ELEKTRİK, VOL.1, NO.1, February 1993, ©TÜBİTAK EMO

A New Algorithm and an Efficient Parallel Implementation of The EM Technique in Pet Imaging

Çağatay Büyükkoç
AT&T Bell Laboratories
Holmdel, NJ, 07733
Guiseppe Persiano
Harvard University
Cambridge, MA, 02138

Abstract

Position Emission Tomography (PET) is a medical diagnostic procedure that enables physicians to visually evaluate the metabolic activity in various organs. A new algorithm based on a modified version of the expectation maximization (EM) technique is proposed for solving the PET imaging problem. The algorithm is parallelized and implemented on AT & T's PIXEL machine which is a 64 processor, mesh-based parallel computer.

1. Introduction

Algorithms that are used in image reconstruction fall broadly into one of two categories [1-2]. First category is based on Fourier methods to reconstruct a function from its line integrals on certain regions. Second category is based on probabilistic reconstruction methods which take into account the stochastic nature of the problem and hence yield more accurate results under a broad range of parameters and data. In most general terms image reconstruction methods take the data that is passed through a general filtering mechanism and reconstruct the original image. In medical applications the amount of data that should be processed for this purpose is usually prohibitive for certain methods due to storage and time limitations. In this paper we will describe methods to help overcome these difficulties using various techniques.

Positron emission tomography (PET) is medical diagnostic procedure [3-4] that enables physicians to visually evaluate the metabolic activity in various organs. Rather than generating 'static' pictures as in X-rays, PET introduces low levels of positron emitting radioactive material in the organ under study, and levels of absorbtion on various parts of the organ is measured by the PET scanner [5]. The type of biochemical and the radioactive material used depends on the organ to be studied. It is known that the brain uses glucose as a primary energy source and therefore glucose 'injected' with radioactive material is used for brain studies, for the study of heart deoxyglucose and palmitic acid injected with the radioactive material is used. Depending on the metabolization of the injected material one can generate a 'dynamic' picture of the organ. For example if the brain is studied, the patient's psychosomatic condition at the time of the study will be different than that at a different time and condition. This hopefully will reveal valuable information on the patient's condition and the effect of various treatments (schizophrenia, etc.). There is a broad literature on PET imaging and the interested reader is referred to [4-5] and the references there.

2. The Model

A simplified model describing the physics of PET is shown in Figure 1.

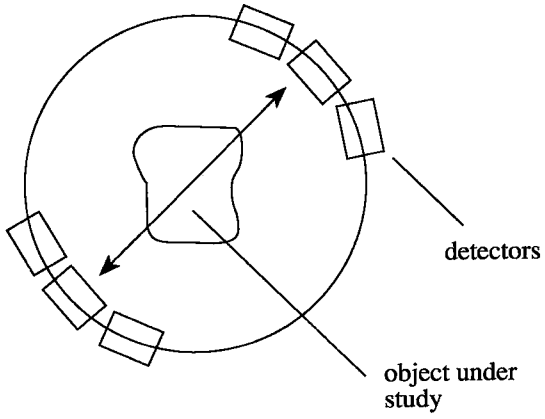

Figure 1. The simplified model of PET camera.

For description purposes we assume that the organ under investigation is the brain. The glucose injected with radioactive material goes around the brain and the substance is deposited in various parts proportional to the uptake mechanism. When the radioactive substance emits a positron, it will annihilate an electron situated nearby and a pair of x-ray photons will be generated. This pair will travel at opposite directions in a line at a uniformly distributed angle around the point of annihilation. The 'PET camera' consists of detectors that are situated around a circle as shown in Figure 1. The mechanism of the detectors allows for only photon pairs to be counted. That is, two opposite detectors will register a count only if both of them receive photons at approximately the same time (within their detection window).

Before going to the data collection mechanism we review some elementary properties of the Poisson distribution [6]. A nonnegative integer valued random variable χ has Poisson distribution with parameter λ if

$$P(\chi = k) = e^{-\lambda}\frac{\lambda^k}{k!}.$$

The following properties are well known.

