UNIVERSIDADE DO VALE DO RIO DOS SINOS — UNISINOS
UNIDADE ACADÊMICA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA
NÍVEL MESTRADO

LUCAS ADAMS SEEWALD

KINECT EM CONJUNTO COM O SRP-PHAT COMO SOLUÇÃO DE LOCALIZAÇÃO DE FONTE SONORA

SÃO LEOPOLDO
2014

Lucas Adams Seewald

KINECT EM CONJUNTO COM O SRP-PHAT COMO SOLUÇÃO DE LOCALIZAÇÃO DE FONTE SONORA

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre pelo Programa de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos — UNISINOS

Orientador:
Prof. Dr. Luiz Gonzaga da Silveira Junior

São Leopoldo
2014

S453k Seewald, Lucas Adams.
Kinect em conjunto com o SRP-PHAT como solução de localização de fonte sonora / Lucas Adams Seewald. – 2014.
88 f. : il. ; 30 cm.

Dissertação (mestrado) – Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, 2014.
"Orientador: Prof. Dr. Luiz Gonzaga da Silveira Junior.”

1. SRP-PHAT. 2. Localização de fonte sonora. 3. Kinect. I. Título.

CDU 004

Dados Internacionais de Catalogação na Publicação (CIP)
(Bibliotecário: Flávio Nunes – CRB 10/1298)

(Esta folha serve somente para guardar o lugar da verdadeira folha de aprovação, que é obtida após a defesa do trabalho. Este item é obrigatório, exceto no caso de TCCs.)

AGRADECIMENTOS

Em primeiro lugar agradeço a Deus por não me largar de mão ao longo destes dois anos. Agradeço em especial a minha esposa Marina, que depois de sofrer todo o processo ela mesma, ainda teve de suportar o meu. Profundos agradecimentos aos meus pais Sadi e Marta, juntamente com a mana Luana, a querida vó Helmi e ao querido vô Bilo, que agora descansa. Da mesma forma agradeço aos meus sogros Sergio e Lilian, à Morgana e ao Marcelo, à vó Rosa e ao trio Kiko, Suzi e Tui. Obrigado por todas as acolhidas, compreensão e paciência. Após viver dois anos com as prioridades invertidas, só posso pedir desculpas e correr atrás do prejuízo.

E como seria possível caminhar sem amigos que compartilhem das mesmas lutas? Obrigado Fabrício e Vanessa, Róbson e Emili, André, Débora e Ricardo! Não sei o que seria desse mestrado sem vocês. E também aos muitos amigos que fizemos: Ivan, Celeste, James, Cláudio, Luana, Naira, Luan, Roberto e tantos mais que nem consegui descobrir o nome. Por todas as ótimas conversas e experiências trocadas, obrigado Leonel, Maurício, Rodolfo e Dormento.

Agradeço ao Vicente Minotto por compartilhar seu conhecimento e experiência na área, me contextualizando na localização de fonte sonora. Também agradeço ao Lucas Kaspary e ao Gonzaga por emprestarem seus Kinects para a implementação do protótipo e realização dos experimentos. Grato ao atencioso pessoal da Geologia trabalhando no Vizlab, em especial os professores Maurício, Marcelo e Leonardo.

Agradecimentos muito fortes aos amigos da V3D, o que inclui meu orientador Gonzaga, o Fábio (e o Mierlo), meu amigão César (conhecido como Afobadik, Pedro Maconheiro, Seaser, Sézar e tantos outros), o oráculo Leandro (LMB) e o companheiro de todas as batalhas Fabrício (o mesmo de antes). Obrigado pela força, amizade e por criarem o melhor ambiente de trabalho ever!

Obrigado ao time da i360 Tecnologia por me aguardarem até o fim da escrita deste trabalho. São parte deste time Roger, Fabrício (sim, é ele outra vez!), Gabriel e Lucas!

Agradeço ao Pink Floyd por Welcome to the Machine, sem a qual o artigo não teria sido escrito. Da mesma forma ao Sambomaster pelas injeções de ânimo quando não haviam forças. Obrigado ao Inafking e à equipe Nigoro pelas boas expectativas.

Obrigado ao professor Dr. Gonzaga por aceitar me orientar e por ser um bom amigo nesses tempos que foram difíceis para ele também. Obrigado aos doutores Paulo Luna, Rodrigo Righi e Cláudio Jung por suas participações e contribuições. Por fim, obrigado à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e à Unisinos por me agraciarem com uma bolsa PROSUP e infraestrutura necessária para a realização do trabalho.

"*Alistem-se, diziam eles... Venham conhecer o mundo, diziam eles...*".
(Asterix, o gaulês)

RESUMO

Este documento apresenta uma avaliação de aplicabilidade do Kinect em conjunto com o SPR-PHAT como solução de Localização de Fonte Sonora. Um protótipo capaz de se comunicar com o aparelho e executar SRP-PHAT foi implementado com a finalidade de testar a precisão da solução. É realizada uma revisão dos fundamentos da Localização de Fonte Sonora e seus princípios matemáticos, com foco específico no SRP-PHAT. Seguindo para o Kinect, são realizadas algumas considerações a respeito de seus componentes e limitações. São apresentados alguns trabalhos que recorrem ao aparelho para localizar fontes sonoras, seguidos de resultados de precisão do SRP-PHAT obtidos por diferentes autores. Foram realizados dois grupos de experimentos, um voltado para as características da fonte sonora e o outro para a qualidade da solução proposta. Os experimentos incluem localização em duas e três dimensões, utilizando dois Kinects no segundo caso. As particularidades de implementação do programa que manipula os Kinects e executa o algoritmo de localização são fornecidas juntamente com descrições dos procedimentos de teste adotados. Os resultados apresentados mostram que a solução é capaz de apontar com precisão para a direção da fonte.

Palavras-chave: SRP-PHAT. Localização de Fonte Sonora. Kinect.

ABSTRACT

This document presents an evaluation of Kinect together with SRP-PHAT as a Sound Source Localization solution. A functional prototype able to communicate with the device and perform SRP-PHAT was implemented in order to test the solution's accuracy. The fundamentals of Sound Source Localization and it's mathematical principles are reviewed, focusing specifically on the SRP-PHAT. Moving on to the Kinect device, some considerations are made about it's components and limitations. Related work which resources to Kinects source localization capabilities is presented, followed by SRP-PHAT precision test results attained by different authors. Two experimental sets were conducted, one focused on the source signal properties and the other on measuring the proposed solutions quality. Performed experiments comprehend two dimensional and three dimensional localization, being a second Kinect needed for the latter. Implementation aspects concerning the software responsible for manipulating both Kinects and executing the localization algorithm are described along with experimental procedure details. Presented results show that the proposed solution can accurately point at the sources direction.

Keywords: SRP-PHAT. Sound Source Localization. Kinect.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

API	*Application Programming Interface*
CC	Correlação Cruzada
CPU	*Central Processing Unit*
GCC	*Generalized Cross-Correlation*
GCC-PHAT	*Generalized Cross-Correlation with PHAse Transform*
GLONASS	*Global Orbiting Navigation Satellite System*
GPS	*Global Positioning System*
GPU	*Graphics Processing Unit*
ML	*Maximum Likelihood*
PHAT	*PHAse Transform*
SDK	*Software Development Kit*
SNR	*Signal-to-Noise Ratio*
SRP	*Steered Response Power*
SRP-PHAT	*Steered Response Power with PHAse Transform*
SSL	*Sound Source Localization*

LISTA DE SÍMBOLOS

cm	Centímetro
kHz	Quilo-hertz
m	Metro
mm	Milímetro
Hz	Hertz

SUMÁRIO

1 INTRODUÇÃO

O campo da Localização de Fonte Sonora, ou SSL (do inglês *Sound Source Localization*), tem experimentado um significativo amadurecimento nas últimas décadas. Com o objetivo de determinar a direção ou posição de uma ou mais fontes sonoras a partir de seus sinais sonoros, uma série de métodos têm sido propostos (HAMON; HANNAN, 1974; KNAPP; CARTER, 1976; BRANDSTEIN; ADCOCK; SILVERMAN, 1995; BENESTY, 2000; DIBIASE, 2000; RUI; FLORENCIO, 2003). Na medida em que os métodos foram se mostrando eficazes, maior se tornou sua popularidade, juntamente com o número de contextos nos quais têm sido utilizados.

Entre as principais áreas de aplicação das soluções de SSL se encontram videoconferências (WANG; CHU, 1997), reconhecimento de fala (NAKADAI et al., 2011), vigilância (GALATAS; FERDOUS; MAKEDON, 2013) e robótica (NAKADAI et al., 2011; HWANG; CHOI, 2011). Também foram fatores benéficos para o desenvolvimento destas soluções o surgimento de processadores com maior poder computacional e o advento de placas gráficas voltadas para o processamento de alto desempenho, pois tornaram viáveis soluções que antes eram consideradas excessivamente custosas computacionalmente. Em alguns casos se tornou possível, inclusive, a execução em tempo real dos algoritmos (POURMOHAMMAD; AHADI, 2012; DO; SILVERMAN; YU, 2007; HWANG; CHOI, 2011).

Tradicionalmente, soluções de SSL empregam arranjos de microfones para a captura de sinais sonoros, a partir dos quais é realizada uma estimativa de posicionamento. O número de microfones que compõem estes arranjos pode variar de unidades (POURMOHAMMAD; AHADI, 2012) a centenas (DO, 2010). O custo computacional e a precisão das estimativas estão diretamente relacionados ao total de microfones utilizados.

1.1 Motivação e apresentação do problema

Dentre os algoritmos de SSL, o SRP-PHAT, ou em inglês *Steered Response Power using the PHAse Transform* (detalhado na Seção 2.2), se destaca pela qualidade observada em seus resultados quando submetido a cenários reais (DIBIASE, 2000) mesmo na presença de reverberação. O método se encontra bem consolidado e já foram propostas várias otimizações que viabilizam sua execução em tempo real (SILVEIRA JR. et al., 2010; DO; SILVERMAN; YU, 2007; DO; SILVERMAN, 2009; COBOS; MARTI; LOPEZ, 2011). À medida que o poder de processamento deixa de ser um gargalo, outros fatores que restringem uma exploração mais ampla dos algoritmos passam a ser relevantes. Entre estes fatores está o custo elevado de arranjos de microfones especializados.

