An automatic algorithm to exploit the symmetries of the system response matrix in PET iterative reconstruction.

Niccolò Camarlinghi[1,2], Giancarlo Sportelli[1,2], Alberto Del Guerra[1,2] and Nicola Belcari[1,2]

1 Department of Physics, Pisa University, Pisa, Italy
2 Istituto Nazionale di Fisica Nucleare, Pisa, Italy

E-mail: niccolo.camarlinghi@unipi.it

Abstract. Positron Emission Tomography (PET) iterative 3D reconstruction is a very computational demanding task. One of the main issues of the iterative reconstruction concerns the management of the System Response Matrix (SRM). The SRM models the relationship between the projection and the voxel space and its memory footprint can easily exceed hundreds of GB. Moreover, in order to make the reconstruction fast enough not to hinder its practical application, the SRM must be stored in the Random Access Memory (RAM) of the workstation used for the reconstruction. This issue is normally solved by implementing efficient storage schemes and by reducing the number of redundant patterns in the SRM through symmetries. However, finding a sufficient number of symmetries is often non-trivial and is typically performed using dedicated solutions that cannot be exported to different detectors and geometries. In this paper, an automatic approach to reduce the memory footprint of a pre-computed SRM is described. The proposed approach was named Symmetry Search Algorithm (SSA) and consists in an algorithm that searches for some of the redundant patterns present in the SRM, leading to its lossy compression. This approach was built to detect translations, reflections and coordinates swap in voxel space. Therefore, it is particularly well suited for those scanners where some of the rotational symmetries are broken, e.g., small animal scanner where the modules are arranged in a polygonal ring made of few elements, and dual head planar PET systems. In order to validate this approach, the SSA is applied to the SRM of a preclinical scanner (the IRIS PET/CT). The data acquired by the scanner were reconstructed with a dedicated Maximum Likelihood Estimation Maximization (MLEM) algorithm with both the uncompressed and the compressed SRMs. The results achieved show that the information lost due to the SSA compression is negligible. Compression factors up to 52 when using the SSA together with manually inserted symmetries and up to 204 when using the SSA alone, can be obtained for the IRIS SRM. These results come without significant differences in the values and in the main quality metrics of the reconstructed images, i.e. spatial resolution and noise. Although the compression factors depend on the system considered, the SSA is applicable to any SRM and therefore it can be considered a general tool to reduce the footprint of a pre-computed SRM.

1. Introduction

Iterative image reconstruction methods can lead to improved performance over analytic methods, when a system model that reproduces accurately the PET physics is used [1, 2]. However, due to their computation burden, they have been gaining popularity only in the last decades, thanks to the computational resources available nowadays. Reconstruction algorithms can be roughly divided into three different categories [2]: algorithms making use of a pre-computed SRM, algorithms that compute the SRM "on-the-fly" during the reconstruction process and hybrid approaches. The pre-computed approach is typically implemented on ordinary multi-cores and clusters architectures [3, 4, 5, 6] whereas "on the fly" SRM calculation is usually performed on many-cores architectures [7, 8]. An SRM can be implemented either by a semi-analytic algorithm [9, 5], by using a Monte Carlo simulation [10, 11, 3], by experimental measurements [12] or by a combination of these methods [13]. A comprehensive review of the methods used to evaluate the SRM can be found in [2]. All the methods using a pre-computed SRM need to deal with its size, that can easily exceed hundreds of Gigabytes [5, 6, 3, 12]. Moreover, in order to make the reconstruction fast enough for clinical applications, a pre-computed SRM must be "memory resident", i.e., it must be stored entirely in the random access memory (RAM) of the workstation [5]. Due to these constraints, a pre-computed SRM can be stored on RAM either using a workstation that supports the needed amount of memory or reducing its footprint by using efficient storage schemes and symmetries. The first approach is straightforward but can be implemented only if a dedicated workstation for the reconstruction is available. The second, is more flexible and suitable for a research environment and was pursued by many authors [3, 4, 11, 6, 14]. Currently, most of the approaches in literature use dedicated implementations that assume, a-priori, the existence of exact symmetry classes such as translations and reflections. In general, finding a sufficient number of symmetries to reduce the SRM footprint is often non-trivial and is typically performed in a way that cannot be exported to different detectors and geometries. The problem of compressing the SRM using manually implemented symmetries has been addressed by several authors using different strategies, including the implementation of quasi-symmetries [3], a-priori symmetries tailored to a specific scanner geometry [4, 11, 3] and implementing polar coordinate system [5, 6]. Moreover, symmetries were used to speed-up the reconstruction on graphics processing unit (GPU) [15] and on multicore architectures [6, 16].

Even if the size of SRM and the compression factors depend on many factors, e.g. the number of the LORs of the scanner, the physics implemented in the SRM, and the FOV sampling, it is useful to review the typical compression factor achieved with other methods. In [3], an exploitation of the symmetries of the eXplore Vista-DR (GE) small animal PET scanner is described. The eXplore Vista-DR is made of 36 modules arranged in two rings where each module is made of 13×13 crystals coupled to a position sensitive PhotoMultiplier Tube (PMT). A compression factor of 40 was obtained, assuming exact axial translation and reflection symmetries. An additional factor 10 was obtained

