Image reconstruction using binary tomography

Lorenzo D’amico, Maria Popa, Vladimir Jovanovic

July 15, 2022

Contents

1 Introduction

Tomography is imaging by sections or sectioning that uses any kind of penetrating wave. In many cases, the production of these images is based on the mathematical procedure tomographic reconstruction, such as X-ray computed tomography technically being produced from multiple projectional radiographs. Many different reconstruction algorithms exist. Most algorithms fall into one of two categories: filtered back projection (FBP) and iterative reconstruction (IR)[6].

Discrete Tomography (DT) focuses on the problem of reconstruction of binary images from a small number of their projections. The problem of tomography is to reconstruct an object from a set of projections. The object corresponds to a function and the problem posed is to reconstruct this function from its integral or sum over subsets of its domain. A special case of discrete tomography deals with the problem of the reconstruction of a binary image from a small number of projections.[5]

2 Reconstruction as an Optimization Problem

2.1 Discrete Tomography

We assume that reconstruction to be in 2D. The slice's size of the reconstructed object is represented by an n x n sized image. We also assume that we are in the condition of a parallel beam therefore the projection values are the linear integration along the X-rays. The general discrete reconstruction problem can be represented by the system of linear equations: $Ax = b$,where$A \in \mathbb{R}^{mxn^2}, x \in L^{n^2}, b \in \mathbb{R}^m$ Where:

- x is the vector of all n^2 unknown image pixels
- b is the vector of all m measured projections
- A describes the projection geometry where a_{ij} is the length of the line segment of the i-th projection line through the j-th pixel
- $L = l_0, l_1, \ldots, l_c$ is the set of possible intensity (assuming that $l_0 < l_1 <$ ... $< l_c$)

If we assume that L = 0,1 we arrive to the well-known model of binary tomography[4]

2.2 Optimization Method

A possible way to solve the equation:

$$Ax = b \tag{1}$$

At least approximately is to reformulate as an optimization problem. We have to find the minimum of the following objective function:

$C(x) = \|Ax - b\|^2 + \gamma \cdot \phi(x)$

The first term ensures that we have an x satisfying the equation, at least approximately. The second term allows us to include some prior information about x into the optimization if there are several good binary candidates that keep $\|Ax - b\|^2$ low.

3 Simulated Annealing

3.1 The Basic Idea

Simulated annealing is one of the most popular local search algorithms, only trailing the evolutionary algorithms (the Genetic algorithm belongs to this family). It is widely used in solving optimization problems. SA was proposed by Kirkpatrick in 1983, who took inspiration from the natural phenomenon known as Annealing, in metallurgy. The process defines the optimal molecular arrangements of metal particles where the potential energy of the mass is minimized and refers cooling the metals gradually after subjected to high heat. The SA algorithm simulates this process by using a temperature variable and conducting actions based on the current value of this variable.

3.2 Convergence

It is possible to use every local search algorithm as an optimization algorithm, so it is only natural to ask what it is that makes Simulated Annealing so efficient.

A simple search algorithm does an iterative comparison of the objective function in the current and all the neighbouring states, and selects the best of them as the next base state for further iterations. If an iteration completes without a change in the state, the algorithm does not search a wider domain for better results, and instead terminates and returns the state[8].

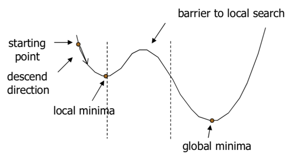


Figure 1: Limitations of simple search algorithms

As can be seen in the figure above, it is often the case that the algorithm gets stuck in the local optima. The SA proposes an efficient solution to this problem by using two iterative nested loops, one for simulating the cooling process in annealing, and the other for using the Metropolis criterion.

3.2.1 Metropolis criterion

The basic idea behind the Metropolis criterion to compute a variable using a certain formula, and compare the value of the variable with a randomly generated value from the segment [0, 1]. In case a condition is fulfilled, the algorithm should visit one of the neighbours of the current state even if the value of that state is smaller than the current one, thus allowing ”bad” steps with the purpose of finding the global optima.

To be more precise, let us define an objective function $f(x_i)$ corresponding to the argument set of $x_i = x_1, x_2, ..., x_n$ with $n \in \mathbb{R}$, and let us consider a minimization problem. As explained in 3.2, if $f(x_i) < f(x_{i+1})$ is met, then take x_{i+1} as the next base for the algorithm. In case that condition is not met let:

$$\Delta = f(x_i) - f(x_{i+1}) \tag{2}$$

and

$$\omega = e^{-\Delta/T_C} \tag{3}$$

where T_C is the current temperature parameter, and generate a random variable s, such that $0 < s < 1$. If $\omega > s$ is true, then we take x_{i+1} as the new base, else go back and generate another s.