Graças a inovações tecnológicas recentes por parte da indústria dos *videogames*, encontram-se popularizados e amplamente disponíveis no mercado dispositivos de baixo custo com arranjos de microfones. É o caso do Kinect (Microsoft Corporation, 2010) especificamente, criado

pela Microsoft com a finalidade de reconhecer comandos de gesto e voz. O aparelho dispõe de 4 microfones e inclui recursos que auxiliam na localização de locutores e realçamento de voz. Todavia é importante destacar que a localização efetuada pelo Kinect apenas aponta para a direção da fonte sonora, não sua posição. Desde seu lançamento em 2010 o dispositivo tem sido explorado por sua comunidade de usuários para os mais diversos fins (Kinect Hacks, 2013).

Apesar da popularidade, são relativamente poucos os projetos que utilizam o aparelho para a Localização de Fonte Sonora propriamente dita. A grande maioria concentra seu foco nas capacidades de captura e processamento de imagens do dispositivo. Conforme será mostrado no Capítulo 3, enquanto alguns atingem seus objetivos apenas com a direção da fonte sonora, outros recorrem às câmeras para efetuar a localização, verificando nos microfones sinais de coerência com os resultados obtidos. Este cenário evidencia a subutilização dos microfones do Kinect como ferramentas de SSL.

Devido ao destaque do SRP-PHAT em relação aos demais algoritmos de SSL, sua escolha parece ser a mais apropriada para execução em conjunto com o Kinect. Uma vez que a qualidade do algoritmo já foi averiguada (DIBIASE, 2000; ZHANG; FLORENCIO; ZHANG, 2008), entram em questão as características do aparelho. Uma série de fatores pode influenciar seus resultados, como a sensibilidade dos microfones, sua taxa de amostragem ou mesmo ruídos introduzidos pelo próprio dispositivo. É necessário, portanto, que o aparelho capture os sinais com fidelidade para que o algoritmo seja eficaz.

Considerando os projetos de localização que já utilizam o Kinect, a combinação proposta tem o potencial de oferecer maior precisão e robustez. Em casos onde a tarefa é realizada pelas câmeras, o método poderia substituir ou atuar em conjunto com as mesmas para superar condições adversas, como por exemplo iluminação insuficiente ou excessiva. Em termos de solução de SSL, a solução ofereceria os benefícios do preço acessível e facilidade de transporte.

Embora uma solução desta natureza possa ser aplicada a diversos campos que já utilizam SSL, a motivação inicial deste trabalho foi a de estabelecer os fundamentos para uma possível solução de Posicionamento *Indoor*. Sistemas de navegação e posicionamento geográfico, tais como o *Global Positioning System* (GPS) dos Estados Unidos (LOCH; CORDINI, 1995) e o Global Orbiting Navigation Satellite System (GLONASS) da Rússia (MONICO, 2007), necessitam de ambientes abertos para manter a comunicação entre seus dispositivos e satélites. Mesmo ambientes externos com grande concentração de edifícios ou vegetação densa podem comprometer seu funcionamento. Sistemas de Posicionamento *Indoor* têm atacado este problema através de uma diversidade de técnicas e tecnologias (LIU et al., 2007). Tendo esta questão em mente, considerou-se a possibilidade de utilizar dispositivos Kinect em conjunto com uma solução de SSL para determinar posições desconhecidas em ambientes fechados. Todavia, antes de aplicá-los a qualquer contexto, cabe avaliar a precisão do dispositivo e algoritmo escolhidos em termos de solução de SSL. Por este motivo ao invés de se concentrar no Posicionamento *Indoor* ou em outra área de aplicação para a solução, o presente trabalho se dedicará a medir sua precisão para que a partir daí seja possível determinar os campos nos quais ela pode

ser aplicada.

1.2 Objetivos

É o objetivo geral deste trabalho estudar a viabilidade do algoritmo SRP-PHAT em conjunto com o Kinect como alternativa de *hardware* para Localização de Fonte Sonora. Com esta finalidade foram estipulados os seguintes objetivos específicos:

- Implementar um protótipo capaz de acessar os microfones do Kinect e executar o SRP-PHAT;
- Avaliar a precisão das estimativas do SRP-PHAT utilizando o Kinect.

Além dos objetivos mencionados, as seguintes contribuições também foram buscadas:

- Estender o protótipo para realizar buscas em espaço tridimensional;
- Avaliar a precisão das estimativas no espaço tridimensional.

1.3 Estrutura da dissertação

Dando aprofundamento ao trabalho proposto, o Capítulo 2 apresenta os conceitos básicos necessários para sua compreensão. Estes se concentram principalmente nos aspectos matemáticos e técnicos envolvidos nos algoritmos de Localização de Fonte Sonora abordados. Também serão descritas as características do Kinect, com maior enfoque em suas funcionalidades de áudio.

O documento prossegue com o Capítulo 3, onde serão brevemente apresentados trabalhos relacionados. Embora existam muitos projetos centrados nas capacidades do Kinect, foram encontrados poucos que realmente compartilhem fatores em comum com o presente trabalho a ponto de justificar sua inclusão no documento. Aparte do Kinect, são também tratados alguns resultados de testes de precisão do SRP-PHAT disponíveis na literatura.

Finalmente, no Capítulo 4 são pormenorizados os aspectos implementacionais e os procedimentos adotados para os testes de precisão da solução proposta neste trabalho. Primeiramente são apresentadas características e escolhas com respeito ao protótipo implementado. Em seguida a ênfase recai sobre os testes de precisão, detalhando seu ambiente, configurações e metodologia.

O Capítulo 5 mostra os resultados obtidos a partir dos experimentos do capítulo anterior. Após apresentá-los o capítulo prossegue com uma série de observações e questões que surgiram ao longo da execução dos testes e após obtidos os resultados. Por fim, o documento encerra no Capítulo 6 onde são apresentadas as conclusões finais do trabalho e são sugeridas etapas futuras para seu desenvolvimento.

2 CONCEITOS BÁSICOS

Neste capítulo serão apresentados os conceitos necessários para a compreensão do trabalho proposto. A Seção 2.1 introduzirá a representação matemática de cenários típicos de Localização de Fonte Sonora e as premissas assumidas pelo SRP-PHAT com respeito a tais cenários. Ainda na mesma seção será explicado o importante conceito de diferença de tempos de chegada, ou TDOA (do inglês *Time Difference of Arrival*), fundamental para a compreensão dos algoritmos de SSL. Estes, por sua vez, serão abordados logo em seguida na Seção 2.2. O objetivo desta seção é detalhar o funcionamento do SRP-PHAT, porém, para melhor compreendê-lo é necessário conhecer os algoritmos que constituem sua base. Por este motivo serão primeiramente expostos os algoritmos de SSL cuja progressão desencadeou na criação do SRP-PHAT. Para encerrar o capítulo, a Seção 2.3 apresentará o dispositivo de *hardware* em questão, chamado Kinect.

2.1 Captação de sons com arranjos de microfones

O algoritmo SRP-PHAT parte de um modelo acústico simplificado. A técnica assume que as ondas sonoras se propagam linearmente, de forma que os caminhos entre o emissor sonoro e os microfones possam ser modelados como um sistema linear. Para assegurar a linearidade deste sistema DiBiase (2000) estabeleceu as seguintes premissas:

- a fonte emite ondas esféricas;
- o meio de propagação é homogêneo;
- o meio de propagação é não dispersivo;
- a propagação pelo meio não causa atenuação;
- o efeito Doppler é desprezível.

A primeira premissa estipula um emissor pontual omnidirecional, evitando a complexidade oriunda de padrões de propagação mais elaborados. As ondas serão, portando, esféricas, devendo manter a velocidade constante para que sua propagação seja linear. Para que a velocidade seja mantida, o sinal não pode sofrer refração ou dispersão durante o seu percurso. Um meio não dispersivo e homogêneo garante que nenhum destes fenômenos alterará o sinal.

Em meios onde ocorre absorção da força do sinal se observa uma atenuação diferenciada para cada faixa de freqüência. Esta diferenciação é desprezada ao assumir que a propagação pelo meio não causa a atenuação do sinal. O fator exclusivo a determinar a atenuação passa a ser o espalhamento da onda sonora.

Por fim, no que diz respeito ao efeito Doppler, o fenômeno descreve a influência que o deslocamento de um emissor ou receptor sonoro exerce sobre o comprimento das ondas. O

modelo não exclui a possibilidade de haver movimento por parte de seus componentes, porém assume que a velocidade destes é insignificante se comparada à velocidade do som, de forma que o efeito possa ser ignorado.

2.1.1 Representação geométrica de microfone e emissor

O espaço físico que compreende o sistema de arranjos de microfones, o emissor e o meio de propagação entre eles pode ser traduzido para o espaço cartesiano (DIBIASE, 2000). Considere-se inicialmente um único microfone m e um emissor sonoro q, ambos pontuais. Suas posições são dadas pelos vetores $\vec{p}_m,\ \vec{p}_q\ \in\ \Re^3$ que descrevem suas distâncias em relação à origem dos eixos cartesianos. Seguem abaixo os vetores:

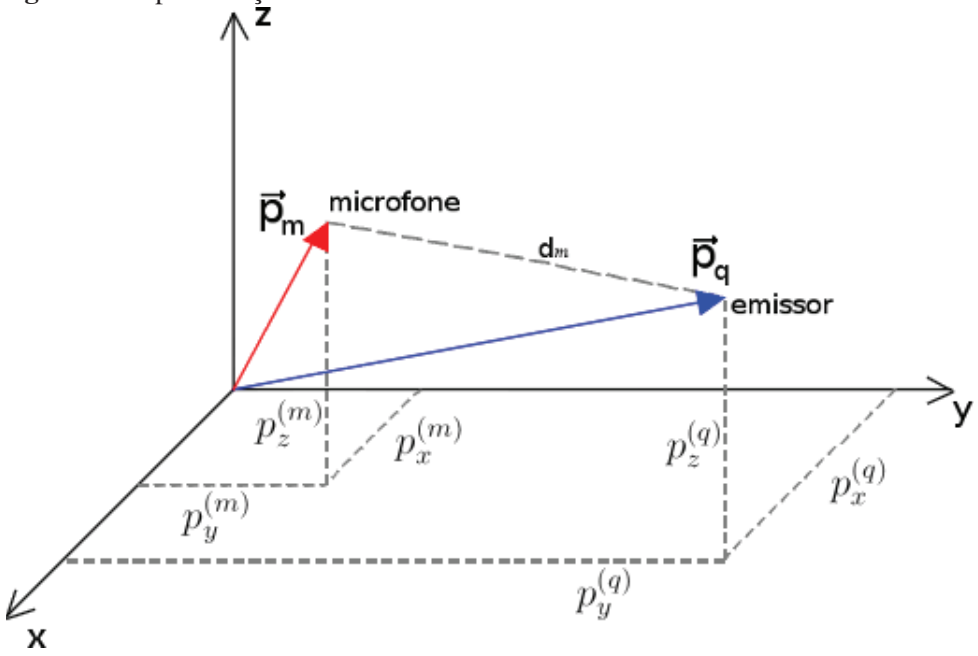
$$\begin{aligned} \vec{p}_m &= (p_x^{(m)},\ p_y^{(m)},\ p_z^{(m)}) \\ \vec{p}_q &= (p_x^{(q)},\ p_y^{(q)},\ p_z^{(q)}) \end{aligned} \tag{2.1}$$

Chame-se d_m à distância entre q e m que, neste modelo, pode ser obtida da seguinte forma:

$$d_m = |\vec{p}_m - \vec{p}_q| \tag{2.2}$$

onde $|\vec{p}_m - \vec{p}_q|$ denota a magnitude do vetor resultante da subtração dos vetores de distância $\vec{p}_m$ e $\vec{p}_q$, ou seja, é a distância euclidiana entre as duas posições. A Figura 1 ilustra $\vec{p}_m$ e $\vec{p}_q$ juntamente com a distância d_m.

