Voxelisation in the 3-D Fly Algorithm for PET

Zainab Ali Abbood, Julien Lavauzelle, Evelyne Lutton, Jean-Marie Rocchisani, Jean Louchet, Franck P. Vidal

Voxelisation in the 3-D Fly Algorithm for PET

Abbood, Zainab; Lavauzelle, Julien; Lutton, Evelyne; Rocchisani, Jean-Marie; Louchet, Jean; Vidal, Franck

Swarm and Evolutionary Computation

DOI:
10.1016/j.swevo.2017.04.001

Published: 01/10/2017

Peer reviewed version

Voxelisation in the 3-D Fly Algorithm for PET

F. P. Vidal [1], and P.-F. Villard [2]
[1] School of Computer Science, Bangor University, LL57 1UT, United Kingdom
[2] LORIA, University of Lorraine, France

Abstract

The Fly Algorithm was initially developed for 3-D robot vision applications. It consists in solving the inverse problem of shape reconstruction from projections by evolving a population of 3-D points in space (the 'flies'), using an evolutionary optimisation strategy. Here, in its version dedicated to tomographic reconstruction in medical imaging, the flies are mimicking radioactive photon sources. Evolution is controlled using a fitness function based on the discrepancy of the projections simulated by the flies with the actual pattern received by the sensors. The reconstructed radioactive concentration is derived from the population of flies, i.e. a collection of points in the 3-D Euclidean space, after convergence. 'Good' flies were previously binned into voxels. In this paper, we study which flies to include in the final solution and how this information can be sampled to provide more accurate datasets in a reduced computation time. We investigate the use of density fields, based on Metaballs and on Gaussian functions respectively, to obtain a realistic output. The spread of each Gaussian kernel is modulated in function of the corresponding fly fitness. The resulting volumes are compared with previous work in terms of normalised-cross correlation. In our test-cases, data fidelity increases by more than 10% when density fields are used instead of binning. Our method also provides reconstructions comparable to those obtained using well-established techniques used in medicine (filtered back-projection and ordered subset expectation-maximisation)

Keywords: Fly algorithm, Evolutionary computation, tomography, reconstruction algorithms, iterative algorithms, inverse problems, iterative reconstruction, co-operative co-evolution.

1 Introduction

This research deals with tomographic reconstruction in nuclear medicine, more particularly positron emission tomography (PET). An unknown radioactive concentration (f) is recovered from known observations ($Y = P[f]$) by solving an ill-posed inverse problem (see Figure 1). In this paper, we exploit the output of a Cooperative Co-evolution algorithm (CCEA), the Fly Algorithm, based on the Parisian Approach to improve quantitative results in reconstructed images. The Fly Algorithm is an evolutionary 3-D image analysis method, which was first developed in the context of robot vision applications [13]. It evolves a population of 'flies', according to the Evolutionary Strategy paradigm. A fly is a 3-D point in the object space. Each fly is used to create projection data (the way this data is generated is problem dependant). The fitness value of each fly, i.e. a quality measurement optimised by the algorithm, is based on the consistency of its calculated projections in the images. This approach has been extended to 3-D tomographic reconstruction in medical imaging ('medical flies') [6, 20, 21, 22].

The problem we address in this paper is related to the extraction of the solution for PET reconstruction using the Fly algorithm (see the last main step in Figure 1). The manuscript particularly focuses on how to voxelise and display the point cloud generated by the final fly population to produce quantatively accurate results. This step is necessary as most medical imaging software use this data representation to store and process tomographic volumes. The voxelisation problem has been overlooked in the previous work mentioned above. The 3-D space was discretised into a regular grid of volume elements (voxels). The intensity of each voxel was proportional to the number of flies located into them. To produce sharper images, only good flies

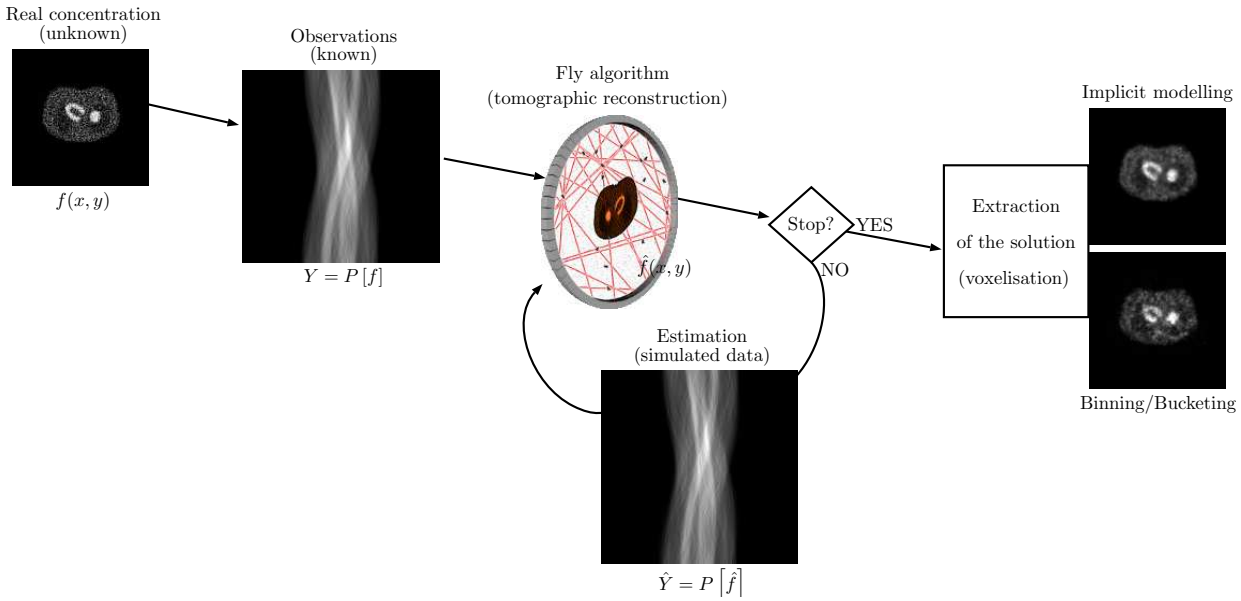


Figure 1: Evolutionary reconstruction using the Fly algorithm. The real radioactive concentration (f) is unknown. P is a projection operator. The projections of f are the known observations ($Y = P[f]$). Individuals of the evolutionary algorithm correspond to 3-D points. The population corresponds to an estimated radioactive concentration $\left(\hat{f}\right)$. Each individual has its own projection data. Together, they produce simulate projections $\left(\hat{Y} = P\left[\hat{f}\right]\right)$. The position of individuals is iteratively optimised using genetic operators to minimise $\mathcal{E}\left(Y, \hat{Y}\right)$ the difference between Y and $\hat{Y}$. After convergence the concentration of individuals is an estimate of the radioactive concentration.

were considered during the voxelisation. In this paper, we study the collective impact of including 'marginally negative' individuals in the final solution and compare evolutionary reconstructions with those obtained using classical algorithms. The aim is to formally determine which flies are necessary in the final solution to produce a volume that is more similar to the real radioactive concentration. We also investigate and compare methods based on implicit modelling to display the fly population in order to properly minimise the difference between the reconstructed pattern and the ground-truth, and get visually realistic rendering. Our method takes advantage of the Fly Algorithm's internal data to efficiently employ implicit modelling. In particular, we investigate how to take advantage of the individual knowledge of each fly after the evolution process to modulate their own appearance in the final image of the whole population. The aim is to reduce the noise level and to retain edges between regions. To ascertain the validity of our new approach, evolutionary reconstructions on two numerical phantoms are eventually compared with those obtained using two of the most popular tomographic reconstruction algorithms in nuclear medicine research, namely filtered backprojection (FBP) and ordered-subset expectation-maximisation (OSEM).

The next section presents the background information necessary to apprehend this paper. It focuses on classical tomographic reconstruction (see Section 2.1), Parisian Evolution (see Section 2.2), and early work on evolutionary reconstruction (see Section 2.3). Section 3 studies how to extract the solution of the optimisation problem in this co-operative co-evolution scheme, i.e. which individuals should be considered during the voxelisation. Section 4 focuses on a technique called Implicit Modelling. An overview is given in Section 4.1. Section 4.2 shows how to apply this technique during the voxelisation and Section 4.3 shows how to exploit the marginal fitness of each fly during the final voxelisation to produce high fidelity reconstructed volumes. In Section 5, we analyze the results obtained using controlled test cases of increasing complexity. We compare the evolutionary reconstruction with classical algorithms used in the nuclear medicine community (FBP and OSEM). The final section provides a discussion and concluding remarks.

2 Background

2.1 Tomographic Reconstruction in PET

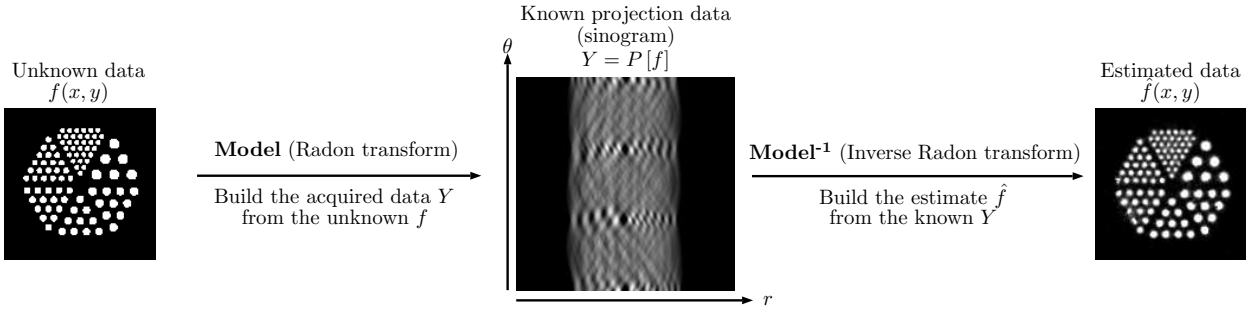


Figure 2: Tomography principle. In emission tomography $f(x, y)$ is an unknown radioactive concentration. Y corresponds to the projections measured by the scanner. It represents integral quantities along straight lines at different angles. Y is inverted analytically (e.g. with FBP) or iteratively (e.g. with MLEM, OSEM or the Fly algorithm): It is the tomographic reconstruction.

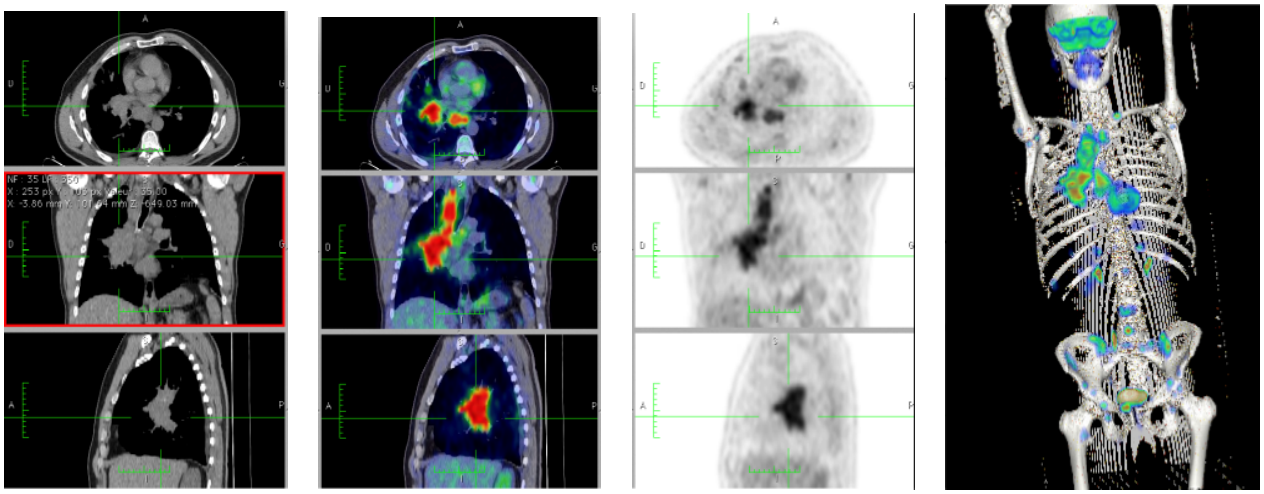
Tomography begins with a multi-angular data acquisition followed by a mathematical reconstruction (see Figure 2). Tomographic reconstruction is an inverse problem that aims at producing cross-sectional images from projection data at successive angles [9]. Various tomography techniques exist in medicine, mainly with modalities using radiations. They include computed tomography (CT), cone-beam computed tomography (CBCT), single-photon emission computed tomography (SPECT), and PET. This paper focuses on PET reconstruction. CT is intensively used in radiology department for examination of any part of the body. It provides anatomical information. The voxel intensity in Figure 3a corresponds to the X-ray attenuation property of tissues (in black air, in grey soft tissues, and in white bones). CBCT is a more recent 3-D technique providing a high spatial resolution useful in radiotherapy or for head examination in dentistry. SPECT and PET are specific to nuclear medicine departments. They are called emission tomography (ET). In this case the source is made of radioactive molecules located within the patient, by means of injection, and rarely by inhalation or ingestion. The reconstruction process aims at providing an estimation of the radioactive distribution within the patient in relation to the uptake of those radioactive molecules and depending on a physiological process (e.g. tumour gross or bone fracture). Figure 3c is displayed in negative: white for low radioactivity concentration and black for high radioactivity concentration. Reconstructed images have a much lower resolution and a poorer signal-to-noise ratio (SNR) than CT and CBCT. SPECT and PET are often combined with CT to provide physiological information co-located with anatomical information. In Figures 3b and 3d the PET data is overlaid in colour onto the CT data. In this example, the red colour of the lookup table (LUT) corresponds to a high concentration and green to a medium concentration. Low concentration is transparent and does not appear.

