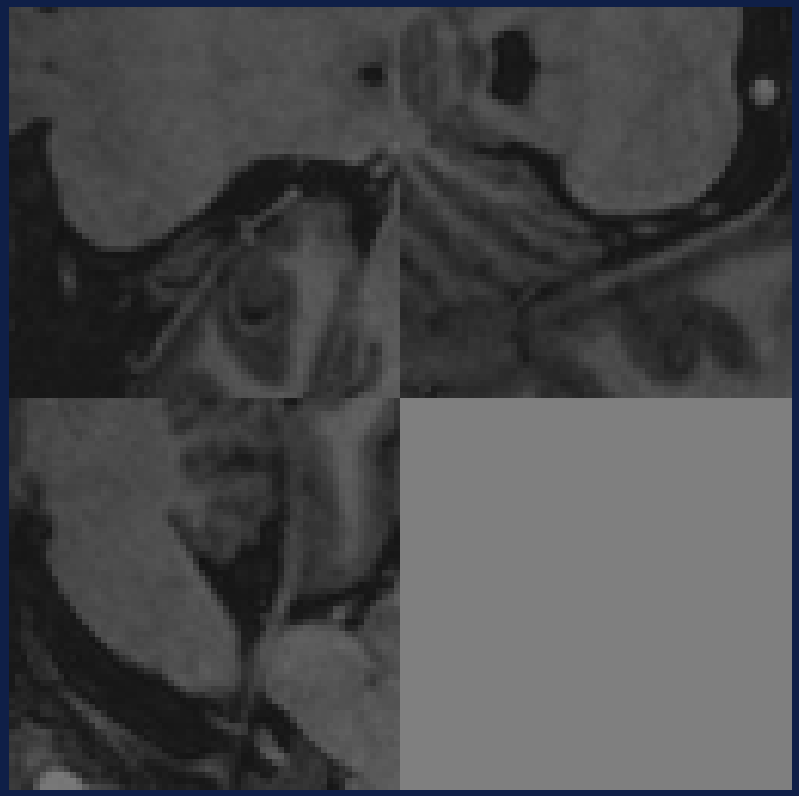


Figure: Original MRI Zoomed

500,000 Level

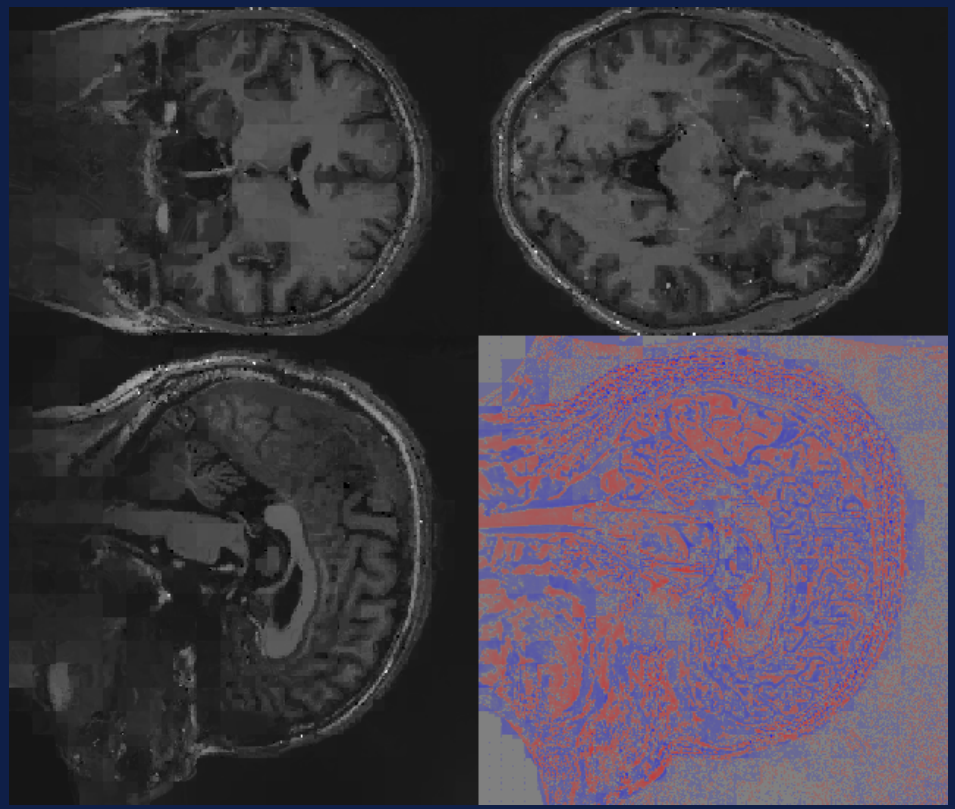


Figure: Level 500,000

500,000 Level Zoomed

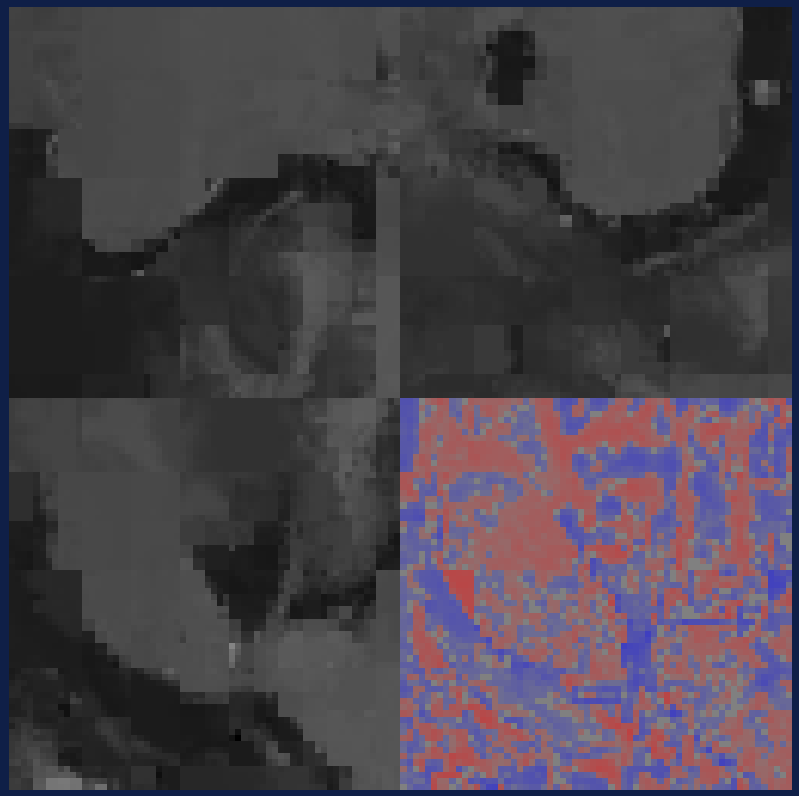


Figure: Level 500,000 Zoomed

100,000 Level

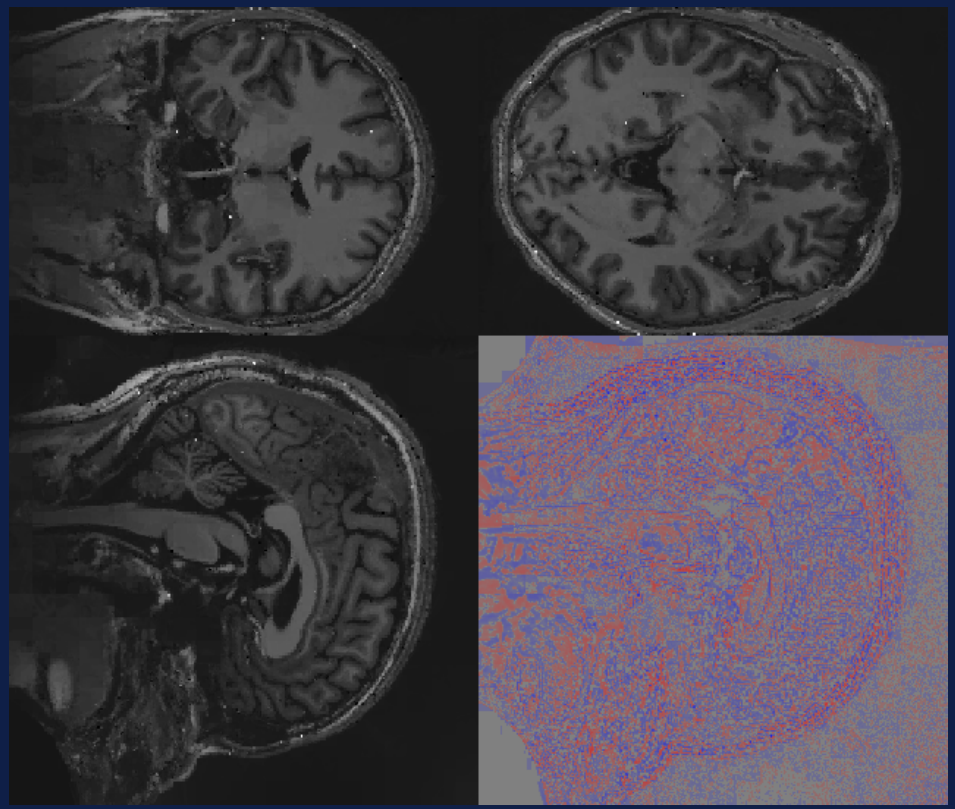


Figure: Level 100,000

100,000 Level Zoomed

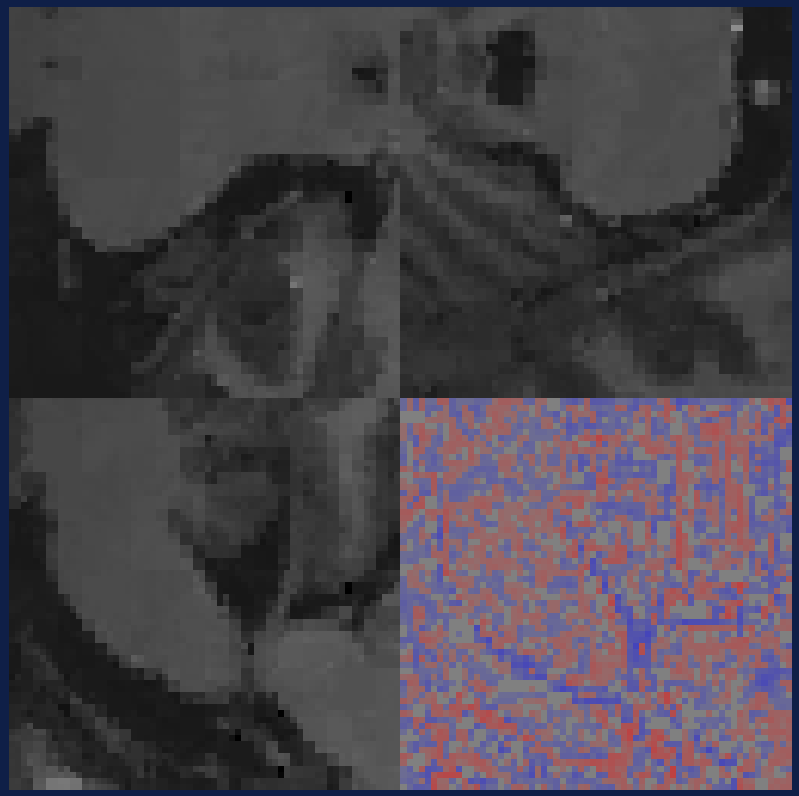


Figure: Level 100,000 Zoomed

8,000 Level

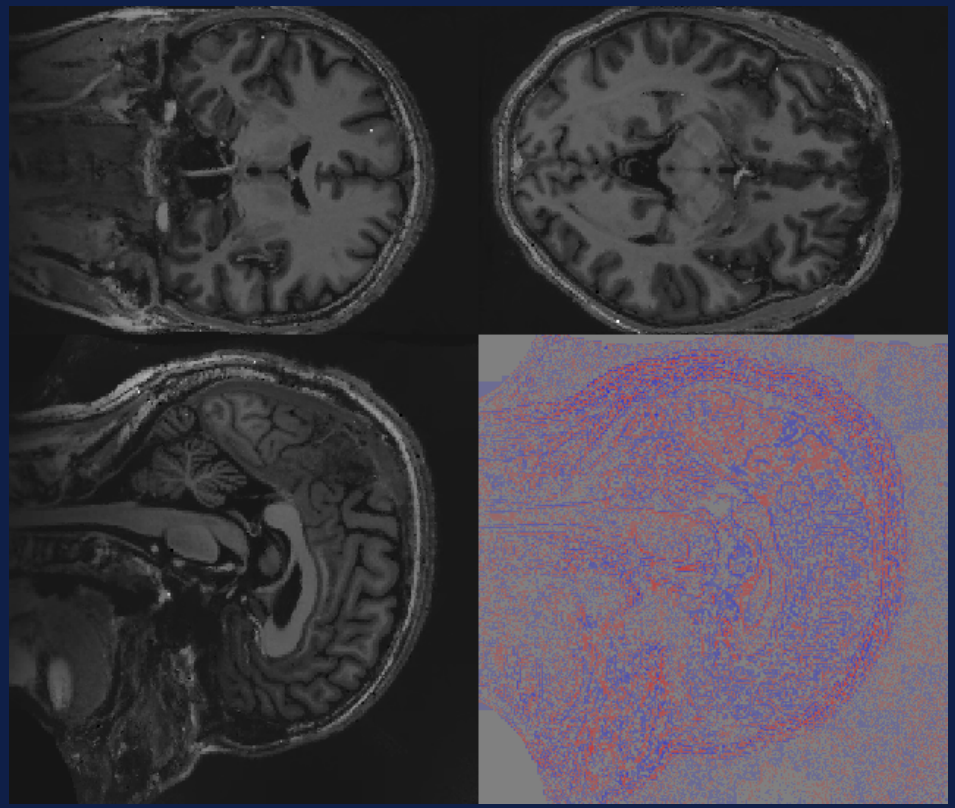


Figure: Level 8,000

8,000 Level Zoomed

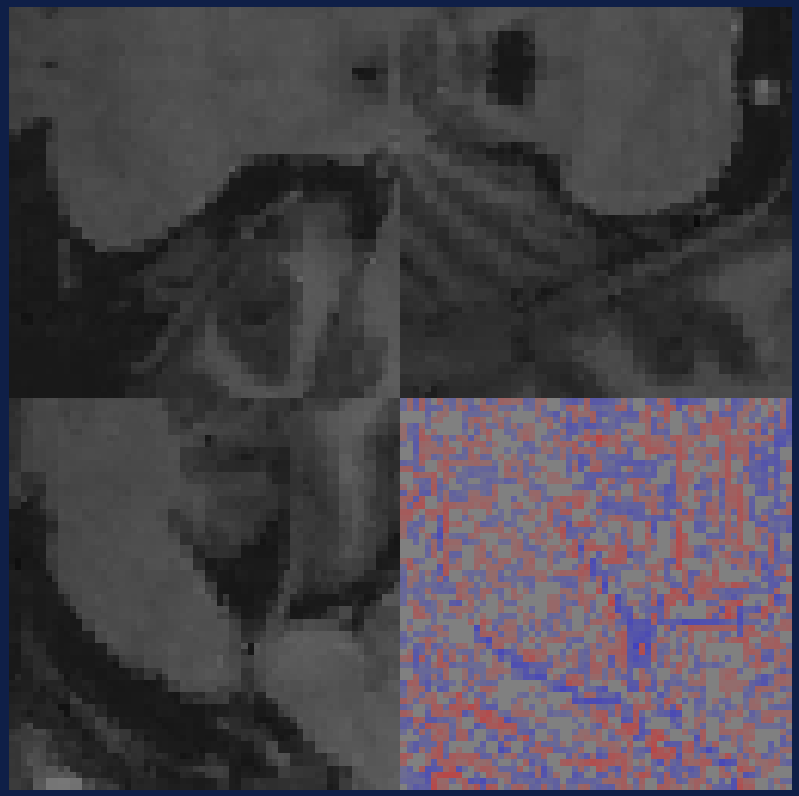


