

**Instituto Biomédico de Investigação da Luz e Imagem – IBILI**

2009

**Reconstrução de Imagem PET  
Por  
Decomposição do Sinograma**



**Tese do Mestrado Integrado em Engenharia Biomédica**  
**elaborada por**

Hugo Alberto da Mota Martins Lages Fernandes

Tese supervisionada por Prof. Doutor Nuno Ferreira e Eng. Francisco  
Caramelo



# Abstract

Positron emission tomography (PET) is a medical imaging technique that provides information about the distribution of radiopharmaceuticals inside a living subject. A radiopharmaceutical is a radioactive compound (radionuclide and pharmaceutical) used for the diagnosis and therapeutic treatment of human diseases.

In the last years some mathematical models have been developed in PET taking into account the Poisson distribution of the detected events. One of these models is used in the popular Maximum Likelihood – Expectation Maximization (ML-EM) algorithm used in tomographic reconstruction.

In a 2D iterative image reconstruction perspective, for this project, we tried to know how two this algorithm relates to an algorithm developed in Instituto de Biofísica/ Biomatemática - IBILI, the Sinocurves algorithm. The Sinocurves algorithm has a particular point of view in which the sinogram is viewed as a stack of sinusoidal curves, so the 2D reconstruction problem can be understood as the decomposition of the sinogram into its sinusoidal curves.

Furthermore this thesis establishes a statistical comparison between two different estimators (mean and median) that can be used in Sinocurves algorithm, to evaluate the pixels convergence to their real value.

The results indicate that the mean is a better estimator than the median, for the image tested. However, these results must be simulated for other images, to get general conclusions and to establish rules.

It would be also interesting to evaluate Sinocurves algorithm for other estimators in the future.

**Keywords:** Positron Emission Tomography (PET), Poisson Distribution, Maximum Likelihood – Expectation Maximization (ML-EM) Algorithm, 2D Iterative Image Reconstruction, Sinocurves Algorithm, Sinogram Decomposition, Mean, Median.



# Resumo

A tomografia por emissão de positrões (PET) é uma técnica de imagem médica que fornece informação acerca da distribuição dos radiofármacos no corpo do paciente. Um radiofármaco consiste num composto radioactivo (radionúclido e fármaco) usado para diagnóstico e tratamento terapêutico de doenças humanas.

Nos últimos anos alguns modelos matemáticos foram desenvolvidos para PET, que tinham em conta a distribuição de Poisson dos eventos detectados. Um destes modelos é usado no popular algoritmo Maximum Likelihood – Expectation Maximization (ML-EM) para reconstrução tomográfica.

Numa perspectiva 2D para a reconstrução iterativa de imagem, neste projecto tentamos saber o quanto este algoritmo se relaciona com um algoritmo desenvolvido no Instituto de Biofísica / Biomatemática – IBILI, o algoritmo das Sinocurvas.

O algoritmo das Sinocurvas tem um ponto de vista particular, assumindo que o sinograma pode ser interpretado como um empilhamento de curvas sinusoidais, de tal forma que o problema de reconstrução em 2D pode ser solucionado por decomposição do sinograma nas suas curvas sinusoidais.

Ainda para esta tese foi estabelecida uma comparação estatística entre dois estimadores diferentes (média e mediana) para o algoritmo das Sinocurvas, para avaliar a convergência dos pixels para o seu valor real.

Os resultados mostram que a média é melhor estimador que a mediana, para a imagem testada. Contudo, estes resultados devem ser simulados para outras imagens, para obter conclusões gerais e estabelecer regras.

Seria interessante, de futuro, simular o algoritmo das Sinocurvas para outros estimadores.

**Palavras-chave:** Tomografia por Emissão de Positrões (PET), Distribuição de Poisson, Algoritmo Maximum Likelihood – Expectation Maximization (ML-EM), Reconstrução de Imagem Iterativa 2D, Algoritmo das Sinocurvas, Decomposição do Sinograma, Média, Mediana.



# Agradecimentos

Gostava de agradecer ao meu supervisor Prof. Nuno Ferreira pela forma como me dirigiu, encorajou e pela paciência que denotou quando discutíamos sobre os mais diversos aspectos relacionados com a reconstrução de imagem, âmbito desta tese.

Também gostava de agradecer ao meu co-supervisor Prof. Francisco Caramelo pela forma como me acompanhou e ajudou ao longo do ano, mantendo-me sempre no rumo certo.

Agradeço à Prof. Bárbara Oliveiros pela disponibilidade que teve em ajudar-me a interpretar de forma eficiente os resultados gerados no software SPSS. Agradeço ao Eng. Luís Mendes pela motivação e interesse que sempre me incutiu.

Agradeço ao Instituto Biomédico de Investigação da Luz e Imagem – IBILI por me fornecerem as condições necessárias à realização do meu trabalho.

Finalmente, agradeço aos meus amigos e à minha família pelo apoio incondicional que me deram no meu percurso universitário.





## Conteúdos

Abstract.....	ii
Resumo .....	iv
Agradecimentos.....	vi
Lista de Figuras .....	x
Lista de Tabelas .....	xiv
Símbolos e Abreviaturas.....	xvi
Introdução.....	1
Teoria.....	3
PET: Conceitos Básicos .....	3
Descrição do algoritmo EM .....	6
Descrição do algoritmo das Sinocurvas .....	10
Métodos .....	17
Resultados.....	21
Simulações de convergência sem ruído.....	25
Simulações de convergência com ruído .....	31
Estimador Média .....	31
Estimador Mediana.....	37
Discussão e Conclusões.....	43
Bibliografia.....	47



## Lista de Figuras

Figura 1-Formação do sinograma.....	4
Figura 2-Tipos de coincidências responsáveis, em parte, pelo ruído.....	5
Figura 3-O gráfico à esquerda representa um <i>pixel</i> numa imagem de reconstrução, denominado <i>imgpixel</i> , que corresponde a um conjunto de <i>pixeis</i> , designados por <i>sinopixeis</i> , no sinograma (gráfico à direita). O conjunto de <i>sinopixeis</i> define a curva sinusoidal - <i>sinocurva</i> . Retirada de [7]......	10
Figura 4-Decomposição do sinograma em <i>sinocurvas</i> .....	11
Figura 5-Imagem testada nos esquemas que mostram a equivalência dos algoritmos no caso particular de 2x2 <i>pixeis</i> . .....	13
Figura 6-Esquema 2x2 <i>pixeis</i> para algoritmo das <i>sinocurvas</i> : cálculo da 2ª estimativa. 14	
Figura 7-Esquema 2x2 <i>pixeis</i> para algoritmo das <i>sinocurvas</i> : cálculo da 3ª estimativa. 14	
Figura 8-Esquema 2x2 <i>pixeis</i> para algoritmo MLEM: Cálculo da 2ª estimativa.....	15
Figura 9-Esquema 2x2 <i>pixeis</i> para algoritmo MLEM: Cálculo da 3ª estimativa.....	15
Figura 10- Imagem de teste, delimitada por uma área a azul e um quadrado pequeno no interior desta (a branco). O quadrado grande (a preto) é designado por “vizinhança de zeros”. A escala de cores revela o valor dos <i>pixeis</i> para cada região. ....	17
Figura 11-Exemplo de <i>boxplot</i> .....	19
Figura 12 -Simulações sem ruído. 10 Iterações. Estimador média. Sem ruído. Eixos dos <i>xx</i> e <i>yy</i> representam o número de <i>pixeis</i> .Escala de cores identifica o valor do <i>pixel</i> . Imagem à esquerda representa a imagem real, imagem central representa imagem reconstruída enquanto que a imagem à direita representa a diferença entre as imagens anteriores. ....	21
Figura 13-Simulações com ruído. 10 Iterações. Estimador média. 10k contagens. Eixos dos <i>xx</i> e <i>yy</i> representam o número de <i>pixeis</i> . Escala de cores distingue o valor do <i>pixel</i> . Imagem à esquerda representa a imagem real, imagem central representa imagem reconstruída enquanto que a imagem à direita representa a diferença entre as imagens anteriores. ....	22
Figura 14-Simulações sem ruído. 10 Iterações. Estimador média. Eixo dos <i>xx</i> representa o índice do <i>pixel</i> e o eixo dos <i>yy</i> representa o número de , cuja primeira iteração 0 corresponde à faixa horizontal negra. Escala de cores identifica o valor do <i>pixel</i> .....	22

Figura 15-Simulações com ruído. 10 iterações. Estimador média. 10k contagens. Eixo dos xx representa o índice do <i>pixel</i> e o eixo dos yy representa o número de iterações, cuja primeira iteração 0 corresponde à faixa horizontal negra. Escala de cores distingue o valor do pixel.....	23
Figura 16-Simulações sem ruído. 50 Iterações. Estimador média. Eixo dos xx representa o número de iterações, eixo dos yy representa o erro quadrado entre a imagem verdadeira e a imagem reconstruída. ....	23
Figura 17-Evolução do valor estimado (eixo dos yy) para o <i>Pixel</i> 1 em função do número de iterações (eixo dos xx). Estimador média (gráficos superiores) e Estimador mediana (gráficos inferiores).....	25
Figura 18-Evolução do valor estimado (eixo dos yy) para o <i>Pixel</i> 2 em função do número de iterações (eixo dos xx). Estimador média (gráficos superiores) e Estimador mediana (gráficos inferiores).....	26
Figura 19-Evolução do valor estimado (eixo dos yy) para o <i>Pixel</i> 3 em função do número de iterações (eixo dos xx). Estimador média (gráficos superiores) e Estimador mediana (gráficos inferiores).....	27
Figura 20-Evolução do valor estimado (eixo dos yy) para o <i>Pixel</i> 5 em função do número de iterações (eixo dos xx). Estimador média (gráficos superiores) e Estimador mediana (gráficos inferiores).....	28
Figura 21-Evolução do valor estimado (eixo dos yy) para o <i>Pixel</i> 10 em função do número de iterações (eixo dos xx). Estimador média (gráficos superiores) e Estimador mediana (gráficos inferiores).....	29
Figura 22-Evolução do valor estimado (eixo dos yy) para o <i>Pixel</i> 1 em função do número de iterações (eixo dos xx). Do topo para baixo: 10k, 100k e 1M contagens.....	31
Figura 23-Evolução do valor estimado (eixo dos yy) para o <i>Pixel</i> 2 em função do número de iterações (eixo dos xx). Do topo para baixo: 10k, 100k e 1M contagens.....	32
Figura 24-Evolução do valor estimado (eixo dos yy) para o <i>Pixel</i> 3 em função do número de iterações (eixo dos xx). Do topo para baixo: 10k, 100k e 1M contagens.....	33
Figura 25-Evolução do valor estimado (eixo dos yy) para o <i>Pixel</i> 5 em função do número de iterações (eixo dos xx). Do topo para baixo: 10k, 100k e 1M contagens.....	34
Figura 26-Evolução do valor estimado (eixo dos yy) para o <i>Pixel</i> 10 em função do número de iterações (eixo dos xx). Do topo para baixo: 10k, 100k e 1M contagens.....	35
Figura 27-Evolução do valor estimado (eixo dos yy) para o <i>Pixel</i> 1 em função do número de iterações (eixo dos xx). Do topo para baixo: 10k, 100k e 1M contagens.....	37

Figura 28-Evolução do valor estimado (eixo dos yy) para o *Pixel 2* em função do número de iterações (eixo dos xx). Do topo para baixo: 10k, 100k e 1M contagens..... 38

Figura 29-Evolução do valor estimado (eixo dos yy) para o *Pixel 3* em função do número de iterações (eixo dos xx). Do topo para baixo: 10k, 100k e 1M contagens..... 39

Figura 30-Evolução do valor estimado (eixo dos yy) para o *Pixel 5* em função do número de iterações (eixo dos xx). Do topo para baixo: 10k, 100k e 1M contagens..... 40

Figura 31-Evolução do valor estimado (eixo dos yy) para o *Pixel 10* em função do número de iterações (eixo dos xx). Do topo para baixo: 10k, 100k e 1M contagens..... 41



## Lista de Tabelas

Tabela 1-Relação linha e coluna com a designação do pixel. ....	18
Tabela 2-Factor utilizado para computação, conforme o número de contagens no sinograma. ....	20
Tabela 3-Posicionamento dos Pixeis 2,3 e 5 em função de linha e coluna. ....	44





## **Símbolos e Abreviaturas**

ALARA	As Low As Reasonably Achievable
EM	Expectation-Maximization
FDG	Fluorodesoxiglucose
IDL	Interactive Data Language
LOR	Line of Response
ML-EM	Maximum Likelihood-Expectation Maximization
PET	Positron Emission Tomography
SPSS	Statistical Package for the Social Sciences



# Capítulo 1

## Introdução

No presente projecto pretendemos numa primeira abordagem demonstrar a equivalência entre o algoritmo das Sinocurvas desenvolvido no Instituto de Biofísica / Biomatemática do IBILI e o algoritmo *Maximum Likelihood-Expectation Maximization* (ML-EM) no plano teórico. A complementar o objectivo deste trabalho, estudamos as diferenças que dois estimadores usados no algoritmo das Sinocurvas possuem no estudo de *pixeis* com posicionamentos previamente definidos, de forma a inferir sobre vantagens/desvantagens no âmbito da convergência para o valor real do *pixel* com a variação do número de iterações.

Foram utilizados dois estimadores diferentes nas simulações alvo deste projecto, ou seja, o algoritmo das sinocurvas numa primeira fase usava como estimador a média e numa segunda fase aplicava a mediana como estimador. Procuramos, assim, averiguar as diferenças de comportamento, por parte de cada *pixel*, registadas em termos de convergência tendo em conta as regiões com diferentes valores de *pixeis*.

A imagem de teste usada permitiu-nos assumir a existência de 3 regiões distintas, nomeadamente, uma região com *pixeis* de valor 0, outra região com *pixeis* que assumiam o valor 1 e por último a região definida por *pixeis* de valor 5. O conceito fronteira/vizinhança de regiões pode assumir um papel primordial na forma como o valor estimado do *pixel* converge para o valor real deste, conforme verificado nas simulações realizadas.

Prosseguindo da computação para o tratamento de dados, foram criados *datasets*<sup>1</sup> que tinham presentes os resultados numéricos obtidos nas simulações. A partir destes dados foram gerados *errorbars*<sup>2</sup> e *boxplots*<sup>3</sup>. Os *errorbars* tinham especial interesse ao mostrar a tendência da média enquanto os *boxplots* demonstravam a tendência da mediana.

Resumindo, ao nível da computação as simulações em IDL compreenderam 2 fases, que ao nível de procedimento tiveram o mesmo tratamento com a única diferença

---

<sup>1</sup> Conjunto de dados

<sup>2</sup> Gráficos de barras com intervalos de confiança

<sup>3</sup> Caixas de bigodes ou diagramas de extremos e *quartis*

em que os estimadores assumidos pelo algoritmo foram diferentes, ora média ora mediana. Ao nível do tratamento dos dados obtidos na fase da simulação, foram gerados *boxplots* e *errorbars* para visualizar a convergência, dos diferentes estimadores.

A tese desenvolve-se em cinco capítulos, cujo primeiro capítulo consiste na presente introdução.

No capítulo 2, é apresentada a teoria subjacente a este projecto. Inicialmente descrevem-se conceitos básicos sobre PET (do inglês, *Positron Emission Tomography*) essenciais à compreensão do trabalho realizado neste projecto. Ainda neste capítulo são fornecidas descrições gerais sobre os algoritmos ML-EM e das Sinocurvas, onde também se procura mostrar que os algoritmos são equivalentes.

No capítulo 3 descrevem-se os métodos usados que possibilitaram atingir os objectivos deste projecto. Começa-se por descrever os aspectos relativos à computação do algoritmo das Sinocurvas e prossegue-se com uma descrição relativa ao tratamento dos dados, em que se procura descrever sucintamente os dois tipos de gráficos utilizados, respectivamente, *boxplots* e *errorbars*.

O capítulo 4, apresenta uma descrição dos resultados gerados para o algoritmo das Sinocurvas, procurando-se comparar e retirar ilações dos dois estimadores testados, média e mediana, para os quais foram realizadas simulações com e sem ruído. Nas simulações com ruído, variou-se o número de contagens no sinograma – 10 mil, 100 mil e 1 milhão de contagens.