1. $E[\chi] = \lambda$

2. If $\{\chi_i\}_1^n$ are independent random variables each with a Poisson distribution with parameter λ_i then $\chi = \sum_1^n \chi_i$ has a Poisson distribution with parameter $\lambda = \sum_1^n \lambda_i$.

3. Let χ be a Poisson distributed random variable with parameter λ, and let χ_i be generated by picking each point of χ independently with probability $P_i (P_i - \text{thinning of } \chi)$ then, χ_i is a Poisson distributed random variable with parameter λP_i.

4. Suppose χ and $\{P_i\}_1^n$ are given and let χ_i be generated as in 3, then the conditional distribution of $(\chi_1, ..., \chi_n)$ is multinomial with parameter $\{P_i\}_1^n$.

We partition the emission area by a grid. In this area the emissions are modeled by a two dimensional Poisson point process, that is, in box b, the emissions are assumed at a rate of λ_b. This is shown in Figure 2.

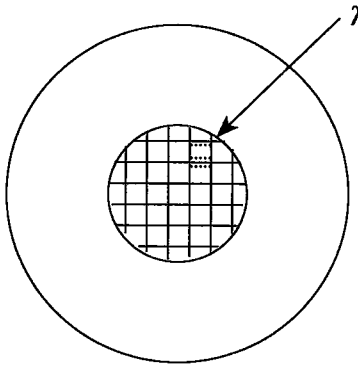

Figure 2. The discretization process

This is a slight oversimplification, however choosing a 'large' number of boxes provides satisfactory resolution [7].

As described above, a detector pair d can register a pair of x-ray photons if line of sight between them includes box b. This fact is used to calculate $\{P_{bd}\}$ as shown in Figure 3 for every box b and every detector pair d.

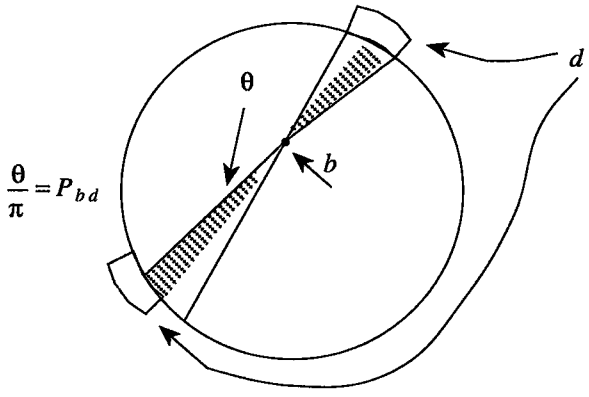

Figure 3. Calculation of P_{bd}

Here P_{bd} denotes the probability that a photon pair emitted by box b will be detected by detector pair d. The inherent assumptions are that each box is emitting photons that follow a Poisson distribution with unknown parameter λ_b, independent of its neighbors and that these photons are independently registered (P_{bd} thinned) by detector pairs.

The relevance of these simplifying assumptions to the properties of Poisson processes described above should be clear. There are various ways to improve the model to accommodate the fact that photon pairs do not follow an exact line of flight, photons are absorbed or deflected in tissue, differential time of flight for a pair of photons might not be negligible etc. Most of these factors can be accommodated into P_{bd}'s [7].

3. Estimation Problem

The model used above almost suggests a maximum likelihood estimation problem. The unknown Poisson random variables χ_b with parameters λ_b are P_{bd}-thinned and n_d, the detector counts, are to be used to estimate the λ_b.

Maximum likelihood methods have good theoretical properties and in a Poisson distribution framework it is especially desirable for its tractability [8]. However, it can be shown that one can equally choose a least squares estimation procedure resulting in the same estimates. Maximum likelihood methods were first used

by Gauss in 1821 and have a clearly intuitive argument behind them. Suppose $p(x,\theta)$ is a density function (or frequency function) where θ is the value of the parameters vector in a parameter space Θ. Now, consider $p(x,\theta)$ as a function of θ for fixed x. This function is called the likelihood function $L(\theta,x)$. If x is discrete then $L(\theta,x)$ is the probability of observing x under the parameter θ. Thus we can think of $L(\theta,x)$ as a measure of how 'likely' θ is to have produced x. Hence the method of maximum likelihood is to find the value of $\hat{\theta}(x)$ if any, which is 'most likely' to have produced the data. The problem is to seek a solution to

$$\begin{aligned} L(\hat{\theta}(x),x) = p(x,\hat{\theta}(x)) &= \max\{p(x,\theta),\theta\in\Theta\} \\ &= \max\{L(\theta,x);\theta\in\Theta\} \end{aligned}$$