Figura 1: Representação cartesiana de um sistema com um microfone e um emissor



Fonte: adaptado de (DIBIASE, 2000)

Uma vez descrita a situação espacial dos componentes do sistema, o foco passa para o

deslocamento de sinais sonoros entre os mesmos. Tanto neste capítulo quanto no restante deste trabalho são considerados emissor e microfone estacionários. Como o sistema é linear, sendo constante a velocidade de propagação de suas ondas sonoras, o tempo de propagação depende somente da distância percorrida e da velocidade de propagação do som pelo meio. Para o propósito deste trabalho, o único meio de interesse é o ar. A velocidade de propagação do som pelo ar será representada pela constante c, cujo valor adotado é de $343,3$ metros por segundo. Para determinar o tempo de propagação τ_m que um sinal leva para percorrer a distância d_m basta resolver:

$$\tau_m = d_m/c \tag{2.3}$$

A distância d_m, que estabelece um caminho direto entre emissor e microfone, será medida em metros ao longo do escopo deste estudo. Conseqüentemente, o valor obtido de τ_m expressa o tempo em segundos.

O conhecimento do tempo de propagação do emissor ao microfone constitui a base para o cálculo da diferença dos tempos de chegada ou TDOA. Existe uma variedade de algoritmos baseados neste valor para estimar a posição da fonte sonora (abordados na Seção 2.2) ou aprimorar o sinal aplicando técnicas de realçamento de voz ou supressão de ruído. A Subseção 2.1.2, a seguir, apresentará o conceito de TDOA em maiores detalhes.

2.1.2 Diferença de tempos de chegada

No exemplo da seção anterior foi considerado um sistema dotado de um único microfone m. A partir de agora se considere o caso geral: um arranjo de M microfones onde m_i e m_k são um par qualquer deste arranjo e $i, k = 1...M$. Uma vez conhecidas as distâncias d_i e d_k, a Equação 2.3 fornece os valores de τ_i e τ_k, que são o tempo de propagação do sinal desde o emissor até os microfones m_i e m_k respectivamente. Considerando apenas o par de microfones, a diferença de tempos de chegada, ou *Time Difference Of Arrival*(TDOA), entre eles pode ser obtida da seguinte forma:

$$\tau_{ik} = \tau_i - \tau_k \tag{2.4}$$

Em conjunto com a Equação 2.3, o TDOA pode ser reescrito em função das distâncias como:

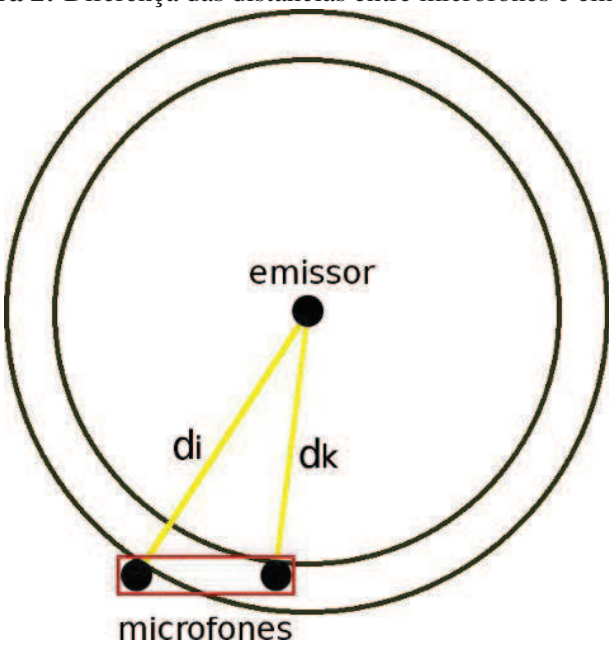
$$\tau_{ik} = \frac{d_i - d_k}{c} \tag{2.5}$$

A Figura 2 ilustra um sistema dotado de um emissor e dois microfones, todos pontuais. Os microfones estão alinhados horizontalmente, porém a distâncias diferentes do emissor, conforme indicado em d_i e d_k.

Ao reescrever a equação uma segunda vez, agora expressando distâncias em termos de TDOA, DiBiase (2000) evidencia que TDOAs parametrizam a localização da fonte:

$$d_i - d_k = c\tau_{ik} \tag{2.6}$$

Figura 2: Diferença das distâncias entre microfones e emissor.



Fonte: Elaborada pelo autor

Sendo assim, é possível estimar a posição do emissor, uma vez que se conheça um conjunto suficientemente grande de TDOAs entre pares de microfones. Conforme demonstra Tellakula (2007), um único par é insuficiente para a realização da estimativa. Um mínimo de três microfones se fazem necessários para efetuar a localização em espaço bidimensional.

Existem infinitas posições no espaço capazes de originar um dado valor de TDOA entre dois microfones. Uma vez conhecido o valor de τ_{ik}, todos os valores de d_i e d_k que satisfazem a Equação 2.5 formam o espaço de posições possíveis. É interessante observar que um único valor de TDOA descreve um hiperbolóide no espaço tridimensional. Conforme Brandstein (1995) este hiperbolóide tem por centro o ponto médio $\vec{a}$:

$$\vec{a} = \frac{\vec{p_i} + \vec{p_k}}{2} \tag{2.7}$$

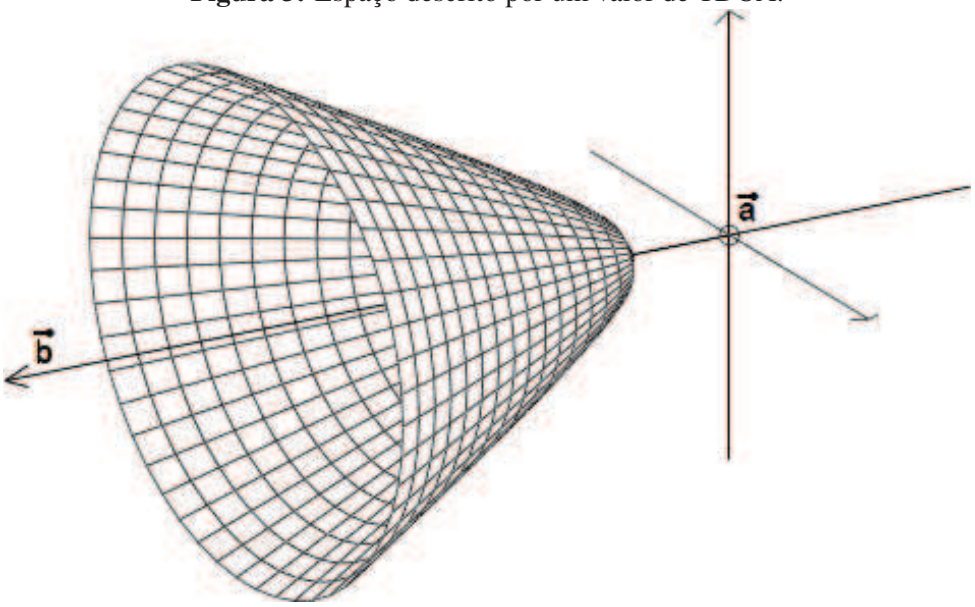
e seu eixo de simetria em $\vec{b}$:

$$\vec{b} = \frac{\vec{p_i} - \vec{p_k}}{|\vec{p_i} - \vec{p_k}|} \tag{2.8}$$

A Figura 3 ilustra a metade de um hiperbolóide no espaço 3D. Os microfones se encontram sobre o eixo $\vec{b}$, eqüidistantes do ponto $\vec{a}$. Já o emissor pode estar situado em qualquer ponto da superfície do hiperbolóide.

O objetivo desta seção se limita a introduzir o conceito de TDOA e sua formulação dentro do modelo adotado. Sua importância se tornará mais clara na Seção 2.2, onde o TDOA constituirá a base dos algoritmos apresentados. Uma vez compreendidas as características do modelo acústico, sua modelagem geométrica e cálculo dos tempos de chegada dos sinais, passa-se agora a abordar a modelagem matemática dos sinais sonoros propagados.

Figura 3: Espaço descrito por um valor de TDOA.



Fonte: Adaptado de (BRANDSTEIN, 1995)

2.1.3 Modelo do sinal

Um cenário típico de captura de áudio inicia com a geração de um sinal (por parte do emissor), sua passagem pelo meio de transmissão e subseqüente chegada aos microfones. Porém, o sinal obtido pelos microfones não será o mesmo tal qual gerado pelo emissor. Espera-se de um ambiente realista a presença de outros sinais sonoros, que por não fazerem parte do sinal de interesse são rotulados como ruído. Sabe-se que o sinal original será recebido por todos os microfones, ainda que não simultaneamente. O mesmo não pode ser afirmado com relação ao ruído, que é oriundo de múltiplas fontes e é capaz de afetar a cada microfone individualmente.