No capítulo 5, apresenta-se um resumo dos resultados obtidos fazendo-se uma discussão final. Evidenciam-se, também, perspectivas futuras que podem/devem ser exploradas. Desta forma, neste capítulo, aborda-se a necessidade em testar novas imagens e utilizar outros estimadores na computação do algoritmo das Sinocurvas, estudando o impacto que estas alterações possam ter.

# Capítulo 2

## Teoria

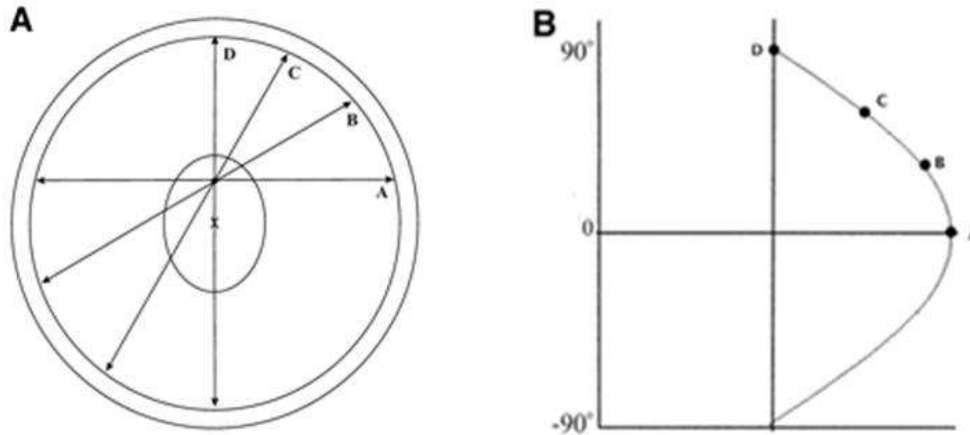
### 2.1 PET: Conceitos Básicos

A reconstrução de imagem é uma técnica que alia a computação à matemática procurando determinar a partir dos dados observados (projeções medidas num exame de tomografia), a distribuição original no paciente ou objecto (distribuição real da grandeza que se pretende medir). É de extraordinária importância ter um bom algoritmo que possibilite reconstruir os dados observados num intervalo de tempo razoável.

A Tomografia por Emissão de Positrões (PET, do inglês *Positron Emission Tomography*) é uma técnica minimamente invasiva que consiste em marcar um composto químico com um radioisótopo emissor de positrões (o conjunto composto químico e radioisótopo é geralmente designado por radiofármaco) para obter uma imagem da concentração do composto químico. Os radioisótopos frequentemente usados são o carbono-11 ( $^{11}\text{C}$ ), azoto-13 ( $^{13}\text{N}$ ), oxigénio-15 ( $^{15}\text{O}$ ) e o flúor-18 ( $^{18}\text{F}$ ) que se correlacionam fisiologicamente com o corpo humano uma vez que os compostos C,N,O existem em todos os tecidos vivos do corpo. São, por isso, de incorporação fácil no processo metabólico, actuando como traçadores sem que as suas propriedades químicas sejam alteradas quando em contacto com o composto químico.

O  $^{18}\text{F}$  é muito utilizado nos estudos de fluorodesoxiglucose (FDG).  $^{18}\text{F}$ -FDG é o radiofármaco mais comum em estudos PET. Após a administração do radiofármaco ao paciente, o radioisótopo, decai emitindo um positrão que é aniquilado com um electrão do meio, originando um par de fotões de 511 keV de energia, os quais se propagam em direcções aproximadamente opostas. Caso estes fotões interajam com os detectores, numa janela temporal curta (tipicamente da ordem do nanosegundo), diz-se que um evento de aniquilação ocorreu num local indeterminado da linha que une a posição da detecção de um fotão à posição de detecção do outro fotão. Esta linha é designada por *Line of Response* (LOR). As LORs organizam-se segundo projecções e acumulam as coincidências verdadeiras, isto é, detecção simultânea das duas emissões de fotões resultantes do mesmo decaimento. Uma das formas mais comuns de representar o conjunto das projecções é o sinograma.

Assim, o sinograma, consiste no registo de eventos de coincidência num *scanner* PET em que as LORs são descritas em função da orientação angular e da posição em relação ao centro do campo de visão (Figura 1).



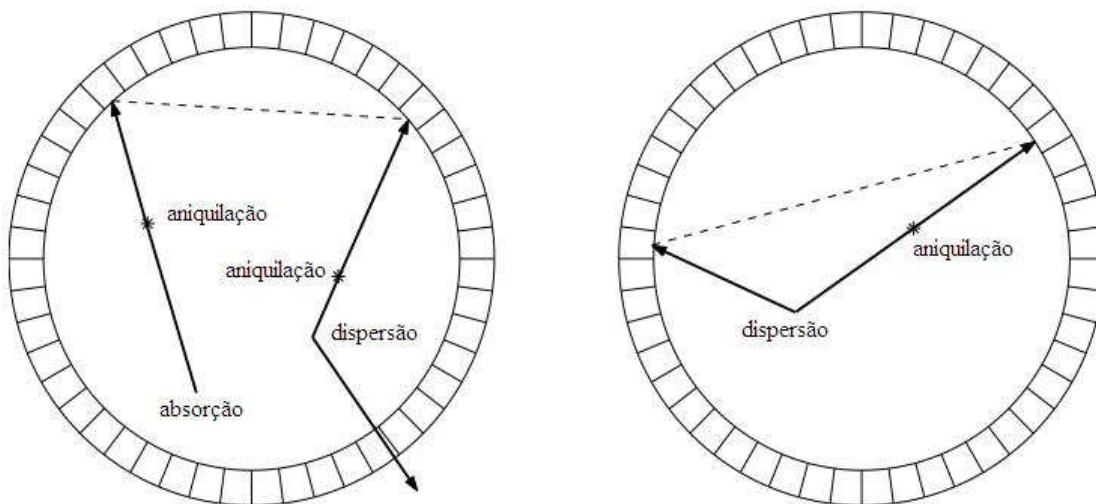
**Figura 1-Formação do sinograma.**

A) As letras A,B,C e D representam 4 LORs que atravessam uma área de interesse (definida por uma elipse). O centro do campo de visão é marcado por x.

B) Sinograma que contém as 4 LORs, eixo dos yy marca a orientação angular e eixo dos xx marca o posicionamento em relação ao centro do campo de visão.

Num estudo PET, em virtude da dose injectada ser baixa (recordando que o princípio básico da protecção radiológica ocupacional - Princípio ALARA estabelece que todas as exposições devem ser mantidas tão baixas quanto possível) o número de contagens está limitado, acarretando um índice elevado de ruído nos dados.

O ruído, em parte, deve-se a coincidências, distintas da já citada coincidência verdadeira, sendo designadas por coincidências aleatórias e de dispersão (Figura 2). Em relação às coincidências aleatórias, se considerarmos que um dos fótons de um determinado par sofre dispersão ou absorção, ele pode nunca vir a ser detectado. Ora se esta situação ocorrer para 2 pares de fótons que ocorram na mesma janela temporal, eles podem ser registados de forma errónea como sendo provenientes da mesma aniquilação[1]. Relativamente às coincidências dispersas, fótons que sofrem dispersão são detectados com menor probabilidade, contudo alguns podem ser detectados em coincidência com os seus pares, motivando erros.



**Figura 2-Tipos de coincidências responsáveis, em parte, pelo ruído. Coincidências aleatórias (esquerda) e coincidências dispersas (direita).**

A problemática da reconstrução em termos matemáticos, ou seja, reconstruir uma função a partir das suas projecções remonta a Johann Radon [2].

Nos últimos anos surgiram técnicas de reconstrução que revelavam o problema da reconstrução em PET como um problema de estimação para os dados incompletos. Da física do PET sabemos que um par de detectores registou um evento de aniquilação, contudo a origem da mesma permanece desconhecida. Algumas das técnicas recentes [3,5,6] tiveram em conta a natureza de Poisson na emissão de positrões.

Neste âmbito, surgiu o algoritmo *Expectation-Maximization* (EM) que foi desenvolvido inicialmente por Dempster [3]. Este algoritmo compreende duas fases: a primeira, onde se calcula o valor de uma função de verosimilhança usando os dados observados e as estimativas actuais dos parâmetros e na segunda fase, maximiza-se a função de verosimilhança obtendo-se assim uma nova estimativa.

A abordagem da máxima verosimilhança na tomografia por emissão foi introduzida por Rockmore e Macovski [4]. Contudo a implementação do EM no âmbito da reconstrução da imagem PET é atribuída a Shepp e Vardi [5], cuja extensão à transmissão CT foi da responsabilidade de Lange e Carson [6].

Neste projecto centrámo-nos sobre o Algoritmo das Sinocurvas [7] que consiste num método de reconstrução de imagem desenvolvida para sinogramas. O sinograma é interpretado como um conjunto de curvas sinusoidais que correspondem aos pixels na imagem. Desta forma o problema bidimensional da reconstrução cinge-se à decomposição do sinograma na sua forma mais elementar – as curvas sinusoidais.

## 2.2 Descrição do algoritmo EM

O algoritmo EM consiste numa técnica iterativa de reconstrução de imagem aplicada à tomografia por emissão que através do método da verosimilhança estima a densidade de actividade no objecto de estudo. Este método surge integrado no modelo matemático que se descreve neste capítulo.

Supondo que os dados observados em PET podem ser representados pelo vector  $y$ , tal que  $[y(1), y(2), y(3), \dots, y(J)]$

em que  $y(j)$  corresponde ao número total de coincidências registadas pelo detector  $j$  e  $J$  é o numero total de detectores.

Representando por  $f(x, y)$  a concentração de actividade da fonte esta é estimada a partir de  $y$ . Assim, se discretizarmos  $f(x, y)$  em regiões  $i=1, 2, \dots, I$  para efeitos de computação e visualização e assumindo que cada região  $i$  possui uma contagem desconhecida  $x(i)$ , que representa o número total de emissões na região delimitada pela região  $i$ , com média  $\lambda(i) = E[x(i)]$  podemos estimar  $\lambda(i)$  para os dados observados  $y(j)$ .

As emissões ocorridas na fonte seguem uma distribuição de Poisson. Assim, para cada região  $i$  existe uma variável de distribuição de Poisson  $x(i)$ , cuja média é  $\lambda(i)$ , gerada de forma independente [5]:

$$P(x(i) = m) = \frac{\lambda(i)^m e^{-\lambda(i)}}{m!}, \quad m = 0, 1, \dots \quad \text{Eq.2-1}$$

Considerando  $C(i, j)$  como a emissão ocorrida na parcela  $i$  registada pelo detector  $j$ , cuja probabilidade é descrita por:

$$C(i, j) = P(\text{evento registado pelo detector } j | \text{evento emitido pela parcela } i) \quad \text{Eq.2-2}$$



tal que  $C(i, j) \geq 0$ .

Assim a probabilidade em detectar a emissão proveniente de  $i$  é:

$$C(i) = \sum_{j=1}^J C(i, j) \leq 1 \quad \text{Eq.2-3}$$

Por outro lado, assume-se, as variáveis  $y(j)$  são independentes e que seguem uma distribuição de Poisson, com esperança  $Y(j)$  em que:

$$Y(j) = E[y(j)] = \sum_{i=1}^I \lambda(i) C(i, j) \quad \text{Eq.2-4}$$

Visto que, as variáveis  $\lambda(i)$  são independentes seguindo uma distribuição de Poisson, então uma combinação linear destas variáveis vai também ser uma variável de Poisson. Por conseguinte, a função de verosimilhança para os dados observados pode ser dada por:

$$L(\lambda) = P(y | \lambda) = \prod_{j=1}^J \frac{Y(j)^{y(j)} e^{-Y(j)}}{y(j)!} \quad \text{Eq.2-5}$$

Fazendo  $l(\lambda) = \log L(\lambda)$  e tendo em conta a Eq.2-4 obtemos [8]:

$$l(\lambda) = \log L(\lambda) = \sum_{j=1}^J -\lambda(i) C(i, j) + y(j) \log \lambda(i) + y(j) \log C(i, j) - \log y(j)! \quad \text{Eq.2-6}$$

Foi demonstrado [9] que a matriz da segunda derivada é negativa semi definida e que  $l(x)$  é côncava (assim todos os seus máximos, são máximos globais). Por outro lado, segundo provado em [10, Teorema 2.19 (e)] as condições suficientes para  $\hat{\lambda}$  maximizar  $l$  são as condições de Kuhn Tucker:

$$0 = \left[ \lambda(i) \frac{\partial l(\lambda)}{\partial \lambda(i)} \right]_{\hat{\lambda}} = -\hat{\lambda}(i) + \sum_{j=1}^J \frac{y(j)\hat{\lambda}(i)c(i,j)}{\sum_{i=1}^I \lambda(i)c(i,j)} \quad \text{Eq.2-7}$$

e

$$\left[ \frac{\partial l(\lambda)}{\partial \lambda(i)} \right]_{\hat{\lambda}} \leq 0, \text{ caso } \hat{\lambda}(i) = 0 \quad i=1,2,\dots,I. \quad \text{Eq.2-8}$$

Um esquema iterativo possível que permitiria a convergência para um máximo de  $l$  seria [5]:

Aplicando o algoritmo EM:

(1) Assume-se como estimativa inicial  $\lambda^{(0)}$ , em que  $\lambda^{(0)}(i) > 0, i = 1, 2, \dots, I$ .

(2) Define-se uma nova estimativa  $\lambda^{(k+1)}$ , sabendo que  $\lambda^{(k)}$  representa a estimativa de  $\lambda^{(k)}$  para a iteração k:

$$\lambda^{(k+1)}(i) = \lambda^{(k)}(i) \sum_{j=1}^J \frac{y(j)c(i,j)}{\sum_{i=1}^I \lambda^{(k)}(i)c(i,j)} \quad \text{Eq.2-9}$$

(3) No caso de atingir a convergência numérica necessária, o algoritmo cessa o seu funcionamento.

O algoritmo EM converge para um máximo global de  $l(\lambda)$  enquanto se verifica o Teorema 1 de Dempster [3] em que:

$$l(\lambda^{(k)}) < l(\lambda^{(k+1)}) \quad \text{Eq.2-10}$$

Quando se verifique a seguinte igualdade:

$$\lambda^{(k)} = \lambda^{(k+1)} \quad \text{Eq.2-11}$$

a Eq.2-9 converge, cessando o desempenho do algoritmo.

Conjugando as equações anteriores Eq.2-4 e Eq.2-9 obtemos o algoritmo EM capaz de ser aplicado a problemas de reconstrução em PET:

$$\lambda^{(k+1)}(i) = \lambda^{(k)}(i) \sum_{j=1}^J C(i,j) \frac{y^{(j)}}{Y^{(k)}(j)}, i = 1, 2, \dots, I \quad \text{Eq.2-12}$$

## 2.3 Descrição do algoritmo das Sinocurvas

Para uma melhor compreensão do algoritmo das sinocurvas, começamos por fazer uma referência breve ao glossário criado pelos autores [7] adaptando-o:

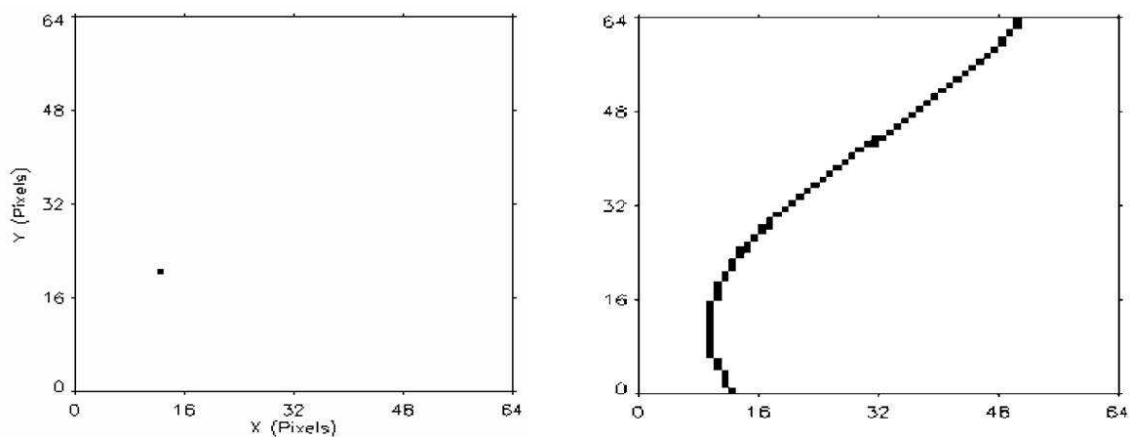
“**sinopixel**” - *pixel* do sinograma.