In practice, in independent Poisson cases, for obvious reasons log-likelihood equation is used. The likelihood equation in our case can be easily seen to be

$$L(n,\lambda) = \prod_d e^{-\sum_{b,d}\lambda_b P_{bd}} \frac{[\sum_b \lambda_b P_{bd}]^{n_d}}{n_d!} \tag{1}$$

after making the proper identification that x is the fixed data vector n, representing the counts at the detectors, and θ is the parameter vector λ. In the sequel summations over λ's and n's are assumed to be over, $1,...,B$ and $1,...,D$ respectively irrespective of what index is used unless otherwise specified. Equation 1 makes use of the properties of Poisson distribution stated earlier.

$$\begin{aligned} \ell(n,\lambda) := \log L(n,\lambda) &= -\textstyle\sum_{b,d}\lambda_b P_{bd} + \sum_d [n_d \log \sum_b \lambda_b P_{bd} - \log n_d!] \\ &= -\textstyle\sum_b \lambda_b + \sum_d [n_d \log \sum_b \lambda_b P_{bd} - \log n_d!] \end{aligned} \tag{2}$$

Second equation follows from the fact that $\sum_d P_{bd} = 1$. This follows from the assumption that all photons are detected. However this is not a serious assumption since a standard change of variable and normalization procedure can be used. Using (2) we can get the first and second derivatives of the log-likelihood equation,

$$\frac{\partial \ell}{\partial \lambda_b} = -1 + \sum_d \frac{n_d P_{bd}}{\sum_i \lambda_i P_{id}} \tag{3}$$

and

$$\frac{\partial^2 \ell}{\partial \lambda_i \partial \lambda_j} = -\sum_d \frac{n_d P_{id} P_{jd}}{[\sum_k \lambda_k P_{kd}]^2} \tag{4}$$

It is easy to show, using Equation 4, that $\ell(\lambda,n)$ ise a concave function of λ, hence if a finite maximum exists it is essentially unique. Hence Kuhn-Tucker theorem, which is a necessary and sufficient condition for $\hat{\lambda}$ to be a maximizer of $\ell(\lambda,n)$, implies that, using Equation 3,

$$-1 + \sum_d \frac{n_d P_{bd}}{\sum_i \hat{\lambda}_i P_{id}} = 0 \quad \text{if} \quad \hat{\lambda}_b \neq 0$$

$$-1 + \sum_d \frac{n_d P_{bd}}{\sum_i \hat{\lambda}_i P_{id}} \leq 0 \quad \text{if} \quad \hat{\lambda}_b = 0. \tag{5}$$

Multiplying both sides in the equality of Equation 5 by $\hat{\lambda}_b \neq 0$, and summing, one gets the identity that the solution vector must satisfy,

$$\sum_k \hat{\lambda}_k = \sum_l n_l.$$

In the sequel we shall use an important information theoretic formula, the Kullback-Leibler Information (KLI) [6]. Let $\mu := \{\mu_i\}$ and $\nu := \{\nu_i\}$ be finite discrete probability distributions on a probability space $\{\Omega, A\}$. (It is not necessary that they be probability distributions as long as $\sum_i \mu_i = \sum_i \nu_i$.) Also, let μ be absolutely continuous with respect to $\nu, \mu \ll \nu$. Then,

$$KLI := \sum_i \mu_i \log \frac{\mu_i}{\nu_i} \tag{6}$$

For many of KLI's properties and applications see [9]. It is easy to see that $KLI \geq 0$.

First, using Equation 3 we propose the following update mechanism that will converge to the optimal value of λ. This result is proved in [7,11] however we present a more concise proof.