Brandstein, Adcock e Silverman (1995) definem os sinais recebidos $x_i(t)$ e $x_k(t)$ referentes ao par de microfones m_i e m_k nas seguintes equações:

$$\begin{aligned} x_i(t) &= s(t) + n_i(t) \\ x_k(t) &= s(t - \tau_{ik}) + n_k(t) \end{aligned} \tag{2.9}$$

onde o parâmetro t indica o instante de tempo, sendo $s(t)$ o sinal em sua forma pura (tal qual foi emitido) conforme se mostra no instante t. As componentes $n_i(t)$ e $n_k(t)$ representam o ruído de fundo capturado pelos microfones.

Cabe destacar que no sinal $x_k(t)$ o atraso de propagação relativo aos microfones τ_{ik} é aplicado somente ao sinal original e não ao ruído. Isto se deve ao fato de que o modelo assume não haver relação entre $s(t)$, $n_i(t)$ e $n_k(t)$. Em outras palavras, os ruídos capturados pelos microfones não possuem relação entre si, ou com o sinal original.

Nesta seção foram apresentados os modelos utilizados para representar matematicamente

um sistema de captura de áudio. Primeiramente foram listadas as premissas e simplificações adotadas para os modelos. Os componentes deste sistema (emissor e microfones) foram representados em um sistema cartesiano, permitindo o cálculo do tempo de propagação do som entre eles. A partir deste tempo foi introduzido o TDOA, que contempla diferenças entre pares de microfones. Por fim, foi também apresentado um modelo simplificado dos sinais capturados pelos microfones. Estes conceitos estabelecem a base para a compreensão dos algoritmos de Localização de Fonte Sonora que serão abordados na seção seguinte.

2.2 Localização de Fonte Sonora

Na Seção 2.1 foi apresentado um modelo acústico simplificado no qual está presente um emissor sonoro e um arranjo de microfones. Também foi dito que a diferença de tempo de chegada do sinal emitido entre os microfones pode ser utilizada para determinar a posição do emissor. Esta diferença de tempo, chamada TDOA, é explorada por diferentes algoritmos de Localização de Fonte Sonora.

Quando a posição do emissor é conhecida pode-se calcular sua distância em relação aos microfones e obter o valor de TDOA entre eles diretamente pela Equação 2.5. Caso nem a posição nem a distância do emissor em relação aos microfones sejam conhecidas, existem técnicas voltadas para a estimativa de TDOA com base nos próprios sinais capturados. Esta seção inicia com a apresentação de algumas destas técnicas, os chamados estimadores de TDOA. Em seguida a ênfase recai sobre o tratamento dos próprios sinais capturados, entrando no tópico das funções de ponderação. Por fim a localização de fonte sonora como um todo toma forma ao serem apresentadas soluções populares para o problema. Entre elas o algoritmo SRP-PHAT escolhido para o desenvolvimento do presente trabalho.

2.2.1 Estimadores de TDOA

Conforme visto na subseção 2.1.2, a partir das distâncias entre o emissor sonoro e os microfones é possível calcular o TDOA de um par de microfones. Ou seja, seu cálculo pode ser feito com base nas posições do emissor e microfones. Não sendo conhecida a posição do emissor têm-se um cenário similar, porém inverso: a partir de valores de TDOA estima-se sua posição. Todavia, normalmente os valores de TDOA também não estão prontamente disponíveis, mas podem ser estimados pela análise dos sinais capturados. A precisão e robustez desta análise são cruciais para a qualidade dos resultados de localização do emissor (BRANDSTEIN, 1995).

Existem diferentes abordagens para a estimativa, como por exemplo o algoritmo *Adaptive Eigenvalue Decomposition* (AED) (BENESTY, 2000). Este algoritmo calcula a matriz de covariância entre sinais de um par de microfones e seus autovetores. A resposta ao impulso entre os sinais da fonte e dos microfones está contida no autovetor de menor autovalor. Com base nas respostas ao impulso, o método prossegue realizando uma estimativa dos tempos de propagação

e, conseqüentemente, dos TDOAs.

As soluções mais populares (BENESTY, 2000), no entanto, são o algoritmo GCC e suas derivações. O próprio SRP, que compõe o SRP-PHAT, foco deste trabalho, tem sua base no GCC, como será mostrado na Subseção 2.2.3.3. Portanto, como parte do estudo das origens do SRP-PHAT, esta seção abordará os estimadores CC, GCC e GCC-PHAT.

2.2.1.1 Correlação Cruzada (CC)

O cômputo da Correlação Cruzada (CC) entre dois sinais de áudio se tornou um método padrão para estimar a diferença de tempo entre eles (HAMON; HANNAN, 1974). Conforme seu nome já indica, a técnica é capaz de quantificar a relação entre sinais. Quanto maior o índice de CC entre dois sinais, maior sua semelhança.

Considerem-se dois sinais contínuos x_i e x_k onde τ representa uma diferença de tempo entre eles, sua Correlação Cruzada é dada por:

$$c_{ik}(\tau) = \int_{-\infty}^{\infty} x_i(t)x_k(t+\tau)dt \tag{2.10}$$

Espera-se que $c_{ik}(\tau)$ atinja seu valor máximo quando τ corresponder ao TDOA entre os dois sinais.Em termos mais práticos, isto significa que os sinais serão mais similares quando estiverem temporalmente alinhados.

A mesma função pode ser expressa no domínio de freqüência por meio da Transformada de Fourier (DIBIASE, 2000; DO, 2010):

$$C_{ik}(\omega) = \int_{-\infty}^{\infty} c_{ik}(\tau)e^{-j\omega\tau}d\tau \tag{2.11}$$

Cabe mencionar que j neste caso representa a unidade imaginária, a constante e o número de Euler e w especifica uma faixa de freqüência em radianos.

O teorema da convolução afirma que a Transformada de Fourier da convolução entre duas funções é equivalente ao produto das transformadas individuais das funções (PRIEMER, 1991). Unindo as Equações 2.10 e 2.11 e aplicando o teorema da convolução C_{ik} pode ser expresso como:

$$C_{ik}(\omega) = X_i(\omega)X_k^*(\omega) \tag{2.12}$$

onde $X_i(\omega)$ é a Transformada de Fourier de $x_i(t)$ e $X_k^*(\omega)$ é o conjugado complexo de $X_k(\omega)$. O operador $*$ neste caso é utilizado para representar o conjugado complexo, tendo em mente que no domínio da freqüência os sinais são representados por números complexos. Para um dado número complexo, diga-se $z = a + ib$, seu conjugado complexo possui a mesma magnitude para as partes real e imaginária, porém a imaginária possui sinal contrário, no caso $z^* = a - ib$.

Por fim, efetuando a inversa da Transformada de Fourier sobre a Equação 2.12 pode-se

escrever a Correlação Cruzada dos sinais em termos de suas transformadas:

$$c_{ik}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X_i(\omega) X_k^*(\omega) e^{j\omega\tau} d\omega \tag{2.13}$$

A CC por si só já pode ser utilizada como uma métrica para localizar fontes sonoras. Porém é desejável antes realizar um pré-processamento dos sinais devido às corrupções que estes naturalmente experimentarão ao longo de suas propagações, o que pode impactar severamente os resultados do método. Tal aspecto conduz com naturalidade à Correlação Cruzada Generalizada, abordada a seguir.

2.2.1.2 Generalized Cross-Correlation (GCC)

Embora a Correlação Cruzada estabeleça uma métrica de similaridade entre sinais, sua vulnerabilidade a ruídos não a torna uma solução robusta para SSL. Uma etapa de filtragem dos sinais capturados para realçá-los ou suprimir ruídos antes de efetuar a CC sobre os mesmos pode aprimorar seus resultados. A adição de filtros foi proposta por Knapp e Carter (1976), dando origem à chamada Correlação Cruzada Generalizada ou *Generalized Cross-Correlation* (GCC).

Considere-se G_i e G_k as Transformadas de Fourier de um par de filtros aplicados sobre os sinais $x_i(t)$ e $x_k(t)$ respectivamente. A GCC dos sinais, representado por $R_{ik}(\tau)$, pode ser formalizada conforme abaixo:

$$R_{ik}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} (G_i(\omega) X_i(\omega))(G_k(\omega) X_k(\omega))^* e^{j\omega\tau} d\omega \tag{2.14}$$

Reorganizando a ordem de seus componentes se obtém:

$$R_{ik}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} G_i(\omega) G_k^*(\omega) X_i(\omega) X_k^*(\omega) e^{j\omega\tau} d\omega \tag{2.15}$$

É possível abstrair os dois filtros em um único termo $\Psi_{ik}(\omega)$, referido como a função de ponderação:

$$\Psi_{ik}(\omega) \equiv G_i(\omega) G_k^*(\omega) \tag{2.16}$$

Aplicando-se a Equação 2.16 à Equação 2.15 a formulação de GCC passa a ser:

$$R_{ik}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \Psi_{ik}(\omega) X_i(\omega) X_k^*(\omega) e^{j\omega\tau} d\omega \tag{2.17}$$

Analogamente ao caso da CC, o valor máximo de R_{ik} corresponde à estimativa de TDOA do método. A procura do intervalo de tempo que maximiza R_{ik} exigirá uma série de testes com diferentes valores de τ. Apesar de ser uma busca unidimensional, a existência de máximos locais a torna computacionalmente custosa (DIBIASE, 2000). A escolha da função de ponderação

pode exercer grande influência sobre os resultados do algoritmo (DO, 2010). Funções de ponderação serão abordadas na Subseção 2.2.2, mas primeiramente será tratado um caso específico de GCC chamado GCC-PHAT.

2.2.1.3 Generalized Cross-Correlation usando PHAse Transform (GCC-PHAT)

De forma objetiva, o GCC-PHAT consiste no algoritmo GCC associado à função de ponderação PHAse Trasnform (PHAT). A função PHAT torna o GCC mais robusto, permitindo melhores estimativas de TDOA entre pares de microfones (ZHANG; FLORENCIO; ZHANG, 2008). Esta combinação alcançou grande popularidade, pois apresentou bons resultados quando submetida a condições acústicas realistas (DIBIASE, 2000; ZHANG; FLORENCIO; ZHANG, 2008).

PHAT será devidamente abordada na Subseção 2.2.2.2, mas para auxiliar na compreensão da formulação de GCC-PHAT, a função de ponderação é apresentada a seguir:

$$\Psi_{ik}(\omega) \equiv \frac{1}{|X_i(\omega)X_k^*(\omega)|} \tag{2.21}$$

sendo $X_k^*(\omega)$ o conjugado complexo de $X_k(\omega)$.