using quasi-symmetries. In [4], the exploitation of the YAP-PET small animal scanner was reported. The YAP-PET consists of two pairs of modules each made of YAP 20×20 crystal matrix coupled to a PMT. A tailored implementation of translations and reflections was used. The compression factor obtained was greater than 80. In [5], the exploitation of the symmetries of the Siemens BrainPET is described. The BrainPET is made of 192 detector blocks arranged in 6 rings of 32 block each. Each block consists of 12×12 LSO crystals couple to Avalanche PhotoDiodes (APD). A compression factor of 320 was obtained using polar voxels and generic cylinder modeling. The performance obtained by all the mentioned approaches refer to different situations and therefore they are not directly comparable. In this paper, an automatic general purpose method to reduce the memory footprint of a disk stored SRM through symmetries is proposed. The algorithm was named SSA (Symmetry Search Algorithm). The SSA method substantially differs from the others, as the symmetries are extracted a-posteriori. No assumption on the SRM, the scanner geometry, and the detector arrangement is done. The proposed approach is applicable to all the geometries and to any SRM stored in Cartesian voxels, requiring little adjustment on a per-case basis. The SSA was built to detect translations, reflections and coordinates swap in voxel space and to provide, at the same time, a very accurate approximation of the original SRM, thus relieving the user from the task of figuring out the symmetries of the system manually. This method is particularly well suited for those scanners where some of the rotational symmetries are broken, e.g., small animal scanner where the modules are arranged in a polygonal ring made of few elements [17, 18] and dual head planar PET systems [19, 20]. In those scanners where a high degree of rotational symmetries is found, e.g. human clinical scanners, the SSA can be used in a semi-automatic way to further reduce the memory footprint of the model part that was actually computed.

2. Materials and Methods

2.1. MLEM Iterative reconstruction and system response matrix representation

Maximum Likelihood Estimation Maximization (MLEM) [21] is one of the most popular algorithms used in iterative PET reconstruction. In its simplest form, the MLEM can be written for a system with N Lines of Response (LOR) and a Field Of View (FOV) of V voxels as

$$X^{q+1} = \frac{X^q}{M^\intercal \cdot 1} M^\intercal \cdot \left(\frac{P}{M \cdot X^q} \right) \tag{1}$$

where $X^q \in R^V$ is a vector containing the image at the q-th iteration, $M \in R^{N \times V}$ is the SRM, $P \in R^N$ is a vector containing the coincidences detected in each LOR. The element a, b of the SRM contains the probability that a photon pair emitted from the bth voxel of the FOV is detected in the ath LOR of the system [22]. The set of values associated with the LOR a, $M_{a,1}, M_{a,2}, .., M_{a,V}$, is often referred to as the Tube Of Response (TOR) T_a. Since the number of nonnull element in each TOR is limited, the size of the SRM can be

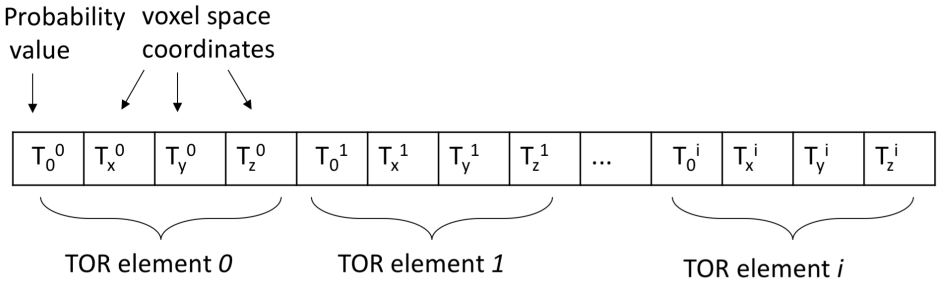


Figure 1. Representation of a Tube Of Response (TOR) with i non-null elements.

reduced exploiting its sparseness, i.e. storing only TOR's non-null elements. In order to exploit symmetries, it is more convenient to represent each voxel of the FOV using a coordinate triplet (b_x, b_y, b_z) rather than the linear index b. A single entry T^i of a TOR can be stored as a probability T_0^i value followed by three voxel coordinates (T_x^i, T_y^i, T_z^i) (see Figure 1). Using this representation, the theoretical SRM memory footprint is $\approx N_0 \cdot (3 \cdot S_{index} + S_{value})$ bytes where N_0 is the number of non-null elements in the whole matrix, S_{index} is the size of the variable representing each index of the triplet and S_{value} is the size of the variable representing the probability. In this work, S_{index} and S_{value} are set to 1 byte (unsigned char) and four bytes (float) respectively. Due to this choice, the size of the FOV is limited to 255 voxels per direction. The generalization of the SSA to other cases is trivial.

2.2. SSA description

Intuitively, two TORs can be considered symmetric if they are equal up to a geometric transformation. This means that their probability values are exactly equal and they can be mapped one into the other with a geometric transformation. However, in practice, due to small differences in their probabilities, the number of TORs that are exactly symmetric, according to this definition, is very limited. Therefore, a less strict condition has to be implemented in order to improve the compression efficacy of the SSA. In this work, two TORs l and m are said to be symmetric within a threshold t if:

(i) It is possible to transform the coordinates of all the elements of l into those of the m using a voxel space transformation.

(ii) All the probability values of the so-obtained TOR lie within a relative threshold t with respect to the original.

When a threshold t=0 is applied, the strict def for the IRIS SRMinition of symmetry is obtained. Only combinations of reflections, translations, and coordinates swap in the voxels space are exploited in this work. Using these hypotheses, the existence of a symmetry between two TORs can be reduced to three sufficient conditions:

A The number of non-null TOR elements n_0 of l and m is the same.