Figure: Level 8,000 Zoomed

500,000 Level Error

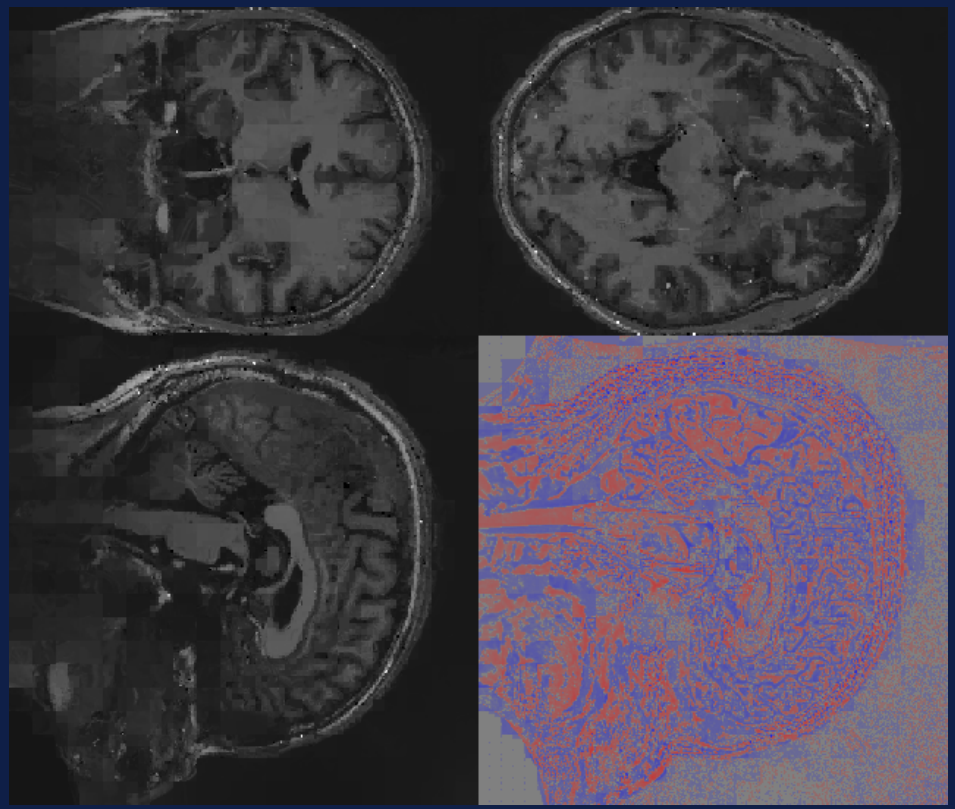


Figure: Level 500,000 Error

500,000 Level Zoomed Error

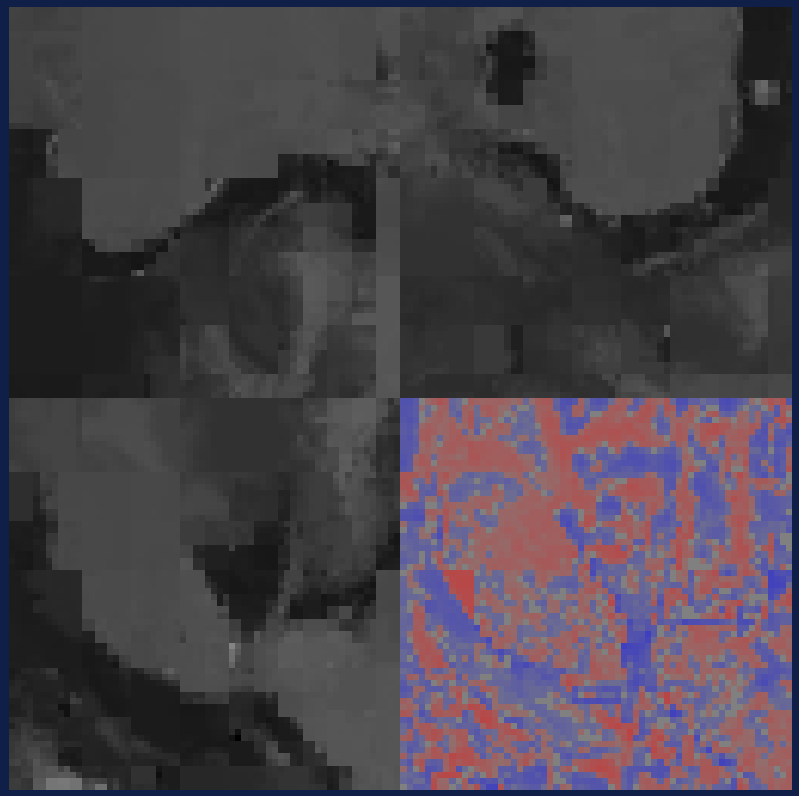


Figure: Level 500,000 Zoomed

100,000 Level Error

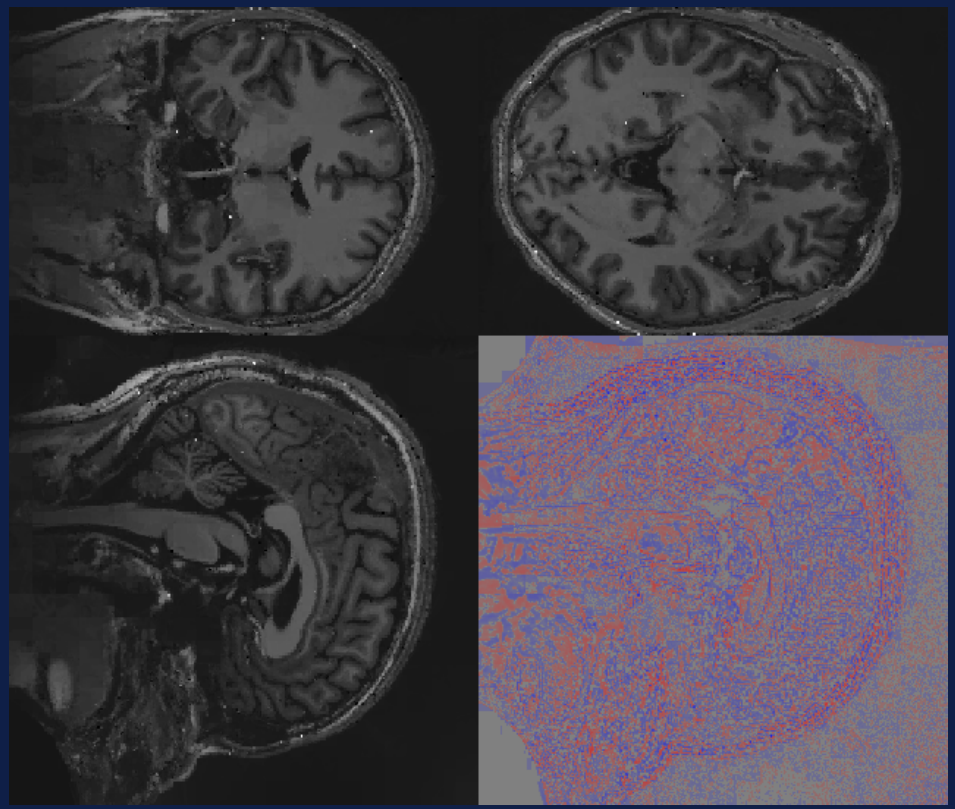


Figure: Level 100,000 Error

100,000 Level Zoomed Error

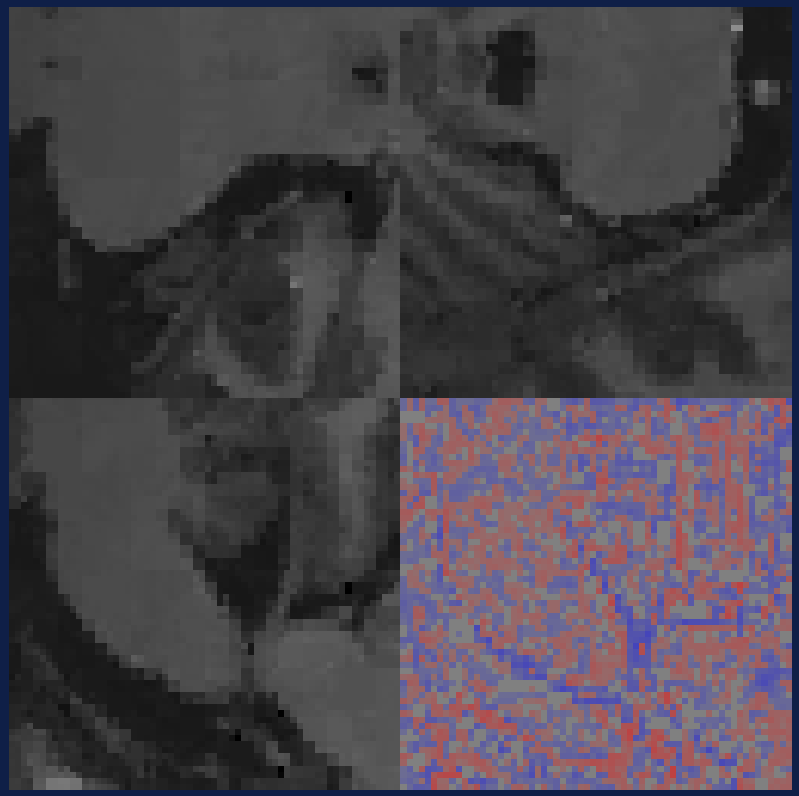


Figure: Level 100,000 Zoomed

8,000 Level Error

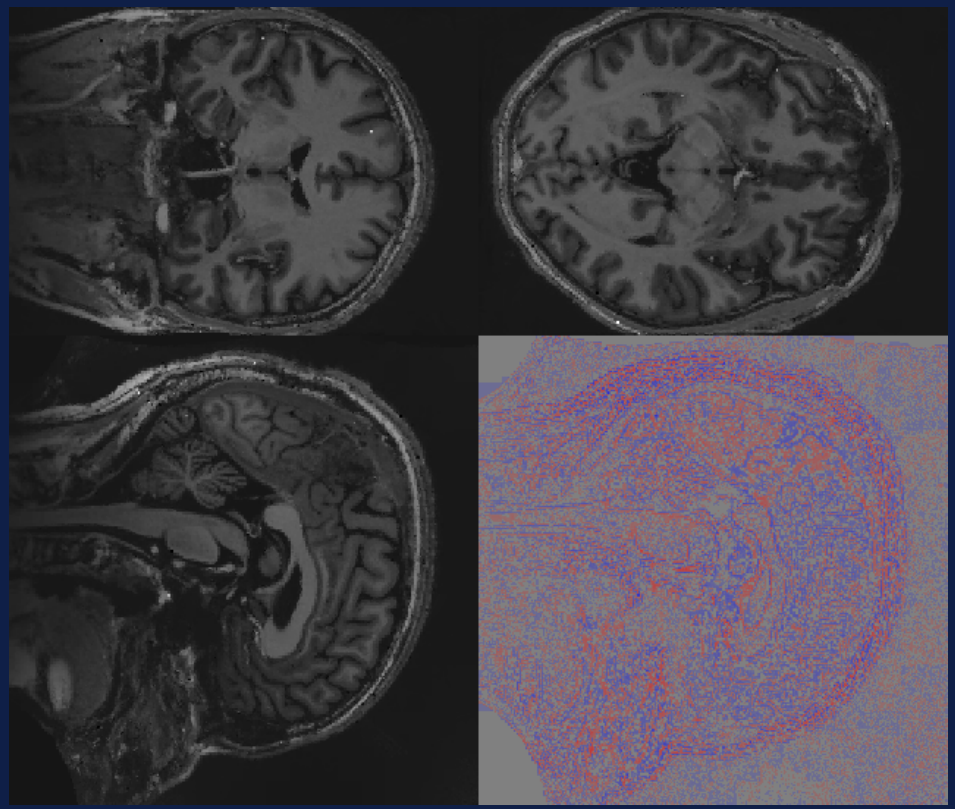


Figure: Level 8,000 Error

8,000 Level Zoomed Error