Substituindo Ψ_{ik} da Equação 2.17 por PHAT, de acordo com a Equação 2.21, o valor de R_{ik} pode ser determinado de acordo com a fórmula:

$$R_{ik}(\tau) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{1}{|X_i(\omega)X_k^*(\omega)|} X_i(\omega)X_k^*(\omega)e^{j\omega\tau}d\omega \tag{2.18}$$

Conforme destaca Do (2010), aplicando-se o GCC-PHAT sobre um subconjunto de Q pares de microfones, adquirem-se Q valores estimados de TDOA. A partir de um ponto hipotético $\vec{p}$ qualquer no espaço tridimensional, onde se supõe estar o emissor sonoro, é possível calcular o valor real dos TDOAs para os Q pares de microfones. Comparando os TDOAs estimados $\tau\hat{Q}(\vec{p})$ com os TDOAs reais $\tau Q(\vec{p})$ para a posição candidata do ponto $\vec{p}$ o grau de erro do ponto pode ser calculado através do valor quadrático médio ou RMS (do inglês *root mean square*):

$$E_{RMS}(\vec{p}) = \sqrt{((\tau\hat{Q}(\vec{p}) - \tau Q(\vec{p}))^2)} \tag{2.19}$$

Neste caso em particular, a partir desta métrica de erro, pode-se interpretar o problema de SSL como a procura do $\vec{p}$ que minimize $E_{RMS}(\vec{p})$ (DO, 2010).

A Subseção 2.2.2 acrescentará outros exemplos de funções de ponderação. O foco central, porém, é a função PHAT, que ainda será explicada em maiores detalhes.

2.2.2 Funções de ponderação

Em essência, funções de ponderação são filtros adicionados aos algoritmos de SSL com o propósito de torná-los mais robustos, pois removem características indesejadas dos sinais como por exemplo determinadas faixas de freqüência. Dentre as funções mais conhecidas podemos citar SCOT (CARTER; NUTTALL; CABLE, 1973), PHAT (KNAPP; CARTER, 1976), Eckart (ECKART; SHIPS. CONTRACT NOBSR-43356, 1952) e ML (HAMON; HANNAN, 1974). Uma comparação entre estas e ainda outras funções foi realizada por Knapp e Carter (1976). Destacam-se, por apresentar maior resistência contra ruídos e reverberação, respectivamente, ML e PHAT (DO, 2010). As duas serão brevemente tratadas a seguir, porém apenas PHAT será utilizada na implementação deste trabalho.

2.2.2.1 Máxima Verossimilhança (ML)

A Máxima Verossimilhança (ou ML do inglês *Maximum Likelihood*) é uma análise estatística utilizada para inferir parâmetros com base em valores observados. O raciocínio por trás do método consiste em analisar resultados obtidos para determinar qual valor de um determinado parâmetro que oferece a maior probabilidade de originar os tais resultados. Em outras palavras, ele maximiza a função de probabilidade dos dados conhecidos em função do parâmetro desconhecido (AGRESTI, 2002).

No contexto de SSL pode-se utilizar um estimador de Máxima Verossimilhança para analisar os sinais capturados e estimar os valores de TDOA capazes de gerá-los. Ao implementar um filtro ML, Knapp e Carter (1976) utilizaram sinais capturados (convertidos para o domínio da freqüência) para definir uma matriz com suas densidades espectrais, isto é, a energia observada em diversas faixas de freqüência dos sinais. Sua implementação parte do princípio de que os componentes de ruído presentes nos sinais de áudio não possuem correlação entre si ou com o sinal original (conforme emitido pela fonte). Embora robusto na presença de ruído, seu modelo não leva em consideração a reverberação. Quando presente, a reverberação cria uma correlação entre sinal e ruído, violando a premissa do método.

Um exemplo de função de ponderação ML apresentado por Brandstein e Silverman (1997) pode ser visto na seguinte equação:

$$\Psi_{ik}(\omega) \equiv \frac{|X_i(\omega)||X_k(\omega)|}{|N_k(\omega)|^2|X_i(\omega)|^2 + |N_i(\omega)|^2|X_k(\omega)|^2} \tag{2.20}$$

As componentes de ruído presentes nos sinais $X_i(\omega)$ e $X_k(\omega)$ estão representadas em $N_i(\omega)$ e $N_k(\omega)$ respectivamente.

De uma forma geral funções de ponderação ML oferecem bom desempenho computacional e se mostram eficazes em relação ao ruído, contanto que não ocorra reverberação. Porém, conforme aponta DiBiase (2000), estas condições ideais dificilmente são encontradas na prática.

O autor também afirma que mesmo pequenos índices de reverberação são capazes de degradar severamente os resultados do método.

Zhang, Zhang e Florencio (2007) apresentam um *framework* de ML para SSL com múltiplos microfones direcionais. Em seu trabalho os autores acrescentaram um termo de reverberação ao modelo do sinal, aprimorando a eficácia do método em cenários desfavoráveis. Sua implementação estima o TDOA partindo do sinal original, de sua atenuação e da matriz de covariância dos ruídos. A covariância dos ruídos é calculada durante períodos de silêncio. Um estudo comparativo entre esta implementação de ML e PHAT escrito pelos mesmos autores pode ser encontrado em (ZHANG; FLORENCIO; ZHANG, 2008).

2.2.2.2 PHAse Transform (PHAT)

A função de ponderação PHAT (do inglês *PHAse Transform*) é conhecida por seu bom desempenho em cenários reais (ZHANG; FLORENCIO; ZHANG, 2008). O filtro é particularmente eficaz para reduzir a influência da reverberação em sinais de áudio. Embora em um primeiro momento o caráter heurístico do método possa parecer desconfortável ao leitor, uma série de trabalhos na literatura atestam sua eficácia com base em resultados experimentais (DIBIASE, 2000; ZHANG; ZHANG; FLORENCIO, 2007; RUI; FLORENCIO, 2004; BENESTY, 2000).

A definição matemática da função PHAT, adiantada na Subseção 2.2.1.2, é reapresentada a seguir:

$$\Psi_{ik}(\omega) \equiv \frac{1}{|X_i(\omega)X_k^*(\omega)|} \tag{2.21}$$

relembrando que $X_k^*(\omega)$ representa o conjugado complexo do sinal $X_k(\omega)$. Como pode ser observado, cada faixa de freqüência é dividida pelas magnitudes de seus componentes. Sua utilização em conjunto com soluções de SSL pode ser observada nas Subseções 2.2.1.3 e 2.2.3.4, para os algoritmos GCC-PHAT e SRP-PHAT respectivamente.

Conforme destacado por Salvati, Canazza e Rodà (2011), o que ocorre na prática é a normalização das amplitudes das densidades espectrais dos sinais. Esta normalização pode ser percebida na equação acima, que efetivamente remove a informação de amplitude. Como as informações de TDOA estão ligadas apenas à fase destes, a perda desta informação não acarreta desvantagens aos algoritmos de SSL. Antes pelo contrário, a medida reduz o impacto de ruídos e reverberação sobre os mesmos.

Em seu estudo comparativo Zhang, Florencio e Zhang (2008) procuram os motivos pelos quais PHAT funciona tão bem na prática. Segundo eles o melhor desempenho de PHAT ocorre em ambientes com pouco ruído, forma independente da quantidade de reverberação presente. Os autores ressaltam ainda que ambientes fechados (escritórios e salas de conferência) normalmente apresentam uma relação sinal-ruído superior a 15 dB, proporcionando um contexto favorável ao método.

2.2.3 Algoritmos de localização

O objetivo desta seção é oferecer uma visão geral dos algoritmos de Localização de Fonte Sonora culminando na apresentação do SRP-PHAT, a técnica adotada para a implementação deste trabalho. Primeiramente serão mostrados alguns algoritmos baseados em hiperbolóides. Estes serão brevemente apresentados, pois possuem uma abordagem distinta àquela adotada pelo SRP-PHAT. Todavia convém mencionar a existência de tais alternativas citando alguns exemplos. Em seguida serão tratados os algoritmos de *beamforming*, cuja aplicação se estende além do contexto de SSL. Sua compreensão é importante para introduzir SRP, abordado logo em seguida, que constitui a base do SRP-PHAT. Finalmente o próprio SRP-PHAT será apresentado.

2.2.3.1 Algoritmos baseados em hipérboles

Existem estratégias de localização que exploram a informação geométrica implícita nos valores de TDOA (BRANDSTEIN; ADCOCK; SILVERMAN, 1997; SMITH; ABEL, 1987). O algoritmo chamado de Intersecção Linear (do inglês *Linear Intersection* ou LI), concebido por Brandstein, Adcock e Silverman (1997), considera o espaço descrito pelo TDOA. Conforme ilustrado na Subseção 2.1.2, um único valor de TDOA descreve um hiperbolóide de posições no espaço tridimensional que poderiam ser ocupadas pelo emissor sonoro. Os autores abstraem o hiperbolóide em dois cones que compartilham um mesmo eixo de simetria. Com base neste eixo e no valor de TDOA entre o par de microfones é estimado o ângulo entre o eixo e o vetor de direção que descreve o cone. Acrescentando mais pares de microfones os autores efetuam a localização de fonte sonora considerando as intersecções entre vetores de direção originados de diferentes pares de microfones.

Outra estratégia, proposta por Smith e Abel (1987), chama-se Interpolação Esférica (do inglês *Spherical Interpolation* ou SI). Em lugar de calcular a posição da fonte sonora a técnica se contenta em oferecer uma aproximação da mesma, alcançando maior eficiência em função desta escolha. De uma perspectiva geométrica, o algoritmo define uma esfera cuja superfície interseccione a posição do microfone referencial e cuja distância perpendicular dos demais microfones seja igual à distância descrita por seus TDOAs em relação ao referencial. O que se observa é a aproximação de uma esfera cujo centro coincida com a posição do emissor.

De uma forma geral métodos que realizam aproximações, como no caso da Interpolação Esférica, são valorizados por seu ganho em desempenho. Todavia existem métodos híbridos onde a solução hiperbólica fornece um ponto de partida a outro algoritmo, mais preciso e custoso, responsável pela procura do emissor, agora, em uma área de busca reduzida (BRANDSTEIN; ADCOCK; SILVERMAN, 1997). Como o SRP-PHAT não tem suas origens em soluções baseadas em hiperbolóides, não serão fornecidos aqui maiores detalhes sobre estas estratégias. A subseção a seguir apresentará outro método chamado *beamforming*, que estabelecerá a base do SRP.