B Sorting the elements of the TORs l and m according to their probabilities, the following relation holds for each element i=1 $\dots n_0$

$$d(l_0^i, m_0^i) = \left|l_0^i - m_0^i\right| / \min\left(l_0^i, m_0^i\right) \leq t.$$

C It is possible to sort l and m and to find $\vec{A} = (A_x, A_y, A_z)$ and $\vec{k} = (k_x, k_y, k_z)$ and z such that the following relation is true for each TOR entry i

$$\vec{l^i} + \vec{A} \odot S_z(\vec{m^i}) = \vec{k} \tag{2}$$

with $A_x, A_y, A_z = \pm 1$, $\vec{k}$ a constant vector not depending on i, the $\odot$ is the element-wise product between vectors and S_z is the "swap operator". S_z transforms a vector into another containing its components permuted. Condition C is a convenient formula to characterize reflections, translations, and coordinate swap. For example, a pair of TORs symmetric up to a reflection along x axis can be described by a transformation with $A = (1, -1, -1)$, $\vec{k} = (0, 0, 0)$ and no coordinate swap (S_z equal to identity). A 90 degree rotation with respect to the z-axis can be implemented swapping x, y, i.e., using S_z such as $(x, y, z) \rightarrow (y, x, z)$, and setting $A = (-1, -1, -1)$ and $\vec{k} = (0, 0, 0)$. Since each component of $\vec{A}$ can assume only two values, i.e., $= \pm 1$ and the number of permutations of 3 indices is 6, 48 different symmetry classes are possible.

2.3. SSA implementation

A naïve implementation of the SSA would require checking all the possible TOR pairs in the SRM against the conditions A,B,C. However, this would lead to a computational complexity of $\sim N^2$, with N equal to the number of TORs contained in the SRM. To reduce the computational burden of the SSA, the TORs are first grouped according to the number of non-null entries, i.e., voxels that had a non-null probability during the model computation, and then each group is stored in a separate file on disk. This reduce the memory demand of the SSA since each TOR group can be analyzed independently and allows for a straightforward multi-core implementation, where each thread searches the symmetries within a group. Using this method, condition A is automatically satisfied within the same group (see algorithm (1)) and the complexity of the SSA is reduced to $\sim G \cdot J^2$, where G is the number of groups and J is the average number of TOR per group, with both G and $J \ll N$. To check a pair of TORs against condition B, both

Algorithm 1 Implementation of condition A

Given the TORs **l**,**m** each with n_l, n_m non-null entries
1: **if** $n_l \neq n_m$ **then**
2: No symmetry between **l** and **m**
3: **end if**
4: Check condition B

the TORs are first sorted according to their probability values. This way, condition B can be checked element-wise (see algorithm (2)). Finally, any TOR pair that meets

condition B is validated against condition C. An algorithm to reduce the complexity of condition C is implemented. This solution allows to validate the condition pixel-wise, therefore reducing its complexity to the number of non-null elements of the involved TORs. The problem of checking condition C is that the values of $\vec{k}$ are not known a-priori. A necessary condition to find $\vec{k}$ can be extracted by summing over the non-null entries on both sides of eq.(2). This yields to eq. (3), which leads to eq. (4).

$$(\sum_{i=1..n_0} \vec{l^i}) + \vec{A} \odot S_z(\sum_{i=j..n_0} \vec{m^j}) = n_0 \cdot \vec{k} \tag{3}$$

$$\vec{k} = \frac{(\sum_{i=1..n_0} \vec{l^i}) + \vec{A} \odot S_z(\sum_{j=1..n_0} \vec{m^j})}{n_0}. \tag{4}$$

Equation (4) gives a necessary but not sufficient condition for $(z, \vec{A}, \vec{k})$ to be a symmetry between two TORs l and m. For a given pair of TORs, 48 values of $\vec{k}$ are found, one for each combination of $\vec{A}$ and z. Some of the values found can be excluded by discarding those transformations that do not correspond to a $\vec{k}$ vector made of integer components. For all the other values, a direct test is needed. The direct test is performed transforming m according to $(z, \vec{A}, \vec{k})$ and then sorting l and m according to their coordinates. Since no tolerance is allowed in the coordinates of two symmetrical TORs, it is possible to check condition C pixel-wise as described in algorithm (3). If $(z, \vec{A}, \vec{k})$ transforms TOR l into TOR m within the threshold, then a symmetry is found. The result of the compression is a new SRM containing only the fundamental TORs (FT) and the transformations $(z, \vec{A}, \vec{k})$ to obtain those that are not fundamental. A non-fundamental TOR m can be retrieved by knowing the $(z, \vec{A}, \vec{k})$ values and the fundamental TOR l, by using the formula

$$\vec{m^i} = S_z^{-1}(\vec{A} \odot (\vec{k} - \vec{l^i})). \tag{5}$$

It is worth noticing that the storage saved depends on the TOR size but not on the symmetry detected.

Algorithm 2 Implementation of condition B

Given the TORs **l**,**m** fulfilling condition A
Sort all entries of **l**,**m** according to their probability values
for i in $1, n_0$ **do**
  if $d(l_0^i, m_0^i) > t$ **then**
    No symmetry between **l** and **m**
  end if
end for
Check condition C

The IRIS PET/CT is a novel pre-clinical system for mice and rats featuring a full ring PET and a high resolution CT system [17]. The PET component of the scanner

Algorithm 3 Implementation of condition C

Given the TORs **l**,**m** fulfilling condition A,B
for Z_* in $Z_1..Z_6$ **do**
  for A_* in $A_1..A_8$ **do**
    Compute $\vec{K}_*$ using (4)
    if All $\vec{K}_*$ components are integer **then**
      Transform all m^i using (5)
      Sort **l** and **m** TORs according to the pixel index
      for i in $1..n_0$ **do**
        if $l^i_{1,2,3} \neq m^i_{1,2,3}$ **or** $d(l^i_0, m^i_0) > t$ **then**
          (Z_*, A_*, K_*) is not a symmetry of **l**,**m**
        end if
      end for
      Symmetry (Z_*, A_*, K_*) found between **l**,**m**
    end if
  end for
end for

consists of 8 heads arranged in an octagonal ring. Each head comprises two modules, each based on a LYSO matrix of 702 crystals of 1.6 mm × 1.6 mm × 12 mm directly coupled to a 64 anodes PMT (Hamamatsu H8500). Each head can acquire coincidences with the three heads facing it, therefore the scanner can acquire coincidences from 12 head pairs and features $(702 \cdot 2)^2 \cdot 12 = 23\ 654\ 592$ LORs. The detector implements a component based normalization procedure [17].

