

# EDGE BASED IMAGE COMPRESSION WITH HOMOGENEOUS DIFFUSION

Debrata Mandal (170050066)  
Manas Shukla (170050073)  
Tushar Agrawal (170100075)

---

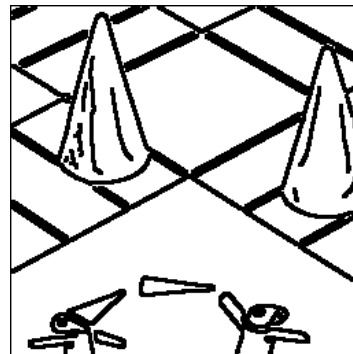
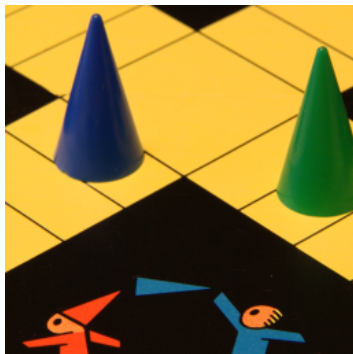
## Introduction:

In this project we have used a lossy compression method for cartoon-like images that exploits information at image edges. These edges are extracted using the Canny edge algorithm. Their locations are stored in a lossless way using JBIG. Moreover, we encode the grey or colour values at both sides of each edge by applying subsampling and PAQ coding. In the decoding step, information outside these encoded data is recovered by solving the Laplace equation, i.e. we in-paint with the steady state of a homogeneous diffusion process.

---

## Encoding:

### Edge detection:



Edge detection algorithms such as Marr Hildreth were used but results were poor, so canny edge algorithm is used.

### Encoding the contour location:

Edge Image (image which contains contours found by canny edge) is encoded and compressed using JBIG. It has been developed as a specialised routine for lossless as well as for lossy compression of bi-level images as mentioned in the paper.

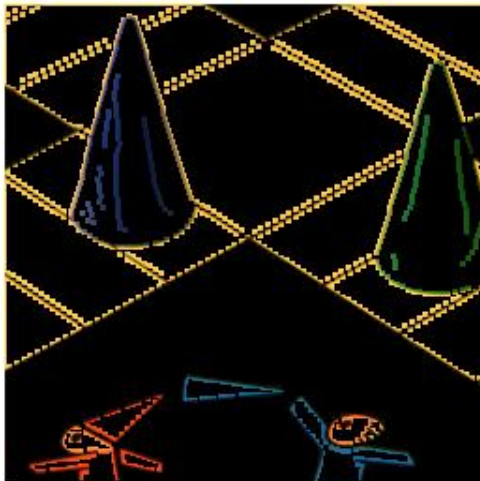
	Original image size (KB)	JBIG compressed edge image (KB)
im1.png	21.2	0.569
im2.png	145.3	1.5

	Original image size (KB)	JBIG compressed edge image (KB)
im3.png	197.6	3.2
im4.png	123.2	1.1
im5.png	71.7	1.3

### Encoding the contour pixel values:

Image is reduced by sampling colour values of pixels that are neighbour to edges, as edges contain most of the information, as proposed in the paper, other colour pixels are blacked out.

Subsampling along edge was experimented but it yielded poor recovery of the edges because of poor super-sampling (which could not be further explored due to time constraints).



### Storing the encoding data

Residual Image is encoded and compressed using PAQ compressor along-with the JBIG file to get one compressed version of the image file.

---

## Decoding:

### Decoding the contour pixel values:

The Archive is decompressed using PAQ decompressor and then the edge image is restored using JBIG decoder.

### Reconstructing Missing Data:

The missing pixels are reconstructed (image in-painting) is performed using homogenous diffusion for interpolation.

$$\partial_t u = \nabla u$$

The reconstructed data satisfies the Laplace equation  $\nabla u = 0$ .

Such a PDE can be discretised in a straightforward way by finite differences.

We tried anisotropic diffusion but could not make it work.



Original Image



Recovered Image

---

## Evaluation:

PSNR value is calculated to measure similarity between restored image and original image.

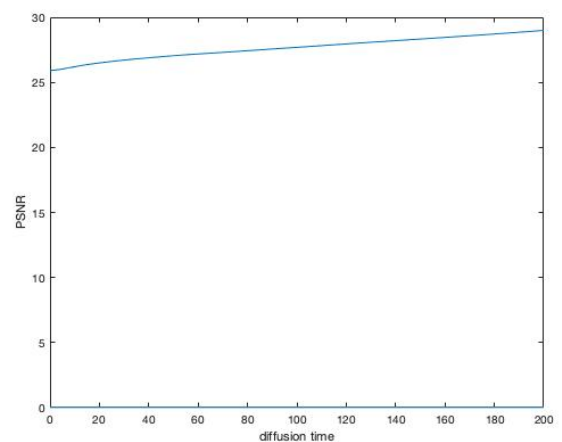
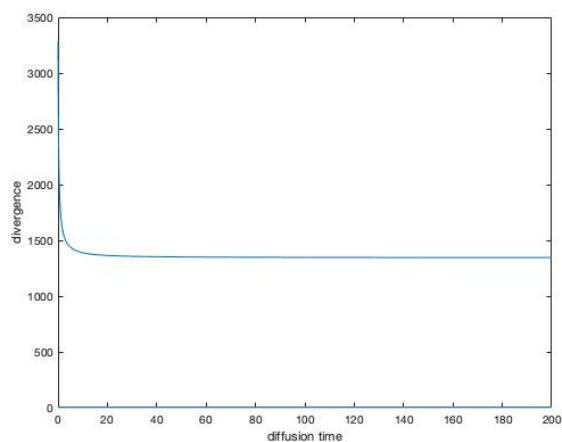
	PSNR	Original image size (KB)	Compressed archive size (KB)	Compression ratio
im1.png	32.003	21.2	14.9	1.42
im2.png	31.425	145.3	59.6	2.43
im3.png	28.832	197.6	84.2	2.34
im4.png	44.208	123.2	40.5	3.04
im5.png	36.302	71.7	45.3	1.58

---

## Hyper paramaters:

- Time step
- Total Time of diffusion

	Diffusion Time step	Total Time of diffusion
im1.png	0.09	500
im2.png	0.25	2000
im3.png	0.25	2000
im4.png	0.25	2000
im5.png	0.15	1500



---

## Analysis:

Huge shortcoming of this compression technique is that it works only for cartoon like images, because of the assumption that area inside a boundary has same colour, which is mostly observed in cartoon images. Also the difference in reconstructed and original image is visible (otherwise diffusion time has to be increased).

The compression ratios are high, note that the compression ratios reported are not on the primitive image, the images are saved in png format which implies image is compressed to 1.5-4 times on top of compression used by PNG.

---

## Steps

1. Run encoder.m with appropriate arguments imagePath, edgeImagePath, resImagePath and run compress.sh from the \*linux\* shell (you may have to install JBIG and PAC encoders).
2. Run decoder.sh from the \*linux\* shell and then decoder.m with appropriate arguments edgeImagePath, resImagePath, imagePath, savePath

---

## References:

1. [Image restoration by partial differential equations- Mirjana Štrboja](#)
2. [Understanding and Advancing PDE-based Image Compression- Pascal Peter](#)
3. [Edge-Based Image Compression with Homogeneous Diffusion- Markus Mainberger and Joachim Weickert](#)
4. [Beating the Quality of JPEG 2000 with Anisotropic Diffusion- Christian Schmaltz, Joachim Weickert, and Andr es Bruhn](#)
5. PDEs for Image Interpolation and Compression - Joachim Weickert