Proposition :

Let

$$\lambda_k^{i+1} = \lambda_k^i \cdot \sum_d \frac{n_d P_{kd}}{\sum_l \lambda_l^i P_{ld}} \tag{7}$$

Then, λ_k^i converges to $\hat{\lambda}_k$ and $\hat{\lambda} = [\hat{\lambda}_1, ..., \hat{\lambda}_B]$ maximizes the likelihood function provided $\lambda_k^0 > 0$. The proof uses the following essential Lemma.

Lemma :

The likelihood function strictly increases in the update rule given by Equation 7 unless λ has already converged, i.e., $L(n, \lambda^{i+1}) \geq L(n, \lambda^i)$ with equality iff $\lambda^{i+1} = \lambda^i$.

Proof :

Let $\Delta := \ell(\lambda^{i+1}) - \ell(\lambda^i)$ denote the difference of the log-likelihood function. Then using Equation 2,

$$\Delta = -\sum_b \lambda_b^{i+1} + \sum_b \lambda_b^i + \sum_d n_d \log \frac{\sum_b \lambda_b^{i+1} P_{bd}}{\sum_j \lambda_j^i P_{jd}} \tag{8}$$

However it is easy to see that $\sum_b \lambda_b^i = \sum_d n_d$ for any i, so that the update rule guarantees that λ vector stays on the simplex

$$\sum_b \lambda_b^0 = constant, \lambda_b^0 > 0. \tag{9}$$

Hence, using Equation 7 in Equation 8,

$$\Delta = \sum_d n_d \log \frac{\sum_b \lambda_b^i [\sum_\ell \frac{n_\ell}{\sum_m \lambda_m^i P_{m\ell}} P_{b\ell}] P_{bd}}{\sum_j \lambda_j^i P_{jd}}$$

Using concavity of log function and applying Jensen's inequality

$$\Delta \geq \sum_d n_d \sum_b \frac{\lambda_b^i P_{bd}}{\sum_j \lambda_j^i P_{jd}} \log [\sum_\ell \frac{n_\ell}{\sum_m \lambda_m^i P_{m\ell}} P_{b\ell}]$$

Now multiplying the numerator and the denominator of the argument of the log function by λ_b^i one obtains using Equation 7 and KLI,

$$\Delta \geq \sum_b \lambda_b^{i+1} \log \frac{\lambda_b^{i+1}}{\lambda_b^i}$$
$$> 0 iff\ \lambda_b^{i+1} \neq \lambda_b^i \text{ for } any\ b.$$

Hence the update rule strictly increases the concave log-likelihood function in the bounded simplex described by Equation 7.

The iteration of Equation 7 implies that each component of λ vector is modified multiplicatively as follows:

$$\lambda_k^{i+1} = \lambda_k^i \cdot \sum_d \frac{n_d}{\hat{n}_d} P_{kd} \tag{10}$$

where $\hat{n}_d = \sum_k \lambda_k^i P_{kd}.n_d$ represents the estimated value of the emission counts based on the current emission intensities. The procedure calls for the calculation of a modification factor as the expected value of the "error" $n_d/\hat{n}_d$. Suppose $n_l/\hat{n}_l$ is greater than 1, then the contribution of this term is larger than other terms where that ratio is less than 1. Note that $\Sigma\hat{n}_d = \Sigma n_d$ and hence both cases always exist unless the algorithm already converged.

This algorithm is the Expectation Maximization (EM) algorithm [10-11] and it is well known that the convergence rate of this algorithm in notoriously slow [12]. To achieve much faster convergence rates we propose the following modification.

$$\lambda_k^{i+1} = C_i \cdot \lambda_k^i \cdot \sum_d [\frac{n_d}{\hat{n}_d}]^{M_i} P_{kd} \tag{11}$$

Here C_i denotes a constant chosen to satisfy $\sum_k \lambda_k^{i+1} = \sum_k \lambda_k^i$ and $M_i \geq 1$ is an integer. The intuition behind Equation 11 is already given above. If $n_l/\hat{n}_l$ is greater than 1 that we could accentuate the effect of this term by raising to a power greater than 1. At initial stages of the algorithm this process intuitively will bring us closer to the solution point and eventually M_i can be set to 1 so that the algorithm is guaranteed to converge to the optimal point.