2.4. SSA validation methods

To show that the SSA can be operated both in an automatic and in a semi-automatic way, two types of experiments were performed in this study. In the first experiment, an SRM, named hereafter SRM_O (Original), including only the TORs corresponding to the head pairs shown in Figure 2 was used. In this configuration, the other TORs are obtained by rotating the image, e.g., the TORs belonging to the pair 3-7 can be obtained by rotating the image by 90° and using the corresponding TORs of the pair 1-5. In this work, a 3rd order b-spline interpolation is used to implement the image rotation. The manual exploitation of this set of rotational symmetries, allows to reduce the SRM computation time and its storage by a factor 6. SRM_O is used in this work as a 'gold standard', i.e. all the reconstructed images are compared to those obtained with SRM_O. In the second experiment, an SRM, named hereafter SRM_C (Complete), that contains all the TORs of the scanner was used. In both cases, the two models were the result of a multi-ray Siddon with 4×4 integration points on the crystal surface and 8 in the crystal depth [9], with the FOV sampled into pixels of 0.855 mm×0.855 mm×1.71 mm. This setup differs considerably from the size of the standard reconstructed image used in pre-

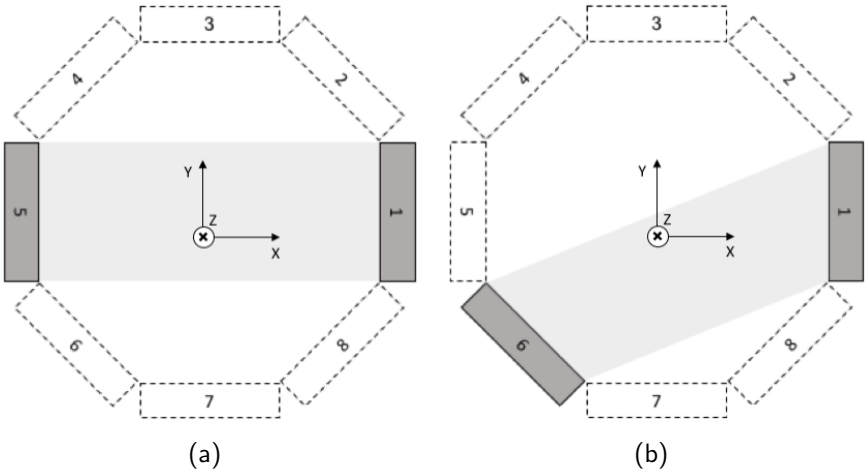


Figure 2. TORs included in SRM_O: the head pairs for which the TOR evaluation was actually performed are shown in gray.

clinical routine (0.42 mm×0.42 mm×0.855 mm). The large pixel size was necessary in order to limit the size of SRM_O and being able to run reconstruction without using the symmetries on the 64 GB RAM workstation available for this work. To show that the SSA does not affect the quality of the reconstructed images, a ^{18}FDG filled NEMA Image Quality (IQ) phantom [23], with an initial activity of 37 MBq (100 μCi), was acquired for 20 minutes with the IRIS PET scanner. The IQ was reconstructed performing 100 MLEM iterations, using the models mentioned above and correcting for decay time, dead-time and using detector normalization [17]. Two analysis were performed on the reconstructed images of the IQ: first, the absolute values of the images reconstructed with the compressed models and those obtained with SRM_O were compared. The differences between the images reconstructed with different models were evaluated using a relative difference metric: for each voxel i and iteration q the following value was computed

$$\Delta I_i^q = \frac{C_i^q - O_i^q}{O_i^q} \tag{6}$$

where O_i^q and C_i^q are the values of the i-th voxel at the q-th iteration obtained with SRM_O and with one of the compressed models (C), respectively. All the relative values reported in this paper are expressed in decimal representation rather than in percentages, e.g., 0.01 instead of 1%. The results of this analysis were resumed in three figures of merit: the maximum relative difference between two images $M_q = \max_i(|\Delta I^q|)$, the average relative difference $A_q = avg_i(|\Delta I^q|)$ and the standard deviation of the relative difference $S_q = std_i(|\Delta I^q|)$. As a second analysis, an evaluation of the image quality according to the NEMA standard NU-2008 for small animal was performed [23]. In order to assess the image quality, the NEMA prescribes to investigate two figures of merit: the recovery coefficients and the image uniformity. For studying the uniformity, a cylindrical region of interest of diameter 22.5 mm and height 10 mm (VOI) was drawn

Threshold	SRM_O size	Compression Factor	# of Fundamental TORs
-	52 GB	-	3942432
0.01	2.4 GB	22	141468
0.05	1.3 GB	40	72786
0.5	1.1 GB	47	57945
1	1.0 GB	52	56975
2	1.0 GB	52	56344
∞	984 MB	54	55449

Table 1. Size and Compression Factors (CF) of SRM_O as obtained at different thresholds.

in the uniform region of the phantom. The image uniformity was then evaluated as

$$IU = \frac{\text{std(VOI)}}{\text{mean(VOI)}} \tag{7}$$

where std(VOI) and mean(VOI) are the standard deviation and mean of the VOI region, respectively. The Recovery Coefficient (RC) of the n-th rod was evaluated as

$$RC_n = \frac{\max(ROD_n)}{\text{mean(VOI)}} \tag{8}$$

where ROD_n is a cylindrical region centered on the n-th rod, with two times the diameter of the rod and with a height of 10 mm. The idea behind this analysis is that RCs are correlated with the spatial resolution of the system, whereas the image uniformity, is intended to quantify the spatial noise perceived in an individual image.