“**imgpixel**” - *pixel* da imagem reconstruída.

“**sinocurva**”-curva sinusoidal que percorre uma série de sinopixeis, correspondentes a um dado imgpixel.

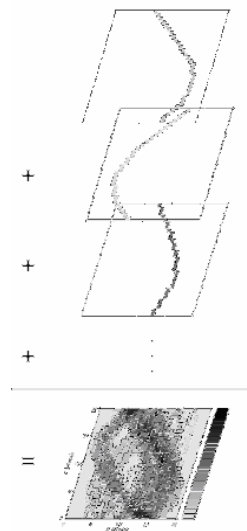
“**contribuições**” = num sinopixel podem passar várias sinocurvas. Por isso por contribuição entende-se o peso que as várias sinocurvas têm no valor do sinopixel.

O algoritmo das sinocurvas assenta no princípio de que cada curva sinusoidal (conjunto de sinopixeis) - “sinocurva” corresponde a um “imgpixel” (Figura 3). Assim, o problema é equivalente a decompor o sinograma nas suas sinocurvas determinando o valor correspondente a cada sinocurva. E, uma vez, que o valor das sinocurvas é constante ao longo do seu perfil e igual ao valor do imgpixel correspondente determina-se desta forma a imagem que se pretende.



**Figura 3-**O gráfico à esquerda representa um pixel numa imagem de reconstrução, denominado *imgpixel*, que corresponde a um conjunto de *pixeis*, designados por *sinopixeis*, no sinograma (gráfico à direita). O conjunto de *sinopixeis* define a curva sinusoidal - *sinocurva*. Retirada de [7].

Esta abordagem pode ser melhor compreendida com base na **Figura 4**, na qual se representa a decomposição do sinograma na sua forma elemental- “ sinocurva”.



**Figura 4-Decomposição do sinograma (imagem inferior) em sinocurvas (imagens superiores).**

O valor de actividade (igual ao valor do imgpixel correspondente à sinocurva) é constante para uma dada sinocurva. Retirada de [7].

A reconstrução é assim facilmente entendida como a resolução de um sistema linear de equações:

$$\begin{cases} \phi_{11}s_1 + \phi_{12}s_2 + \dots + \phi_{1n}s_n & = & p_1 \\ & \vdots & \\ \phi_{m1}s_1 + \phi_{m2}s_2 + \dots + \phi_{mn}s_n & = & p_m \end{cases} \quad \text{Eq.3-1}$$

em que assumimos  $n$  sinocurvas  $s_i$  e  $m$  sinopixeis  $p_i$ . Por facilidade de implementação mas sem perda de generalidade assumimos que a função  $\phi_{ji}$  adquire valor 1 se a sinocurva  $i$  contém o sinopixel  $j$ , caso contrário assume valor 0.

Em termos matriciais:

$$\phi S = P \quad \text{Eq.3-2}$$

onde  $\phi$  consiste numa matriz com dimensões  $m \times n$  (designada por matriz sistema),  $S$  é um vector com dimensão  $n$  e  $P$  é um vector de dimensão  $m$ . A matriz sistema representa no problema a correspondência entre os imgpixeis e os sinopixeis, a qual é calculada previamente à reconstrução de imagem com base em assumpções geométricas.

A abordagem iterativa usada para resolver a equação 3.2 pode ser descrita da seguinte forma.

O valor de cada “sinopixel” ( $S_{ij}$ ) é distribuído por todas as “sinocurvas” que o intersectam tendo em conta o peso ( $W_i$ ) destas últimas. O peso desta distribuição (inicialmente assume-se o peso como sendo 1) ocorre tendo em conta os valores actuais dos pixeis de maneira iterativa.

Assim para cada “sinocurva”:

$$s(i, j) = \frac{w(i) p(j)}{\sum_{k=1}^n \phi(j, k) w(k)} \quad \text{em que } \phi_{ji} \text{ seja } 1. \quad \text{Eq.3-3}$$

No final de cada iteração, procede-se a um reajustamento baseado nos valores obtidos para  $S_{ij}$ .

Desta forma, o novo peso é dado por:

$$w(i) = \frac{\sum_{k=1}^m s(i, k)}{1} \quad \text{Eq.3-4}$$

Conforme verificado na Eq.3-4, este novo peso é uma média dos valores das sinocurvas, onde l consiste no número de valores únicos para a sinocurva i.

O algoritmo para dados com ruído poderia ser expresso por:

$$\lambda^{k+1}(i) = \frac{\lambda^k(i)}{l} \sum_{j=1}^l \frac{y(j)}{\sum_{i=1}^N c(i,j) \lambda^k(j)} \quad \text{Eq.3-5}$$

Em que  $\lambda$  é o valor do imgpixel, y é o valor do sinopixel, C indica a contribuição da sinocurva i para o sinopixel j, l representa o número de estimativas e N o número de contribuições. O novo valor do “imgpixel” é calculado a partir do “imgpixel” anterior e de todos os “sinopixeis” correspondentes. As semelhanças desta equação com a Eq.2-9 são evidentes. De tal forma que podemos dizer que o ML-EM numa perspectiva de “sinocurvas” poderia ser compreendido como uma média de todas as estimativas obtidas para os “sinopixeis” das respectivas “sinocurvas”.

Outra forma de encarar o problema é a partir das próprias projecções. Para o efeito consideremos uma imagem muito simples como a que se mostra na Figura 5.

4	3
2	1

**Figura 5-Imagem testada nos esquemas que mostram a equivalência dos algoritmos no caso particular de 2x2 pixels.**

Apresentam-se os esquemas para o algoritmo das Sinocurvas (Figura 6 e Figura 7) bem como os esquemas para o algoritmo MLEM (Figura 8 e Figura 9).

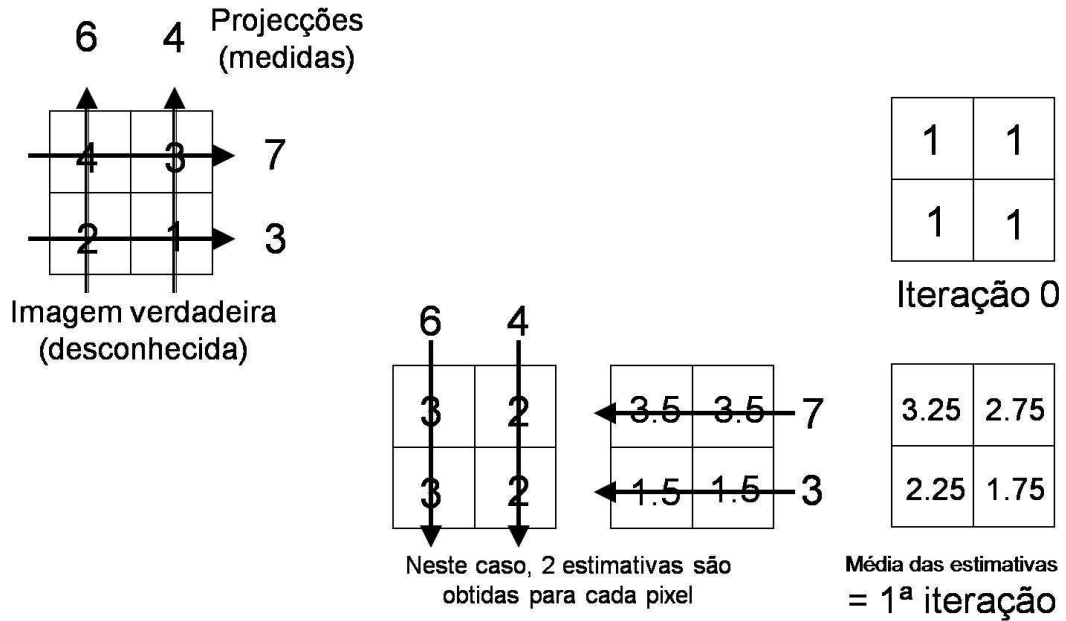


Figura 6-Esquema 2x2 pixels para algoritmo das sinocurvas: cálculo da 2ª estimativa.

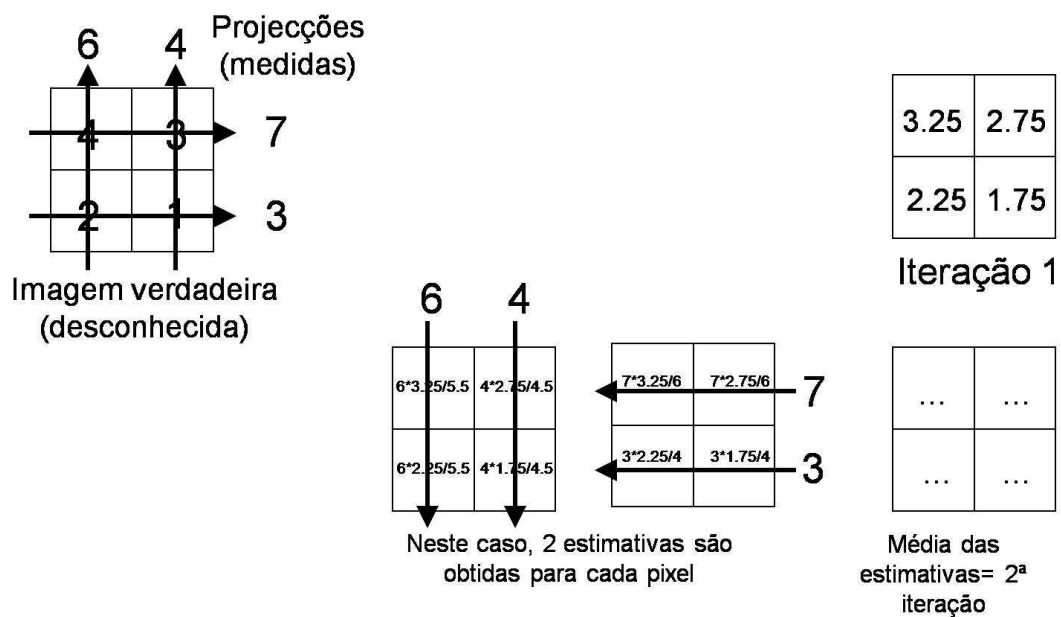


Figura 7-Esquema 2x2 pixels para algoritmo das sinocurvas: cálculo da 3ª estimativa.



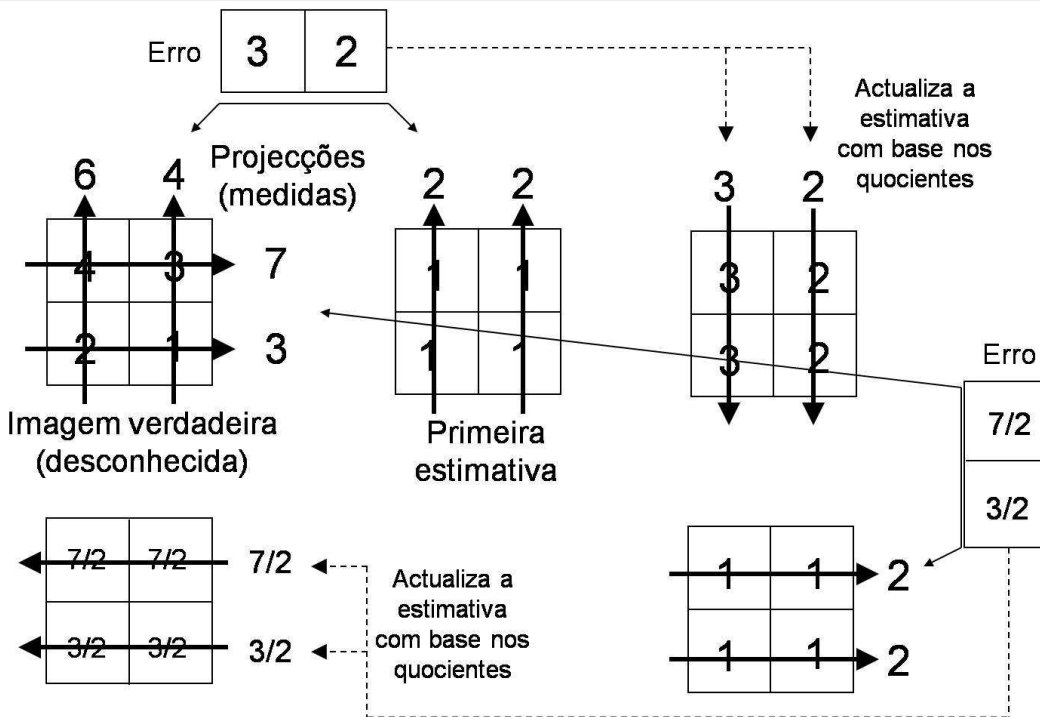


Figura 8-Esquema 2x2 pixels para algoritmo MLEM: Cálculo da 2ª estimativa

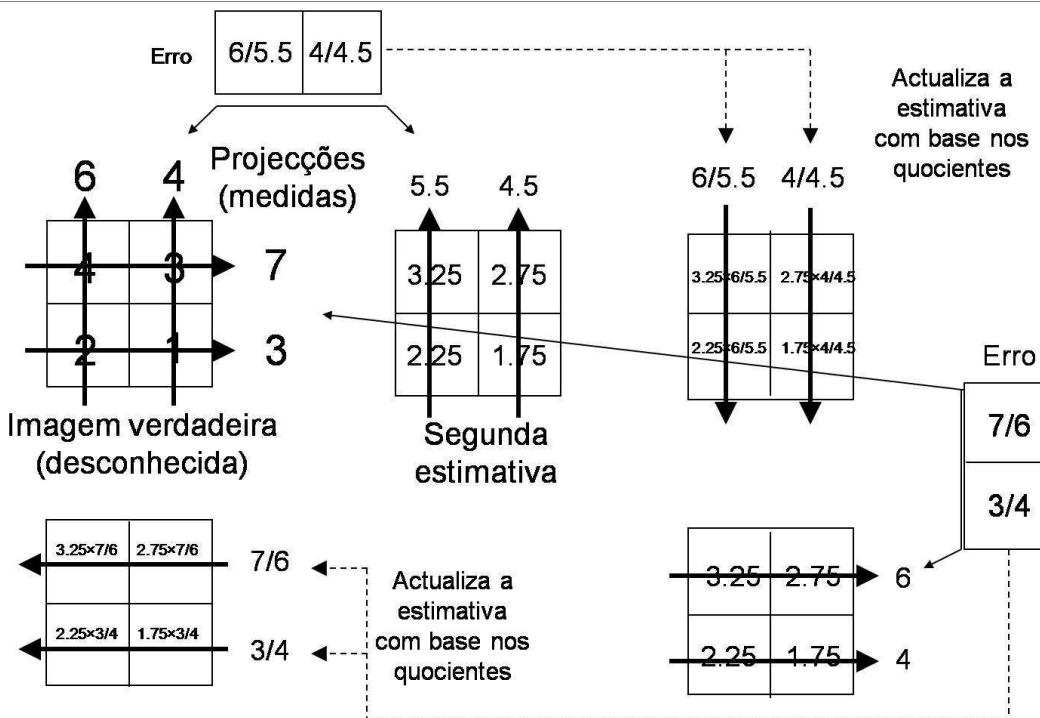


Figura 9-Esquema 2x2 pixels para algoritmo MLEM: Cálculo da 3ª estimativa

Com base nas projecções medidas e actualizando a estimativa a calcular, com base nos coeficientes, observa-se nos esquemas que ao fim de uma iteração a actualização da estimativa é igual, ou seja, a 1ª estimativa para o algoritmo das Sinocurvas corresponde à 1ª estimativa do algoritmo MLEM.

# Capítulo 3

## Métodos

O algoritmo de reconstrução de imagem, algoritmo das Sinocurvas, assim como os métodos para calcular a matriz sistema foram implementados em IDL para possibilitar a reconstrução de imagem e visualização de sinogramas assim como gráficos que mostravam o erro existente entre o valor real e o valor estimado para os pixels estudados.

A imagem de teste utilizada possuía dimensões 20 por 20 *pixels*, que se apresenta na Figura 10.

Figura 10 Durante o cálculo da matriz sistema definiu-se que o número de projecções no sinograma era 20.