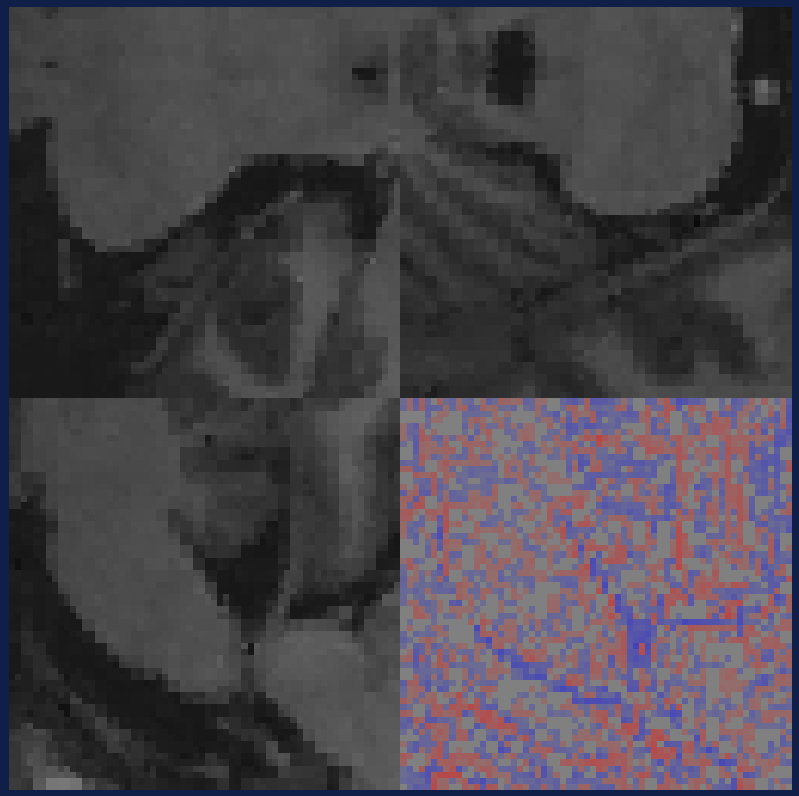


Figure: Level 8,000 Zoomed

Fractal Interpolation

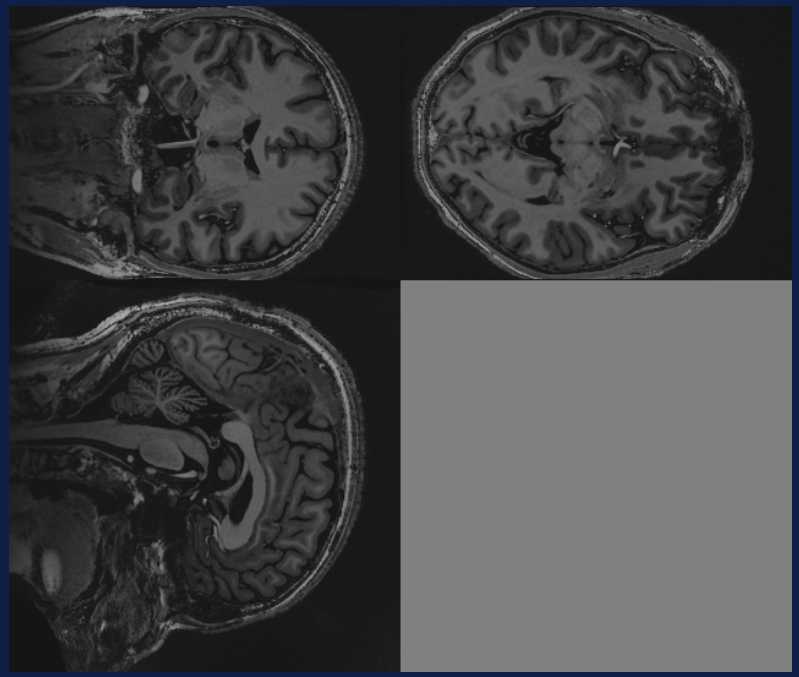


Figure: Interpolated

Fractal Interpolation

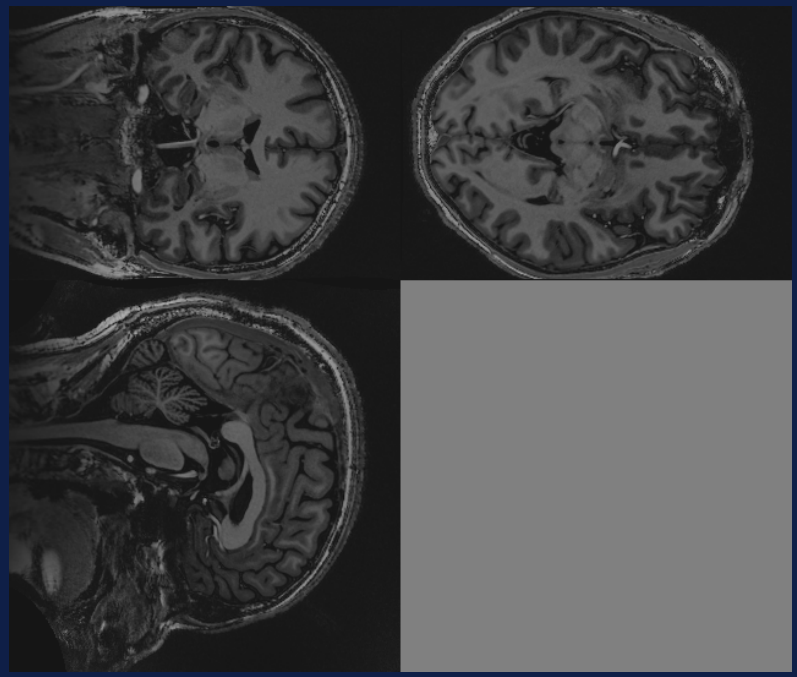


Figure: Interpolated

Fractal Interpolation

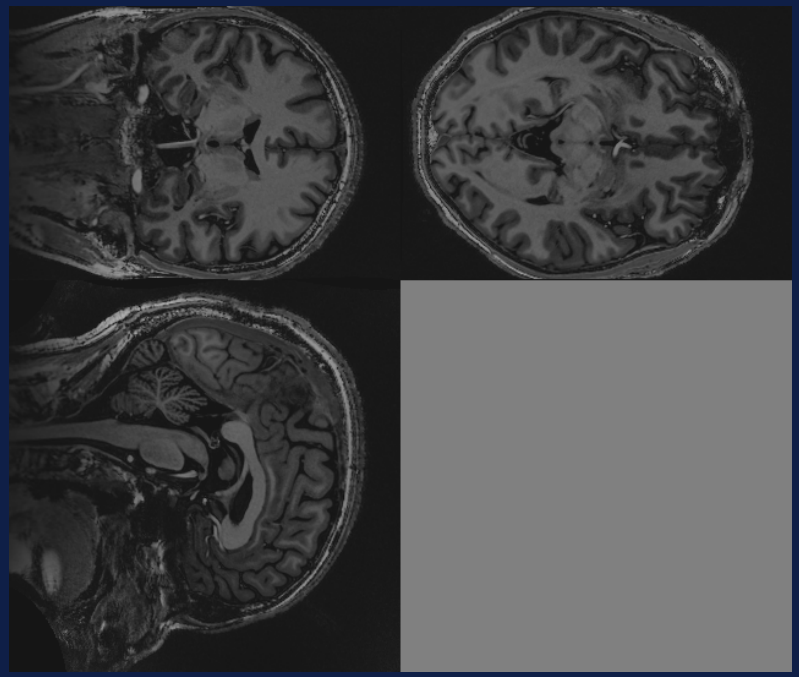


Figure: Interpolated

Video Compression and Questions?

Figure: Video

References

- M. Barnsley, "Fractals everywhere, acad," *Press, New York*, 1988.
- J. Hutchinson, "Fractals and self-similarity," *Indiana Univ. Math. J.*, vol. 30, pp. 713–747, 1981.