3. Results

3.1. Compression factors

The sizes of SRM_O and SRM_C are respectively 52 GB and 346 GB. SRM_O and SRM_C were compressed using different thresholds ranging from 0.01 to 2 and discarding condition B, which is equivalent to set $t = \infty$. Each model produced by the SSA was named according to the model compressed and the threshold used, e.g., the name of the model obtained compressing SRM_O with $t = 0.01$ is $SRM_{O_{0.01}}$. The results achieved in terms of compression factors and number of fundamental TORs at different thresholds are shown in Table 1 and 2.

3.2. Computational performance

The SSA was implemented in C++ and was run on an Intel(R) Xeon(R) @ 3.50GHz. The time needed to compress a SRM depends on the threshold used and reaches a maximum when $t = \infty$ is used. The maximum compression time was found to be 3000 s for SRM_O and 15000 s for SRM_C. These numbers have to be compared with the time

Threshold	SRM_C size	Compression Factor	# of Fundamental TORs
-	346 GB	-	23654592
0.01	11 GB	31	605957
0.05	3.6 GB	96	194819
0.5	1.9 GB	182	106745
1	1.8 GB	192	101090
2	1.8 GB	192	98145
∞	1.7 GB	204	94015

Table 2. Size and Compression Factors (CF) of SRM_C as obtained at different thresholds.

needed to compute SRM_O and SRM_C, which are 42000 s and 250000 s, respectively. All these results refer to single core implementations of the SSA. To quantify the overhead due to the implementation of the symmetries detected by the SSA in the reconstruction software, two multi-core MLEM algorithms were implemented: the first does not make use of the SSA detected symmetries, the second implements equation (5) before projection and retro-projection operations. The reconstructions were run on the same workstation used for the compression tests. The time needed to perform a single MLEM iteration using any of the models compressed with the SSA was 67 s, regardless of the threshold used to produce the model. The time needed to perform the a single MLEM iteration using SRM_O, i.e. without the use of SSA detected symmetries, was 56 s.

3.3. Image reconstruction results

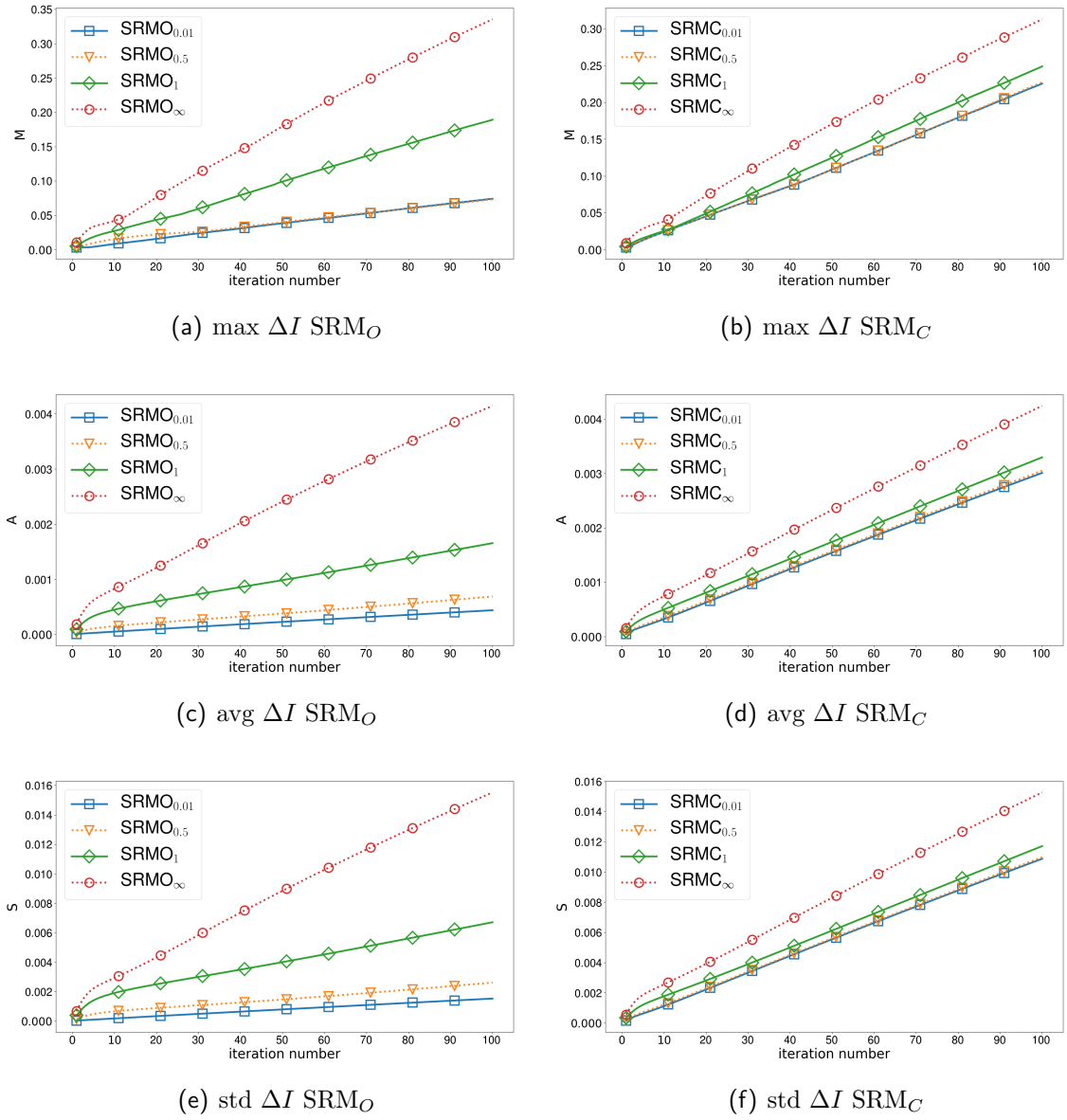
The plots of M_q, A_q, S_q vs the number of iterations (q) for some of the models produced by compressing SRM_O and SRM_C are reported in Figure 3. As an example, the distribution of values of ΔI^{100} are shown in Figure 4 for four models obtained compressing SRM_O.

