

# MÉTODO COMPUTACIONAL DE CONVERSÃO DE VARREDURA PARA APRESENTAÇÃO DE IMAGENS SETORIAIS POR ULTRASSOM

A. A. Assef\*, J. M. Maia\*, F. K. Schneider\*, P. M. Gewehr\*, R. da Costa\*, A. F. de Moraes Filho\*\*, V. L. S. N. Button\*\*\* e E. T. Costa\*\*\*

\*CPGEI-DAELT-DAELN/Universidade Tecnológica Federal do Paraná - UTFPR, Curitiba, Brasil

\*\*Instituto Federal do Paraná - IFPR, Curitiba, Brasil

\*\*\*DEB-FEEC e CEB/Universidade Estadual de Campinas - UNICAMP, Campinas, Brasil

e-mail: amauriassef@utfpr.edu.br

**Resumo:** A conversão de varredura é um procedimento padrão realizado em equipamentos comerciais e de pesquisa do ultrassom que consiste em mapear e interpolar os dados amostrados segundo um sistema de coordenadas polares em coordenadas retangulares para que a imagem possa ser exibida de forma correta em um monitor. Apesar de bem estabelecido, o método requer a utilização de técnicas complexas de processamento de vídeo para adequar as linhas de varredura pré-processadas em escala de cinza para dados em escala de cinza baseado em pixel (elementos de imagem). Neste trabalho é apresentado um método computacional simples e eficiente, implementado com o software Matlab® para conversão de varredura de imagens setoriais. Os resultados obtidos indicam que a aplicação poderá ser usada como ferramenta educacional para estudantes de engenharia biomédica, bem como auxiliar pesquisadores no desenvolvimento de novas aplicações do ultrassom.

**Palavras-chave:** Ultrassom, conversão de varredura, pesquisa do ultrassom.

**Abstract:** The scan conversion is a standard procedure performed on commercial and research ultrasound equipment to translate sampled data captured in Polar coordinates into Cartesian coordinates to be displayed correctly on a monitor. Although well established, the method requires the use of complex video processing techniques to adjust the preprocessing grayscale scanlines data to pixel based grayscale data. This work presents a simple and efficient computational method for sector image scan conversion, implemented in Matlab® software. The results show that the application can be used as an educational tool for students of biomedical engineering, as well as to assist researchers to develop new ultrasound applications.

**Keywords:** Ultrasound, scan conversion, ultrasound research.

## Introdução

Entre as diversas técnicas de processamento empregadas em um equipamento de geração de imagem por ultrassom pulso-eco, a conversão de varredura (*scan conversion*) representa uma etapa fundamental na apresentação correta das imagens reconstruídas para

auxílio ao diagnóstico médico [1,2]. O método é particularmente útil durante o processo de conversão de coordenadas e reconstrução de imagens bidimensionais setoriais [3], conforme ilustrado na Figura 1.

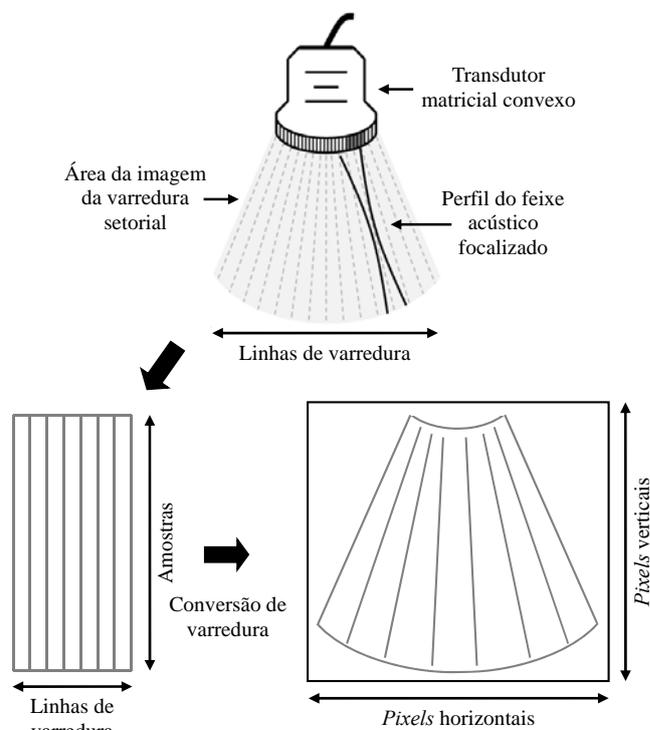


Figura 1: Processo de formação de imagem em corte transversal polar com ilustração da conversão de varredura para apresentação no monitor de imagem.