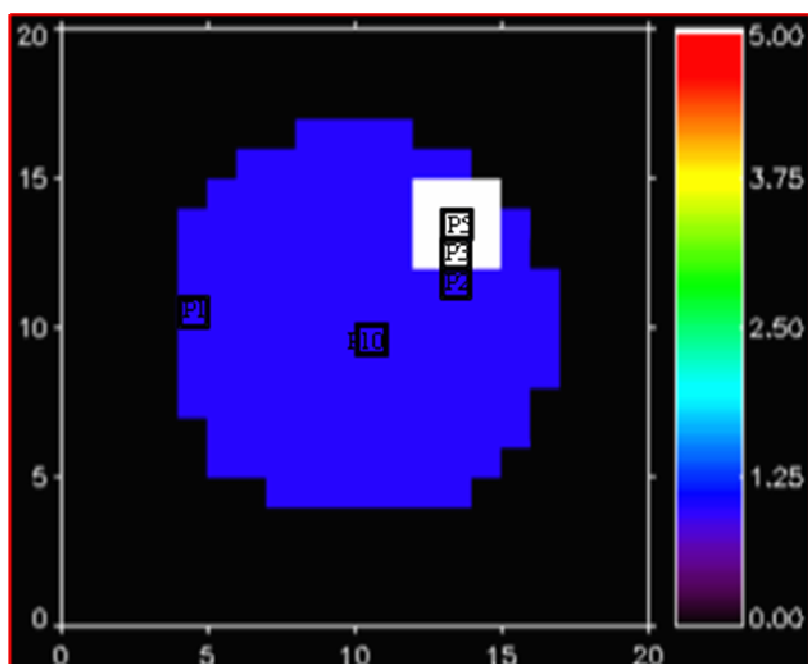


Figura 10- Imagem de teste, delimitada por uma área a azul e um quadrado pequeno no interior desta (a branco). O quadrado grande (a preto) é designado por “vizinhança de zeros”. A escala de cores revela o valor dos *pixels* para cada região.

Na Figura 10 surgem os *pixeis* que foram alvo de estudo. Na imagem de teste podemos considerar três regiões de diferente valor:

- uma região a preto onde os *pixeis* assumem valor 0.
- uma região definida por um quadrado (a branco) onde os *pixeis* assumem valor 5.
- uma região demarcada pela área azul onde os *pixeis* assumem valor 1.

Na Tabela 1 recorreu-se a uma nomenclatura concreta para tornar referências futuras mais simples.

Nomenclatura	Linha	Coluna	Índice
Pixel 1	10	4	90
Pixel 2	11	13	271
Pixel 3	12	13	272
Pixel 5	13	13	273
Pixel 10	9	10	209

**Tabela 1- Relação linha e coluna com a designação do pixel.**

O índice de cada pixel é dado por:  $\text{índice} = \text{linha} + (20 \cdot \text{coluna})$ . A importância do índice do *pixel* é fundamental para uma fácil identificação do *pixel* em determinados outputs gerados como veremos adiante (Capítulo 4, Secção 4.1). O critério utilizado na selecção dos *pixeis* foi posicional, era do nosso interesse perceber qual o papel que a distância, de um determinado *pixel* em relação a uma região com *pixeis* de valor diferente, assumia na convergência do *pixel* para o seu valor real.

O Pixel 1 é aquele que tem um posicionamento mais periférico sendo o mais próximo da região de *pixeis* de valor 0, o Pixel 10 aproximadamente central é o *pixel* deste estudo mais distante de regiões de valor diferente de *pixeis*, o Pixel 2 encontra-se próximo de uma região onde o valor dos *pixeis* sofre uma variação, o Pixel 3 demarca a fronteira entre duas zonas com diferentes valores de *pixeis*. Enquanto o Pixel 5 se estabelece como um *pixel* aproximadamente central na área quadrada. A posição de

cada *pixel* acaba por ter influência sobre o número de LORs que passa pelo *pixel*. Em princípio passam menos LORs num *pixel* periférico do que num *pixel* central. De realçar que as simulações realizadas tiveram por base o algoritmo das sinocurvas aplicado a dados sem ruído numa primeira instância e aplicado a dados com ruído numa segunda fase.

As simulações compreenderam numa fase inicial 10 iterações, para observação da tendência de convergência. Numa fase posterior o número de iterações foi alargado às 50 iterações, não sendo superior por questões de computação (tempo).

Foram testados vários níveis de ruído. Para isso foi adicionado ruído de Poisson aos sinogramas com 10k contagens, 100k e 1M contagens.

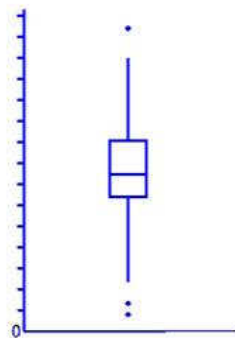
Todos os testes foram repetidos usando dois estimadores distintos: a média e a mediana.

Os resultados obtidos, para os *pixels*, por computação do algoritmo das Sinocurvas para a imagem de teste foram coligidos em bases de dados.

Os resultados foram então tratados com o software SPSS (Statistical Package for the Social Sciences), tendo sido gerados gráficos do tipo *boxplots* e *error bars* ( para um intervalo de confiança de 95%).

Os *boxplots* resumizam as seguintes medições estatísticas:

- média
- *quartil* superior e *quartil* inferior
- mínimo e máximo para os dados analisados



**Figura 11-Exemplo de *boxplot***

A interpretação de um *boxplot* revela-nos:

- A caixa propriamente dita, contém 50% dos dados tratados. A parte superior desta indica o percentil 75º do *dataset* enquanto que a parte inferior indica o percentil 25º. O intervalo entre os dois *quartis* anteriormente citados é designado por Amplitude Inter-Quartil
- A linha na caixa indica a mediana para os dados estudados. Caso esta linha não seja equidistante em relação aos *quartis*, os dados dizem-se enviesados.
- As pontas das linhas verticais indicam o valor mínimo e valor máximo para os dados. Os valores marcados acima destas linhas são os *outliers*.

Em relação aos *errorbars* com um intervalo de confiança de 95%, indicam a probabilidade do intervalo conter a média verdadeira.

Resumindo, os *boxplots* indicam a progressão da mediana em função do número de iteração enquanto que os *errorbars* indicam a progressão da média em função do número de iteração.

As simulações de convergência sem ruído foram realizadas para os 5 pixels definidos na imagem de teste para os diferentes estimadores – média e mediana.

De notar que os gráficos referentes às 50 iterações, no eixo das abcissas apenas são marcadas as iterações ímpares.

Para ser possível estudar a convergência em moldes semelhantes ao estudo realizado para as simulações sem ruído, foi necessário dividir os resultados gerados por um factor conforme o número de contagens no sinograma de forma a normalizar os resultados (Tabela 2).

<b>Número de contagens no sinograma</b>	<b>Factor</b>
10k	2,87356
100k	28,7356
1M	287,356

**Tabela 2-Factor utilizado para computação, conforme o número de contagens no sinograma.**

# Capítulo 4

## Resultados

Neste capítulo apresentam-se alguns dos *outputs* (Secção 4.1) gerados durante a fase de computação bem como os gráficos tratados em SPSS obtidos nas simulações realizadas neste projecto, nomeadamente, simulações de convergência sem ruído (Secção 4.2) para ambos estimadores, e simulações de convergência com ruído para estimador média (Secção 4.3.1) e simulações de convergência com ruído para estimador mediana (Secção 4.3.2). Para todas as simulações os gráficos gerados denotam uma linha horizontal a ponteadado que representa o valor real do *pixel*.

### 4.1 Exemplos de *outputs* gerados

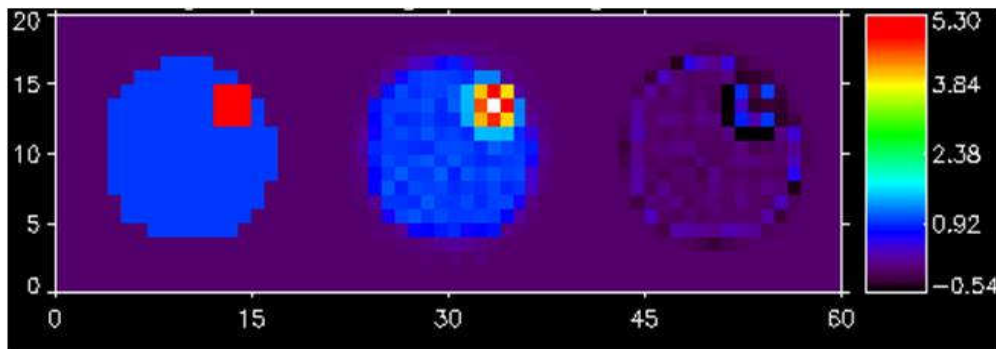


Figura 12 -Simulações sem ruído. 10 Iterações. Estimador média. Sem ruído. Eixos dos xx e yy representam o número de *pixels*.Escala de cores identifica o valor do *pixel*. Imagem à esquerda representa a imagem real, imagem central representa imagem reconstruída enquanto que a imagem à direita representa a diferença entre as imagens anteriores.

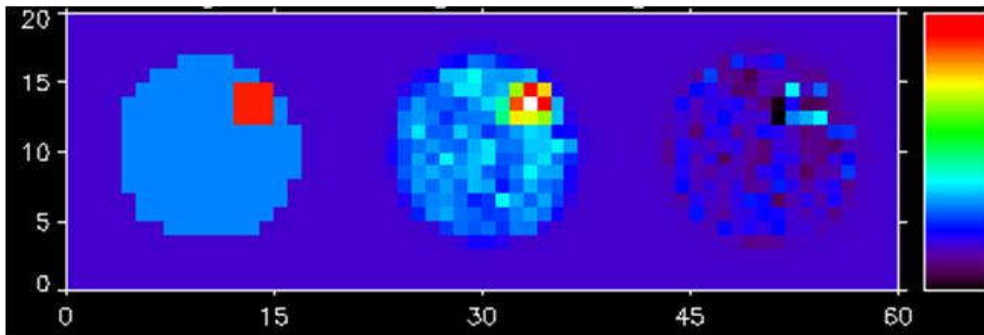


Figura 13-Simulações com ruído. 10 Iterações. Estimador média. 10k contagens. Eixos dos xx e yy representam o número de *pixels*. Escala de cores distingue o valor do *pixel*. Imagem à esquerda representa a imagem real, imagem central representa imagem reconstruída enquanto que a imagem à direita representa a diferença entre as imagens anteriores.

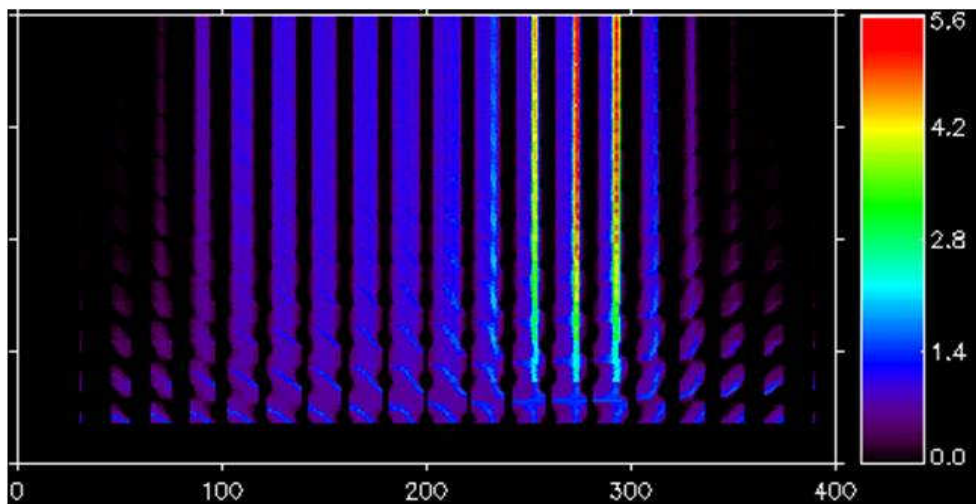


Figura 14-Simulações sem ruído. 10 Iterações. Estimador média. Eixo dos xx representa o índice do *pixel* e o eixo dos yy representa o número de , cuja primeira iteração 0 corresponde à faixa horizontal negra. Escala de cores identifica o valor do *pixel*.



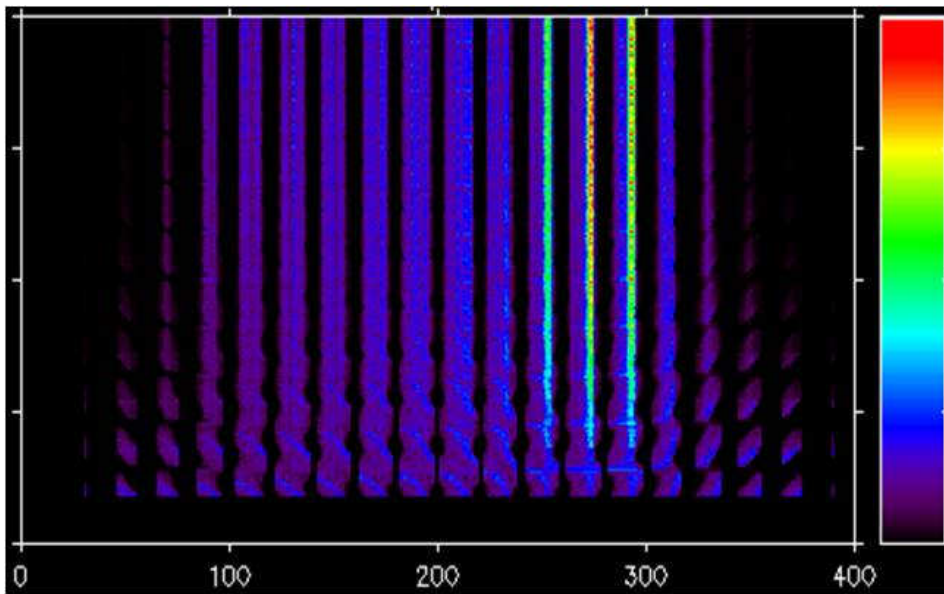


Figura 15-Simulações com ruído. 10 iterações. Estimador média. 10k contagens. Eixo dos xx representa o índice do *pixel* e o eixo dos yy representa o número de iterações, cuja primeira iteração 0 corresponde à faixa horizontal negra. Escala de cores distingue o valor do pixel.

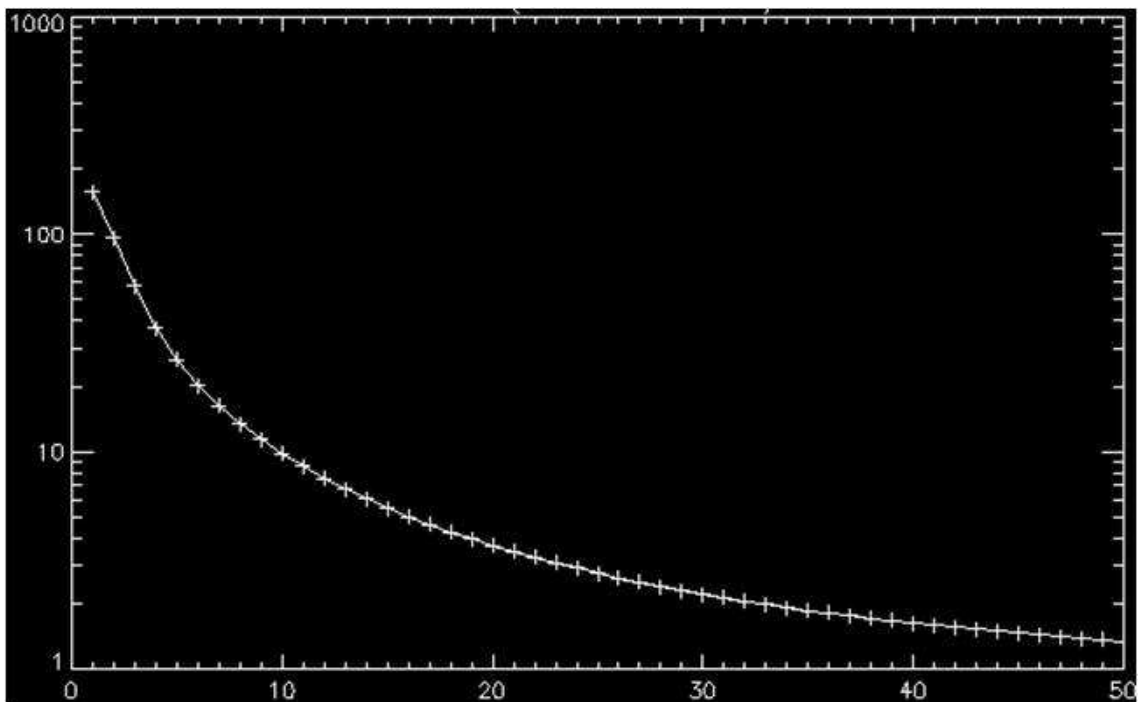


Figura 16 -Simulações sem ruído. 50 Iterações. Estimador média. Eixo dos xx representa o número de iterações, eixo dos yy representa o erro quadrado entre a imagem verdadeira e a imagem reconstruída.