3.4. NEMA Image quality

The Maximum Recovery Coefficients MRC obtained within the first 100 iterations and the iteration where the maximum was obtained are reported in Table 3. The results of all the models compressed using a finite threshold are not reported as they were exactly the same as those obtained with SRM_O. An IU of 6.75 % at the 100^{th} MLEM iteration was measured for all the models.

3.5. Compression of the YAP-PET small animal scanner model

An experiment using the model of the YAP-PET small animal scanner was performed. The model of the YAP-PET was computed as described in [4] and its size was 5.9 GB with the FOV segmented into 128×128×80 voxels of 0.375×0.375×0.5 mm^3. The



(a) max ΔI SRM_O (b) max ΔI SRM_C

(c) avg ΔI SRM_O (d) avg ΔI SRM_C

(e) std ΔI SRM_O (f) std ΔI SRM_C

Figure 3. Plot of M_q (top) and A_q (center) and S_q (bottom) vs the number of iterations with the models obtained compressing $\mathrm{SRM_O}$ (left column) and $\mathrm{SRM_C}$ (right column).

Rod / MRC	$\mathrm{SRM_O}$ (iteration)	$\mathrm{SRM_{O_\infty}}$	$\mathrm{SRM_{C_\infty}}$ (iteration)
1 mm	0.21±0.03 (100)	0.21±0.03 (100)	0.21±0.03 (100)
2 mm	0.50±0.07 (100)	0.50 ±0.07 (100)	0.51±0.07 (100)
3 mm	0.81±0.06 (47)	0.80 ±0.06 (47)	0.81±0.06 (46)
4 mm	0.93±0.05 (25)	0.92±0.05 (25)	0.93±0.05 (25)
5 mm	0.91±0.05 (19)	0.91±0.05 (19)	0.91±0.05 (18)

Table 3. Maximum recovery coefficients of the NEMA IQ phantom obtained using $\mathrm{SRM_O}$, $\mathrm{SRM_{O_\infty}}$ and $\mathrm{SRM_{C_\infty}}$.

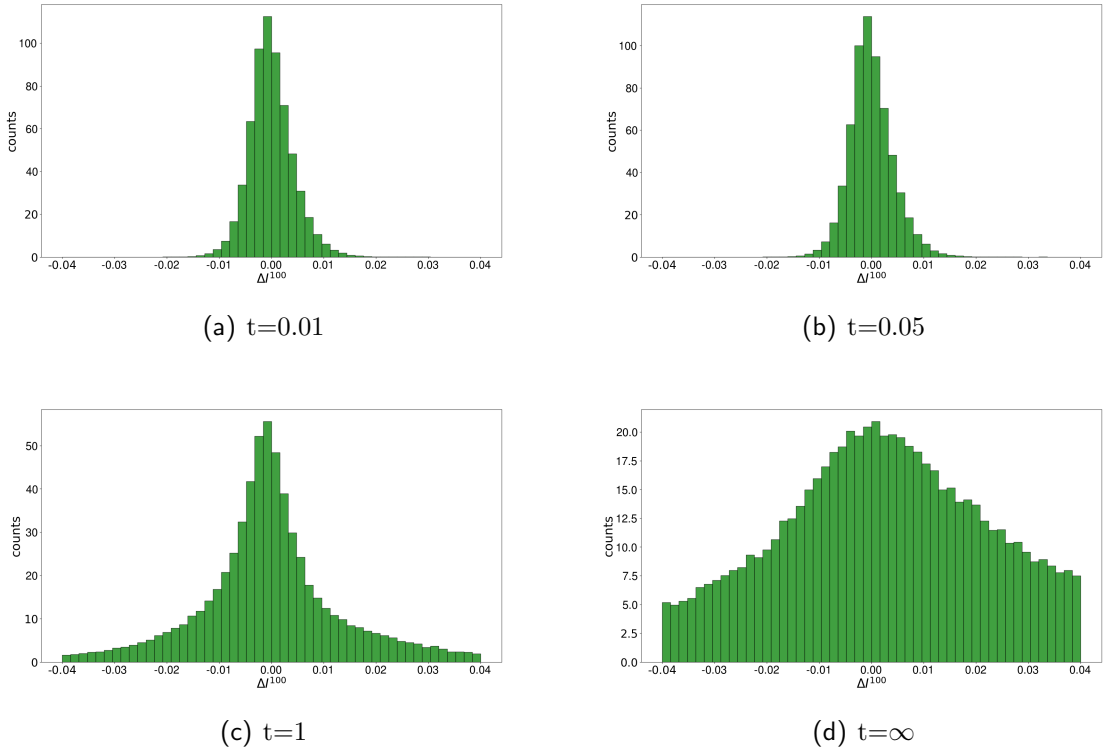


Figure 4. Histograms of the pixel-wise relative difference at the 100^{th} iteration (ΔI^{100}) for $\mathrm{SRM}_{\mathrm{O}_{0.01}}$, $\mathrm{SRM}_{\mathrm{O}_{0.05}}$, $\mathrm{SRM}_{\mathrm{O}_1}$, $\mathrm{SRM}_{\mathrm{O}_\infty}$.