Apesar de ser uma técnica bem estabelecida, o método requer tipicamente a utilização de funções complexas de processamento de vídeo para converter e interpolar dados de sinais ultrassônicos amostrados e processados em coordenadas polares para o formato cartesiano, isto é, linhas de varredura em escala de cinza para dados em escala de cinza baseado em *pixel* (elementos de imagem), compatível com monitores convencionais de imagem [4,5].

Neste trabalho é apresentado um método computacional simples e eficiente baseado no *software* Matlab® para conversão de varredura de imagens

setoriais em Modo B, que poderá ser utilizado como ferramenta de auxílio durante as atividades de ensino e pesquisa de imageamento por ultrassom.

**Materiais e métodos**

O procedimento de conversão de varredura setorial da coordenada polar ( $r; \theta$ ) para cartesiana ( $x, z$ ) de um sistema de ultrassom com interpolação baseada nos quatro vizinhos mais próximos ( $2 \times 2$ ), considerando a geometria de um transdutor matricial convexo, é ilustrado na Figura 2. Também são apresentados os passos básicos para geração de um dado interpolado:

- Identificar os quatro pontos vizinhos de *pixel* nas adjacências de duas linhas de varredura;
- Computar os quatro coeficientes de interpolação;
- Interpoler a amostra desejada utilizando os quatro vizinhos mais próximos e computar os coeficientes de interpolação [4].

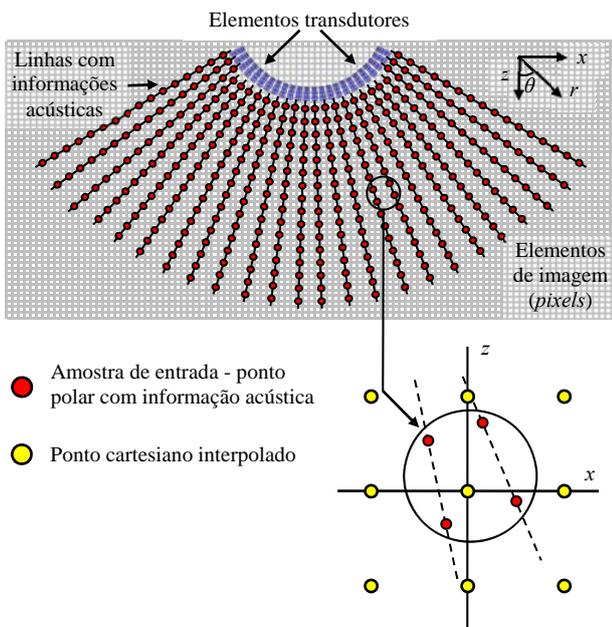


Figura 2: Exemplo ilustrativo de conversão de varredura setorial com interpolação baseada nos quatro vizinhos mais próximos.

Para ilustrar a metodologia adotada neste trabalho é apresentada na Figura 3 a geometria característica de um transdutor matricial convexo com abertura (número de elementos utilizados) de 8 elementos. Nesta, os parâmetros de raio convexo ( $R_{convexo}$ ), largura do elemento ( $d$ ) e *kerf*, que representa a distância entre elementos adjacentes, são especificados pelo fabricante do transdutor. O índice  $i$  representa o número de cada elemento transdutor da abertura de tamanho  $ne$  ( $i = 1, 2, 3, \dots, ne$ ).