The Metropolis criterion, with larger values of T_C, allows a bigger search range area, which raises the probability to reach the global minimum.

It is key to start with a good value for T_C, because too big of a value will always satisfy the condition $w > s$, so the algorithm will most likely be terminated at a random minimum which may be a local minimum, and too small of a value results in a lack of flexibility, so the algorithm is also very likely to get stuck at the first minimum it reaches, which can easily be a local and not a global minimum[8].

3.3 Implementation

The code below represents the pseudo-code of the simulated annealing algorithm for the binary image reconstruction problem.

Algorithm 1 SA Algorithm

```
Require: $\gamma \geq 0$
Require: $T_{start} > 0$
Require: $T_{min} > 0$
Require: $1 > \alpha > 0$
  $x := (0, 0, ..., 0)^T$
  $T := T_{start}$
  $C_{start} := \|Ax - b\|^2 + \gamma \cdot \phi(x)$
  while $T > T_{min}$ do
    for $i = 0$ to $sizeof(x)$ do
      choose a random position $j$ in the vector x
      $x[j] := 1 - x[j]$
      $C_{new} = \|Ax - b\|^2 + \gamma \cdot \phi(x)$
      $s = random(0, 1)$
      $\Delta C = C_{new} - C_{old}$
      if !($\Delta C < 0$ or $e^{-\Delta C/T} > s$) then
        $x[j] := 1 - x[j]$
      end if
    end for
    $T = \alpha \cdot T$
  end while
```

3.4 Regularization

Often, it is possible to improve the results of the SA algorithm by adding some *a priori* information about the image we are trying to reconstruct in order to keep $\|Ax - b\|^2$ low.

In our experiments, we have used the following formula:

$$\phi(x) = \sum_{i=0}^{n}((x_i - x_r)^2 + (x_i - x_b)^2) \tag{4}$$

In formula (3), x_i denotes the i-th pixel in the vector x, x_r denotes the first pixel on the right side of that pixel and x_b denotes the first bottom pixel of x_i.

4 Evaluation

4.1 Data sets

In our experiments, both custom and downloaded binary images were used. Some of the images were downloaded from `https://game-icons.net/`. The images were then transformed to 2D and thresholding was applied.

All of the images were taken with 20 projections, and resized to have the size of 64x64. Directions of the projections were uniformly chosen within [0°, 180°].

Aside from noiseless projection data, we also used noisy data. The projection data was superimposed by noise with Gaussian distribution $\mathcal{N}$ (0, σ), $\sigma \in \{1, 3\}$.

4.2 Performance measures

In the implementation of the task, the following numbers were computed for the white values of the matrix [7]:

- TP - true positives - the number of white pixels our reconstruction has that are actually white
- TN - true negatives - the number of black pixels our reconstruction has that are actually black
- FP - false positives - the number of black pixels our reconstruction has that are actually white
- FN - false negatives - the number of white pixels our reconstruction has that are actually black

The main performance measure of our experiment is the RME - relative mean error, which is computed using the following formula:

$$RME = M(x)/O(x) \tag{5}$$

where

$$M(x) = FP + FN \tag{6}$$

i.e. M(x) is the number of all the misguessed pixels, and O(x) is the total number of white pixels in the original image.

4.3 Parameter Settings

A fixed set of parameters were used in every run of the algorithm. The algorithm was run 10 times for each scenario (no noise, noise with variance 1, noise with variance 3).

$$\gamma = 14.0$$

$$T_{start} = 4.0$$

$$T_{min} = 10^{-14}$$

$$T_{factor} = 0.97$$

5 Results

In the following table, the original images as well as the results of our experiments can be seen, alongside mean error values.