After obtaining many projections at different angles, the image system produces an image called "sinogram". It is a visual representation of the Radon transform produced by the concatenation of the successive projections (see middle image in Figure 2). The projections correspond to integral quantities along straight lines from the source to the detector.

Tomographic reconstruction is a mathematical process that consists in inversing the Radon transform by back-projecting the measurement data into the object space. This process is an inverse problem, with ill-conditioned data due to the noise affecting those data (noise is a major concern in emission tomography). It is usual to describe two classes of reconstruction algorithms:

- Analytic reconstruction methods,
- Iterative reconstruction methods (including algebraic based methods and statistical based methods).

Analytic reconstruction is the mainstream one in CT, while statistical reconstruction has become a standard in emission tomography. Analytic reconstruction methods are based on continuous modelling. They inverse



(a) CT. *(b)* PET+CT. *(c)* PET. *(d)* Visualisation.

Figure 3: PET-CT scan of the lungs: CT provides anatomical information, PET physiological information. CT and PET are complementary. They are often displayed side-by-side as well as superimposed. The gross of a tumour, black mark in (c), is highlighted in red in (b). Images created using Osirix [16].

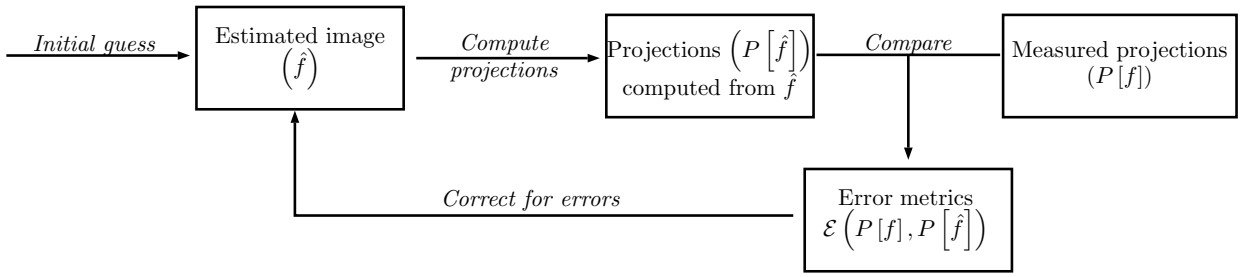


Figure 4: Iterative algorithm principles: Projection data $\left(P\left[\hat{f}\right]\right)$ is generated from an estimated image $(\hat{f})$. It is iteratively corrected to minimise the discrepencies between its projections and the known observations $(P[f])$.

the Radon transform using the Fourier slice theorem: the 1-D Fourier transform of a projection is equal to a slice of the 2-D Fourier transform of the original image. The complete 2-D Fourier transform of the image is reconstructed from these 1-D Fourier transforms. Then, the image is obtained by inverting its Fourier transform. The most frequently used method is the filtered backprojection (FBP) algorithm. Before the back-projection, the sinogram is initially filtered in the Fourier domain to produce sharper images.

Iterative reconstruction techniques are based on iterative correction algorithms. They usually follow the general scheme presented in Figure 4. It consists in estimating a new image $(\hat{f})$ at a certain step by combining the image estimated at the previous step and the projection data $\left(P\left[\hat{f}\right]\right)$ generated from this estimate to minimise the error between these estimated projections and the projections measured by the scanner $(P[f])$. The process is repeated until a given criterion is satisfied.

In SPECT and PET, maximum-likelihood expectation-maximisation (MLEM) [17] and its derivatives are now more popular than the analytic reconstruction methods [15]. The main reason is that they take into account Poisson noise in the measured photon count. MLEM is known to be a very slow algorithm, and alternatives have been proposed to obviate this drawback. This is why OSEM has become the reference reconstruction method in PET [8]. Its principle is to reduce the amount of projections used at each iteration of the EM algorithm by dividing the projections in K sub-groups. One of the main issues with MLEM and its derivatives, including OSEM, is the difficulty to chose a good stopping criterion [4]: they are known to be

relying on a non-converging algorithm. MLEM provides an acceptable estimation of tracer distribution with a few iterations. Then, when the number of iterations increases the reconstruction offers a better resolution but becomes noisier.

2.2 Parisian Approach and Fly Algorithm

Our evolutionary reconstruction algorithm follows the iterative scheme presented above. Our method heavily relies on the Fly algorithm, which follows the Parisian Approach. Whereas classical evolutionary algorithms (EAs) evolve a population in order to keep only the best individual of the final population as the optimal solution of the problem, in the Parisian Approach the solution is the whole population itself. We thus refer to the Cooperative Co-evolution (CoCo) family of algorithms. The aim is to take advantage of an EA to help its best individual of the population towards the global optimum and the other individuals into attractive areas of the search space [7]. The population is considered as a society of individuals who are collaborating to build something, whatever it may be. To achieve this goal the fitness landscape is designed so that the solution to the optimisation problem is given by a set of individuals or the whole population.

A Parisian EA usually contains all the usual components of an EA (e.g. selection, mutation, recombination, etc.) and includes some optional ingredients:

Global fitness function: calculated on the whole population.

Local fitness function(s): measuring each individual's contribution to the global solution.

Diversity mechanism: To avoid solutions where most individuals gather in a few areas of the search space.

A special case of Parisian EA is the Fly algorithm. It was initially developed for 3-D image analysis in the context of robot vision applications [13]. While classical stereo-vision algorithms build a 3-D representation of the scene by matching 2-D primitives previously extracted from image segmentation, the Fly Algorithm works directly in the 3-D space. A fly is relatively simple: It is a 3-D point in the object space. A population of 'flies' evolves using the repetitive application of genetic operators according to the Evolutionary Strategy scheme. Each fly is used to create projection data in a way that depends on the problem being solved. The fitness value of each fly is a quality measurement optimised by the algorithm. It is usually based on the consistency of its calculated projections in the images.

CoCo and particle swarm optimisation (PSO) share many similarities. PSO is inspired by the social behaviour of bird flocking or fish schooling [10, 18], where every particle somehow follows its own random path biased toward the best particle of the swarm. Both are search methods that start with a set of random solutions, which are iteratively corrected toward a global optimum. In the Fly algorithm, the solution of the optimisation problem is the population (or a subset of the population): The flies collaborate to build the solution. In PSO the solution is a single particle, the one with the best fitness. A typical volume in nuclear medicine includes $128 \times 128 \times 128$ voxels. In other medical modalities, it is not uncommon to have volumes of $512 \times 512 \times 512$ voxels. The final solution is therefore an array of several millions of numerical values. This is an extremely complex search space that cannot be solved in an acceptable computing time by a classic EA or PSO. Also small regions of low, medium or high concentrations should not be missed in the reconstruction. It is therefore necessary to keep a level of diversity and ensure that the new individuals are not forced to follow the best one(s). Having diversity built-in and having many individuals who collaborate to build the estimate of the radioactive concentration are therefore attractive features of the Fly algorithm for the tomographic reconstruction problem. Due to the intrinsic following behaviour of the particles toward the current best solution, it is not practical to adapt PSO for tomographic reconstruction.

2.3 Early Evolutionary Reconstruction

The Fly algorithm has been extended more recently to 3-D reconstruction in medical imaging ('medical flies') [6, 20, 21, 22]. Figure 5 illustrates how the Fly algorithm works in this case. Here, each fly emulates a radioactive emitter and has its own illumination pattern. The projection data they create can be stored as a sinogram. It is 2-D image made of a set of 1-D projections at successive angles. Figure 6 shows how the flies are orthogonally projected to generate parts of a sinogram. The data created by all the flies $\left(\hat{f}\right)$ is

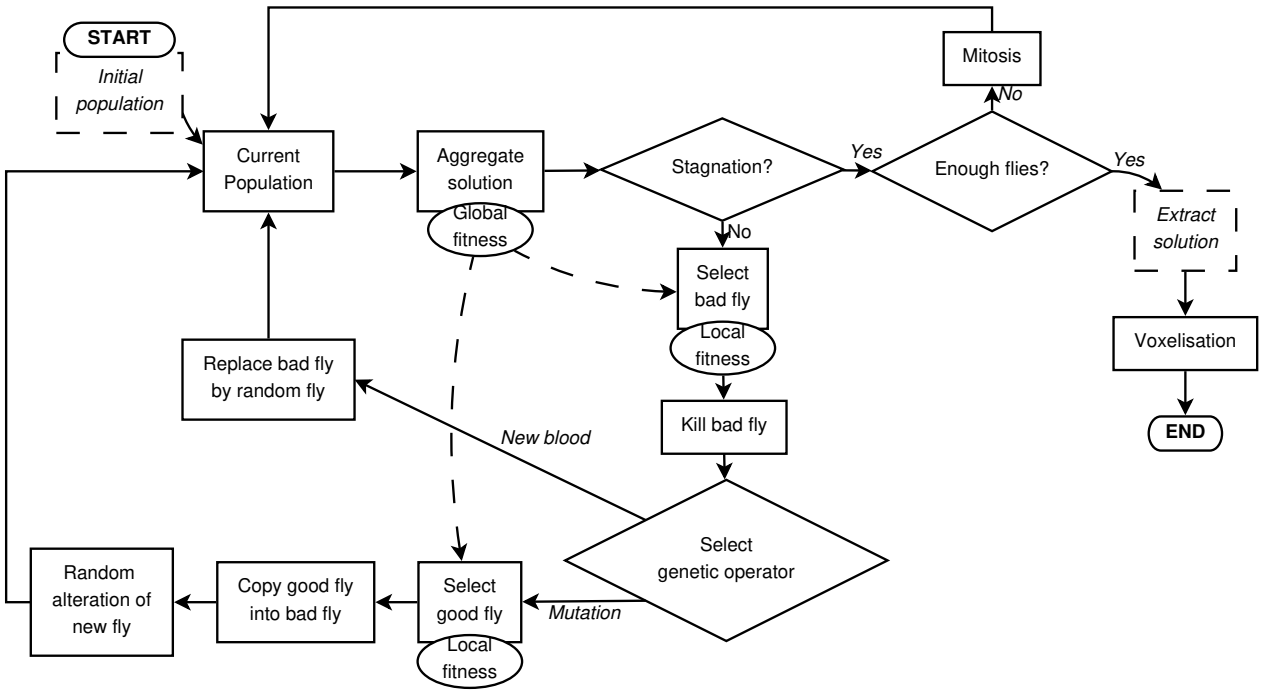


Figure 5: The Fly algorithm is an iterative method (as described in Figure 4). Here genetic operators (new blood and mutations) are applied to correct the position of flies and minimise the error between the known observations ($P[f]$) and the projection data ($P\left[\hat{f}\right]$) generated by the population of flies ($\hat{f}$). After convergence the concentration of flies is an estimate of the radioactive concentration.

aggregated to build the sinogram simulated by the population $\left(P\left[\hat{f}\right]\right)$. It is used to compute a metrics useful in the calculation of the fitness functions. At the start of the reconstruction, flies are randomly scattered in the search space (see *Initial population* in Figure 5). The evolutionary algorithm optimises their positions to minimise the global fitness function. After convergence, the sum of illumination patterns of all the flies closely matches the input data and the final population of flies gives an estimate of the unknown radioactive concentration (f).

To optimise the flies' position, our algorithm relies on mutation and new blood operators (see *Select genetic operator* in Figure 5). No cross-over is used as it is not particularly useful in the Fly algorithm for image reconstruction. Let us consider two flies that are well positioned. If there are in two separate areas, then it does not make sense to create a new fly in between as it would probably lead to a bad fly. The new blood operator aims at maintaining some diversity in the population. This is particularly important at the early stages of the reconstruction to make sure no small object of low intensity are missed. Its principle is relatively simple (see *Replace bad fly by random fly* in Figure 5):

1. a bad fly is randomly selected using our Threshold selection (see below for an explanation)
2. its projections are removed from $P\left[\hat{f}\right]$
3. the fly is replaced a new fly randomly positioned in the search space, and
4. the projections of the new fly are added to $P\left[\hat{f}\right]$.