Comparando a Figura 12 com Figura 13 observamos que a primeira apresenta mais *pixels* roxos na imagem mais à direita, que revela as diferenças entre a imagem real e a imagem reconstruída. Uma vez que os *pixels* roxos tem um valor aproximadamente zero, revela-nos que a simulação sem ruído apresenta um melhor resultado reconstutivo do que a simulação com ruído.

A Figura 14 e Figura 15 permitem ter uma perspectiva das variações ocorridas nas estimativas para todos os *pixels* durante todas as iterações.

A Figura 16 revela-nos que para as simulações sem ruído, ao longo de 50 iterações, o erro entre a imagem real e a imagem reconstruída decresce significativamente. Ao fim de 10 iterações esta percepção ainda não seria significativa.



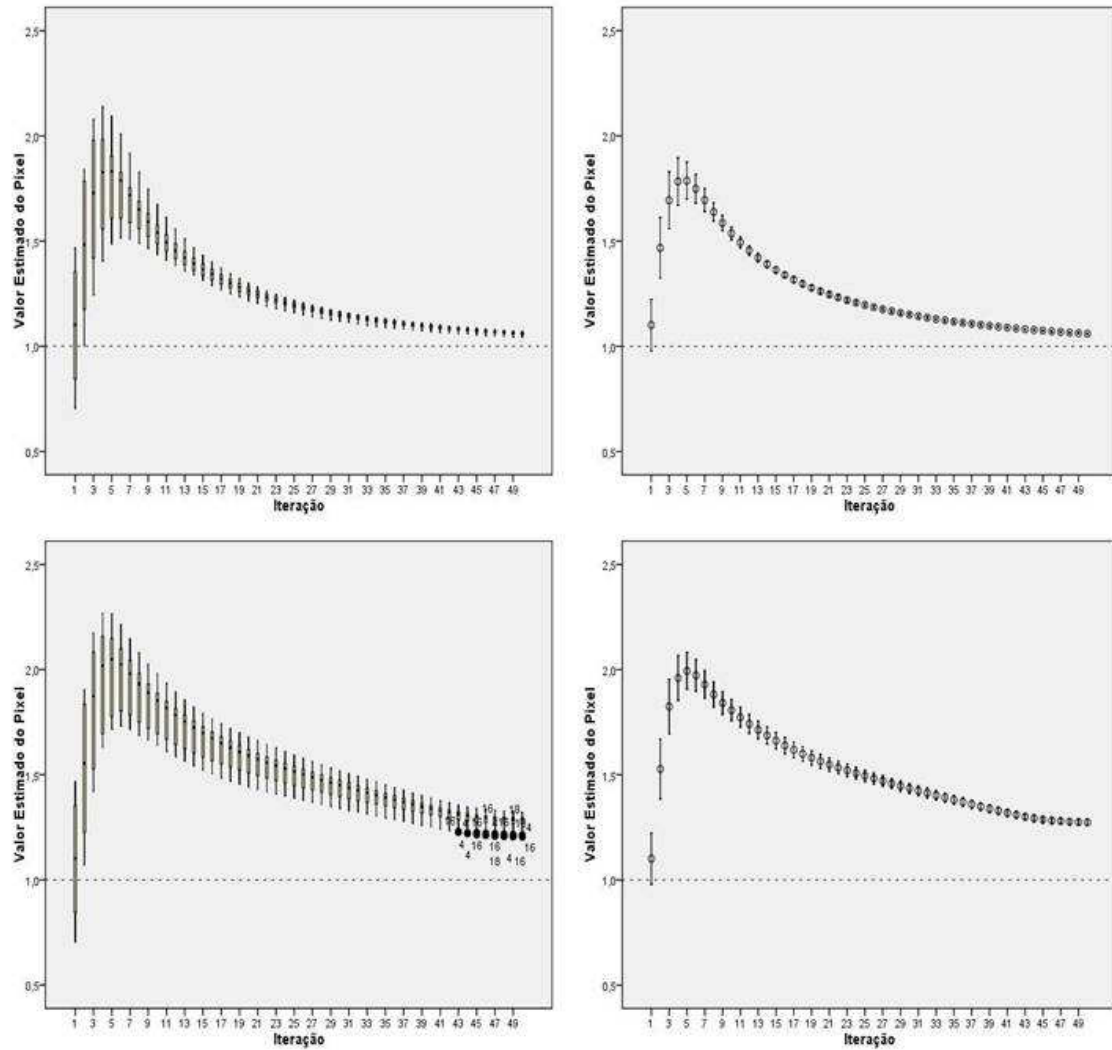


Figura 18- Evolução do valor estimado (eixo dos yy) para o *Pixel 2* em função do número de iterações (eixo dos xx). Estimador média (gráficos superiores) e Estimador mediana (gráficos inferiores).

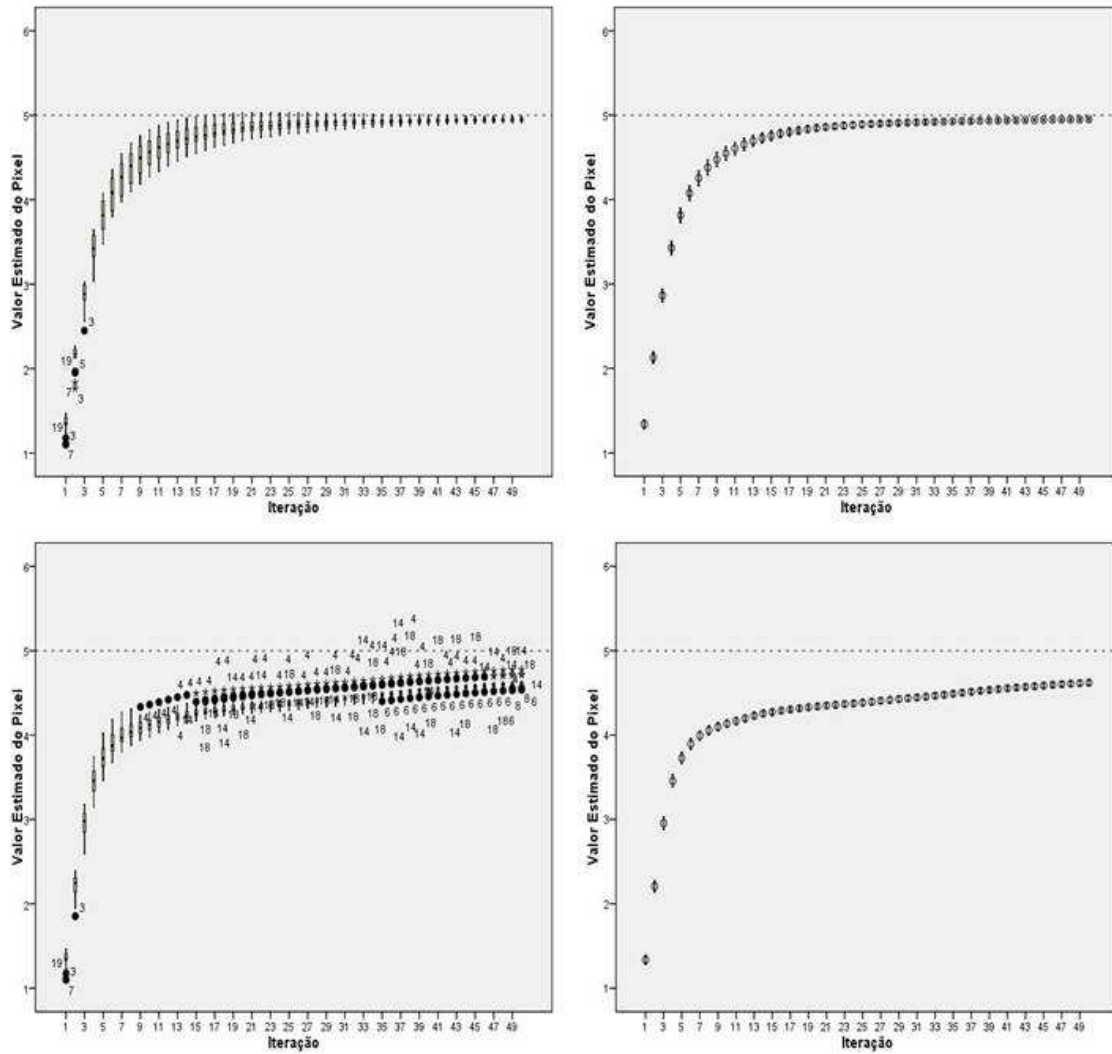


Figura 19-Evolução do valor estimado (eixo dos yy) para o *Pixel 3* em função do número de iterações (eixo dos xx). Estimador média (gráficos superiores) e Estimador mediana (gráficos inferiores).



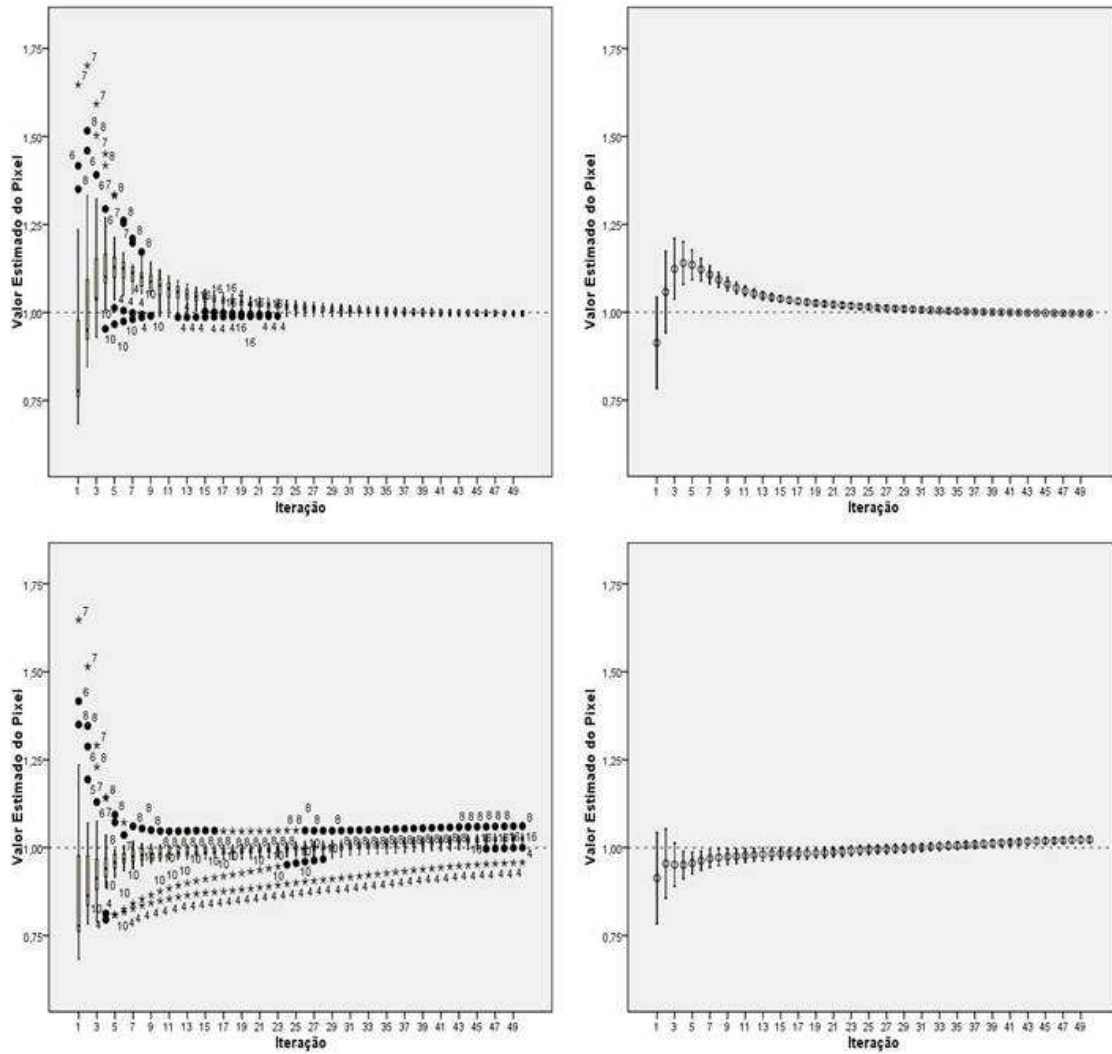


Figura 21- Evolução do valor estimado (eixo dos yy) para o *Pixel 10* em função do número de iterações (eixo dos xx). Estimador média (gráficos superiores) e Estimador mediana (gráficos inferiores).

Para o estimador média (gráficos superiores), observamos que o Pixel 1 e o Pixel 3 crescem sempre por valores inferiores ao valor real do *pixel*, enquanto os outros *pixeis*, nomeadamente Pixel 2, Pixel 5 e Pixel 10 revelam uma primeira fase de crescimento seguida de um decrescimento, convergindo para o valor real do *pixel*. De notar que o Pixel 1 e o Pixel 3 estão, aproximadamente, em zonas fronteira, isto é, zonas que dividem regiões de diferente valor, em termos de *pixeis*. Assim, por exemplo o Pixel 1 possui uma região próxima composta por *pixeis* de valor 0 enquanto a região onde se situa é composta por *pixeis* de valor 1. O Pixel 3 apresenta a mesma situação, mas neste caso particular divide região de valor 5 das regiões com valor 1. De entre os Pixeis 2,5 e 10 que como já foi dito experimentam um crescimento seguido de um decrescimento, destacamos o Pixel 2 que cresce muito durante as primeiras iterações, provavelmente porque está relativamente perto de uma zona com *pixeis* com valores elevados. O Pixel 5 comporta-se de um modo semelhante, embora cresça muito pouco quando comparado ao Pixel 2, mas importa notar que este crescimento nas primeiras iterações ultrapassa o valor real do *pixel*, convergindo de seguida para esse valor, provavelmente por influência dos *pixeis* de valor 1 que se encontram na região próxima.

Em relação ao estimador mediana (gráficos inferiores), os gráficos revelaram-nos que as curvas sem ruído não convergem para o valor real do *pixel*, ao contrário do ocorrido para a média. Assim o Pixel 1 apresentou um decrescimento logo nas primeiras iterações, seguindo-se um crescimento, contudo desviado do valor real do *pixel*. Para o Pixel 2 registamos um grande crescimento acompanhado de um decrescimento, contudo não converge para o valor real. O grande crescimento poderá estar novamente aliado ao facto do *pixel* se encontrar perto de uma zona com *pixeis* de valores elevados. No caso do Pixel 3 assistimos a um crescimento constante, sempre por valores inferiores aos do valor real do *pixel*. Tal como no caso do estimador média, verificamos que o Pixel 1 e Pixel 3 possuem estimativas sempre abaixo do valor real do *pixel* o que mais uma vez poderá estar relacionado com as regiões fronteira de regiões com valores diferentes para os *pixeis*. O Pixel 5 cresce um pouco acima do valor real, e embora não convergindo para o valor real, poderá convergir para um valor próximo ou, aparentemente, aumentar o valor da estimativa com o aumento do número das iterações. Sucedendo o mesmo para o Pixel 10.