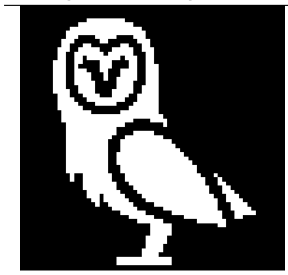
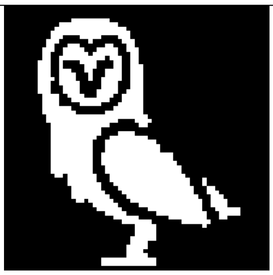
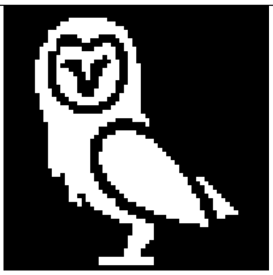
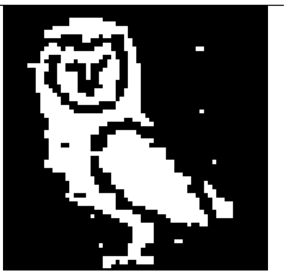
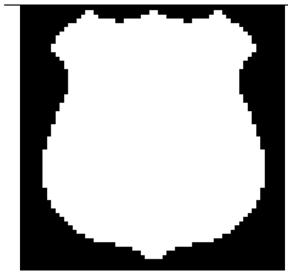
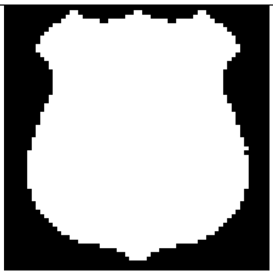
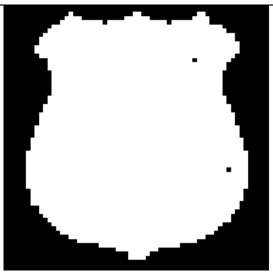
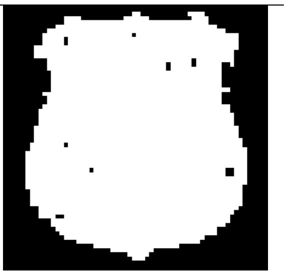
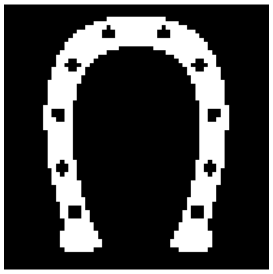
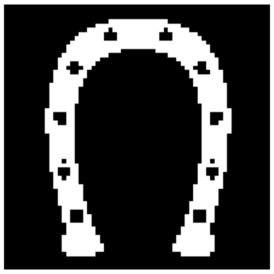
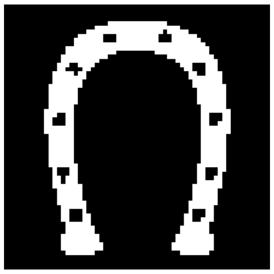
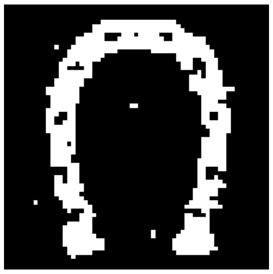
Original Image	Without Noise	$\mathcal{N}$ (0, 1)	$\mathcal{N}$ (0, 3)
	 0.49%	 2.91%	 16.47%
	 0.08%	 0.28%	 2.94%
	 0.37%	 2.15%	 16.08%

Table 1: Results comparison

* The percentage values in the table show the median of the error across all 10 instances.

6 Conclusion

The patterns in the experiment are according to expectations, but the algorithm has performed better than anticipated. It is clear from the examples in 1 that there is a big difference between the reconstruction of compact shapes and icons that have ”cuts” in them. The compact shape errors are much lower than those of other icons, even when noise is added. In the future, the parameters could be changed, and the number of projections may be increased or lowered, depending on the size of the image and the speed of the hardware used for the experiment.

References

[1] A. Kadu and T. van Leeuwen, ‘A Convex Formulation for Binary Tomography’, IEEE Trans.Comput. Imaging, vol. 6, pp. 1–11, 2020, doi: 10.1109/TCI.2019.2898333.

[2] N. Hantos, S. Iván, P. Balázs, and K. Palágyi, ‘Binary image reconstruction from a small number of projections and the morphological skeleton’, Ann Math Artif Intell, vol. 75, no. 1–2, pp. 195–216, Oct. 2015, doi: 10.1007/s10472-014-9440-8

[3] W. Cai and L. Ma, ‘Comparison of approaches based on optimization and algebraic iteration for binary tomography’, Computer Physics Communications, vol. 181, no. 12, pp. 1974–1981, Dec. 2010, doi: 10.1016/j.cpc.2010.09.004

[4] L. Varga, P. Balázs, and A. Nagy, ‘Discrete tomographic reconstruction via adaptive weighting of gradient descents’, Computer Methods in Biomechanics and Biomedical Engineering: Imaging Visualization, vol. 3, no. 2, pp. 101–109, Apr. 2015, doi: 10.1080/21681163.2013.853624

[5] https://en.wikipedia.org/wiki/Discrete_tomography

[6] https://en.wikipedia.org/wiki/Tomography

[7] S. Weber, A. Nagy, T. Sch u1 ,C. SchN or1, A. Kuba, ’A Benchmark Evaluation of Large-Scale Optimization Approaches to Binary Tomography’

[8] Y. Eren, İ. Üstoğlu, ’Optimization in Renewable Energy Systems’, 2017