This idea is similar to the 'cooling schedules' in simulated annealing [13] and care should be exerciced in choosing M_i. Obviously decreasing M_i 'too slow' can take us too far from the optimal point. In Section 4 we shall present our implementation of the above algorithm on a mesh architecture AT&T PIXEL Machine.

4. Implementation of the Algorithm

In this section we describe our implementation of the algorithm described above on the AT&T PIXEL Machine. Before going into details, we briefly discuss the important features of the AT&T PIXEL Machine architecture.

4.1. The AT&T Pixel Machine

The PIXEL Machine is a mesh-based parallel computer. The version we use has 64 processors arranged in a 8×8 mesh. Each processor is connected to its North, East, West, South neighbor with wrap-around on each column and row; i.e. each column or row is actually a ring of 8 processors. The communication between processors is synchronous: each processor writes the data to be transferred into a (previously declared) output buffer and calls a synchronizing primitive that returns after all processors have called it; the transfer of data is

then performed and the incoming data is put in a (previously declared) input buffer. Before this exchange of data can take place processors have to declare to which neighbor the data is going and this is done by calling a direction-setting primitive. Because of hardware-design reasons, it is required that for any global exchange of data, all transfer between any pair of processors should be performed along the same direction; e.g. all processors send data to their North neighbor and receive data from the South neighbor. The direction of communication can be changed by calling the direction-setting procedure passing as parameter the desired direction. However, this is a timewise expensive operation and its use should be minimized.

Each processor is a DSP 32 processor. It has 36 Kbyte of static memory, where the program and variables used by the program reside. In addition it has a Video-Memory and a dynamic memory. The dynamic memory is a bank of 64 K addresses arranged in a 256 by 256 array. Each individual address can store a floating point number in the DSP format and can be accessed using the row and column reference. The Video-Memory contains the data relative to the pixels assigned to each processor and is arranged in a 256 by 256 array. In our implementation we had to content ourselves with 256 by 256 images, with each processor being assigned 1024 points and 128 detectors (that gave 8192 tubes).

4.2. Memory

As we have already mentioned, the main problem we had to face in our implementation was the limited amount of memory we could use and that had an important effect on the running time of the program. In the ideal situation, where each processor has enough memory to store all the data it needs, the following data would be needed by each processor,

P_{ij} : the probability that an emission at point i is detected in tube j. Each processor needs P_{ij} for all points assigned to it (1024) and all tubes (8192).

m_i : The estimated count in tube i after the current iteration. Each processor has to store m_i for all 8192 tubes.

n_i : the count of the observed number of emission in tube i. Each processor needs n_i for all 8192 tubes.

λ_i : the current value of the intensity at point i. Each processor needs to store the value λ_i for all 1024 points i assigned to it.

It is clear that, given the current memory limitation on AT&T PIXEL Machine, the above cannot be stored in the local memory of the processors. We choose not to store the P_{ij} and to recompute them each time we need it. However, one should be careful as the computation of a P_{ij} is a very time consuming computation and, moreover, for most of i and j this value is 0.

4.3. Computation of P_{ij}

The computation of the P_{ij} will turn out to be the most time consuming computation in our program and we have paid extra attention to reducing it. The following observation is crucial to reduce the cost of computing the P_{ij}. Let us fix a point i and a detector s_1. Now, there are very few other detectors s_2 (in our example no more than 6, with an average of around 3) for which the probability that an emission at location i be detected in the tube defined by the detectors s_1, s_2 in nonzero. Actually, it is straightforward to compute these detectors in the following way. Consider P_1 and P_2, the two endpoints of detector s_1, and $Q^1_{s_1}$ (respectively $Q^2_{s_2}$) be the intersection of the lines passing through the point i and P_1 (respectively P_2) with the circle of detectors. Then, by denoting by $L[s_1]$ and $H[s_1]$ the sectors of $Q^1_{s_1}$ and $Q^2_{s_2}$, the detectors s_2 are those that are between $L[s_1]$ and $H[s_1]$.