### 4.3.1 Simulações de convergência com ruído Estimador Média

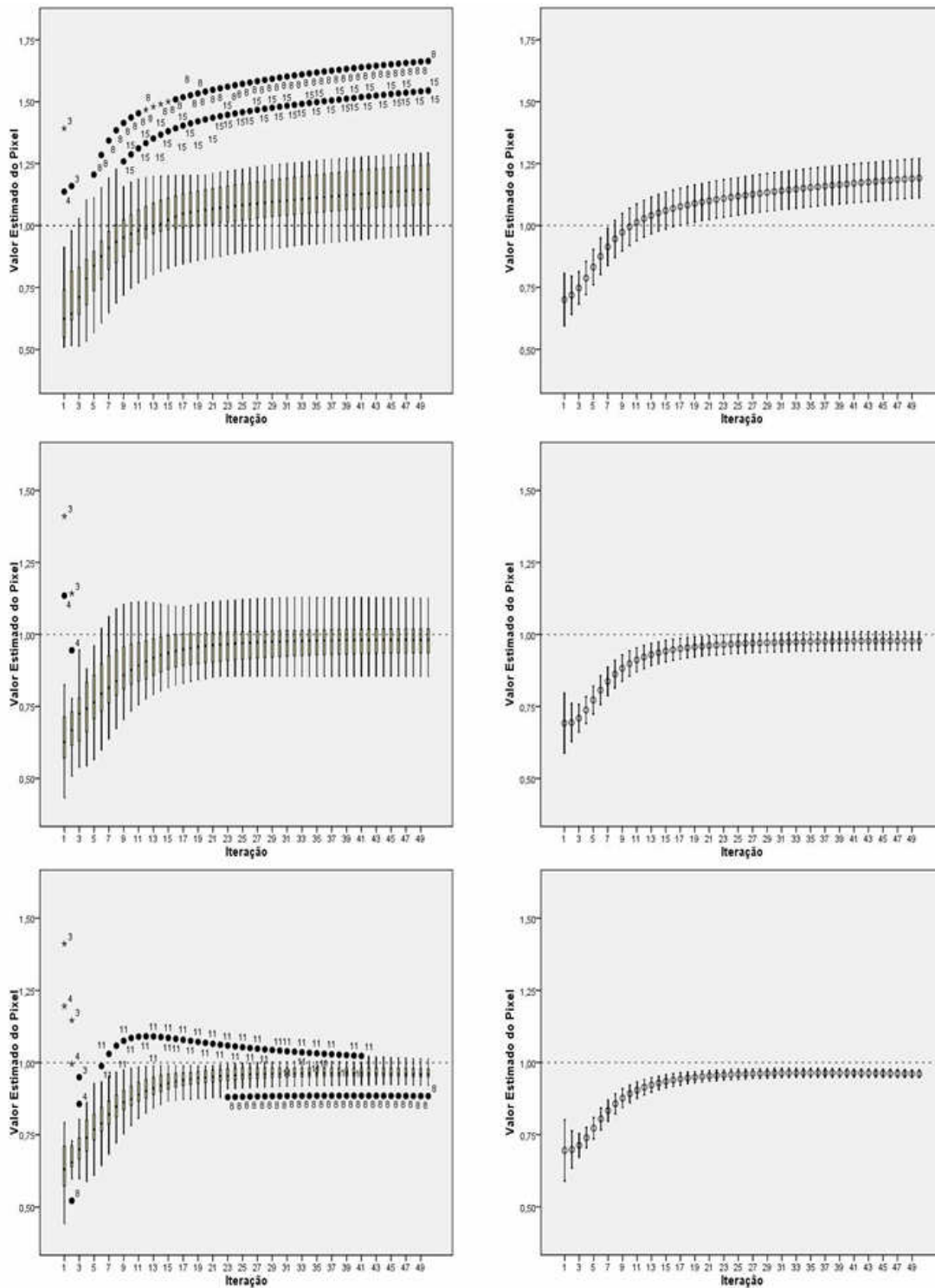


Figura 22-Evolução do valor estimado (eixo dos yy) para o *Pixel 1* em função do número de iterações (eixo dos xx). Do topo para baixo: 10k, 100k e 1M contagens.

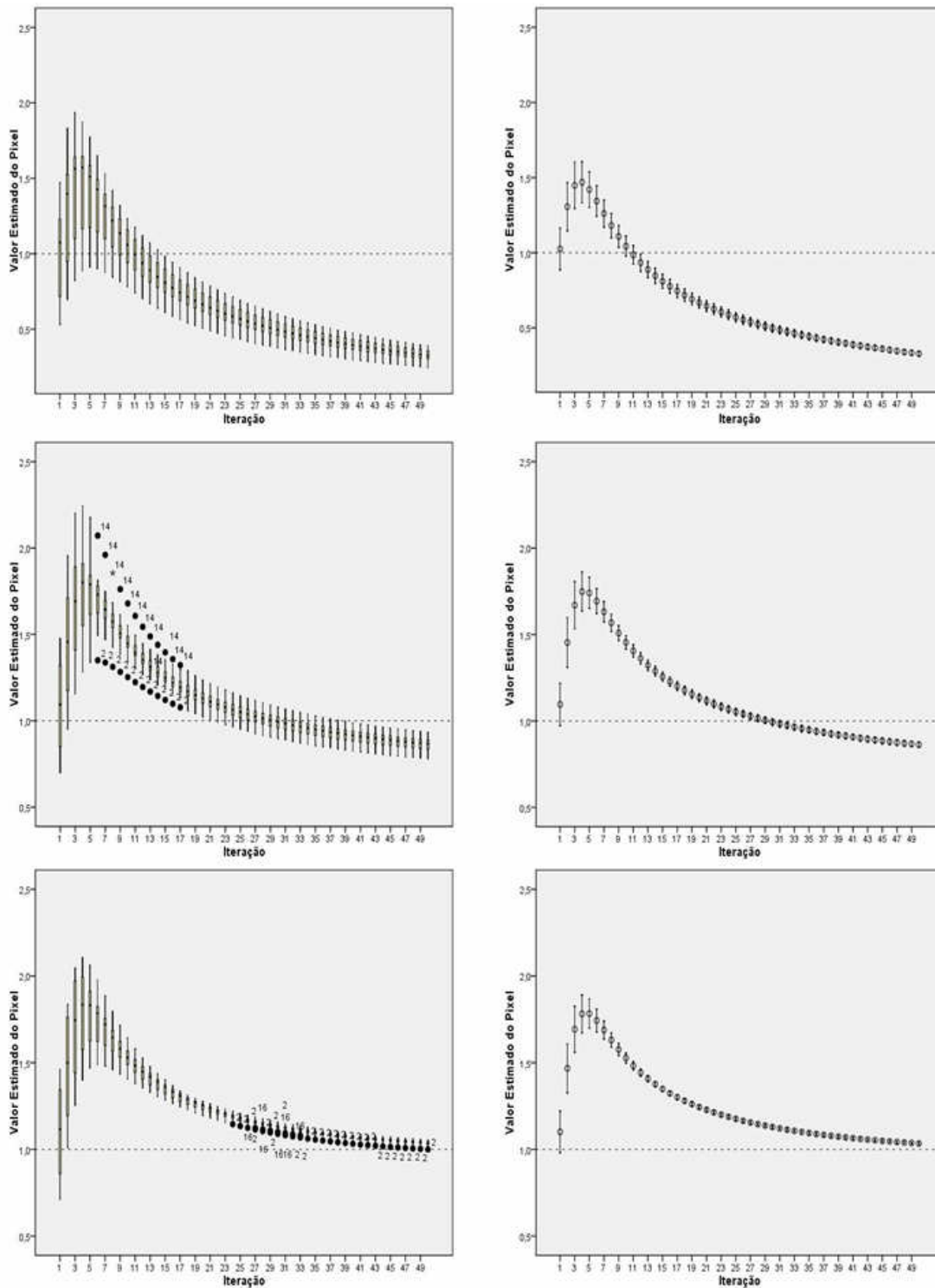


Figura 23-Evolução do valor estimado (eixo dos yy) para o *Pixel 2* em função do número de iterações (eixo dos xx). Do topo para baixo: 10k, 100k e 1M contagens.

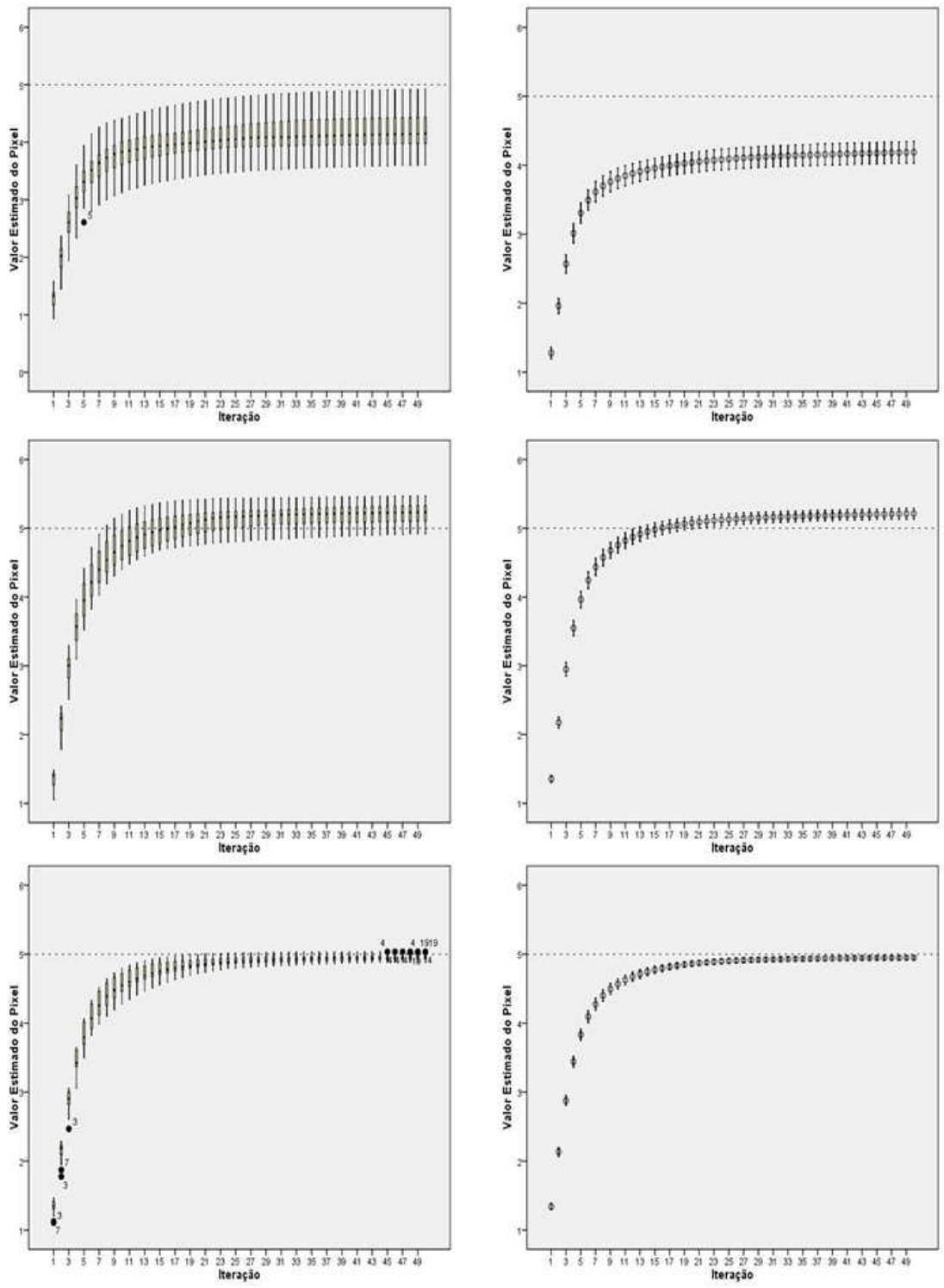


Figura 24-Evolução do valor estimado (eixo dos yy) para o *Pixel 3* em função do número de iterações (eixo dos xx). Do topo para baixo: 10k, 100k e 1M contagens.

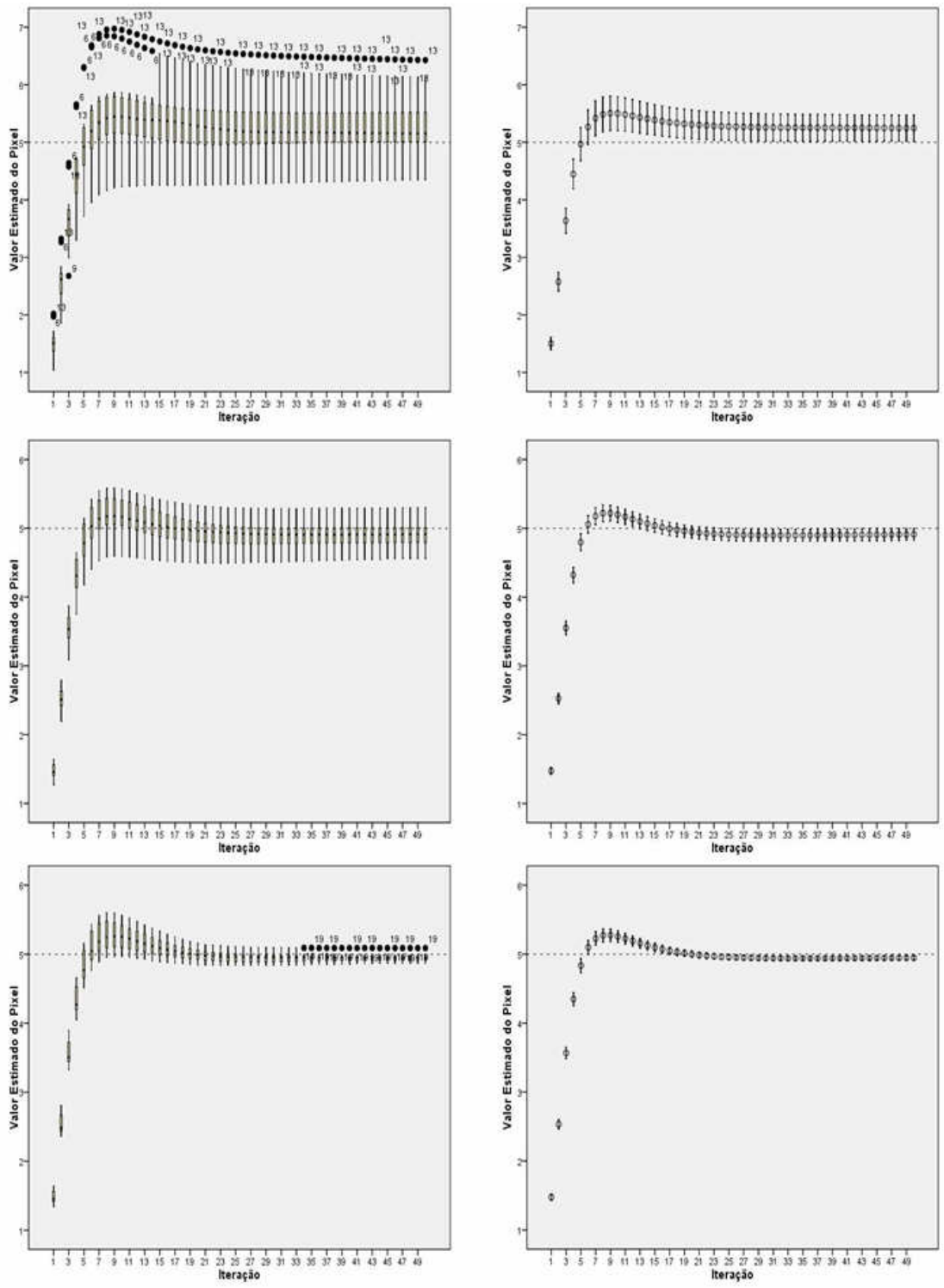


Figura 25-Evolução do valor estimado (eixo dos yy) para o *Pixel 5* em função do número de iterações (eixo dos xx). Do topo para baixo: 10k, 100k e 1M contagens.