2.2.3.2 Beamforming

Algoritmos de *beamforming* são caracterizados pela capacidade de focar em uma determinada posição, reforçando sons originados nesta e suprimindo os demais. Em contextos de videoconferência, por exemplo, quando conhecida a posição do locutor, pode-se realçar sua voz e reduzir sons originados por demais fontes presentes no ambiente. Este processo normalmente envolve a captura dos sinais por arranjos de microfones, o tratamento destes sinais por parte de filtros temporais e a composição de um único sinal sonoro resultante da soma dos demais, razão pela qual é conhecido como *filter-and-sum*. Os filtros em questão são procedimentos responsáveis pela remoção de características indesejadas presentes nos sinais, como por exemplo determinadas faixas de freqüência. Versões sofisticadas do método efetuam maiores aprimoramentos dos sinais por meio de múltiplos filtros antes da etapa de soma. A versão básica, porém, chamada *delay-and-sum*, se limita a sincronizar temporalmente e somar os sinais (DIBIASE, 2000).

Considerando novamente um arranjo formado por M microfones, o *delay-and-sum* pode ser definido matematicamente como:

$$y(t, \Delta_1 ... \Delta_M) \equiv \sum_{i=1}^{M} x_i(t - \Delta_i) \tag{2.22}$$

onde os sinais $x_i(t)$ podem ser descritos pela Equação 2.9 do sinal. A lista de valores $\Delta_1 ... \Delta_M$ corresponde aos atrasos de cada sinal, sendo responsável por sua sincronia. Acusticamente sincronizar os sinais significa alterar o foco dos microfones para uma dada posição espacial. Para esclarecer o significado de foco neste contexto, tome-se por exemplo um cenário com duas fontes sonoras, A e B, em posições distintas. Suponha-se que os sinais emitidos por ambas alcançaram um mesmo arranjo de microfones durante um mesmo intervalo de tempo, porém com diferentes tempos de chegada para cada microfone. Agora, assuma-se que os sinais capturados pelo arranjo serão tratados por um *delay-and-sum* focado na posição da fonte A. Neste caso o *beamformer* em questão compensará as diferenças de tempo entre sinais capturados de forma que sua soma será construtiva em relação ao sinal emitido por A e destrutiva para aquele emitido em B. O que se percebe na prática é o realçamento de um sinal em detrimento dos demais.

Em um sistema como o do exemplo anterior, cada posição espacial origina seu próprio conjunto de valores de atraso. Semelhantemente os conjuntos de valores de atraso aponta para a sua posição de origem (dado um número suficientemente grande de microfones). Sendo assim, manipular os valores $\Delta_1 ... \Delta_M$ equivale a manipular o posição na qual o arranjo está focado. Para sincronizar sinais em função de seus tempos de chegada aos microfones, pode-se definir Δ_i da seguinte maneira:

$$\Delta_i = \tau_i - \tau_0 \qquad \text{para } i = 1..M \tag{2.23}$$

Do (2010) estabelece τ_0 como um referencial representando o menor tempo de chegada entre os

microfones. Desta forma Δ_i nunca será negativo e o sistema será causal. Em outras palavras, o estado do sistema em um instante t não depende de estados futuros, mas apenas do instante t e de instantes anteriores.

No que tange ao *filter-and-sum*, além da sincronização dos sinais, primeiramente ocorre a aplicação de filtros sobre os mesmos. Sua modelagem no domínio da freqüência é dada por:

$$Y(\omega, \Delta_1...\Delta_M) = \sum_{i=1}^{M} G_i(\omega)X_i(\omega)e^{-j\omega\Delta_i} \tag{2.24}$$

Os sinais $X_1(\omega)...X_M(\omega)$ equivalem às transformadas de Fourier dos sinais capturados. Semelhantemente $G_1(\omega)...G_M(\omega)$ representam as transformadas de Fourier dos filtros temporais.

No exemplo anterior (uma videoconferência) a atenção estava voltada para a voz do locutor, a qual se desejava realçar. Obter um sinal sonoro aprimorado era o objetivo da aplicação do *beamformer*. No contexto da localização de fonte sonora o sinal resultante em si é de menor importância, porém sua informação de energia é de grande valor. Alterando-se a função do *beamformer* para retornar a potência de resposta do sinal, é possível fazer uma varredura de um espaço amostral com o fim de encontrar a posição que resulte na maior potência de resposta. Espera-se que seu pico seja máximo sobre a posição do locutor. Esta potência de resposta será abordada na próxima subseção.

2.2.3.3 Steered Response Power (SRP)

O *Steered Response Power* (SRP) pode ser definido como a potência de resposta da saída de um *filter-and-sum* iterado sobre todos os pontos de uma região predefinida (DO, 2010). A posição sobre a qual o *beamformer* está focada é determinada pelos valores de $\Delta_1...\Delta_M$.

O SRP é escrito no domínio da freqüência em função de $\Delta_1...\Delta_M$ da seguinte forma:

$$P(\Delta_1...\Delta_M) \equiv \int_{-\infty}^{\infty} Y(\omega, \Delta_1...\Delta_M)Y^*(\omega, \Delta_1...\Delta_M)d\omega \tag{2.25}$$

onde $Y(\omega, \Delta_1...\Delta_M)$ é a saída do *filter-and-sum* e $Y^*(\omega, \Delta_1...\Delta_M)$ é seu conjugado complexo. Espera-se que a Equação 2.25 seja máxima quando os valores de $\Delta_1...\Delta_M$ sejam equivalentes aos TDOAs entre os microfones. Isto corresponde a dizer que os microfones estão focados na posição do emissor.

Seguindo o desenvolvimento matemático demonstrado por Do (2010), as Equações 2.24 e 2.25 podem ser combinadas para formar:

$$P(\Delta_1...\Delta_M) \equiv \int_{-\infty}^{\infty} (\sum_{i=1}^{M} G_i(\omega)X_i(\omega)e^{-j\omega\Delta_i})(\sum_{k=1}^{M} G_k^*(\omega)X_k^*(\omega)e^{j\omega\Delta_k})d\omega \tag{2.26}$$

Reorganizando os membros da Equação 2.26 ela pode ser reescrita como:

$$P(\Delta_1 ... \Delta_M) \equiv \int_{-\infty}^{\infty} \sum_{i=1}^{M} \sum_{k=1}^{M} (G_i(\omega) G_k^*(\omega))(X_i(\omega) X_k^*(\omega)) e^{j\omega(\Delta_k - \Delta_i)} d\omega \tag{2.27}$$

A partir da Equação 2.23 pode-se afirmar:

$$\Delta_k - \Delta_i = \tau_k - \tau_i \tag{2.28}$$

Aplicando-se a Equação 2.28 à 2.27:

$$P(\Delta_1 ... \Delta_M) \equiv \int_{-\infty}^{\infty} \sum_{i=1}^{M} \sum_{k=1}^{M} (G_i(\omega) G_k^*(\omega))(X_i(\omega) X_k^*(\omega)) e^{j\omega(\tau_k - \tau_i)} d\omega \tag{2.29}$$

De acordo com Do (2010), a integral converge porque a energia dos sinais e dos filtros é finita. Sendo assim é possível passar os somatórios para fora da integral:

$$P(\Delta_1 ... \Delta_M) \equiv \sum_{i=1}^{M} \sum_{k=1}^{M} \int_{-\infty}^{\infty} (G_i(\omega) G_k^*(\omega))(X_i(\omega) X_k^*(\omega)) e^{j\omega(\tau_k - \tau_i)} d\omega \tag{2.30}$$

Para simplificar, represente-se a função de ponderação da seguinte forma:

$$\Psi_{ik}(\omega) \equiv G_i(\omega) G_k^*(\omega) \tag{2.31}$$

Considere-se também a Equação 2.4 apresentada na Subseção 2.1.2:

$$\tau_k - \tau_i = \tau_{ki}$$

Ao combinar as Equações 2.30, 2.31 e 2.4, SRP pode ser reescrito como:

$$P(\Delta_1 ... \Delta_M) \equiv \sum_{i=1}^{M} \sum_{k=1}^{M} \int_{-\infty}^{\infty} \Psi_{ik}(\omega) X_i(\omega) X_k^*(\omega) e^{j\omega\tau_{ki}} d\omega \tag{2.32}$$

Neste formato é mais fácil visualizar a semelhança entre as expressões do SRP e GCC. Relembrando o GCC conforme apresentado na Equação 2.17:

$$R_{ik}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \Psi_{ik}(\omega) X_i(\omega) X_k^*(\omega) e^{j\omega\tau} d\omega$$

DiBiase (2000) explicita a relação dos dois métodos mostrando que o SRP equivale à soma dos GCCs de todas as combinações possíveis de pares de microfones. Para este fim, ele prossegue substituindo τ pela diferença de atraso entre dois microfones Δ_{ik}, onde $\Delta_{ik} \equiv \Delta_i - \Delta_k$,

de forma que o GCC fique com o seguinte aspecto:

$$R_{ik}(\Delta_{ik}) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \Psi_{ik}(\omega)X_i(\omega)X_k^*(\omega)e^{j\omega\Delta ik}d\omega \tag{2.33}$$

A partir deste ponto a relação entre os dois métodos está evidenciada o suficiente para se expressar o SRP como uma função de GCCs:

$$P(\Delta_1...\Delta_M) = 2\pi\sum_{i=1}^{M}\sum_{k=1}^{M} R_{ik}(\Delta_k - \Delta_i) \tag{2.34}$$

Por fim, embora esteja demonstrado que o SRP é uma soma de GCCs, o autor aponta ainda dois aspectos da Equação 2.34. Primeiramente a presença da constante 2π que multiplica o resultado dos somatórios. E em seguida há o fato de que as combinações entre microfones incluem combinações equivalentes como $R_{ik}(\Delta_i - \Delta_k)$ e $R_{ki}(\Delta_k - \Delta_i)$. Neste caso é como se as combinações recebessem fator de escala 2. Conseqüentemente, a Equação 2.34 expressa SRP como a soma de todas as combinações possíveis de GCC multiplicada por 2π.