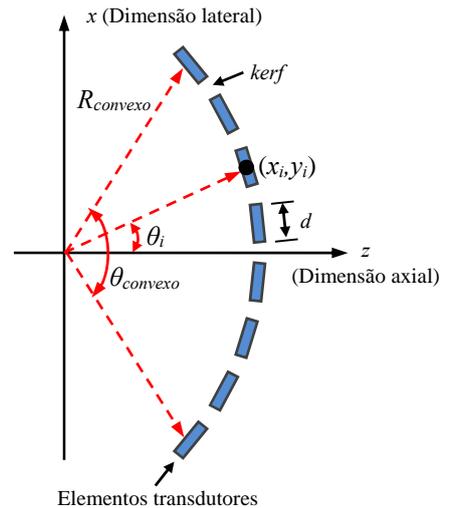


Figura 3: Geometria característica de um transdutor matricial convexo com indicação dos parâmetros utilizados para o processo de conversão de varredura.

O ângulo de abertura do transdutor convexo  $\theta_{convexo}$  é obtido pela Equação 1. O parâmetro é calculado pela relação entre o tamanho total da abertura e o comprimento da circunferência do transdutor de raio  $R_{convexo}$ .

$$\theta_{convexo} = \frac{360 \cdot C_{abertura}}{C_{circunferencia}} = \frac{360 \cdot (ne - 1) \cdot (d + kerf)}{2\pi R_{convexo}} \quad (1)$$

onde  $ne$  representa o número de elementos da abertura.

Dessa forma, o ângulo de cada elemento  $i$  em relação ao eixo  $x$  no centro da abertura pode ser calculado através da Equação 2:

$$\theta(i) = \left( i - \left( \frac{ne + 1}{2} \right) \right) \cdot \frac{\theta_{convexo}}{(ne - 1)} \quad (2)$$

A próxima etapa do método compreende ao cálculo das posições relativas das amostras dos vetores correspondentes às linhas de varredura. A partir destes dados e da velocidade do som no meio é possível calcular o vetor de profundidade que será utilizado para a correção das coordenadas finais.

Com isso, os vetores de tempo  $t(n)$  e profundidade  $z(n)$  discretos no tempo e amostrados na frequência  $f_s$  podem ser descritos, respectivamente, pelas Equações 3 e 4:

$$t(n) = \frac{n}{f_s}, \quad 1 \leq n \leq N \quad (3)$$

$$z(n) = c \cdot \frac{t(n)}{2} + R_{convexo} \quad (4)$$

onde  $n$  denota o índice da amostra discreto no tempo,  $N$  é o número total de amostras de cada linha e  $c$  é a velocidade do som no meio.

Na etapa final são exploradas as funções *meshgrid*, *pol2cap* e *surf* do software Matlab® para:

- Transformar o domínio especificado pelos vetores  $\theta(i)$  e  $z(n)$  em duas matrizes *THETA* e *RHO*, respectivamente, contendo linhas e colunas repetidas, para funcionarem como novo domínio da função. Essas matrizes serão usadas para preparar a plotagem de superfície do gráfico;
- Converter os pontos de coordenadas polares (*THETA*, *RHO*) para cartesianas ( $x, y, z$ );
- Plotar a superfície do gráfico tridimensional da função das variáveis.

## Resultados

Os principais parâmetros utilizados para simulação e validação do método de conversão de varredura são apresentados na Tabela 1, levando-se em conta a geometria de um transdutor matricial convexo comercial (AT3C52B, Broadsound Corporation, Taiwan).

Tabela 1: Especificações dos parâmetros utilizados nas simulações.

Parâmetro	Notação	Valor	Unidade
Velocidade do som no meio	$c$	1540	m/s
Frequência de amostragem	$f_s$	40	MHz
Número de amostras	$N$	4096	-
Número de elementos	$ne$	128	-
<i>Kerf</i>	$kerf$	115	$\mu\text{m}$
Largura do elemento	$d$	0,41	mm
Raio de curvatura	$R_{convexo}$	40	mm

Nas Figuras 4(a) e 4(b) são apresentadas as imagens bidimensionais em Modo B com compressão dinâmica de 50 dB e profundidade total 78,84 mm, resultantes da simulação do método utilizado. Na Figura 4(a) é mostrada a imagem original nas coordenadas polares com inclusões nas faixas de 0 dB (cor branca), -25 dB (cor cinza) e -50 dB (cor preta), enquanto na Figura 4(b) é mostrada a imagem resultante nas coordenadas retangulares após a transformação de coordenadas.