We have thus reduced the computation of 8192 P_{ij}'s per point to around 3 $\times$ 128 with a saving factor of 25. The algorithm above also fits naturally in the computation we have to perform. In fact, if we denote P_{ij} by

$P(i, s_1, s_2)$ where the $j-$th tube is defined by detectors s_1 and s_2, we obtain that the main computation of the algorithm

$$\lambda_k \leftarrow \lambda_k \cdot \sum_{s_1} \sum_{s_2} P(i, s_1, s_2) \frac{n[s_1, s_2]}{m[s_1, s_2]} \tag{12}$$

$$\lambda_k \leftarrow \lambda_k \cdot \sum_{s_1=1}^{128} \sum_{s_2=L[s_1]}^{H[s_1]} P(i, s_1, s_2) \frac{n[s_1, s_2]}{m[s_1, s_2]} \tag{13}$$

4.4. Computation of the Estimated Count

Our algorithm, being essentially a modified EM algorithm, can be seen as consisting of two different phases: the Expectation and the Maximization phase. In the Expectation phase the number of emissions detected by a tube is estimated, by its expected value, by assuming that the current solution is the right one. This is the only phase of our algorithm in which inter-processor communication is required. By denoting the estimate for the j-th tube by m_j, we have $m_j = \sum_i \lambda_i P_{ij}$ which can be written as $m_j = \sum_{p=1}^{64} \sum_{i \in I_p} \lambda_i P_{ij}$ where I_p denotes the set of points assigned to processor p. This gives a direct way of computing these estimates. Each processor p computes its contribution $\sum_{i \in I_p} \lambda_i P_{ij}$ to m_j and the contributions will be summed over all processors to give m_j. The summation needed to compute m_j can be performed in the following way. First, each processor passes to its NORTH neighbor his contribution while receiving the contribution of its SOUTH neighbor. The processor sums these two contribution and passes NORTH what he received from SOUTH. In this way, after 8 iteration each processor has the sum of the contributions of all the processors in its column. After that, the same process is repeated over the rows so that the final value m_j can be computed by each processor. The procedure we have just described involves just 2 changes of direction in the communication. To avoid having to do 2 changes of direction for each of the 8192 m_j's, we let each processor compute its contribution to all the m_j's before starting summing the contributions of all the processors. This way, we only need to change direction of communication twice in order to compute all the m_j's. This can be further reduced to just one change per each iteration, with the first iteration having 2 changes.

4.5. An Example

In this section we give the result on the example described below. For comparison purposes we chose a fixed exponent of 3 in the MEM. The results of the EM algorithm for each iteration are shown in Figure 4, top to bottom and left to right. The first picture is the target image in each of the figures. Figure 5 shows the same experiment with the MEM. It is clear that convergence to the target image is much faster in MEM. We have experimented with various exponent structures for the MEM and the results are very encouraging and will be reported elsewhere together with applications in other fields.