For mutation (see *Copy good fly into bad fly* and *Random alteration of new fly* in Figure 5),

1. a bad fly is selected using the Threshold selection,
2. its projections are removed from $P\left[\hat{f}\right]$,

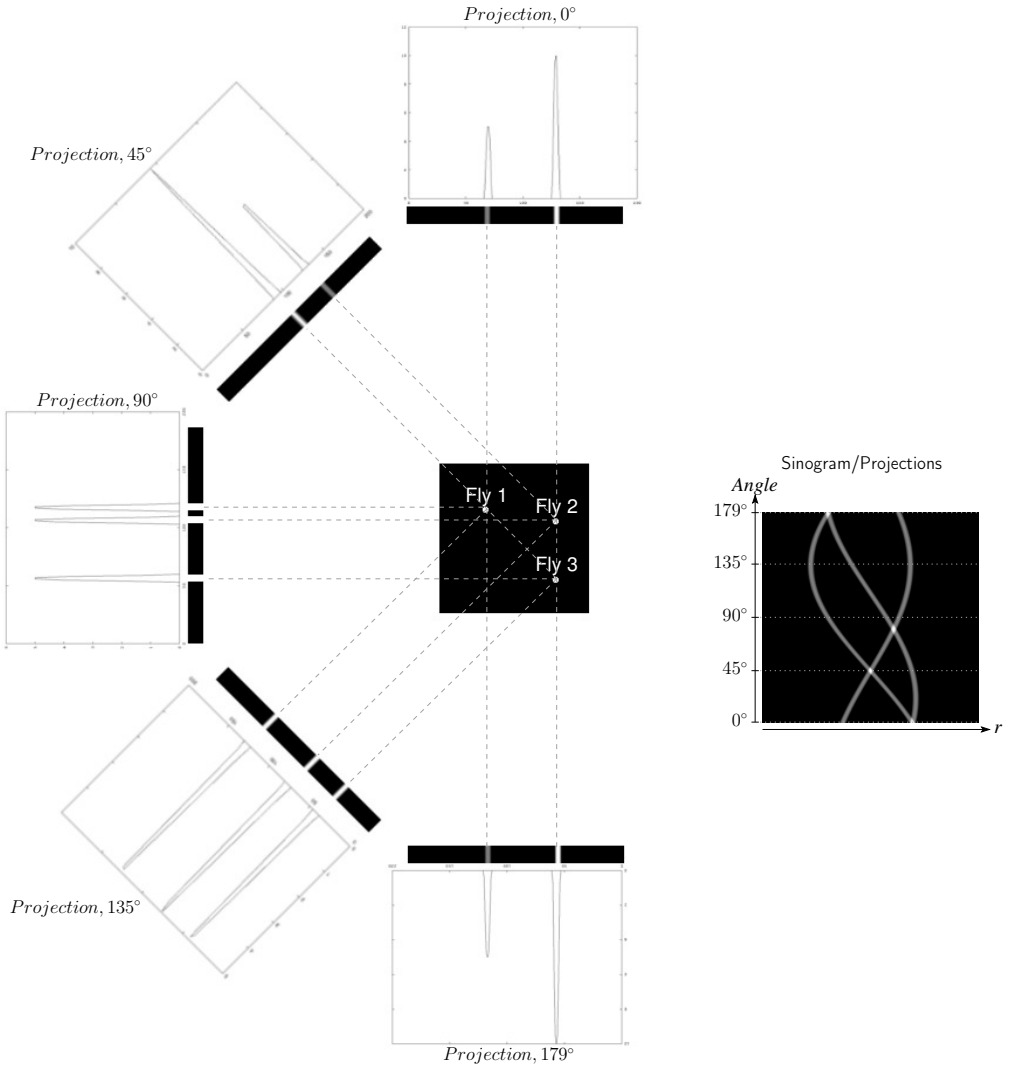


Figure 6: From a population of flies ($\hat{f}$) to an estimated sinogram $\left(P\left[\hat{f}\right]\right)$. Each fly has it own projection data at different angles. Put together, they produce the estimated sinogram.

3. a good fly is selected using the Threshold selection,
4. the bad fly is replaced by the good fly,
5. all the genes of the newly created fly are altered by some small random changes, and
6. the projections of the mutated fly are added to $P\left[\hat{f}\right]$.

To control the amount of random changes during the mutation, we favour adaptive schemes to limit user inputs. In our implementation the probability of all the operators is encoded in the genome of each fly. In our previous work, only one type of mutation operator was used, the Dual-mutation [22]. Here we have several types of mutation operators:

Basic: A mutation variance is included in the genome of each fly. At the beginning of the reconstruction, all the flies have the same value. The mutation variance is subject to an adaptive pressure itself and is self-adapted [2, 3].

Dual: In [22], we proposed to use two alternative mutation variances, one being greater than the other one. The algorithm keeps track of which variance provides the best results in term of global fitness. At

regular intervals, the performance of each variance is assessed. If the biggest value is the best, then both variances are multiplied a constant value. If the smallest value is the best, then both variances are divided the constant value.

Adaptive: In [23], we use stress mutation to take into account the local fitness value of the individual being mutated. When the local fitness value of the fly is low, the amount of changes is large to favour exploration. When the local fitness value of the fly is high, the amount of changes is small to refine the results.

To evaluate the performance of successive populations toward more realistic images, we use a distance measurement ($\mathcal{E}$) between global images $\left(P\left[\hat{f}\right]\right)$ resulting from the photon emissions of the population of flies $\left(\hat{f}\right)$ and the real images ($Y = P[f]$) obtained from the sensors. It is the global fitness of the population (see *Aggregate solution* in Figure 5):

$$fitness = \mathcal{E}\left(\hat{f}\right) = \frac{1}{w \times h}\left\|Y - P\left[\hat{f}\right]\right\|_2^2 \tag{1}$$

with $\mathcal{E}$ based on the ℓ^2-norm (also known as the Euclidean distance) between the observations (Y) and the data simulated $\left(P\left[\hat{f}\right]\right)$ by the flies; w and h the number of pixels in Y and $P\left[\hat{f}\right]$ along the horizontal and vertical axis respectively; $\hat{f}$ the fly population (i.e. an estimate of the unknown f), which corresponds to the population of flies; P is the projection operator, which projects flies to simulate an estimate of Y. The algorithm minimises the fitness as follows:

$$\hat{f} = \underset{f \in \mathbb{R}^2}{\arg\min}\left(\frac{1}{w \times h}\left\|Y - P\left[\hat{f}\right]\right\|_2^2\right) \tag{2}$$

$\mathcal{E}$ measures the discrepancies between the observations and the data simulated from flies. Lower values of $\mathcal{E}$ correspond to lower errors in the data simulated by the flies, i.e. an image of the population $\hat{f}$ that better matches the observations.

The fitness of each individual fly is calculated as its (positive or negative) contribution to the collective fitness of the population (which is called 'marginal fitness evaluation') [6]. It is based on the "leave-one-out-cross-validation" principle. In practice, we measure the population's performance twice: once taking into account all the individuals (i.e. the global fitness); and once without the fly (i) that is being assessed. The two values are then compared. This way, we know if a fly helps the population improve its performance or not:

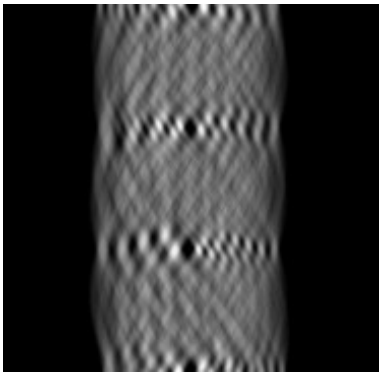
$$F_m(i) = \mathcal{E}\left(\hat{f} - \{i\}\right) - \mathcal{E}\left(\hat{f}\right) \tag{3}$$

where: $F_m(i)$ the marginal fitness of Fly i, $\left(\hat{f} - \{i\}\right)$ is the population without Fly i. The marginal fitness makes it possible to detect if a fly is positively or negatively contributing to the population's performance:

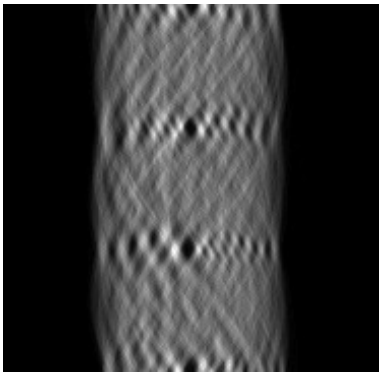
- If the local fitness is positive, then the fly improves the population;
- If it is negative, then the fly reduces the population's performance;
- If it is null, then the fly has no impact on the population's performance.

The population of flies is then evolved based on this marginal fitness calculation using a steady state evolution strategy [20], in which at each loop one individual (fly) has to be eliminated and replaced with a new fly. In the evolution process, flies are picked up randomly. We developed a specific selection process called 'Threshold Selection' [21] that matches the definition of marginal fitness: If the fly's fitness is above the threshold, the fly will survive; otherwise, it will die and be replaced using the genetic operators (mutation, or new blood). The selection process gradually eliminates flies with a negative fitness (see *Kill bad fly* in Figure 5). As this process tends to generate more and more good flies (see *Copy good fly into bad fly* in Figure 5), there are less and less 'bad' flies to replace: The Threshold Selection will get stuck when the algorithm converges ('stagnation'). This provides an excellent stopping criterion.

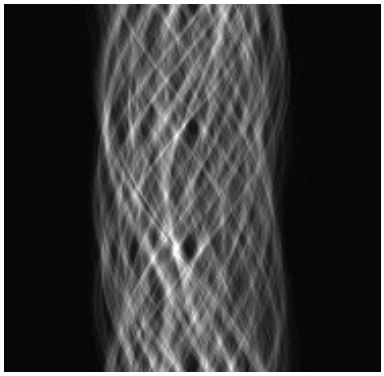
We are using this feature to introduce progressive multi-resolution processing [22] (see *Enough flies?* in Figure 5). The evolution starts with a relatively low number of flies (see *Mitosis* in Figure 5). Each time



(a) Known observations ($Y = P[f]$).



(b) Estimation with all the flies $\left(P\left[\hat{f}\right]\right)$ (NCC with (a): 99.19%).



(c) Estimation with the good flies only $\left(P\left[\hat{f}^{+}\right]\right)$ (NCC with (a): 90.97%).

Figure 7: Sinograms with 185 pixels per projection, 1st angle: 0°, angular step: 1°, and last angle: 179°. Corresponding radioactive concentrations are given in Figure 8. The geometrical relationship link between the simulated sinogram $\left(P\left[\hat{f}\right]\right)$ from the position of flies ($\hat{f}$) is given in Figure 6.

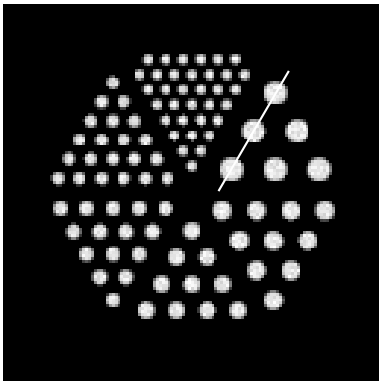
stagnation is detected, evolution is paused and a mitosis process is launched. Similar to biological mitosis, for each fly a new fly is created by mutation and added to the population: This doubles the population size. Evolution eventually resumes after the mitosis. The whole evolution-mitosis process is stopped when after two successive mitosis the global fitness does not improve anymore.

Once the optimisation loop ends, the solution has to be extracted and encoded (see *Extract solution* and *Voxelisation* in Figure 5). Contrary to mainstream tomographic reconstruction algorithms whose output is a 3-D rectilinear grid of voxels, our Fly-based approach delivers a set of 3-D points as an output. Deciding which type of representation is more legible to the user is another story. [19] shows the advantages of a representation based on discrete 3-D points. In order to enable a genuine (/trusty) comparison between the outputs of the Fly algorithms and mainstream methods, we show in this paper how to do the opposite way and build a continuous representation of the Fly output. It also makes it possible to use a multitude of image processing and visualisation tools developed for voxelised data.

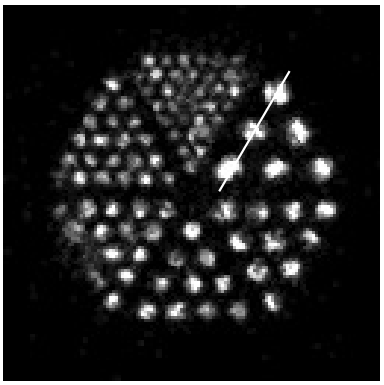
To date, only the sub-population of flies with a positive fitness $\left(\hat{f}^{+}\right)$ was taken into account to build the final estimate of the distribution of 3-D points [21]. It is sampled into voxels. Data binning, also called bucketing, has been used so far to produce the voxel map. The 3-D space is divided into a regular grid. Each element of the grid is called a voxel. With data binning, the value of a voxel is given by the number of flies that it contains. In this paper, we study which flies have to be included in the final result to improve accuracy, and how to best voxelise the fly data using implicit modelling to further improve quantitative results.