## Discussão

Como as imagens de ultrassom em Modo B são bidimensionais, após a aplicação da função de plotagem de superfície *surf*, definida por uma matriz de quatro argumentos, é realizado o ajuste do ângulo de visualização do gráfico para o modo padrão 2-D através do comando do Matlab *view(90,90)*. O tempo total para a execução da rotina em computador com processador

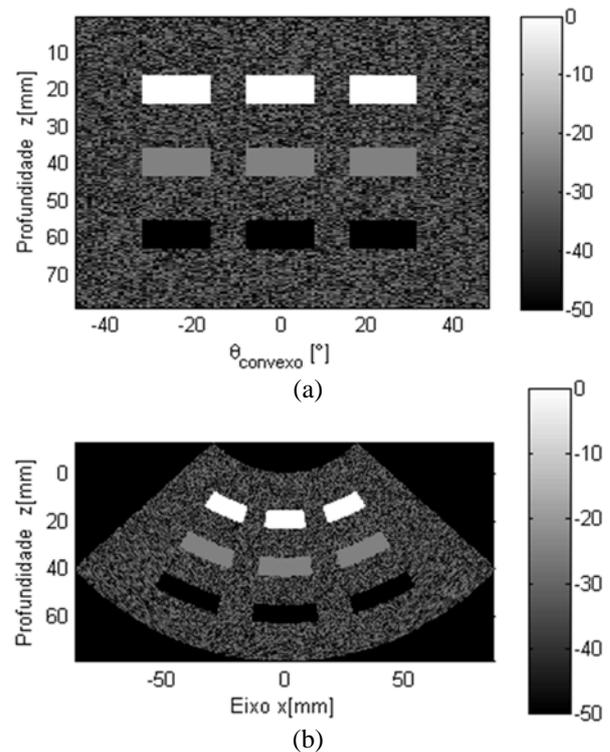


Figura 4: (a) Representação de imagem simulada nas coordenadas polares. (b) Imagem em coordenadas retangulares obtida após a conversão de varredura pela técnica proposta.

Core i3, frequência de 2,4 GHz e memória RAM de 4 GB é inferior a 1 segundo. Entretanto esse tempo pode ser ainda menor, uma vez que os dados das linhas de varredura, após as técnicas de demodulação, detecção de envoltória e compressão dinâmica, já estejam disponíveis para processamento. A rotina completa para a demonstração do método de conversão de varredura é apresentado como apêndice no final deste trabalho.

Estudos para a implementação de novos algoritmos de conversão de varredura utilizando o software Microsoft Visual C Studio, bem como, dispositivos de lógica reconfigurável FPGA (*Field-Programmable Gate Array*), já estão em andamento e poderão ser utilizados posteriormente como base comparativa para avaliação do custo computacional das técnicas empregadas.

## Conclusão

A temática do método de conversão de varredura apresentado tem como principal diferencial a utilização de funções proprietárias do Matlab que permitem a otimização dos algoritmos de preenchimento de polígonos e superfícies, organizado em torno de um paradigma de rastreamento de bordas. Através do exemplo de simulação, ilustrado nas Figuras 4(a) e (b) e no código em apêndice, é possível verificar que o método é bastante simples e adequado como uma ferramenta pedagógica para estudantes de engenharia biomédica, bem como para pesquisadores no

desenvolvimento de novas aplicações do ultrassom.

## Agradecimentos

Ao CNPq, FINEP, Fundação Araucária, CAPES e Ministério da Saúde pelo apoio financeiro para o desenvolvimento da pesquisa.