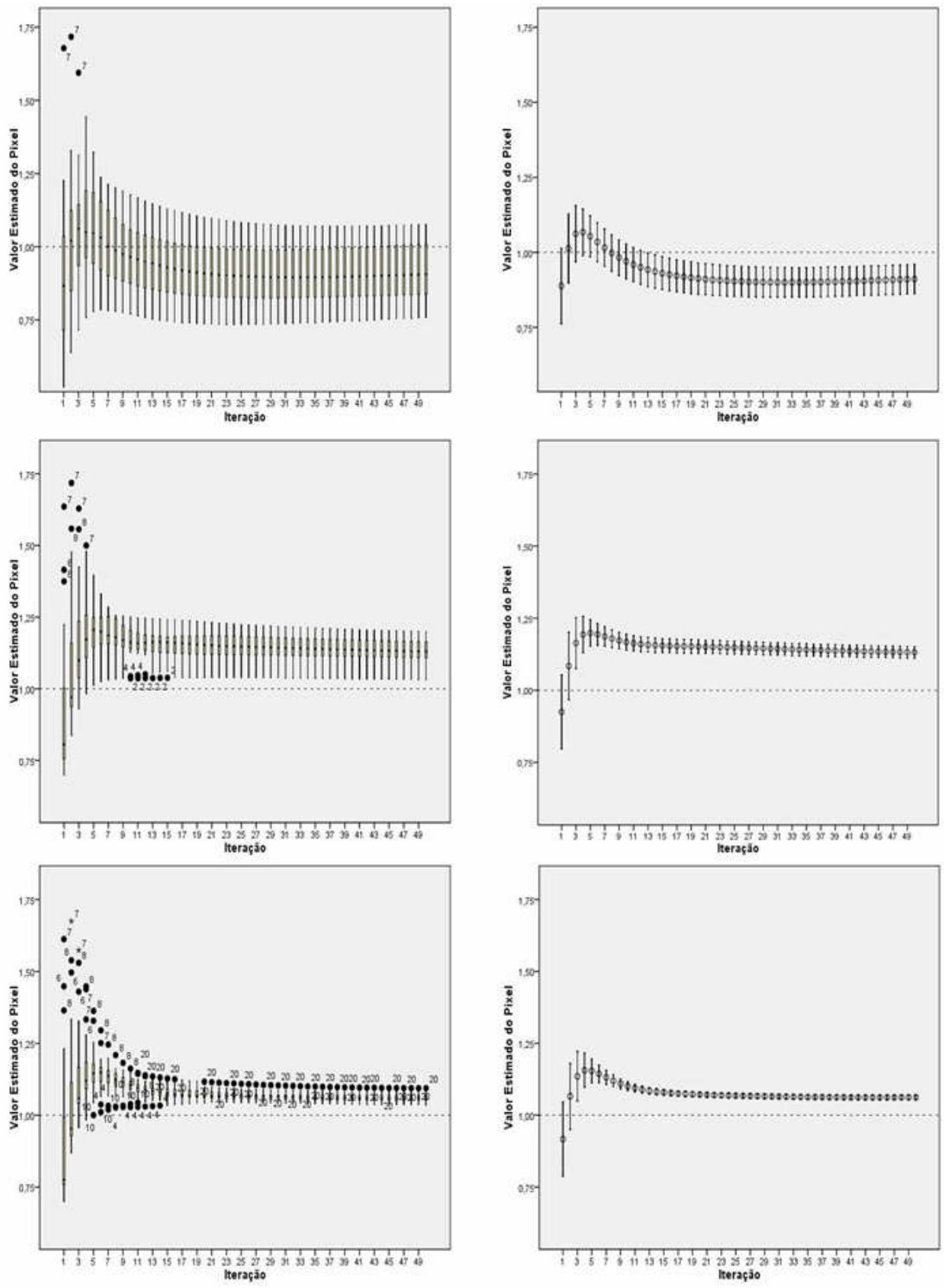


Figura 26-Evolução do valor estimado (eixo dos yy) para o *Pixel 10* em função do número de iterações (eixo dos xx). Do topo para baixo: 10k, 100k e 1M contagens.

No geral para simulações de convergência sem ruído e com ruído, para o estimador média, apontamos como principais diferenças:

- as simulações de convergência sem ruído apresentam uma convergência mais precisa do que as simulações de convergência com ruído.
- As barras de erro, nos *errorbars* para convergências com ruído, diminuem conforme o aumento do número de contagens. Ao fim de 50 iterações a barra de erro não é visível a olho nu para 1M de contagens.

A observação das curvas para o estimador média, para simulações de convergência com ruído, leva-nos a assumir que o comportamento das curvas é muito próximo da situação anterior verificada para as simulações sem ruído, tirando um caso especial.

Assim, o Pixel 1 e Pixel 3 revelam-nos um crescimento permanente.

Para o Pixel 2 tal como em situações sem ruído, verificamos que cresce bastante nas primeiras iterações, decrescendo de seguida.

O Pixel 5 cresce ligeiramente acima do valor real, decrescendo ligeiramente de seguida.

Para todos os *pixels* anteriormente citados com o aumento do número de contagens assistimos à convergência para o valor real do *pixel*.

No caso do Pixel 10 a situação é diferente, a curva verifica um crescimento e decrescimento tal como para situações sem ruído, no entanto com o aumento do número de contagens o valor estimado não converge para o valor real.

### 4.3.2 Estimador Mediana

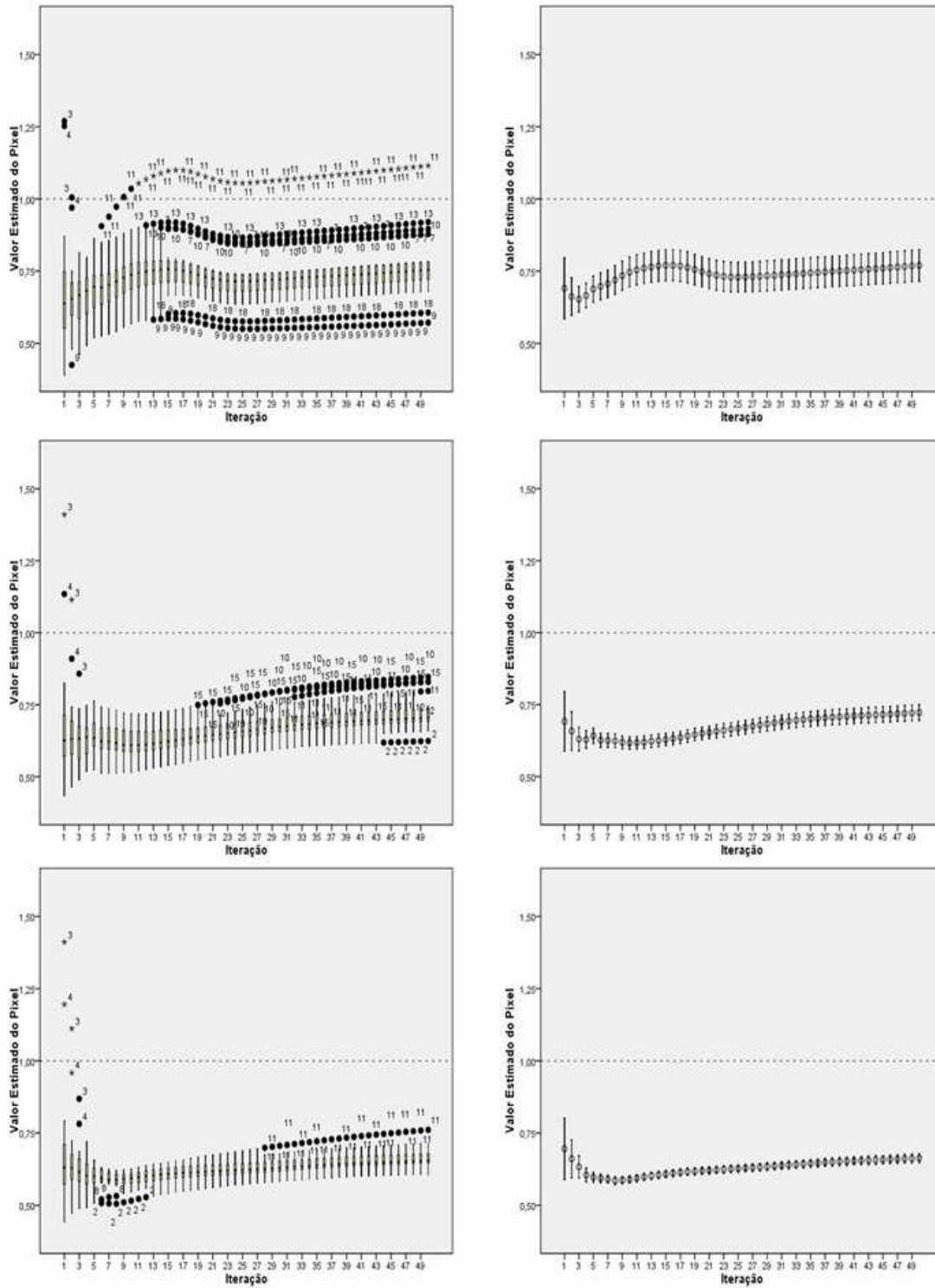


Figura 27-Evolução do valor estimado (eixo dos yy) para o *Pixel 1* em função do número de iterações (eixo dos xx). Do topo para baixo: 10k, 100k e 1M contagens.

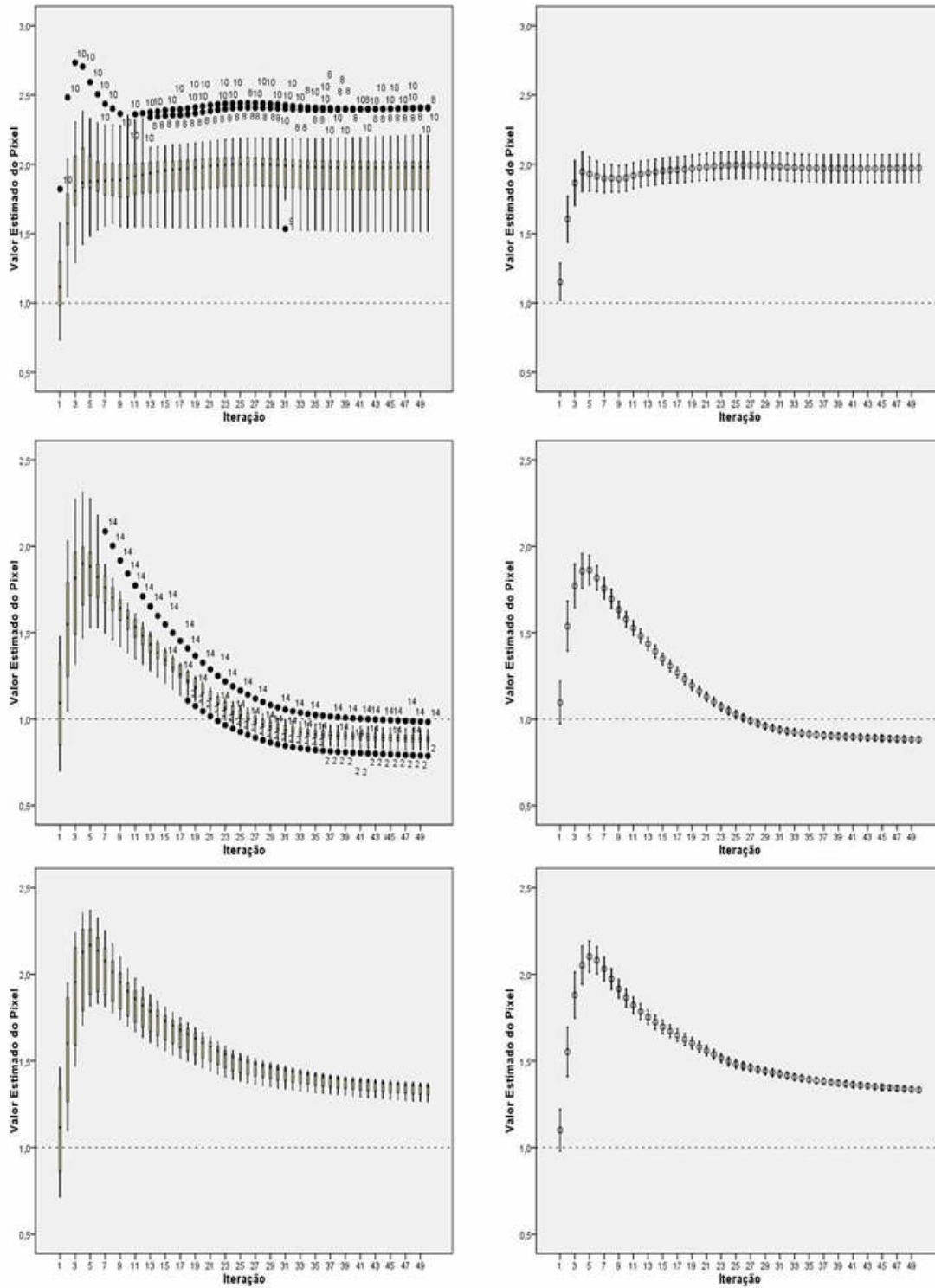


Figura 28-Evolução do valor estimado (eixo dos yy) para o *Pixel 2* em função do número de iterações (eixo dos xx). Do topo para baixo: 10k, 100k e 1M contagens.



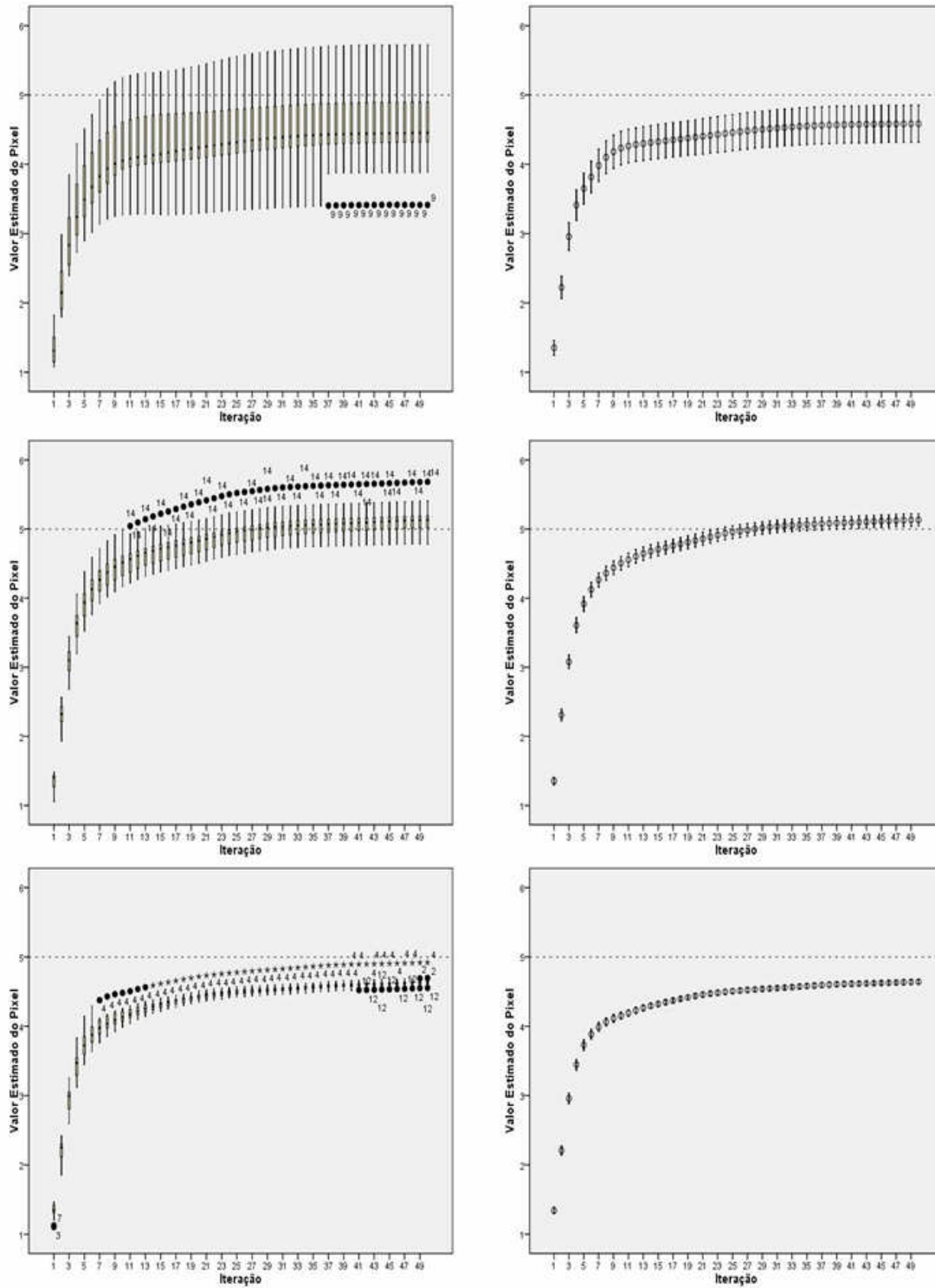


Figura 29-Evolução do valor estimado (eixo dos yy) para o *Pixel 3* em função do número de iterações (eixo dos xx). Do topo para baixo: 10k, 100k e 1M contagens.