- S. Banach, "Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales," *Fund. math*, vol. 3, no. 1, pp. 133–181, 1922.
- F. Hausdorff, *Grundzüge der mengenlehre*, vol. 7. von Veit, 1914.
- A. E. Jacquin, *A fractal theory of iterated Markov operators with applications to digital image coding*. Georgia Institute of Technology, 1989.
- G. Oien, S. Lepsoy, and T. A. Ramstad, "An inner product space approach to image coding by contractive transformations," in *[Proceedings] ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing*, pp. 2773–2776, IEEE, 1991.
- B. Wohlberg and G. De Jager, "A review of the fractal image coding literature," *IEEE Transactions on Image Processing*, vol. 8, no. 12, pp. 1716–1729, 1999.
- G. Lu, "Fractal image compression," *Signal Processing: Image Communication*, vol. 5, no. 4, pp. 327–343, 1993.
- D. Saupe, M. Ruhl, R. Hamzaoui, L. Grandi, and D. Marini, "Optimal hierarchical partitions for fractal image compression," in *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No. 98CB36269)*, vol. 1, pp. 737–741, IEEE, 1998.
- L. Thomas and F. Deravi, "Region-based fractal image compression using heuristic search," *IEEE transactions on Image processing*, vol. 4, no. 6, pp. 832–838, 1995.
- Y. Fisher, E. Jacobs, and R. Boss, "Fractal image compression using iterated transforms," in *Image and text compression*, pp. 35–61, Springer, 1992.
- T. Bedford, F. M. Dekking, and M. S. Keane, "Fractal image coding techniques and contraction operators," *Nieuw Archief voor Wiskunde*, 1992.
- D. Saupe, "Lean domain pools for fractal image compression," in *Still-Image Compression II*, vol. 2669, pp. 150–157, SPIE, 1996.
- M. H. Loew, D. Li, and R. L. Pickholtz, "Adaptive pifs model in fractal image compression," in *Medical Imaging 1996: Image Display*, vol. 2707, pp. 284–293, SPIE, 1996.
- J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Transactions on Mathematical Software (TOMS)*, vol. 3, no. 3, pp. 209–226, 1977.
- K. Carlsson, D. Karrasch, N. Bauer, O. Samuel, T. Kelman, E. Schmerling, J. Hoffimann, M. Visser, P. San-Jose, J. Christie, A. Ferris, P. Anthony Blaom, B. Pasquier, C. Lucibello, C. Foster, E. Saba, G. Goretkin, I. Orson, S. Choudhury, and T. Nagy, "Kristofferc/nearestneighbors.jl: v0.4.11," June 2022.
- J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM review*, vol. 59, no. 1, pp. 65–98, 2017.
- Y. Fisher, *Fractal image compression: theory and application*. Springer Science & Business Media, 2012.
- J. C. Lau, Y. Xiao, R. A. M. Haast, G. Gilmore, K. Uludağ, K. W. MacDougall, R. S. Menon, A. G. Parrent, T. M. Peters, and A. R. Khan, ""stereotactic neurosurgery 7t control dataset (snsx-32)"," 2022.