## Referências

- [1] Schueler CF, Lee H, Wade, G. Fundamentals of digital ultrasonic imaging. IEEE transactions on sonics and Ultrasonics. 1984; 31(4):195-217.
- [2] Berkhoff AP, Huisman HJ, Thijssen JM, Jacobs EM, Homan RJ. Fast scan conversion algorithms for displaying ultrasound sector images. Ultrasonic Imaging. 1994; 16(2):87-108.
- [3] Kassem A, Sawan M, Boukadoum M. A scan conversion CMOS implementation for a portable ultrasonic system. In: Proceedings of the Canadian Conference on Electrical and Computer Engineering - IEEE CCECE 2003; 2003 May; 3:1461-1464.
- [4] Xiaohui Li. Ultrasound Scan Conversion on TI's C64x+ DSPs. Texas Instruments Incorporated – White Paper [internet]. 2009 Mar. Available from: <http://www.ti.com/lit/an/sprab32/sprab32.pdf>.
- [5] Robinson DE, Knight PC. Interpolation scan conversion in pulse-echo ultrasound. Ultrasonic Imaging, 1982; 4(4):297–310.

## Apêndice

A seguir é apresentada a rotina de conversão de varredura desenvolvida no *software* Matlab versão 7.10.

```

%-----
% Rotina de conversão de varredura
%-----
clear all;
close all;
clc;

tic;                % inicia temporizador

% Definição de parâmetros de entrada
ne = 128;           % numero de elementos
i=1:ne;            % índice dos elementos
kerf = 115e-6;     % kerf
d = 0.41e-3;       % largura do elemento
R = 40e-3;         % raio do transdutor
N = 4096;          % numero de amostras
n = 1:N;           % índice das amostras
c = 1540;          % velocidade do som
fs = 40e6;         % frequência de amostragem
t = (n/fs)';      % escala de tempo

%-----
% Ângulo de abertura do transdutor convexo
%-----
c_transdutor = 2*pi*R;
c_abertura = (ne-1)*(kerf + d);
theta_convexo = 360*c_abertura/c_transdutor;

%-----
% Ângulo de abertura do transdutor convexo
% de cada elemento
%-----

```

```

theta_convexo_i = (i-
(ne+1)/2)*theta_convexo/(ne-1);

%-----
% Escala de profundidade
%-----
z = c*t/2+R;

%-----
% Buffer de dados para simulação
%-----
valor_final_Env_Log = (rand(N,ne)-2)*25;

%-----
% Dados das inclusões para teste
%-----
a = 200; b = 10;

%-----
% Equação das inclusões
%-----
valor_final_Env_Log ((N/4)-a:(N/4)+a,ne/4-
b:ne/4+b)= 0;
valor_final_Env_Log ((N/4)-a:(N/4)+a,ne/2-
b:ne/2+b)= 0;
valor_final_Env_Log ((N/4)-a:(N/4)+a,ne*3/4-
b:ne*3/4+b)= 0;

valor_final_Env_Log ((N/2)-a:(N/2)+a,ne/4-
b:ne/4+b)= -25;
valor_final_Env_Log ((N/2)-a:(N/2)+a,ne/2-
b:ne/2+b)= -25;
valor_final_Env_Log ((N/2)-a:(N/2)+a,ne*3/4-
b:ne*3/4+b)= -25;

valor_final_Env_Log ((N*3/4)-a:(N*3/4)+a,ne/4-
b:ne/4+b)= -50;
valor_final_Env_Log ((N*3/4)-a:(N*3/4)+a,ne/2-
b:ne/2+b)= -50;
valor_final_Env_Log ((N*3/4)-
a:(N*3/4)+a,ne*3/4-b:ne*3/4+b)= -50;

[THETA,RHO] =
meshgrid(deg2rad(theta_convexo_i),z);

[xc,yc] = pol2cart(THETA,RHO);

%-----
% Imagem simulada antes da conversão de varre-
dura
%-----
figure; colormap(gray);

imagesc(theta_convexo_i,(z-
R)*1e3,valor_final_Env_Log);

xlabel('\theta_c_o_n_v_e_x_o [\circ]');
ylabel('Profundidade z[mm]');
axis tight; colorbar;

%-----
% Imagem simulada após a conversão de varredura
%-----
figure; colormap(gray);

h=surf((xc-
R)*1e3,(yc)*1e3,valor_final_Env_Log,'edgecolor'
,'none');

view(90,90);
xlabel('Profundidade z[mm]');
ylabel('Eixo x[mm]');
axis image; colorbar;

toc                % finaliza temporizador
%-----

```