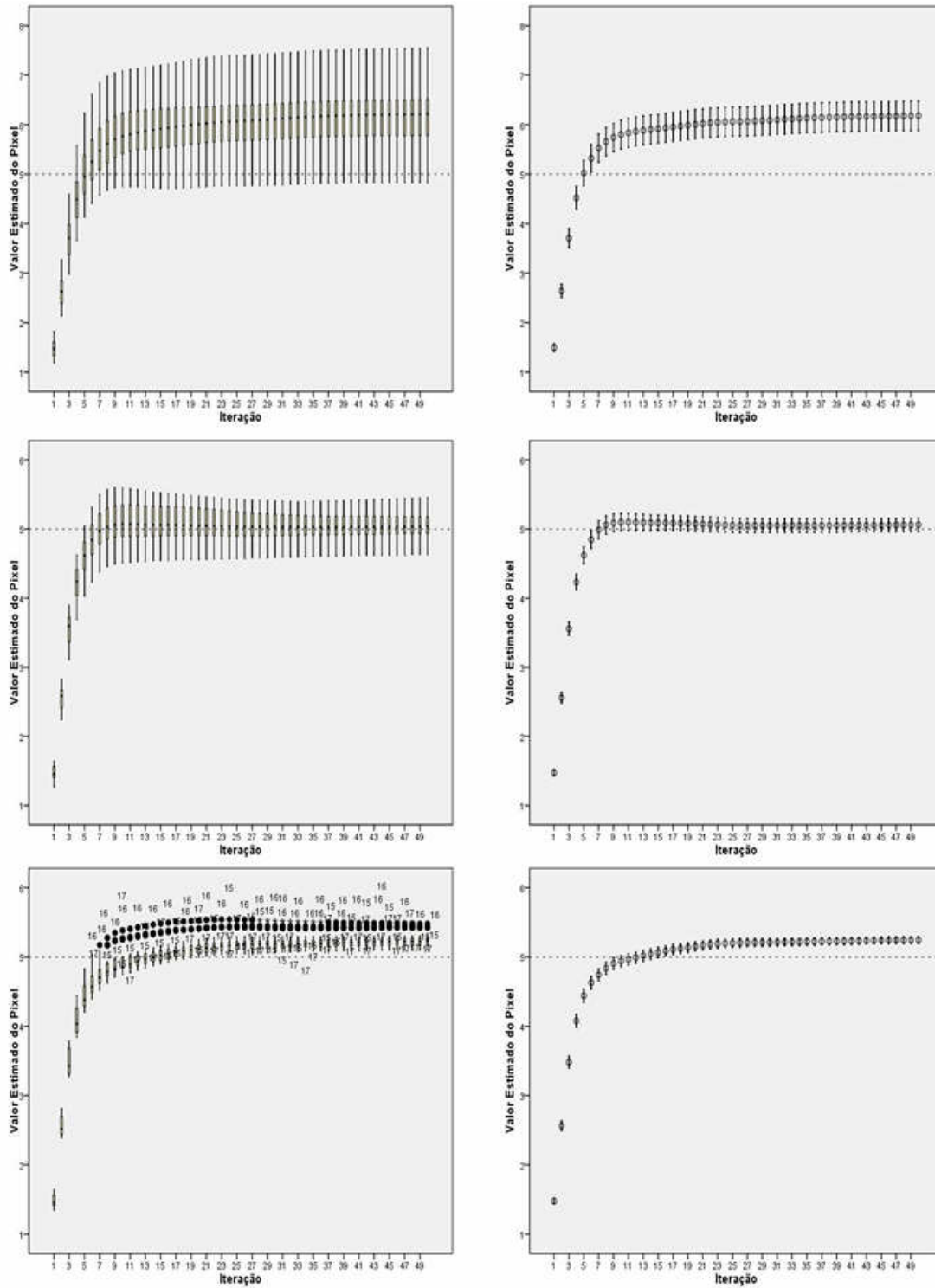


Figura 30-Evolução do valor estimado (eixo dos yy) para o *Pixel 5* em função do número de iterações (eixo dos xx). Do topo para baixo: 10k, 100k e 1M contagens.

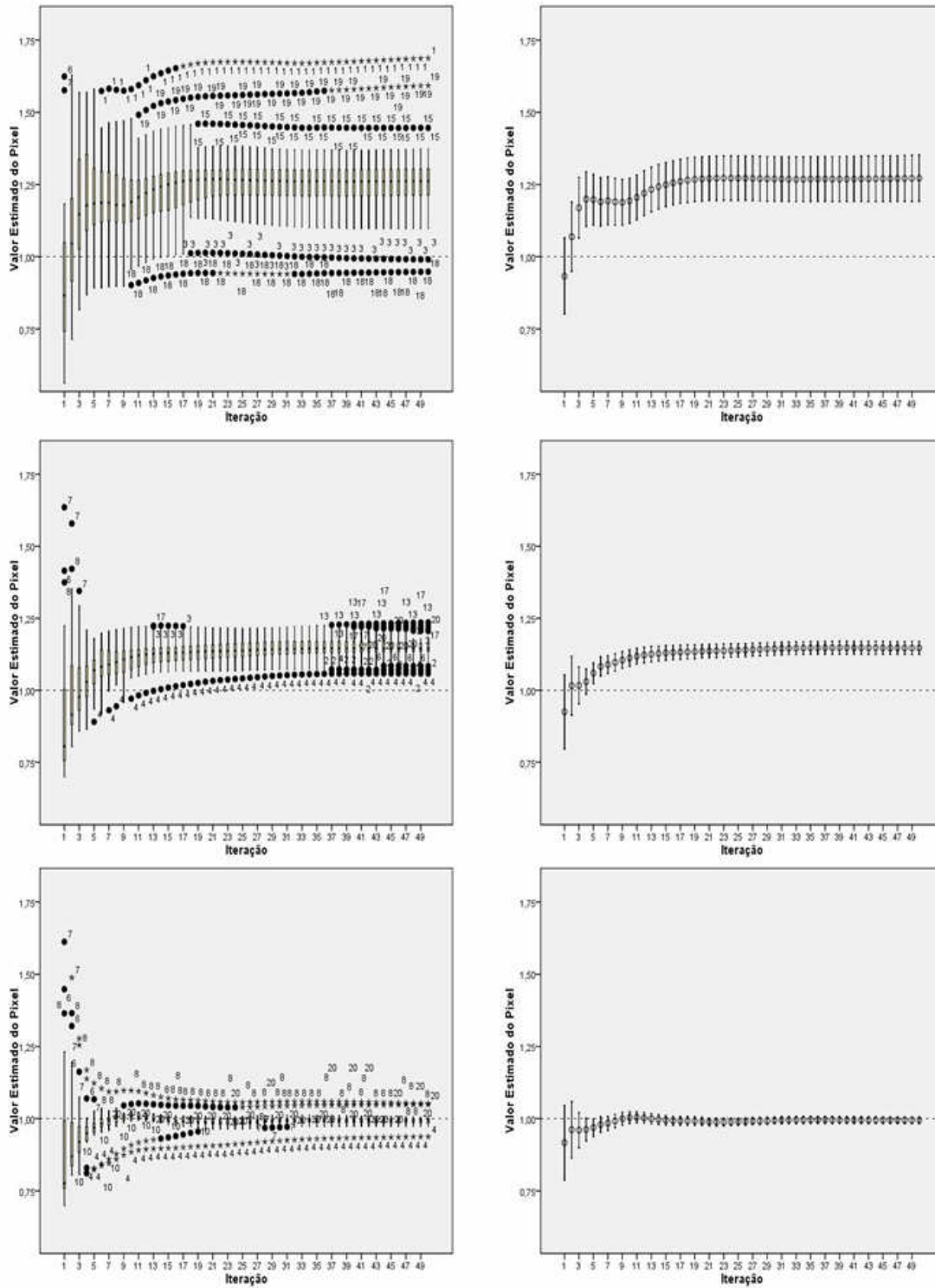


Figura 31- Evolução do valor estimado (eixo dos yy) para o *Pixel 10* em função do número de iterações (eixo dos xx). Do topo para baixo: 10k, 100k e 1M contagens.

A observação das curvas para o estimador mediana leva-nos a inferir:

O Pixel 1 denota um comportamento incerto para o aumento de ruído, 10 mil contagens, num alternar entre crescimento e decrescimento, nunca convergindo para o valor real mesmo para as situações de 100 mil ou 1 milhão de contagens.

O Pixel 2 também nunca converge para o valor real, apresentando-se muito desviado desse valor para as 10 mil contagens, bem como para 1 milhão de contagens, se bem que neste caso o desvio é menor. A curva para as 100 mil contagens é a que se pode dizer que está mais próxima do valor real.

O Pixel 3 e Pixel 5 apresentam um crescimento constante, nunca convergindo por completo para o valor real. E à semelhança do Pixel 2, a curva mais próxima do valor real, ultrapassa ligeiramente esse valor, é estabelecida para as 100 mil contagens.

O Pixel 10 apresenta ligeiros decrescimentos por entre os crescimentos, e denotamos que para situações de pouco ruído, 1 milhão de contagens, chega a convergir para o valor real.

As barras de erros, nos *errorbars* também diminuem conforme o aumento do número de contagens, não sendo visíveis a olho nu para 1 milhão de contagens. Tal como sucedeu para o estimador média.

# Capítulo 5

## Discussão e Conclusões

O algoritmo ML-EM e o algoritmo das sinocurvas são equivalentes, podendo o ML-EM ser interpretado à luz dos conceitos das curvas sinusoidais como uma média iterativa de todas as estimativas obtidas para os “sinopixéis” das respectivas “sinocurvas”.

Baseando-nos nos resultados expostos nesta tese e reportando todas as conclusões exclusivamente à imagem de teste utilizada podemos começar por assumir a importância do conceito da zona fronteira que já foi introduzido nos resultados quando se comentavam os gráficos gerados para os diferentes tipos de simulação de convergência.

No fundo por zona fronteira (ou vizinhança), entende-se uma zona que divide regiões de diferente valor, e no caso da imagem de teste estudada podíamos considerar três regiões de diferente valor:

- uma região fora da circunferência onde os *pixeis* assumiam valor 0.
- uma região definida por um quadrado onde os *pixeis* assumiam valor 5.
- uma região demarcada pela circunferência onde os *pixeis* assumiam valor 1, excepto para a região definida pelo quadrado, que se encontrava inserido na circunferência.

Quando correlacionamos o conceito de zonas fronteira com o posicionamento dos *pixeis* estudados, em particular para o estimador média (porque no geral os resultados são melhores), o Pixel 1 e o Pixel 3 são os que assumem maior predominância. O Pixel 1 enquanto *pixel* mais periférico no que à circunferência diz respeito, e pela sua proximidade com a região composta por *pixeis* de valor zero. O Pixel 3 enquanto zona fronteira entre uma região quadrada composta por *pixeis* de valor cinco que está inserida numa zona caracterizada por *pixeis* de valor um.

Para os casos estudados, tanto para as simulações de convergência sem ruído como para as simulações de convergência com ruído, o Pixel 3 apresentou um crescimento sempre por valores inferiores ao valor real do *pixel*.

A vizinhança de “zeros” assume especial importância no caso do Pixel 1, enquanto para o Pixel 3 é a vizinhança de “uns”. Podemos concluir então que como há

um crescimento por valores inferiores para ambos os *pixeis*, podemos falar num efeito limitador causado por essas vizinhanças. Assim, o valor real do Pixel 1 sendo um, não é ultrapassado por causa de um efeito limitador que se deve à vizinhança existente, zeros. O mesmo raciocínio pode ser aplicado ao Pixel 3, de valor real cinco, nunca ultrapassa este valor pelo efeito limitador motivado pela vizinhança presente, uns.

Para o Pixel 2, no geral, observamos sempre um grande crescimento logo nas primeiras iterações, prosseguindo-se um decrescimento até convergir para o valor real. Este crescimento também pode estar relacionado com o tal conceito de vizinhança.

Podemos afirmar que o Pixel 2 está relativamente perto da região definida pelo quadrado, ora possuindo o *pixel* valor real um, o crescimento acima deste valor pode relacionar-se com a proximidade dos *pixeis* com valores mais elevados, de valor cinco.

Para o Pixel 5, aproximadamente centrado no quadrado, quando ultrapassa o valor real, o desvio nunca é grande, podemos mesmo afirmar que a amplitude de desvio é mínima se tivermos como ponto de comparação a amplitude de desvio no caso do Pixel 2. Situação que faz todo o sentido, já que o impacto de uma vizinhança de valor cinco sobre um *pixel* de valor um tenderá sempre a ser superior do que a situação inversa, vizinhança de valor um sobre um *pixel* de valor cinco.

Além disso conforme podemos ver na Tabela 3, se imaginarmos o raio da vizinhança com alcance máximo de uma linha, percebemos que o Pixel 5 alcançava a posição do Pixel 3, de resto uma posição com *pixeis* de valor 5 tal como o Pixel 5 apresentava no seu posicionamento original. Já o Pixel 2 com o alcance máximo de uma linha alcançava a zona fronteira definida pelo Pixel 3. Este simples facto também explica a diferença que os *pixeis* apresentam no crescimento nas iterações iniciais.

Nomenclatura	Linha	Coluna
Pixel 2	11	13
Pixel 3	12	13
Pixel 5	13	13

**Tabela 3-Posicionamento dos *Pixel*s 2,3 e 5 em função de linha e coluna.**

O Pixel 10 acaba por ser o caso especial deste estudo, isto porque, não atinge a convergência para o estimador média nas simulações de convergência com ruído. Com 1 milhão de contagens, situação de pouco ruído, não se atingiu a convergência. Além disso na única situação deste trabalho em que assistimos a uma convergência com ruído (para 1 milhão de contagens) para o valor real, para o estimador mediana, ocorreu precisamente para o Pixel 10.

Na ausência de outros *pixeis* tão centrais e sem vizinhanças próximas, tal como sucede para o Pixel 10, assumindo que não houve erros ao gerar resultados nem como no tratamento deste somos levados a pôr a hipótese que este tipo de *pixeis* poderia ser melhor descrito pelo estimador mediana.

De futuro, acreditamos que este trabalho possa adquirir um carácter mais geral testando outras imagens diferentes daquela usada neste estudo particular, bem como utilizar um estimador diferente da média para o algoritmo das sinocurvas. O Pixel 10 é um indicador em como é necessário estudar *pixeis* afastados de vizinhanças diferentes, para perceber se a hipótese aqui exposta se pode tornar numa regra do género: *pixeis* com vizinhanças diferentes próximas são melhor descritos pelo estimador média, enquanto *pixeis* com vizinhanças diferentes muito afastadas são melhor descritos pelo estimador mediana. Verificar esta hipótese para outras imagens de teste seria muito útil.





## Bibliografia

- [1] Hoffman, E.J.; Huang, S.C.; Phelps, M.E. e Kuhl, D.E. *Quantitation in positron emission tomography: 4. Effect of accidental coincidences. Journ. Comput. Assist. Tomog.* 1981; 5(3); pp:391-400
- [2] Radon, J., *Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten.* Berichte Sächsische Akademie der Wissenschaften. Leipzig, Math-Phys.1917;69; pp: 262-267.
- [3].Dempster, A.P.; Laird, N.M.; e Rubin, D.B. *Maximum Likelihood from Incomplete Data Via Em Algorithm.* Journal of the Royal Statistical Society Series B-Methodological, 1977; 39(1); pp: 1-38.
- [4] Rockmore, A. J.; Macovski, A. *A maximum likelihood approach to emission image reconstruction from projections,* IEEE Trans. Nucl. Sci., 1976; 23; pp:1428-1432
- [5] Shepp, L. A. e Vardi, Y. *Maximum likelihood reconstruction in Positron Emission Tomography.* IEEE Transactions on Medical Imaging, 1982; 1; pp:113-122
- [6] Lange, K. e Carson R. *EM reconstruction algorithms for Emission and Transmission Tomography,* J. Comp. Ass. Tomogr., 1984; 8; pp:302-316
- [7] Caramelo, F. J; Ferreira, N. C. e Fazendeiro, L., *Image Reconstruction by Sinogram Decomposition into Sinusoidal Curves, Conf. Rec. 8th Intl. Meeting on Fully 3D Image Reconstruction in Radiology and Nuclear Medicine,* Salt Lake City, Utah, USA, 2ª submissão.
- [8] Moon, T.K. *The Expectation-Maximization Algorithm,* IEEE Signal Processing Magazine, 13,6, pp:47-60, 1996.

[9] Vardi, Y.; Shepp, L.A. e Kaufman, L. *A Statistical-Model for Positron Emission Tomography*. Journal of the American Statistical Association, 80(389): pp: 8-20, 1985.

[10] Zangwill, W. I. *Nonlinear Programming*, Prentice-Hall, Englewood Cliffs, N. J., 1969.