maximum compression obtained with the SSA using $t = \infty$ is 92. This compression factor is comparable with that provided by the authors using manual implementation of reflections and translations (∼80).

4. Discussion

The data contained in Table 1 and Table 2 show that by varying the SSA threshold it is possible to control the size of the compressed models. A reduction of the size of a compressed model corresponds to a reduction of the number of its fundamental TORs. The compression factors obtained using $t = \infty$ represent the highest achievable by the SSA using a specific configuration, e.g. projector model, geometry and FOV sampling. Thanks to the strategies described in Sec. 2.3, the time needed to perform an SRM compression is negligible with respect to its computation time, e.g. 3000 s vs 42000 s for $\mathrm{SRM}_{\mathrm{O}}$. Moreover, the time needed to compress $\mathrm{SRM}_{\mathrm{C}}$, which has 6 times the TORSs of $\mathrm{SRM}_{\mathrm{O}}$, increases linearly with number TORs involved. This suggests that performing compressions of larger models is feasible. Therefore, this aspect does not limit the applicability of the SSA.

In principle, the number of symmetries found by the SSA could have an impact on the MLEM iteration time, since equation 5 needs to be implemented before projection and

retro-projection operations in order to obtain the non-fundamental TORs. However, the experiments show that the time needed to perform an MLEM iteration is roughly independent from the compressed model used and each iteration is 11 sec slower when using a model compressed with the SSA.
The models compressed with the SSA were also characterized in terms of the image quality. For this aim, experiments using several thresholds were carried out. Figure 3 shows the maximum relative difference (M_q), the average (A_q) and the standard deviation (S_q) of the relative difference with respect to the images reconstructed with $\mathrm{SRM_O}$. Both M_q, A_q and S_q were found to increase with the number of iterations. The value of M_q at the 100^{th} iteration is, in general, non-negligible. However, despite the maximum difference found, both the A_q and S_q are relatively small. This shows that large deviations from the original model take place only in a limited number of voxels. This is also confirmed by the results of NEMA IQ analysis: the Maximum Recovery Coefficients (MRC) obtained for each rod, together with the iteration where the maximum value is obtained, are reported in Table 3. The MRC of the 4^{th} and 5^{th} rod obtained with $\mathrm{SRM_{O_\infty}}$ are instead slightly worse than those obtained with $\mathrm{SRM_O}$ but still within the error. A similar behavior is found for $\mathrm{SRM_{C_\infty}}$. No appreciable differences were found comparing the results obtained with $\mathrm{SRM_O}$ and all the models obtained with a finite threshold. These experiments show that the image quality is not compromised by the compression performed with the SSA. In particular, with the IRIS small animal PET scanner configuration, a maximum compression factor of 54 was obtained when using a combination of manual symmetries and the SSA. A maximum compression factor of 204 was found when using fully automatic SSA compression. It is worth to notice, that the SSA is designed to detect symmetries in SRM components that are not object-dependent. This aspect does not represent a limitation as the object dependent corrections, e.g. attenuation, scatter and positron range, can be accounted for into the reconstruction through other mechanisms: attenuation can be factored out in a diagonal matrix, an estimation of the scatter counts can be inserted directly into the MLEM algorithm [24] and positron range can be modeled directly in the image space [25].

5. Conclusions

In this paper, an algorithm that automatically detects the symmetries encoded in the SRM is proposed. This algorithm was named Symmetry Search Algorithm (SSA) and provides, at the same time, a very accurate approximation of the original model and relieves the user from the task of figuring out the symmetries of the system manually. Even if the algorithm is not able, by design, to detect all the symmetries present in the SRM, i.e. it works for translations, reflections, and coordinate swap symmetries, it could provide enough compression to allow for storing the SRM on RAM. The experiments described in this paper were performed using an SRM with a rather big pixel size with respect to that used in pre-clinical routine. Nonetheless, similar compression factors

and a similar image metrics are observed regardless of the SRM pixel size.

References

[1] S. Tong, A. M. Alessio, and P. E. Kinahan. Image reconstruction for PET/CT scanners: past achievements and future challenges. *Imaging in Medicine*, 2(5):529–545, October 2010.

[2] A. Iriarte, R. Marabini, S. Matej, C.O.S. Sorzano, and R.M. Lewitt. System models for pet statistical iterative reconstruction: A review. *Computerized Medical Imaging and Graphics*, 48:30 – 48, 2016.

[3] J. L. Herraiz, S. Espaa, J. J. Vaquero, M. Desco, and J. M. Udas. First: Fast iterative reconstruction software for (pet) tomography. *Physics in Medicine and Biology*, 51(18):4547, 2006.

[4] A. Motta, C. Damiani, A. Del Guerra, G. Di Domenico, and G. Zavattini. Use of a fast EM algorithm for 3d image reconstruction with the yappet tomograph. *Computerized Medical Imaging and Graphics*, 26(5):293 – 302, 2002.