2.2.3.4 Steered Response Power usando PHAse Transform(SRP-PHAT)

Proposta por DiBiase (2000), o SRP-PHAT adiciona ao SRP a robustez do filtro PHAT mencionado na Subseção 2.2.2.2. Novamente considere-se o Ψ_{ik} especificado na Equação 2.21, desta vez inserido na Equação 2.32:

$$P(\Delta_1...\Delta_M) \equiv \sum_{i=1}^{M}\sum_{k=1}^{M}\int_{-\infty}^{\infty} \frac{1}{|X_i(\omega)X_k^*(\omega)|}X_i(\omega)X_k^*(\omega)e^{j\omega\tau_{ki}}d\omega \tag{2.35}$$

Uma formulação computacionalmente mais eficiente do método é proposta por Zhang, Zhang e Florencio (2007):

$$P(\Delta_1...\Delta_M) = \int_{-\infty}^{\infty}\left|\sum_{i=1}^{M}\frac{X_i(\omega)e^{j\omega\tau_i}}{|X_i(\omega)|}\right|^2 d\omega \tag{2.36}$$

o que reduz o número de ponderações e somas de M^2 para M.

O desempenho do SRP-PHAT é influenciado diretamente por dois fatores fundamentais: o total de microfones M que compõem o arranjo e o número de posições a serem testadas. O aumento do número de microfones confere maior robustez ao algoritmo, porém demanda o processamento de um número maior de sinais, o pode ser bastante custoso. Quanto às posições candidatas a serem testadas, é prática comum definir a área de busca como um *grid* de pontos uniformemente distribuídos (DIBIASE, 2000; DO, 2010; COBOS; MARTI; LOPEZ, 2011). Similarmente, o aumento do número de pontos nesta área tende a gerar estimativas mais precisas, porém a custo de um número superior de testes. Um exemplo ilustrativo de uma área de busca

pode ser visto na Subseção 4.1.2.1.

Muitos trabalhos foram desenvolvidos com o objetivo de otimizar o algoritmo, inclusive viabilizando-o para execução em tempo real. Do, Silverman e Yu (2007) aplicaram *Stochastic Region Contraction* (SRC), originalmente proposto por Berger e Silverman (1991), ao SRP-PHAT. No lugar de um *grid* o método faz a seleção estocástica de um conjunto de pontos a serem verificados. A área de busca onde tais pontos são selecionados vai sendo reduzida iterativamente, até que seja considerada suficientemente precisa. Do e Silverman (2007) também propuseram uma alternativa semelhante chamada *Coarse-to-Fine Region Contraction* (CFRC). Esta técnica retoma a busca em *grid*, abandonando o caráter estocástico do SRC e mantendo a redução iterativa da área de busca. Tal medida reduz o custo de execução do SRP-PHAT em ambientes com taxa de ruído elevada. Posteriormente, Do e Silverman (2009) introduziram outro método chamado *Stochastic Particle Filtering* (SPF). Neste foi utilizada uma versão modificada de PHAT, chamada β-PHAT, proposta por Donohue, Hannemann e Dietz (2007). O algoritmo trabalha de forma similar ao SRC, porém é capaz de selecionar pontos de maior relevância para verificação. Resultados experimentais alcançados indicam um custo de apenas 0,03% do custo total de uma procura em todo o *grid*.

Além de otimizações quanto à complexidade do algoritmo, outros autores aplicaram recursos de alto desempenho para atacar o problema. Lee e Kalker (2010) dividiram a solução em duas etapas: primeiro o cômputo de valores imutáveis, depois o cômputo em tempo de execução dos demais valores. Também foram agrupadas etapas similares, com a intenção de maximizar o uso de instruções vetoriais providas por uma biblioteca de alto desempenho. Silveira Jr. et al. (2010) paralelizaram o SRP-PHAT, dividindo-o em tarefas paralelas e executando-as em GPU. Em sua abordagem a potência de resposta de cada ponto do *grid* é calculada paralela e independentemente dos demais.

Nesta seção foi exposta a base teórica do algoritmo de SSL SRP-PHAT. Começando pelos estimadores de TDOA, foram tratadas a Correlação Cruzada, sua generalização em GCC e combinação com funções de ponderação. Em seguida foram abordadas as funções de ponderação, enfocando ML e PHAT por seu destaque em ambientes reais. Finalmente as soluções de SSL como um todo foram contempladas, sendo mencionadas as estratégias baseadas em hiperbolóides e *beamforming*. A última estratégia constitui a base para o algoritmo SRP, que possui relação direta com o GCC e estabelece os fundamentos do SRP-PHAT. A seção atinge o seu objetivo apresentando o SRP-PHAT, acrescentando também algumas otimizações propostas por diferentes autores. Em seguida serão tratadas questões pertinentes ao *hardware* responsável pela captura de sinais para execução do algoritmo.

2.3 Kinect

Em 2006 a Nintendo lançou o *videogame* Nintendo Wii, caracterizado pelo seu controle sensível aos movimentos do usuário. A partir desta iniciativa ressurgiu uma forte tendência no

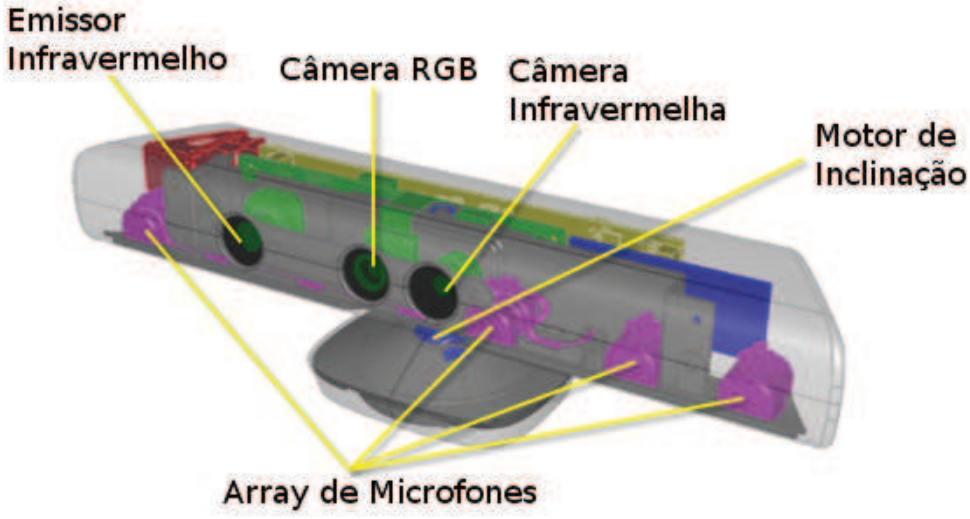
mercado dos consoles domésticos de oferecer aos jogadores meios diferenciados de interação. Para competir com as concorrentes Nintendo e Sony, no final de 2010 a Microsoft lançou o Kinect (Microsoft Corporation, 2010) para seu console Xbox 360. Em fevereiro de 2012 uma versão para computadores intitulada "*Kinect for Windows*" foi lançada. O aparelho pode ser visualizado na Figura 4.

Figura 4: Kinect

Fonte: http://www.microsoft.com/en-us/kinectforwindows/purchase

A proposta inicial do Kinect é permitir interação com o usuário sem o uso de controles, apenas movimentos corporais e comandos de voz. Para este fim o dispositivo conta com uma câmera RGB, um emissor e câmera de profundidade (ambos infravermelhos) e um arranjo composto por quatro microfones supercardióides (TASHEV, 2012) (mais sensíveis a sons frontais), conforme ilustrado na Figura 5. Provavelmente sua característica mais famosa é a capacidade de capturar e processar imagens de profundidade em tempo real, como pode ser observado em seus jogos de dança.

Figura 5: Componentes do Kinect



Fonte: http://msdn.microsoft.com/en-us/library/jj131033.aspx

Voltando a atenção para os recursos sonoros do aparelho, em conformidade com o escopo deste trabalho, já vêm integrados métodos de supressão de eco e reconhecimento de fala. Os microfones são voltados para baixo e se encontram afastados entre si de forma assimétrica. Suas distâncias relativas da esquerda para a direita são de 149 mm, 40 mm e 37 mm (JANA, 2012). Cada microfone suporta uma taxa de amostragem de som de 16 kHz com 32 bits de precisão.

O *pipeline* de processamento de áudio do Kinect inclui uma série de processadores de sinais digitais para a execução de diversas tarefas. Estas incluem cancelamento de eco, *beamforming* (vide Subseção 2.2.3.2), supressão de ruídos e controle de ganho automático (JANA, 2012). É importante ter em consideração que o aparelho foi projetado para uso em ambientes fechados, relativamente pequenos, especificamente para a captura de voz humana. Como a principal função do Kinect é controlar jogos, é esperado um elevado grau de ruído ambiente provindo do próprio jogo exibido em um televisor. Por este motivo faixas de freqüência fora do espectro da voz (entre 80 Hz e 1100 Hz) são suprimidas ao longo do *pipeline*. Outro aspecto importante com relação ao seu *design* diz respeito à distância em relação ao usuário. Embora estas restrições estejam mais atreladas às câmeras do que aos microfones, convém considerar que o aparelho não prevê uma distância maior do que 4 metros por parte do usuário.

Quanto às funcionalidades de *beamforming* e SSL, o Kinect fornece métodos para realçar a voz do locutor e apontar sua direção. O aparelho varre o espaço num ângulo de 100º a sua frente, em incrementos de 10º, conforme documentado em sua API (Microsoft, 2013). Conseqüentemente, seu espaço de busca é discretizado em um total de 11 direções possíveis.

A versão original para Xbox 360 do Kinect é comercializado por $ 99,99. O "*Kinect for Windows*" custa $ 249,99 sendo também vendido por $ 149,99 para estudantes (Microsoft Corporation, 2012). Arranjos profissionais muitas vezes são vendidos em kits para videoconferência, cujo custo pode ser elevado. Cada microfone do *array* de teto da ClearOne (ClearOne, 2013) é comercializado por cerca de $ 480,00. Outro exemplo é o *Polycom CX5000* (Polycom, 2013) da Microsoft, que inclui 6 microfones direcionais e uma câmera de 360º, custando entre $ 3.000,00 e $ 3.600,00. Sendo assim, o Kinect se destaca como uma alternativa econômica quando comparado a soluções profissionais e *hardware* especializado. Além do mais é facilmente encontrado no mercado.

3 TRABALHOS RELACIONADOS

Neste capítulo serão apresentados de forma sucinta trabalhos em certo grau relacionados à proposta aqui apresentada, sendo ao mesmo tempo potenciais áreas de aplicação da mesma. Não foram encontrados trabalhos que empregassem especificamente o Kinect em conjunto com o SRP-PHAT, o que pode ser indício de uma lacuna a ser preenchida. Todavia existem projetos que o utilizam para apoiar a localização por meio de imagens ou mesmo que dependam exclusivamente de áudio, porém por meio de outros algoritmos de SSL. Embora seus objetivos não coincidam com os aqui apresentados, o emprego do Kinect e a ênfase em suas capacidades sonoras demonstram certa similaridade.