3 Extraction of the Solution

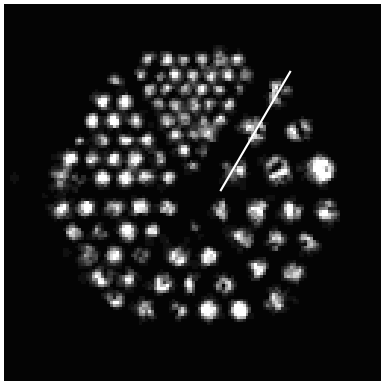
Traditionally, the answer to an optimisation problem modelled using artificial evolution is the best individual of the whole population after convergence. Using the cooperative co-evolution scheme of the Fly algorithm, the solution of the optimisation problem is embedded within the population [13]. We use tomographic reconstruction as an application example of the Fly algorithm but other applications could be considered. As a proof-of-concept, below we will consider the 2-D case only. However, note that the algorithm is actually developed for 3-D and the notations can be extended to account for the Z-dimension. Figure 7a shows the input data. It is the known observations stored as a sinogram. The projection operator P is designed to project the population of flies $\left(\hat{f}\right)$ in order to simulate the sinogram $\left(P\left[\hat{f}\right]\right)$ according to the illustration in Figure 6. Figure 7b shows the corresponding simulated sinogram after convergence of 12,800 flies. Figure 7c shows the sinogram generated by flies $\left(\hat{f}^{+}\right)$ with a positive fitness only. We can observe that the sinogram produced by all the flies is visually closer to the ground-thruth than the sinogram simulated by good flies



(a) Ground-truth (f). It is unknown.



(b) Concentration estimation with all the flies $\left(\hat{f}\right)$ (NCC with (a): 82.74%).



(c) Concentration estimation with the good flies only $\left(\hat{f}^{+}\right)$ (NCC with (a): 82.24%).

Figure 8: Tomographic reconstruction using 12,800 flies. Corresponding sinograms are given in Figure 7. The geometrical relationship link between the position of flies ($\hat{f}$) and the simulated sinogram ($P\left[\hat{f}\right]$) is illustrated in Figure 6.

only.

To measure the level of similarity between two images I_1 and I_2, we use the normalised cross-correlation (NCC):

$$NCC(I_1, I_2) = \frac{1}{m \times n} \sum_{i=0}^{i<m} \sum_{j=0}^{j<n} \frac{\left(I_1(i,j) - \bar{I_1}\right)\left(I_2(i,j) - \bar{I_2}\right)}{\sigma_1 \sigma_2} \tag{4}$$

with $\bar{I_1}$ and $\bar{I_2}$ the average values of all the pixels in I_1 and I_2 respectively, such as

$$\bar{I} = \frac{1}{m \times n} \sum_{i=0}^{i<m} \sum_{j=0}^{j<n} \left[I(i,j)\right] \tag{5}$$

and σ_1 and σ_2 the standard deviations of all the pixel values in I_1 and I_2 respectively, such as:

$$\sigma = \sqrt{\frac{1}{m \times n} \sum_{i=0}^{i<m} \sum_{j=0}^{j<n} \left[I(i,j) - \bar{I}\right]^2} \tag{6}$$

Due to the stochastic nature of the algorithm, 15 evolutionary reconstructions have been performed in total to provide statistically meaningful results. The image simulated using both good and bad flies leads to a NCC of 99.27% ±0.06%. The image simulated by the good flies only has a NCC of 91.40% ±0.75%. In other words, keeping the marginally negative flies leads to more accurate and more stable results.

The concentration of flies is then sampled into voxels to generate the tomography volume. For data binning, we considered a fly as a Dirac delta function (δ): The value of each voxel is incremented for each fly it contains. Figure 8 shows the ground-truth image and the corresponding reconstructions (qualitative validation). Figure 9 presents the NCC between the ground-truth and the reconstructed images (quantitative validation). Both figures complement each other and show that reconstructions including flies with negative fitness generally produce images that are visually and numerically closer to the ground-truth. In past papers, we usually kept good flies only as it resulted into visually sharper reconstructed images.

In order to measure how sharp images are, we compute the sum of gradient magnitudes for each reconstruction:

$$Sharpness(I) = \sum_{i=0}^{w-1} \sum_{j=0}^{h-1} \left|\nabla I\right|(i,j) \tag{7}$$

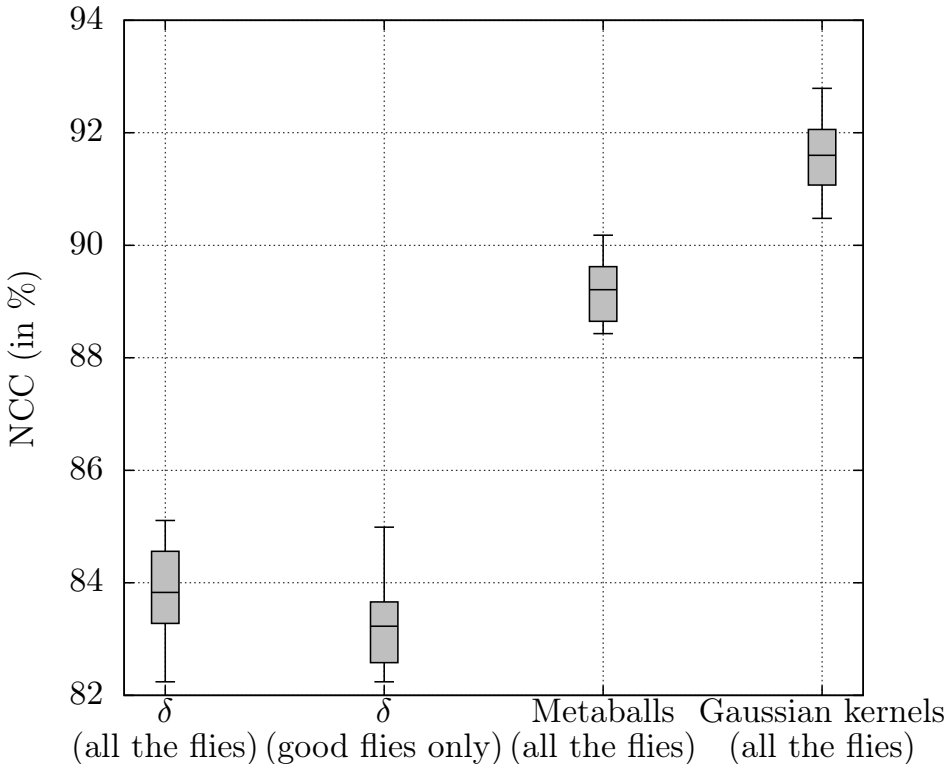


Figure 9: Similarity metrics (NCC) between the ground-truth (f) and the images of the fly population ($\hat{f}$) using different voxelisation methods. Due to the stochastic nature of the evolutionary reconstruction, the reconstruction is performed 15 times for each voxelisation method to produce statistically meaningful results.

It is 4120 ±320 with good flies only; 4088 ±497 with all the flies. Removing negative flies leads to sharper reconstructions. Note that we use other metrics below to ascertain this assumption as Eq. 7 is sensitive to noise.

Removing all the flies with a negative fitness is wrong as the Fly algorithm is based on a co-operative scheme. When a fly is killed, i) its contribution to the population is removed, ii) the global fitness changes, iii) which also modifies the local fitness of every other fly. In other words, when any fly is killed, a good fly may become bad, and *vice versa*, a bad fly may become good. Because of this phenomenon, we can eventually say that bad flies have to be included in the final solution. This is why the NCC of the whole population (including bad and good flies) is better on average than the sub-population of good flies only.

Figure 10 corresponds to profiles (also known as intensity profiles) extracted from white lines in Figure 8. In the imaging context, a profile corresponds to a set of intensity values taken from regularly spaced points along an arbitrary line segment within the image. It is often plotted as a 1-D function. We quantify how steep edges are using the rise time from 10% to 90% and fall time from 90% to 10%. These metrics assess how many pixels are required to change from the minimum to maximum values and *vice versa*. We limit the values to 10% to 90% of the whole interval to account for noise. The values for each edge of every profile are summarised in Table 1. Edges are usually steeper for the ground-truth and the reconstruction using the good flies only. The transition from extreme values is twice as large for the reconstruction with all the flies than the ground-truth.

In addition we check the total sum of values of each profile. It is: 29.33 for the ground-truth; 29.09 ±4.34 for all the flies; 21.88 ±4.34 for the good flies only. It indicates that there is approximately as much information in the profiles of the ground-truth and all the flies, and some information is missing in the profile of the good flies only.

The results above can be summarised as follows:

1. Edges are much more blurred when bad flies are included.

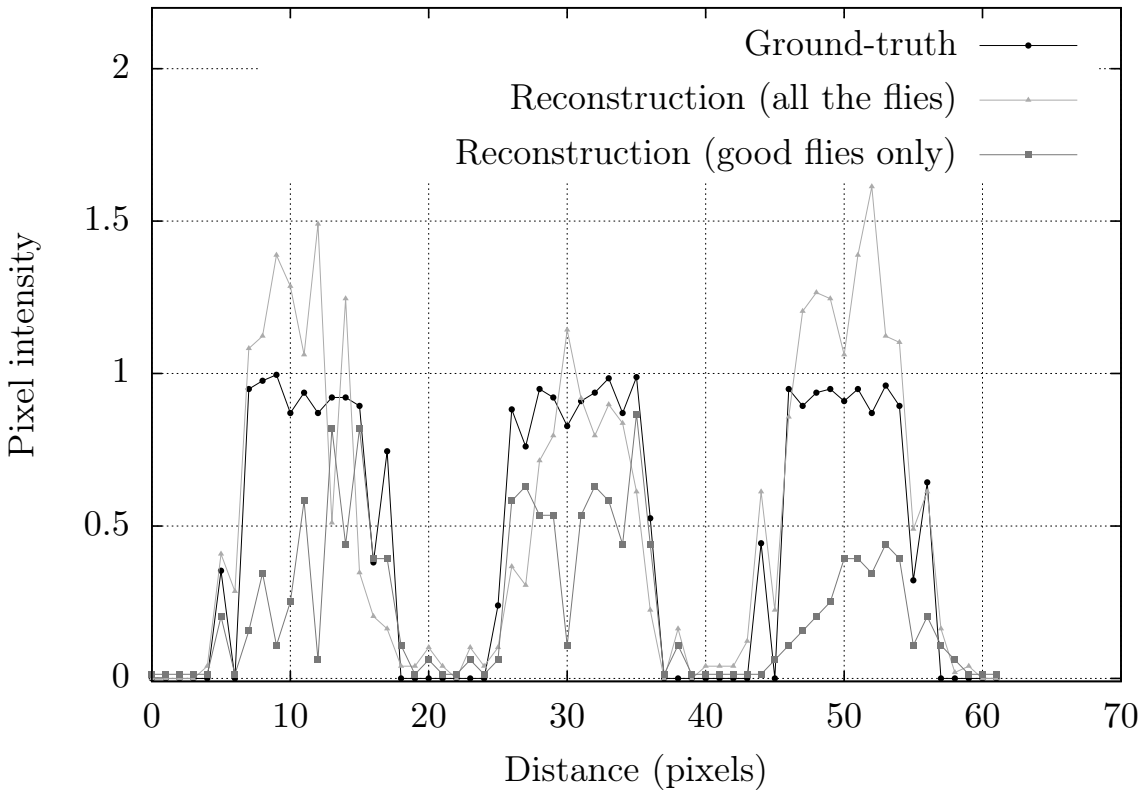


Figure 10: Intensity profiles corresponding to the white lines in Figures 8a, 8b and 8c.

2. Gaps appear when bad flies are not included.

It could be seen as a dilemma,

- Bad flies should be excluded to preserve edges;
- Bad flies should be included to avoid holes in the data.

In the next section of this paper, we demonstrate how to solve this dilemma: How to include bad flies to produce a more accurate image, whilst still retaining its sharpness.

4 Voxelisation using Implicit Modelling

We saw in the previous section that the solution that is extracted should contain all the flies, including the ones with a negative marginal fitness. Previously the contribution of each fly in the final volume was one and it was assigned to a single voxel. It could lead to noise. It is not uncommon to post-process tomographic images with a low-pass convolution filter. However, with our pixel/voxel-less approach it is possible to use the internal data of the Fly algorithm, here the flies' position, to remove the need for a smoothing filter. Below we demonstrate how to spread this contribution over several voxels using implicit modelling.

4.1 Definition

It is a computer graphics (CG) technique used to defined the surface of geometric objects using control primitives (e.g. points or line segments) and a few equations [5]. Blobby Molecules, Metaballs and Soft Objects are well known types of implicit modelling techniques. In computer graphics, it consists of the steps below:

Table 1: Image and profile comparison between the ground-truth (Figure 8a) and the evolutionary reconstructions (Figures 8b, 8c, 15, and 17). Numerical values in bold characters are the ones closest to the ground-truth.