[5] J.J. Scheins, H. Herzog, and N.J. Shah. Fully-3d pet image reconstruction using scanner-independent, adaptive projection data and highly rotation-symmetric voxel assemblies. *Medical Imaging, IEEE Transactions on*, 30(3):879–892, March 2011.

[6] J. J. Scheins, K. Vahedipour, U. Pietrzyk, and N. J. Shah. High performance volume-of-intersection projectors for 3d-pet image reconstruction based on polar symmetries and simd vectorisation. *Physics in Medicine & Biology*, 60(24):9349, 2015.

[7] G. Sportelli, J. E. Ortuo, J. J. Vaquero, M. Desco, and A. Santos. Massively parallelizable list-mode reconstruction using a monte carlo-based elliptical gaussian model. *Medical Physics*, 40(1):–, 2013.

[8] G. Pratx, G. Chinn, P.D. Olcott, and C.S. Levin. Fast, accurate and shift-varying line projections for iterative reconstruction using the gpu. *Medical Imaging, IEEE Transactions on*, 28(3):435–445, March 2009.

[9] S. Moehrs, M. Defrise, N. Belcari, A. Del Guerra, A. Bartoli, S. Fabbri, and G. Zanetti. Multi-ray-based system matrix generation for 3d PET reconstruction. *Physics in Medicine and Biology*, 53(23):6925, 2008.

[10] J. Cabello and M. Rafecas. Comparison of basis functions for 3d pet reconstruction using a monte carlo system matrix. *Physics in Medicine and Biology*, 57(7):1759, 2012.

[11] Chien-Min Kao, Yun Dong, Qingguo Xie, and Chin-Tu Chen. Accurate image reconstruction with computed system response matrix for a high-sensitivity dual-head pet scanner. *IEEE Trans. Med. Imaging*, pages 1346–1358, 2008.

[12] V.Y. Panin, F. Kehren, C. Michel, and M. Casey. Fully 3-d pet reconstruction with system matrix derived from point source measurements. *Medical Imaging, IEEE Transactions on*, 25(7):907–921, July 2006.

[13] M. S. Tohme and J. Qi. Iterative image reconstruction for positron emission tomography based on detector response function estimated from point source measurements. *Physics in medicine and biology*, 54(12):3709–3725, May 2009.

[14] K Li, M Safavi-Naeini, D R Franklin, Z Han, A B Rosenfeld, B Hutton, and M L F Lerch. A new virtual ring-based system matrix generator for iterative image reconstruction in high resolution small volume pet systems. *Physics in Medicine & Biology*, 60(17):6949, 2015.

[15] C. Chou, Y. Dong, Y. Hung, Y. Kao, W. Wang, C. Kao, and C. Chen. Accelerating image reconstruction in dual-head pet system by gpu and symmetry properties. *PLos ONE*, 7(12):1–12, 12 2012.

[16] I. K. Hong, S. T. Chung, H. K. Kim, Y. B. Kim, Y. D. Son, and Z. H. Cho. Ultra fast symmetry and simd-based projection-backprojection (ssp) algorithm for 3-d pet image reconstruction. *IEEE Transactions on Medical Imaging*, 26(6):789–803, June 2007.

[17] N. Belcari, N. Camarlinghi, E. Fabbiani, S. Ferretti, P. Iozzo, D. Panetta, P. A. Salvadori,

G. Sportelli, and A. Del Guerra. NEMA NU-4 performance evaluation of the iris pet/ct preclinical scanner. *IEEE Transactions on Radiation and Plasma Medical Sciences*, PP(99):1–1, 2017.

[18] P. Mollet, K. Deprez, B. Vandeghinste, S. Neyt, R. Marcinkowski, S. Vandenberghe, and R. Van Holen. The -cube, a high-end compact preclinical benchtop pet for total body imaging. *Journal of Nuclear Medicine*, 58(supplement 1):393, 2017.

[19] N Camarlinghi, G Sportelli, G Battistoni, N Belcari, M Cecchetti, G A P Cirrone, G Cuttone, S Ferretti, A Kraan, A Retico, F Romano, P Sala, K Straub, A Tramontana, A Del Guerra, and V Rosso. An in-beam pet system for monitoring ion-beam therapy: test on phantoms using clinical 62 mev protons. *Journal of Instrumentation*, 9(04):C04005, 2014.

[20] M.G. Bisogni, A. Attili, G. Battistoni, N. Belcari, N. Camarlinghi, P. Cerello, S. Coli, A. Del Guerra, M.A. Piliero, G. Pirrone, A. Rivetti, M.D. Rolo, V. Rosso, P. Sala, G. Sportelli, and R. Wheadon. INSIDE in-beam positron emission tomography system for particle range monitoring in hadrontherapy. *Journal of Medical Imaging*, 4:4 – 4 – 12, 2016.

[21] Y. Vardi, L. A. Shepp, and L. Kaufman. A statistical model for positron emission tomography: Rejoinder. *Journal of the American Statistical Association*, 80(389):34–37, March 1985.

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

[23] *National Electrical Manufacturers Association. NEMA Standard Publication NU 4-2008: Performance Measurements of Small Animal Positron Emission Tomographs. Rosslyn, VA: National Electrical Manufacturers Association; 2008.*

[24] P.E. Valk. *Positron Emission Tomography: Basic Sciences*. Springer, 2003.

[25] O. Bertolli, A. Eleftheriou, M. Cecchetti, N. Camarlinghi, N. Belcari, and C. Tsoumpas. Pet iterative reconstruction incorporating an efficient positron range correction method. *Physica Medica*, 32(2):323 – 330, 2016.