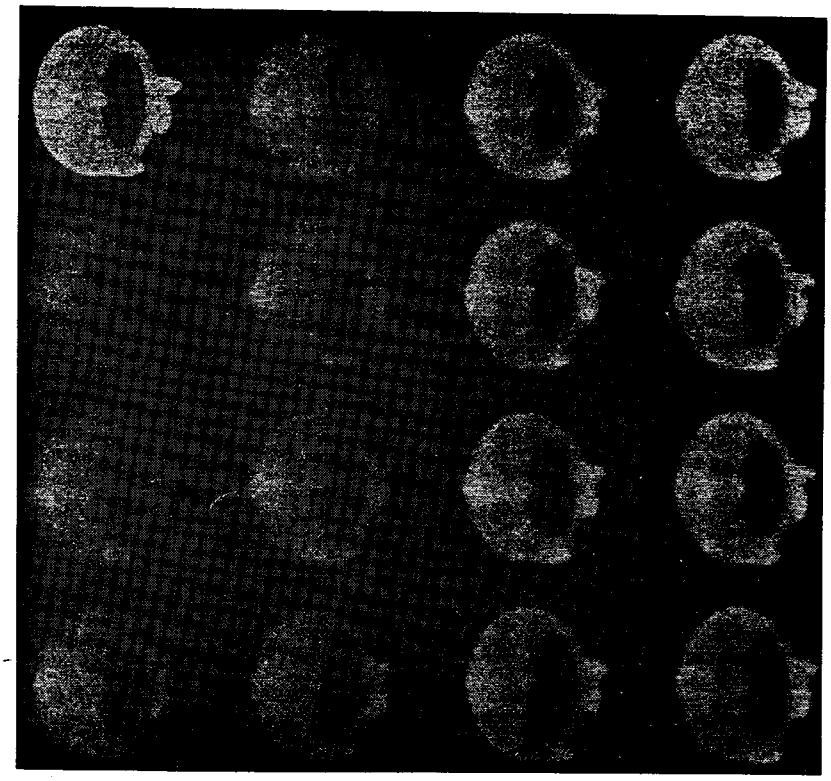

Figure 4. Iterations of EM algorithm

Figure 5. Iterations of MEM algorithm

5. Conclusion

The EM algorithm is used in many branches of applied probability [7,10,14,15], and MEM that we describe similarly can be used in those areas where the convergence rate is very crucial. The basic EM algorithm is a special case of broad range of algorithms described in [16]. Although the conditions of convergence of MEM are difficult to obtain in general, 'pessimistic' exponent structures can be used to guarantee convergence.For example, an exponent that converges very rapidly to 1 is guaranteed to converge, however the full benefits of the MEM will not be seen in this case. Further experimentation is necessary for MEM. Case by case fine tuning of MEM will be very beneficial in applications and is recommended.

REFERENCES

[1] Herman G.T., ed., "Topics in Applied Physics, vol. 32, Image reconstruction from projections: Implementation and Applications", Springer Verlag, Berlin 1979.

[2] Herman G. T., "Image reconstruction from projections: the fundamentals of computerized tomography", Academic Press, New York 1980.

[3] Ter-Pogossian M.M., Raichle M.E. and Sobel B.E., "Positron Emission Tomography", Scientific American, 243 (4), 170-181, 1980.

[4] Lange K. and Carson R., "EM Reconstruction Algorithms for Emission and Transmission Tomography", Journal of Computer Assisted Tomography (8), 302-316, 1984.

[5] Reivich M. and Alavi A., "Positron Emission Tomography", Alan R. Liss, Inc., New York 1985.

[6] Feller W., "An Introduction of Probability Theory and its Applications", Vol. 1 Wiley, New York 1968.

[7] Vardi Y., Shepp L. A. and Kaufman L., "A Statistical Model for Positron Emission Tomography", Journal of the American Statistical Association, (80), No. 389, 8-37, 1985.

[8] Bickel J.J. and Doksum K.A., "Mathematical Statistics", Holden Day, Oakland 1977.

[9] Kullback S., Keegel J.C. and Kullback J.H., "Topics in Statistical Information Theory", Springer-Verlag, Lecture Notes in Statistics, 1987.

[10] Little R.J.A. and Rubin D.B., "Statistical Analysis with Missing Data", Wiley, 1987.

[11] Dempster A.P., Laird N.M. and Rubin D.B., "Maximum Likelihood from Incomplete Data via the EM Algorithm", J. Royal Statistical Society, (B39), 1-38, 1977.

[12] Wu C.F.J., "On the convergence properties of the EM Algorithm", Annals of Statistics, (11), 95-103, 1983.

[13] Kirkpatrics S, Gelatt C.D. and Vecchi M.P., "Optimization by simulated annealing", Science 220, 671-680, 1983.

[14] Cover M.C., "An Algorithm for Maximizing Expected Log Investment Return", IEEE Tran. Inf. Th., (30), 369-373, 1984.

[15] Bell R. M and Cover T.M., "Competitive Optimality of Logarithmic Investment", Math. Op. Res., (5), 161-166, 1980.

[16] Csiszar I. and Tusnady G., "Information Geometry and Alternative Minimization Procedures", Statistics and Decisions, Supp. Is. 1, 205-237, 1984.