	NCC	**Sharpness**	**Rise time**	**Fall time**	**$\sum$ Profile**
Ground-truth	N/A%	9443	2.22	2.35	29.33
All the flies	82.74%	3857	4.17	3.44	32.13
Good flies only	82.24%	3931	**3.15**	**3.08**	14.34
Metaballs	89.69%	5150	3.90	3.90	31.96
Gaussian	**92.79%**	**6303**	3.57	3.36	**31.63**

1. Positioning control primitives (usually points or line segments) in the 3-D space;
2. Computing the corresponding density field using a given equation (e.g. Eqs. 8 or 9) (see Figures 11a and 11c);
3. Selecting a threshold value (see Figures 11b and 11d);
4. Reconstructing the isosurface corresponding to the threshold using either raycasting [11] or marching cubes [12] (see Figure 12).

Blobby Molecules [5] uses the electron density distribution of the hydrogen atoms (Gaussian Distribution):

$$f(r) = a\mathrm{e}^{-br^2} \tag{8}$$

b is related to the standard deviation of a Gaussian curve, a is the height of the curve, and r is the distance to the atom centre (see Figure 13).

For Metaballs [14], the density field is modelled using a piecewise function:

$$f(r) = \begin{cases} a\left(1 - \frac{3r^2}{b^2}\right) & \forall r \in [0; b/3] \\ \frac{3a}{2}\left(1 - \frac{r}{b}\right)^2 & \forall r \in [b/3; b] \\ 0 & otherwise \end{cases} \tag{9}$$

Figure 14 illustrates how the three sub-functions from Eq. 9 are combined to produce a smooth falling curve. The influence of the parameters a and b are presented in Fig 13. a is a scaling factor, and b is the maximum distance that a control primitive contributes to the field.

The value of the density field at a any point $[x, y, z]$ is given by:

$$F(x, y, z) = \sum_{i=1}^{N} f(\sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2}) \tag{10}$$

with N the number of control points and $[x_i, y_i, z_i]$ the position of the i-th control point. When two particles are close to each other, their density fields are merging in a smooth manner (see Figure 11c). When they are sufficiently far apart, their density fields stay separated. Evaluating $F(x, y, z)$ becomes computationally expensive when the number of control primitives increases. To limit this effect, the field function in Eq. 9 does not make use of the exponential and it is bounded as $f(r)$ does not contribute much to the field when r increases. It makes density fields using Metaballs faster to compute than those with Blobby molecules.

Other field functions, such as Soft Objects [24], are of course possible but will not be investigated in this paper. We focus here on Blobby Molecules as they rely on Gaussian kernels; and on Metaballs as they are well known in the CG community.

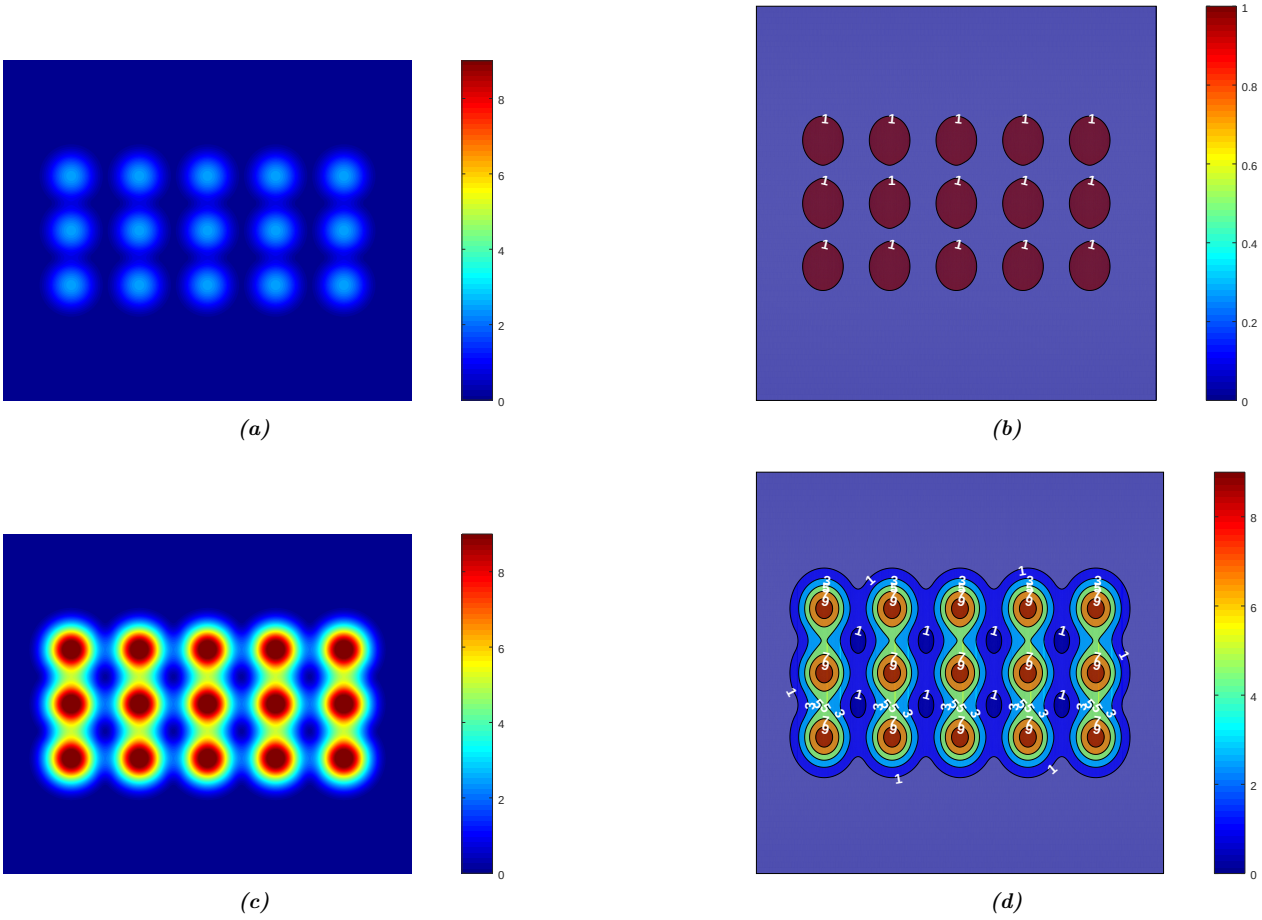


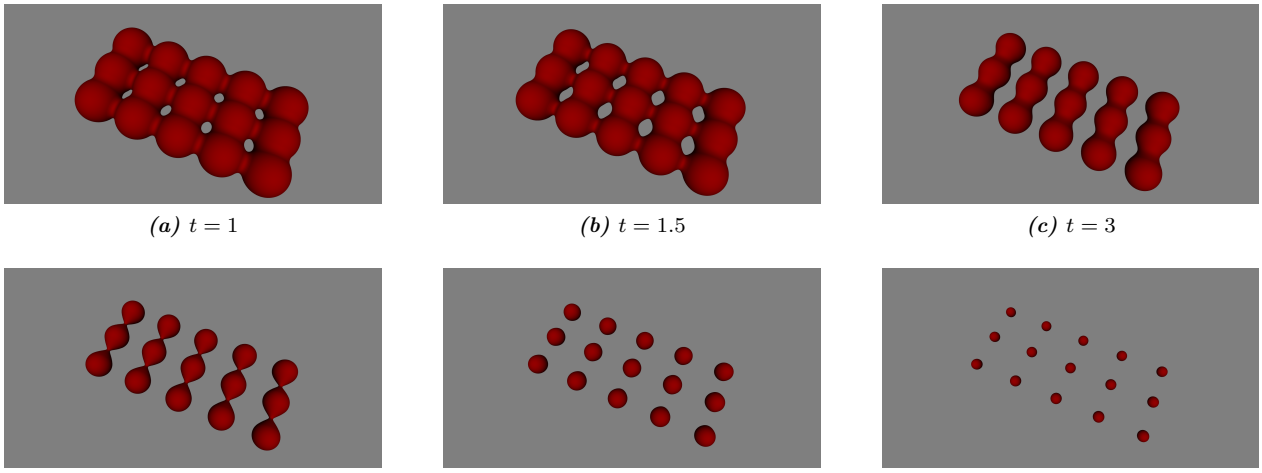
Figure 11: 3-D density field using 15 points as control primitives using Eq. 9. When the particles are in close proximity, their density fields are joining each other smoothly without discontinuity. (a) and (c): cross-sections of the density field at different heights. (b) and (d) corresponding isolines. See Figure 12 for corresponding 3-D isosurfaces.

4.2 Voxelisation using Metaball as Density Field Function

The stochastic nature of the evolutionary algorithm leads to noisy PET volumes (see Figures 8 and 10). To limit noise, i) voxelised volume could be post-processed by a low-pass filter, it would lead to a loss of information, or ii) more flies can be used in the reconstruction, the computing time will significantly increase.

The aim of the Fly algorithm is to estimate the radioactive concentration. As an output it produces a 'point cloud'. This point cloud can be described as a density field. Instead of the δ function, an implicit function ($f(r)$) is used: Here, a fly corresponds to a particle surrounded by a density field. Using Equations 9 and 10, the influence of the particle decreases with the distance from the particle location.

Figure 15 shows the reconstruction results when Metaballs are used. The NCC with the ground-truth is 89.69% in this case. In Figure 9, we can see that the NCC with 12,800 flies used to generate a density field is now much better than using our previous voxelisation method based on binning. Corresponding profiles are in Figure 16. The total sum of values of the profile for metaballs is 31.76 $\pm$3.44, which is relatively close to the corresponding value in the ground-truth (29.33). The average rise time and fall time between 10% and 90% are both 3.9 pixels. It indicates that the sharpness around large objects is still not well recovered. It is because a fly is spread over several voxels. As a consequence, flies leak at edges, which leads to unsharpness. However, the overall sharpness metrics (see Eq. 7) is 4964 $\pm$429. This is because smaller structures are recovered.



(a) $t = 1$ *(b)* $t = 1.5$ *(c)* $t = 3$

(d) $t = 5$ *(e)* $t = 7$ *(f)* $t = 9$

Figure 12: Implicit surfaces corresponding to the density field defined with the 15 metaballs of Figure 11. Triangle meshes are extracted from the density field using the Marching Cubes algorithm [12] with various threshold (t) values.

4.3 Adaptive Gaussian kernels to Exploit the Fly's Individual Knowledge

Although the images produced using Metaballs are quantitatively much better than using a naïve approach based on binning (e.g. noise reduction), the final output could be better if edges between areas of different concentrations could be preserved. In this section, we will used Gaussian kernels instead of Metaballs. For each fly, the spread of the Gaussian function will depend on the fly's marginal fitness. This is because we can consider this numerical value as a level of confidence in the fly's position. We chose Gaussian kernels as the Gaussian function is very well known and it is used for Blobby Molecules [5] (see Eq. 8).

The total contribution of every fly to the final volume is 1. When we were using (δ), each fly was embedded into one voxel only. Using metaballs, it was spread over several voxels. In this case it is not straightforward to normalise the fly's contribution to account for its performance by modulating a and b in Eq. 9 depending on F_m. It would require to compute for each fly:

$$\int_0^b f(r)\,\mathrm{d}r \,\forall\, a \,\&\, b \tag{11}$$

As an alternative, we consider each fly as a Gaussian kernel whose standard deviation is linearly proportional to F_m. The greater F_m, the more accurate the fly position. The lower F_m, the lower the trust in the fly position.

$f_i(r)$ in Eq. 10 becomes:

$$f_i(x, y, z, \sigma_i) = \frac{1}{V_i} \exp\left(-\left(\frac{(x-x_i)^2}{2\sigma_i^2} + \frac{(y-y_i)^2}{2\sigma_i^2} + \frac{(z-z_i)^2}{2\sigma_i^2}\right)\right) \tag{12}$$

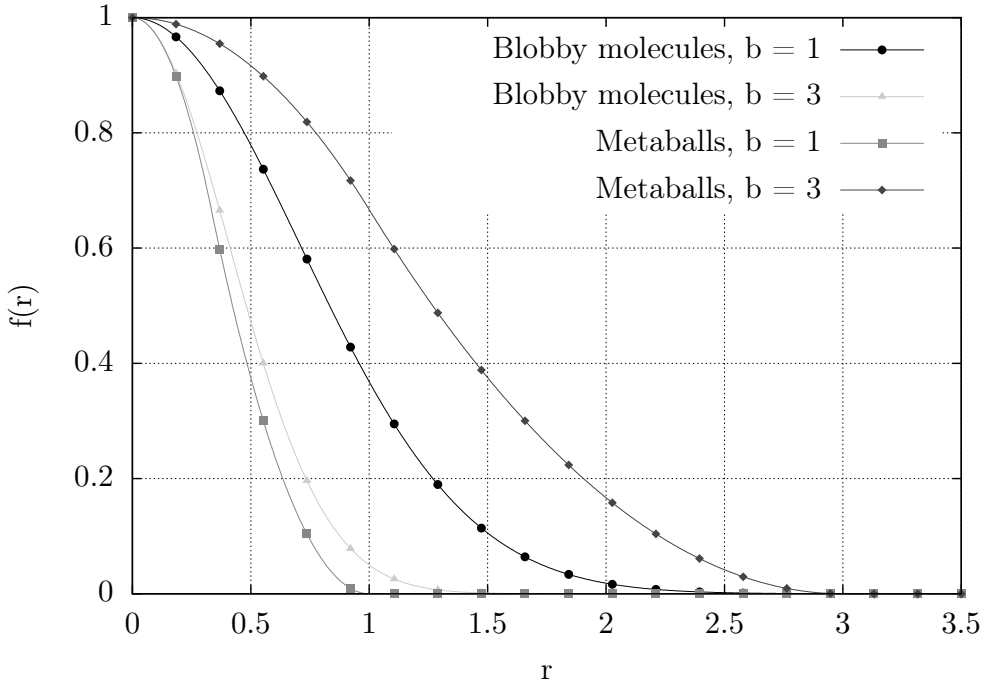
with

$$V_i = 2\pi\sigma_i^3 \times \sqrt{2\pi} \tag{13}$$

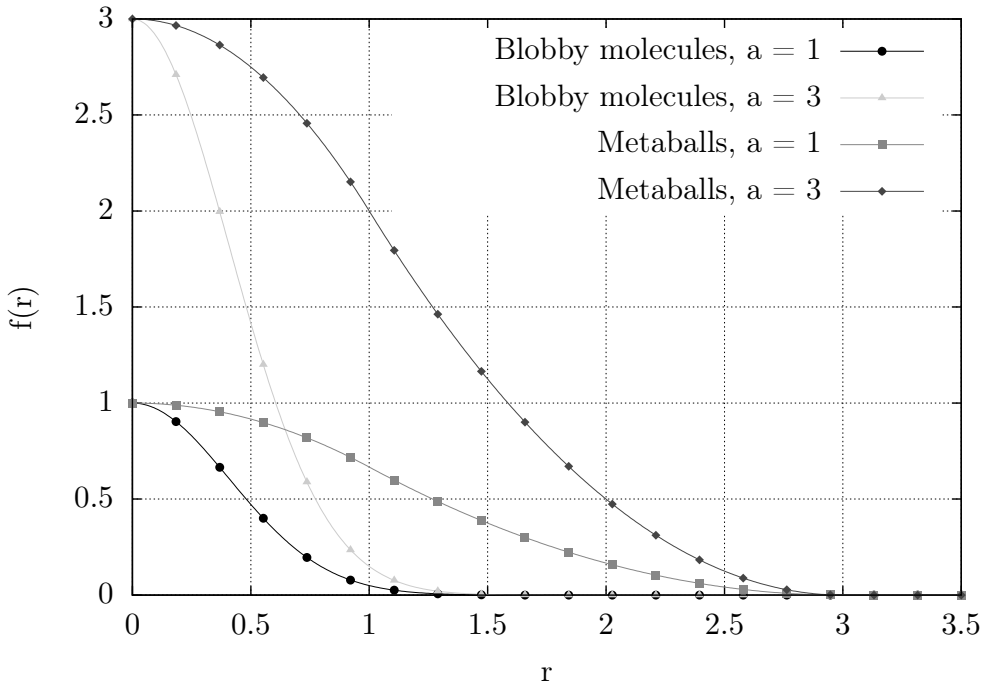
and

$$\sigma_i = \sigma_m + (\sigma_M - \sigma_m) \times \left(1 - \frac{F_m(i) - \min(F_m)}{\max(F_m) - \min F_m}\right) \tag{14}$$

where (x, y, z) is the position of the voxel in the object space, (x_i, y_i, z_i) is the position of Fly i in the object space, σ_i is the standard deviation for Fly i, σ_m and σ_M are the lowest and largest possible standard



(a) With $a = 1$.



(b) With $b = 3$.

Figure 13: Density field control functions from Eqs. 8 and 9. Parameters a and b are used to control the height and the width of the curve. For a given value of b, the shape of the curve can be more or less wide depending on the density field function used.

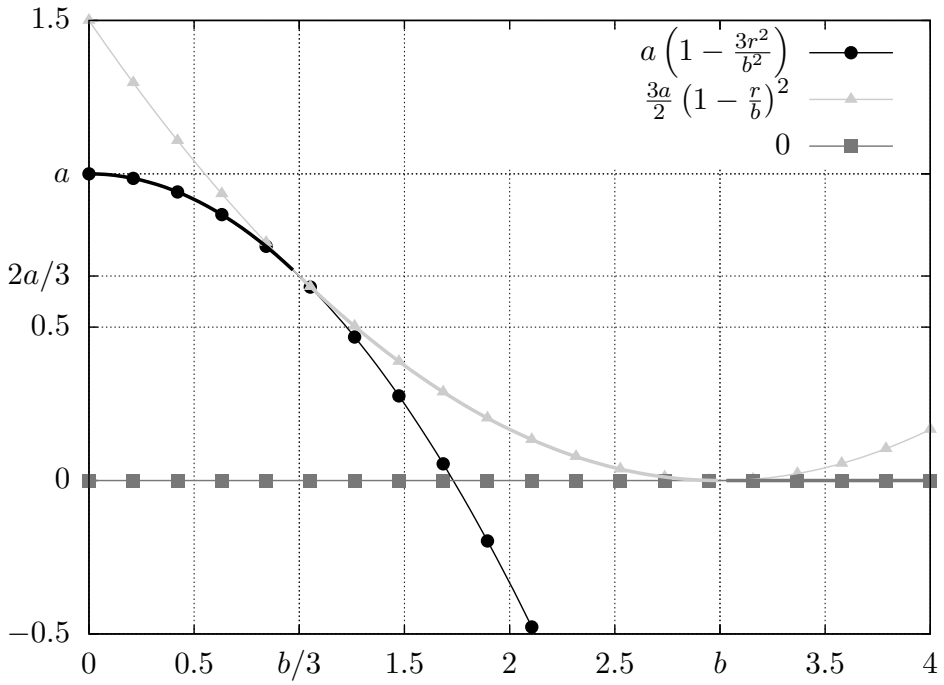


Figure 14: Decomposition of $f(r)$ from Eq. 9 with $a = 1$ and $b = 3$. Eq. 9 is a piecewise function with 3 sub-functions that join each other in $b/3$ and b to produce a smooth falling curve.

deviations, and $\min(F_m)$ and $\max(F_m)$ the smallest and biggest marginal fitness of flies. V_i is used to ensure that the total contribution of each fly to the final volume is 1.

We can see the corresponding reconstruction in Figure 17. In this case the NCC with the ground-truth is 92.79%. It looks visually closer to the ground-truth than any of the previous reconstructions. Figure 9 shows that the NCC between the new reconstruction and the ground-thruth is further improved. It is 10% higher than our previous method. The average rise time and fall time in the profile are 3.57 and 3.36 pixels respectively (see Figure 18). The overall sharpness metrics is 5710 ±309, which is better than any of the previous values. The total sum of values of the profile for Gaussian kernels is 32.18 ±2.36, which is also close to the corresponding value in the ground-truth.

One of our main aims in this paper was to demonstrate that more sophisticated voxelisation in the Fly algorithm could lead to better reconstructions. In the test-case considered so far, the final reconstruction is closer to the ground-truth and edges have been preserved. The fly population is now considered as a density field. The spread of Gaussian kernels is individually modulated depending on the marginal fitness of each fly. The outcome is a resolution-free model that is truly scalable, e.g. for a given reconstruction the noise level does not increase with the number of voxels.

5 Evaluation and Comparative Study

In this section, we evaluate our method in several ways. The first step is to compare our various voxelisation methods in term of speed and accuracy. The second step is to reconstruct volumes using a degraded sinogram from the same phantom to increase the problem's complexity. We also compare our reconstructions with those obtained with FBP and OSEM. Finally, we try the reconstruction methods on a more anatomically realistic phantom.

Figure 19 is an illustration of the reconstructions obtained at different stages of the algorithm with the sampling techniques presented above. It can be seen that using the local fitness to adapt the width and height of Gaussian kernels for each fly provide the most visually realistic results and also the most accurate results in term of NCC. Computations were performed on HPC Wales' supercomputer using a single node

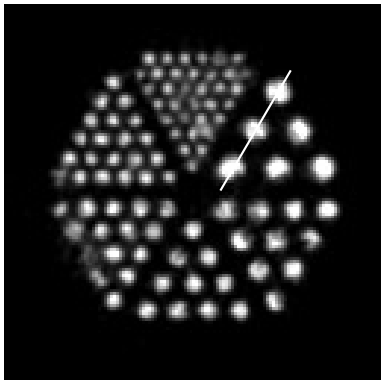


Figure 15: Voxelisation of the fly population ($\hat{f}$) using 12,800 metaballs (NCC with ground-truth: 89.69%) (see Figure 8a for the corresponding ground-truth). The same fly population as in Figure 8b was used.

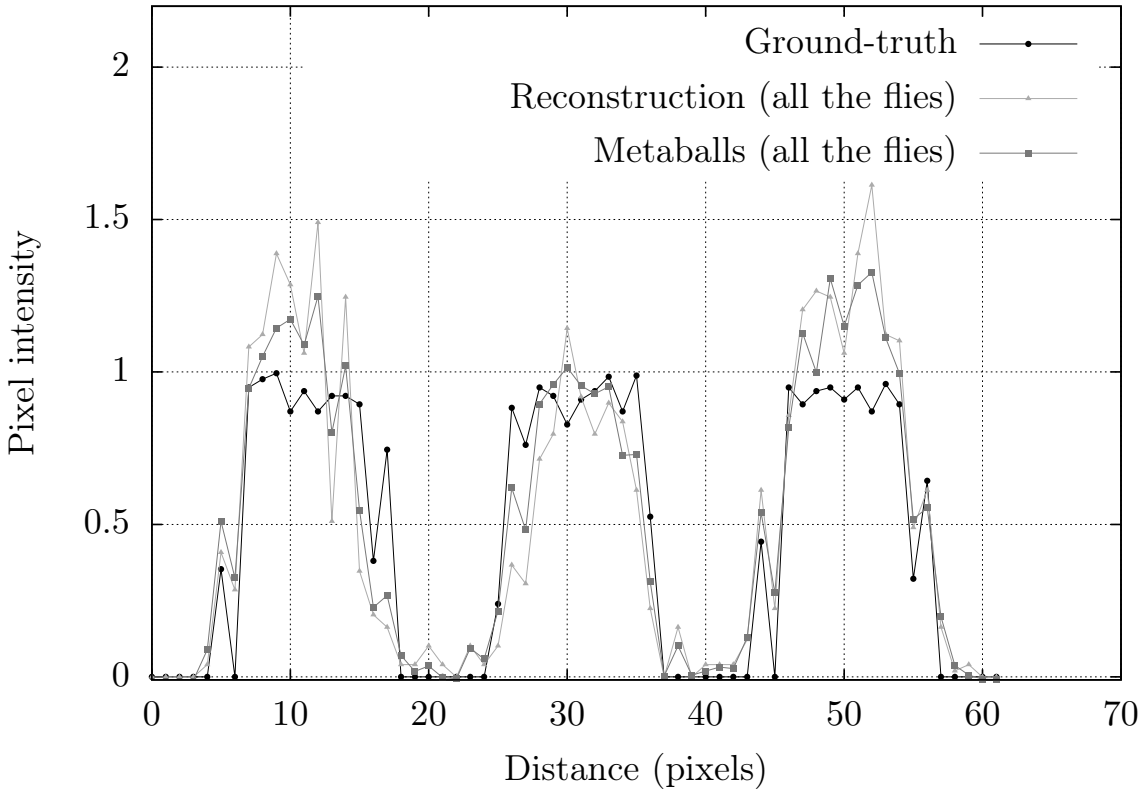


Figure 16: Intensity profiles corresponding to the white lines in Figures 8a, 8b and 15.

with an Intel Xeon Westmere X5650 @ 2.67 GHz processor. A quick succession of mitosis happens in less than 5 minutes, when the NCC reaches a threshold. Optimal results were obtained in 8:41 minutes using 6,400 flies (NCC of 92.76%). More processing time did not yield to better results. In fact, using 12,800 flies (17:57 minutes) leads to results that are comparable to using 3,200 flies (4:50 minutes): NCC of 91.67% and 91.33% respectively.

Using our naive approach as in [22, 21], the NCC would have been limited to 78.27% or 84.75% only. By exploiting the local fitness of flies to adapt Gaussian kernels, the image quality improves by about 10%. At similar levels of quality, the computing time is significantly reduced. For example, with our initial voxelisation method 17:57 minutes were required to obtain 82.59%. Only 2:12 minutes are needed to reach an even better

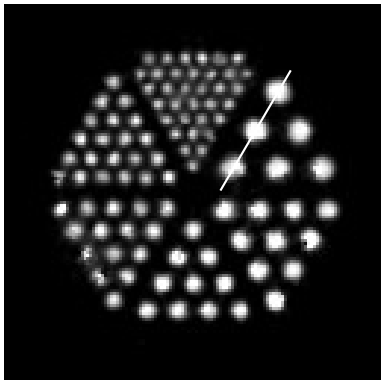


Figure 17: Voxelisation of the fly population ($\hat{f}$) using 12,800 gaussian kernels (NCC with ground-truth: 92.79%) (see Figure 8a for the corresponding ground-truth). The same fly population as in Figures 8b and 15 was used.

Table 2: NCC between the ground-truth (Figure 8a) and the reconstructions of Figure 21. Numerical values in bold characters are the ones closest to the ground-truth.

Reconstruction type	**NCC**
All flies	79.79% ± 1.13%
Good flies only	78.94% ± 1.26%
Metaballs	83.82% ± 1.06%
Gaussian	**85.92% ± 0.60%**
FBP	72.39%
OSEM	78.50%

NCC with adaptive Gaussian kernels.

To assess the algorithm in more difficult conditions, the number of angles in the sinogram is lowered and noise is included (see Figures 20 and 21a). The initial sinogram in Figure 7a was made of 180 rows with an angular step of 1°. The new sinogram contains 37 rows and the angular step is 5°. The sinogram estimated by the final population of flies (Figure 20b) is almost perfect (NCC of 99.63%). Figure 21 shows the corresponding reconstructions. Table 2 summarises the NCC values between the ground-truth (see Figure 8a) and the reconstructions of Figure 21. It shows that the Fly algorithm with density fields, both with Metaballs and Gaussian kernels, outperforms the traditional FBP and OSEM algorithms when the input data is of low resolution and noisy. One of the reasons of the outstanding performance of the Fly algorithm in the presence of noise is the stochastic nature of artificial evolution.

In Figure 22a, we use another numerical phantom that is more anatomically realistic. It corresponds to cardiac PET data. Noise is included in the phantom (see Figure 22b) to produce the sinogram of Figure 23a. Again, the sinogram estimated by the final population of flies (Figure 23b) is almost perfect (NCC of 99.87%). FBP, OSEM and an evolutionary reconstruction are given in Figure 22. In this test case, the NCCs of all the reconstructions are within less that 2% from each other (see Table 3).

6 Conclusion

In the research presented here, we addressed the complex problem of medical tomographic reconstruction using evolutionary computing, by transposing the Fly Algorithm technique originally developed in a stereo-vision context for robotics. In classical EAs at the end of the algorithm the best individual is extracted and considered the solution of the optimisation problem, while all the rest of the population is discarded. The Fly Algorithm, which is a Cooperative Co-evolution algorithm relies on a different philosophy, where each individual is a part of the solution. An individual corresponds to a 3-D point. The whole population is a representation of the reconstructed tomographic images. The evaluation of the performance of each individual

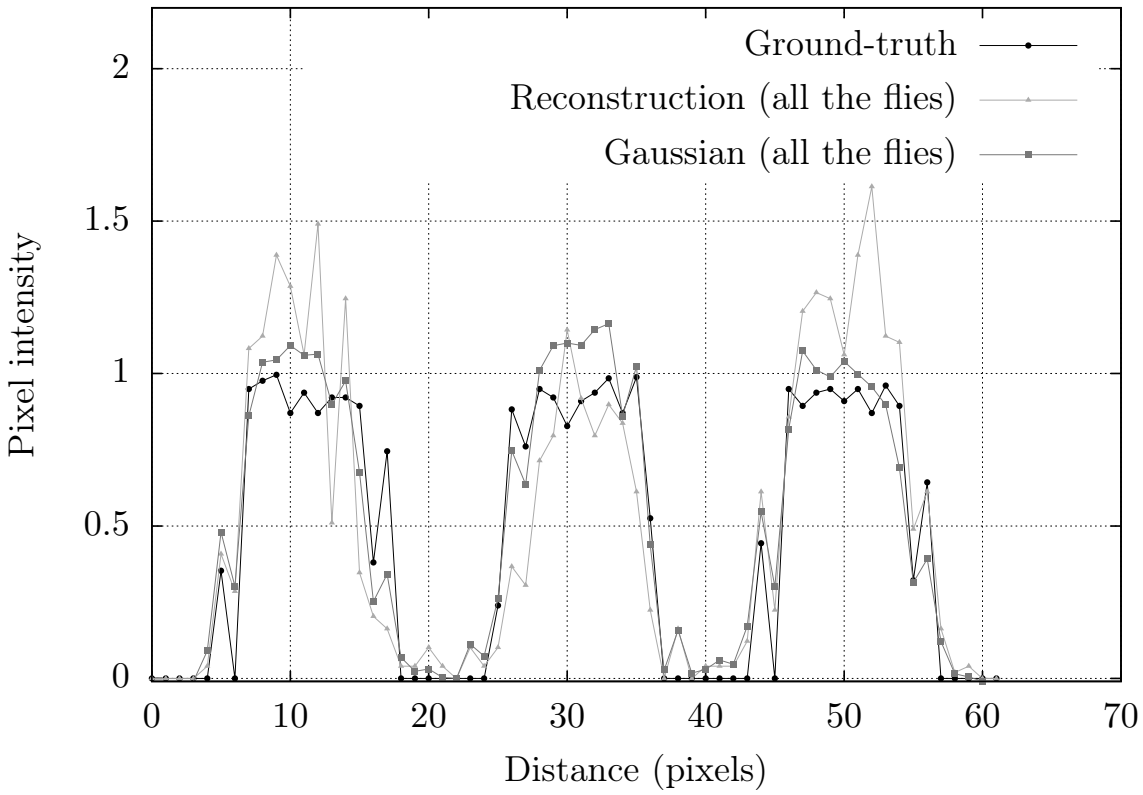


Figure 18: Intensity profiles corresponding to the white lines in Figures 8a, 8b and 17.

is performed using a fitness function based on the leave-one-out-cross-validation method. If the fitness is positive, then the fly is improving the performance of the population; if it is negative, it is deteriorating the performance of the population. In the nuclear medicine context, the concentration of flies will be an estimation of the radioactive concentration.

In our previous developments we were only keeping the flies with a positive fitness. Binning (also call 'bucketing') was used to convert this point cloud into a discrete 3-D volume made of voxels: The space was divided in a regular 3-D grid and the voxel intensity corresponded to the number of good flies located into it. The local fitness of flies was not exploited during the voxelisation. No or very little comparison with traditional tomographic reconstruction algorithms in nuclear medicine was provided.

We saw in this paper that keeping the good flies only does not necessarily lead to the best quantitative results: Retaining the flies with a negative fitness yields more accurate results. The natural output of the Fly algorithm is a population of 3-D points. Here, we also exploit the point cloud one step further and use

Table 3: NCC between the ground-truth and the reconstructions in the case of the cardiac example (Figure 22). Numerical values in bold characters are the ones closest to the ground-truth.

Reconstruction type	**NCC**
All flies	90.75% ± 0.92%
Good flies only	86.78% ± 1.63%
Metaballs	93.97% ± 0.68%
Gaussian	95.39% ± 0.45%
FBP	96.81%
OSEM	**97.07%**

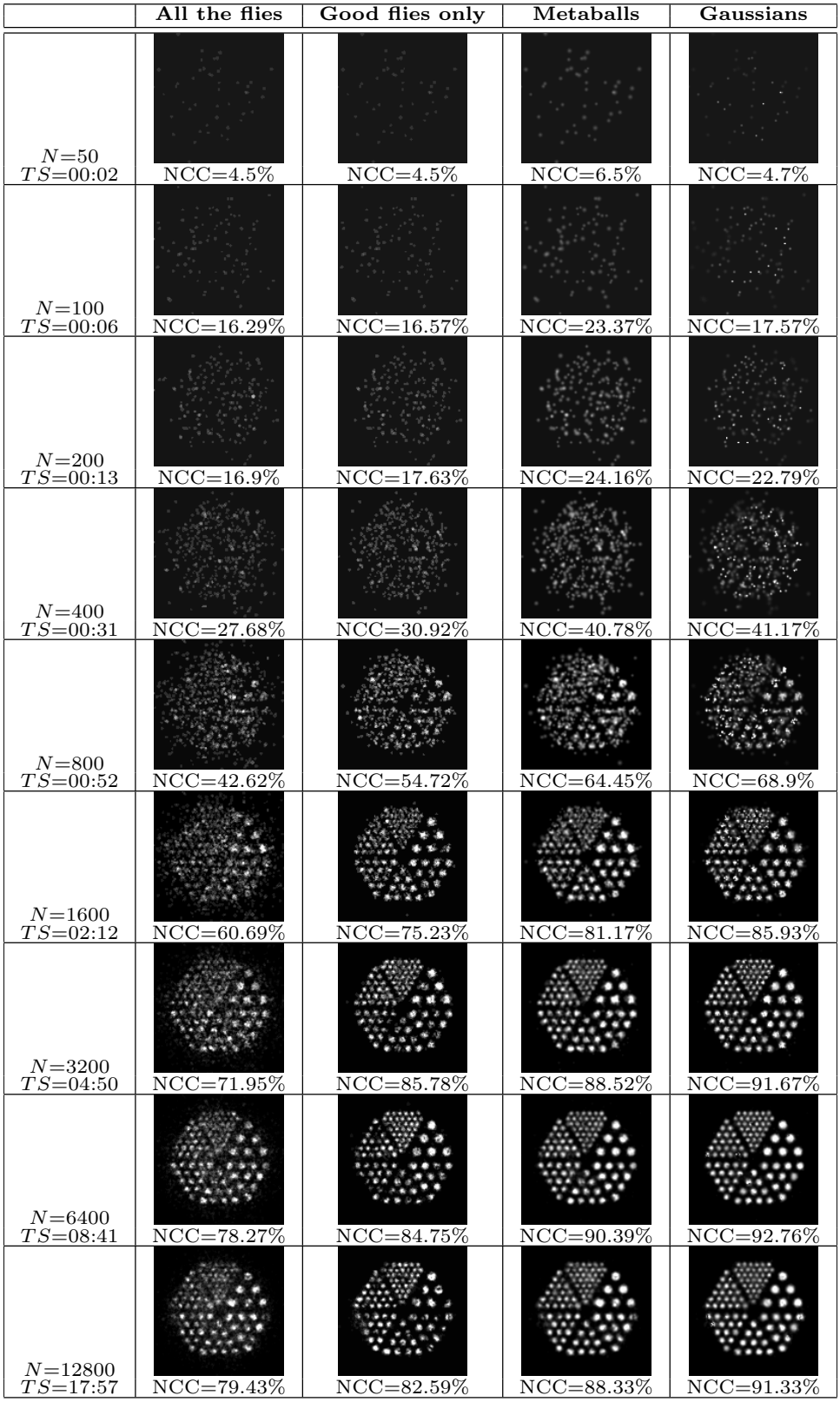
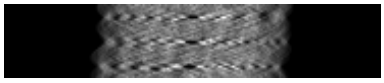
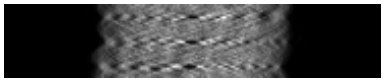


Figure 19: Evolutionary reconstructions at successive resolutions (with N the number of flies and TS the time-stamp in minutes) using an Intel Xeon Westmere X5650 @ 2.67 GHz processor.

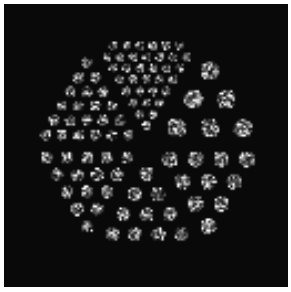


(a) Observations (known data).

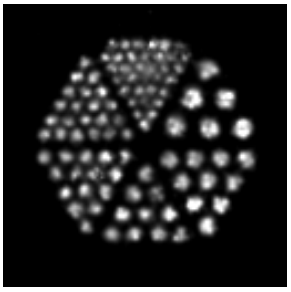


(b) Estimation (data simulated with all the flies) (NCC with (a): 99.63%).

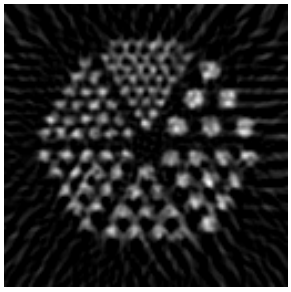
Figure 20: Sinograms from Figure 21a corresponding to the hot rode phantom with a low resolution and with noise. Images with 185 pixels per projection, 1st angle: 0°, angular step: 5°, and last angle: 175°.



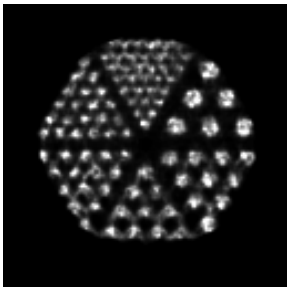
(a) Actual noisy phantom (NCC with ground-truth: 73.97%).



(b) Reconstruction with Gaussians (NCC with ground-truth: 85.92%).



(c) Reconstruction with FBP (NCC with ground-truth: 72.39%).



(d) Reconstruction with OSEM using 18 iterations and 2 subsets (NCC with ground-truth: 78.50%).

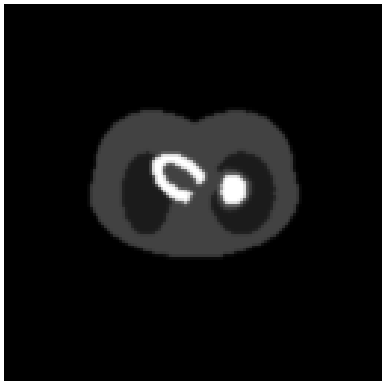
Figure 21: Tomographic reconstructions of the sinogram in Figure 20a corresponding to the hot rod example with a low number of angles and noise (see Figure 8a for the corresponding ground-truth).

implicit modelling to voxelise the data from a density field. To this end, this paper has investigated the use of Metaballs and Gaussian kernels - where the height and width of each Gaussian is computed to take into account the corresponding fly's performance. During the evolution, flies are discrete (from a numerical point of view) particle emitters: Flies are trying to replicate the observed data (what actually happened). At the end of the reconstruction, i.e. after convergence and during the extraction of the solution, each fly is considered as a given realisation of a stochastic process: A fly is an approximation of a random variable and, as such, can be modelled as a density field. It is actually intuitive to prefer implicit modelling over data binning as this is these stochastic events that we are actually trying to estimate. The marginal fitness can be considered as a confidence level in the fly's position. We demonstrate here that it is possible to take advantage of the fitness of each individual after the optimisation process to modulate the spread of flies depending on their respective performance. It improves quantitative results in all our test-cases by more than 10% in term of NCC compared to binning. A more accurate reconstruction is also achieved using less computational power. In this paper, our reconstructions are also compared with those of FBP and OSEM, which are traditionally used in nuclear medicine. Results show that density fields using Gaussian kernels lead to similar, if not better, reconstructions depending on the input data.

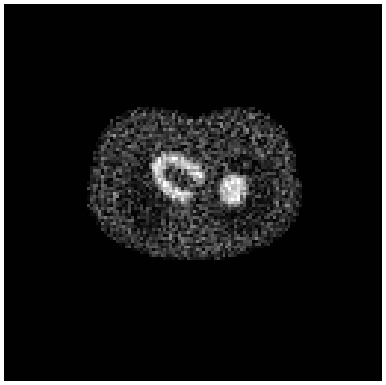
Future work will include testing our methods using more clinically realistic data and investigating the use of the latest advances in signal processing such as compressed sensing (also known as compressive sensing, compressive sampling, or sparse sampling).

Acknowledgements

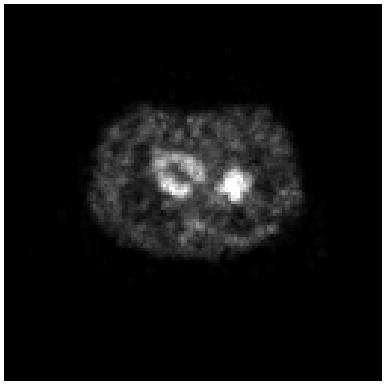
The authors would like to thank Yanhua Hong from the School of Electronic Engineering at Bangor University for her help with data analysis, and HPC Wales (`http://www.hpcwales.co.uk/`) for providing some of the computing facilities used in this study. This work was partially supported by the European Commission, Grant no. 321968 (`http://fly4pet.fpvidal.net`).



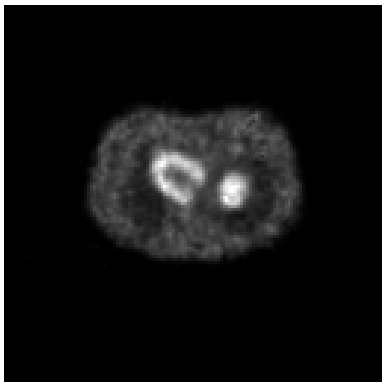
(a) Ground-truth.



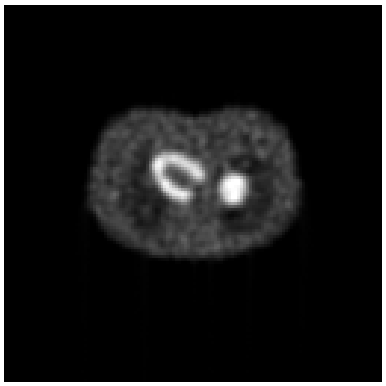
(b) Actual noisy phantom (NCC with (a): 91.12%).



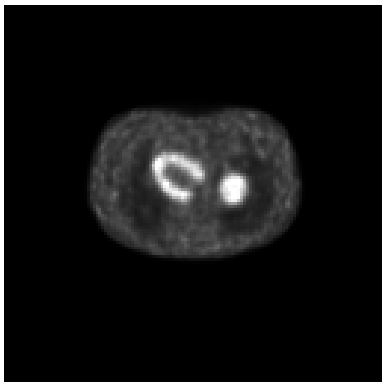
(c) Binning with all the flies (NCC with (a): 91.44%).



(d) Reconstruction with Gaussians (NCC with (a): 96.14%).



(e) Reconstruction with FBP (NCC with (a): 96.81%).



(f) Reconstruction with OSEM using 7 iterations and 4 subsets (NCC with (a): 97.07%).

Figure 22: Tomographic reconstructions of the sinogram in Figure 23a corresponding to the cardiac example, i.e. a more anatomically realistic sinogram with noise.

Acronyms

AE artificial evolution.

CBCT cone-beam computed tomography.

CCEA Cooperative Co-evolution algorithm.

CG computer graphics.

CoCo Cooperative Co-evolution.

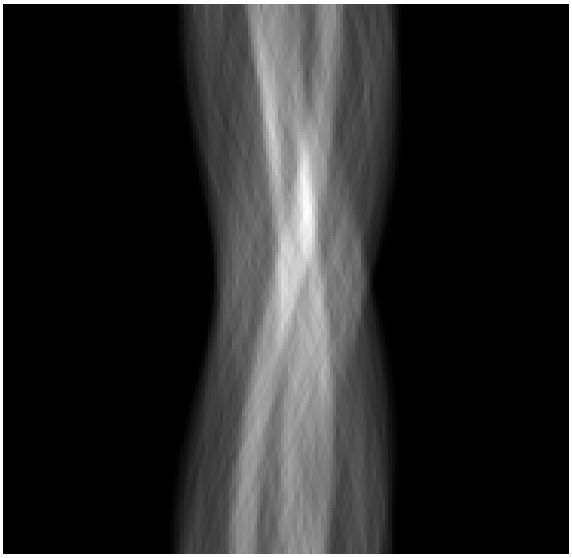
CT computed tomography.

EA evolutionary algorithm.

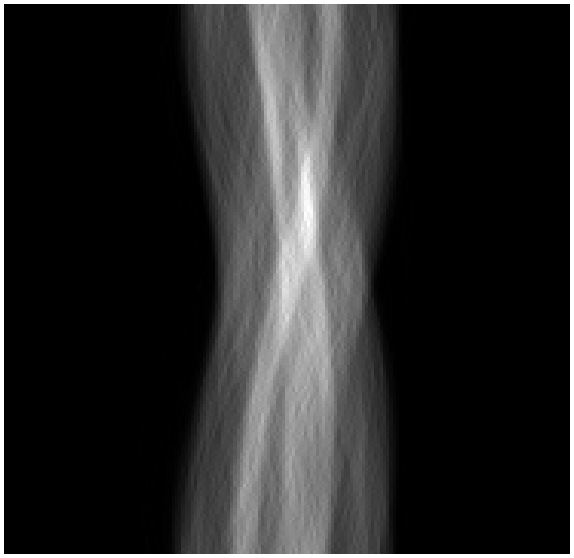
EM expectation-maximisation.

ET emission tomography.

FBP filtered backprojection.



(a) Observations (known data).



(b) Estimation (data simulated with all the flies) (NCC with (a): 99.87%).

Figure 23: Sinograms of the cardiac example from Figure 22a. Images with 185 pixels per projection, 1st angle: 0°, angular step: 1°, and last angle: 179°

LUT lookup table.

MLEM maximum-likelihood expectation-maximisation.

NCC normalised cross-correlation.

OSEM ordered-subset expectation-maximisation.

PET positron emission tomography.

PSO particle swarm optimisation.

SNR signal-to-noise ratio.

SPECT single-photon emission computed tomography.

voxel volume element.

References

[1] Z. A. Abbood, J. Lavauzelle, E. Lutton, J.-M. Rocchisani, J. Louchet, and F. P. Vidal. Voxelisation in the 3-D Fly algorithm for PET. *Swarm and Evolutionary Computation*, 2017. To appear.

[2] T. Bäck. Self-adaptation in genetic algorithms. In *Proc 1st European Conf Artif Life*, pages 263–271. MIT Press, 1992.

[3] T. Bäck. Optimal mutation rates in genetic search. In *Proc 5th Int Conf Genetic Algorithms*, pages 2–8, 1993.

[4] N. Bissantz, B. A. Mair, and A. Munk. A statistical stopping rule for MLEM reconstructions in PET. In *IEEE Nuclear Science Symposium Conference Record*, pages 4198–4200, Oct 2008.

[5] J. F. Blinn. A generalization of algebraic surface drawing. *ACM Trans. Graph.*, 1(3):235–256, July 1982.

[6] A. Bousquet, J. Louchet, and J.-M. Rocchisani. Fully three-dimensional tomographic evolutionary reconstruction in nuclear medicine. In *Proceedings of the Evolution Artificielle, 8th International Conference on Artificial Evolution*, EA'07, pages 231–242, Berlin, Heidelberg, 2008. Springer-Verlag.

[7] P. Collet and J. Louchet. *Applications in the Processing of Signals and Images*, chapter Chapter 2. Artificial Evolution and the Parisian Approach, pages 15–44. Wiley, 2010.

[8] H. M. Hudson and R. S. Larkin. Accelerated image reconstruction using ordered subsets of projection data. *IEEE Transactions on Medical Imaging*, 13(4):601–609, Dec 1994.

[9] A. C. Kak and M. Slaney. *Principles of Computerized Tomographic Imaging*. Society of Industrial and Applied Mathematics, 2001.

[10] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks, 1995*, volume 4, pages 1942–1948, Nov. 1995.

[11] M. Levoy. Efficient ray tracing of volume data. *ACM Trans. Graph.*, 9(3):245–261, July 1990.

[12] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, Aug. 1987.

[13] J. Louchet. Stereo analysis using individual evolution strategy. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 1, pages 908–911 vol.1, 2000.

[14] H. Nishimura, M. Hirai, T. Kawai, T. Kawata, I. Shirakawa, and K. Omura. Object modeling by distribution function and a method of image generation. *Trans. Inst. Elect. Commun. Eng. Japan J68-D*, 4:718–725, 1985.

[15] J. Qi and R. M. Leahy. Iterative reconstruction techniques in emission computed tomography. *Physics in Medicine and Biology*, 51(15):R541, 2006.

[16] A. Rosset, L. Spadola, and O. Ratib. Osirix: An open-source software for navigating in multidimensional dicom images. *Journal of Digital Imaging*, 17(3):205–216, 2004.

[17] L. Shepp and Y. Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE Transactions on Medical Imaging*, 1(2):113–122, Oct 1982.

[18] Y. Shi and R. Eberhart. A modified particle swarm optimizer. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 69–73, May 1998.

[19] A. Sitek, R. H. Huesman, and G. T. Gullberg. Tomographic reconstruction using an adaptive tetrahedral mesh defined by a point cloud. *IEEE Transactions on Medical Imaging*, 25(9):1172–1179, Sept. 2006.

[20] F. P. Vidal, D. Lazaro-Ponthus, S. Legoupil, J. Louchet, E. Lutton, and J. Rocchisani. Artificial evolution for 3D PET reconstruction. In *Proceedings of the 9th international conference on Artificial Evolution (EA'09)*, volume 5975 of *Lecture Notes in Computer Science*, pages 37–48, Strasbourg, France, Oct. 2009. Springer, Heidelberg.

[21] F. P. Vidal, J. Louchet, J. Rocchisani, and E. Lutton. New genetic operators in the Fly algorithm: application to medical PET image reconstruction. In *Applications of Evolutionary Computation*, volume 6024 of *Lecture Notes in Computer Science*, pages 292–301, Istanbul, Turkey, Apr. 2010. Springer, Heidelberg.

[22] F. P. Vidal, E. Lutton, J. Louchet, and J. Rocchisani. Threshold selection, mitosis and dual mutation in cooperative coevolution: application to medical 3D tomography. In *International Conference on Parallel Problem Solving From Nature (PPSN'10)*, volume 6238 of *Lecture Notes in Computer Science*, pages 414–423, Krakow, Poland, Sept. 2010. Springer, Heidelberg.

[23] F. P. Vidal, P. Villard, and E. Lutton. Tuning of patient specific deformable models using an adaptive evolutionary optimization strategy. *IEEE Transactions on Biomedical Engineering*, 59(10):2942–2949, Oct. 2012.

[24] G. Wyvill, C. McPheeters, and B. Wyvill. Data structure for soft objects. *The Visual Computer*, 2(4):227–234, 1986.