Também são brevemente tratados no final do capítulo alguns trabalhos voltados para o SRP-PHAT onde são apresentadas medidas de precisão da técnica, ainda que não possuam nenhuma ligação com o Kinect. O motivo de sua inclusão é a apresentação de seus resultados para posterior comparação com aqueles apresentados no presente documento.

Na primeira seção foram agrupados projetos que adotam o Kinect como parte de soluções de análise multimodal. Como nestas soluções muita ênfase recai sobre as câmeras, não são requeridas estratégias tão complexas de SSL. A segunda seção foi reservada para estudos mais alinhados com a proposta deste trabalho, onde o Kinect é aplicado especificamente como solução de SSL. Já a terceira seção apresenta um apanhado de testes de precisão a respeito do SRP-PHAT conforme encontrados na literatura.

3.1 Kinect e análise multimodal

Grande parte dos projetos que focam nas capacidades auditivas do Kinect as utilizam como uma faceta para soluções de análise multimodal (ANZALONE et al., 2013; GALATAS; FERDOUS; MAKEDON, 2013). Neste contexto a análise multimodal normalmente consiste em utilizar conjuntamente as informações das câmeras (RGB e infravermelha) e microfones para extrair uma informação de interesse. Graças ao auxílio das câmeras, tarefas como localizar um locutor humano podem utilizar algoritmos mais simples sobre os sinais de áudio e complementar a falta de precisão com algoritmos de análise de imagens (ANZALONE et al., 2013). Em alguns casos, onde a maior parte da complexidade recai sobre as câmeras, basta ao algoritmo de áudio apontar a direção da fonte sonora. Conforme consta na Seção 2.3, a API do Kinect possui um método capaz de determinar a direção (dentro de um ângulo de 100º) da fonte.

No trabalho desenvolvido por Anzalone et al. (2013) a análise multimodal tem por finalidade atribuir reações humanas a um robô. Os autores capacitam o robô a identificar pessoas, de forma que ele olhe sempre na direção de quem estiver falando. A detecção e rastreamento dos usuários são realizados através de detecção facial com base nas imagens capturadas pela câmera RGB. Embora os sinais de áudio sejam utilizados para determinar a direção onde se encontra o locutor, sua localização propriamente dita é obtida por meio de processamento de imagens e do mapa de

profundidade fornecido pelos sensores infravermelhos do dispositivo. O papel dos microfones é meramente indicar qual das pessoas detectadas está falando em um dado momento.

Já em (GALATAS; FERDOUS; MAKEDON, 2013) a análise multimodal e dois Kinects são aplicados à criação de um ambiente inteligente. Tendo usuários de idade avançada em mente, o sistema dos autores identifica humanos em situações de emergência, alertando em caso de acidentes domésticos. Na ausência de informação visual, cabe ao módulo de localização sonora determinar a presença e posição de pessoas no ambiente. Usando o método provido pela API do Kinect, cada aparelho, posicionado estrategicamente, origina uma reta que cruza seu centro e a posição da fonte sonora. A intersecção das duas retas demarca o local da fonte sonora no espaço bidimensional.

Embora ambos os trabalhos mencionados sejam fundamentalmente diferentes deste aqui apresentado, o par Kinect e SRP-PHAT poderia ser aplicado aos mesmos. No caso de (ANZALONE et al., 2013), a localização bidimensional poderia substituir a análise do mapa de profundidade, porém ainda seria necessária a detecção facial para determinar a altura do locutor. Todavia a inclusão de um segundo Kinect, possibilitando a localização tridimensional, poderia substituir inteiramente o processamento de imagens, capacitando o robô a olhar na direção do usuário com base unicamente no som.

Já no caso de (GALATAS; FERDOUS; MAKEDON, 2013) o uso do método aqui proposto poderia significar a utilização de um único Kinect. Os microfones são utilizados para reconhecer pedidos de socorro, sons de volume muito elevado e determinar a posição em espaço bidimensional da pessoa necessitada. As duas primeiras funções podem ser realizadas naturalmente por um Kinect só. O segundo Kinect foi adicionado especificamente para auxiliar na localização, o que não seria necessário se fosse utilizado o SRP-PHAT para a mesma finalidade.

3.2 Kinect, SSL e identificação

O trabalho desenvolvido por Tómasson (2012) utiliza o Kinect para identificar e localizar múltiplos locutores humanos simultaneamente. Semelhante a (ANZALONE et al., 2013), seu propósito também é o de aprimorar as interações entre robôs e humanos. Diferente dos trabalhos mencionados anteriormente, sua análise não é multimodal, ou seja, a localização recai exclusivamente sobre a análise dos sinais de áudio.

O processo de identificação considerou diferentes estratégias de extração de características vocais. Foram criados bancos de dados, treinados com amostras vocais de múltiplos usuários. Para a etapa de localização foi implementado o GCC-PHAT (apresentado na Subseção 2.2.1.3), porém apenas para estimar o ângulo de origem do som e não sua posição exata. O autor apresenta ainda os resultados de sua implementação quando submetida a cenários de teste em tempo real.

O presente trabalho compartilha semelhanças com o desenvolvido por Tómasson (2012) no que diz respeito ao uso do Kinect. Em ambos os casos foram implementados algoritmos

tradicionais de SSL para atuar sobre os microfones do aparelho. Porém, enquanto Tómasson procura identificar e determinar a direção de origem de múltiplas fontes sonoras, este trabalho se concentra em uma única fonte e na estimativa de sua posição no espaço. O autor poderia ter utilizado o SRP-PHAT em lugar do GCC-PHAT, porém o problema atacado não exige posicionamento. Já este documento procura avaliar a técnica de forma independente de seu contexto de aplicação.

Um último aspecto de interesse relacionando os dois trabalhos é a representação do espaço tridimensional. Tómasson manifesta a intenção de incluir a terceira dimensão em seus trabalhos futuros. Faz parte do escopo deste documento contemplar também a avaliação da solução dentro do espaço das três dimensões.

3.3 Avaliações do SRP-PHAT

Nesta seção são apresentados resultados oferecidos por diversos autores com respeito à precisão do SRP-PHAT. Os trabalhos em questão não têm relação com o Kinect, sendo normalmente utilizados arranjos de microfones profissionais. No entanto, seja para propôr otimizações ao algoritmo ou usá-lo como base de comparação para outros métodos de SSL, os testes apresentados oferecem alguma medida das taxas de acerto do SRP-PHAT. Com a finalidade de comparar os resultados destes autores com os obtidos no presente trabalho, uma visão geral de suas configurações, procedimentos e resultados são aqui apresentados.

Um ambiente virtual de 7 x 6 x 2,50 metros foi simulado por Zhang, Florencio e Zhang (2008), utilizando gravações de fontes sonoras reais. O sinal de interesse contém voz humana, enquanto os ruídos são formados por um ventilador e computador em funcionamento. O arranjo é composto por 6 microfones, dispostos em um círculo próximo ao centro do ambiente. O emissor sonoro foi posicionado a 1,50 metros de distância do arranjo, sendo testado em 10 posições uniformemente distribuídas em torno do mesmo. Em cada posição foi realizado um teste de 30 segundos, sendo gerados 100 quadros de fala por posição. A janela de tempo usada pelo algoritmo foi de 40 ms, com 20 ms de sobreposição. Dois tempos de reverberação foram simulados: 100 ms e 500ms. Ao todo são analisadas as estimativas de direção de 1000 quadros.

Observa-se nos resultados dos autores uma sensível melhoria na taxa de acertos do algoritmo com o aumento da relação sinal-ruído. Simulando 100 ms de reverberação e uma relação sinal-ruído de 5 dB, 76,1% das estimativas exibiram erro inferior a 2º e 82,0% inferior a 10º. Aumentando a relação para 15 dB as taxas de acerto subiram para 89% e 91,4% respectivamente. Por fim, uma relação de 25 dB elevou as mesmas taxas para 97,6% 98,9%.

A reverberação simulada também exibiu grande impacto sobre a precisão do algoritmo. Utilizando um tempo de 500 ms para a reverberação do sinal, mesmo com uma relação sinal-ruído de 25 dB os quadros que apresentaram erro inferior a 2º foram reduzidos para 60,1% do total e para 79,2% do mesmo os quadros com erro inferior a 10º.

Os mesmos autores em (ZHANG; ZHANG; FLORENCIO, 2007) realizaram testes em ce-

nário real, utilizando um dispositivo RoundTable dotado de 6 microfones e criado pela Microsoft. Foram utilizadas 99 gravações de conferências com 4 minutos de duração. As gravações foram realizadas em 50 ambientes diferentes, com taxas de sinal-ruído variando entre 5 dB e 25 dB. Os autores verificaram o erro de direção das estimativas com base nas imagens retornadas pela câmera panorâmica do RoundTable. Segundo os mesmos, erros de até 14º são aceitáveis para indicar ao dispositivo a direção do locutor. Ao todo foram analisados 6706 quadros dos quais 81,73% indicaram erro abaixo de 6º e 88,13% abaixo de 14º.

Os testes apresentados em (DO; SILVERMAN, 2007) foram realizados em um laboratório especializado da Universidade de Brown, capaz de suportar um arranjo de 512 microfones. Suas medidas são 8 x 8 x 3 metros, com tempo de reverberação de 450 ms. Para os testes dos autores, um conjunto de 24 microfones foi adotado, estando estes distribuídos nas paredes de um dos cantos do laboratório. Os autores definiram um *grid* de $2,4x10^7$ pontos e 1 cm^3 de resolução. A janela de tempo utilizada pelo algoritmo foi de 102,4 ms, avançando 25,6 ms por iteração. Novamente a voz humana representou o sinal de interesse, sendo registrada em períodos de 4 segundos a partir de 4 posições diferentes. Quadros com predominância de ruído foram descartados, totalizando cerca de 40% do total amostrado. As posições escolhidas para o emissor distam 2,14 m, 2,76 m, 3,47 m e 4,25 m dos microfones. Suas taxas de sinal-ruído, na mesma ordem, são: 7,9 dB, 5,7 dB, 3,1 dB e 1,9 dB. A taxa de acerto das posições estimadas, em ordem crescente de distância dos microfones, foi de 100%, 96,6%, 87,8% e 67,3%.

Em seus testes de localização 3D DiBiase (2000) utilizou o mesmo laboratório que Do e Silverman (2007), porém colocou em uso 128 de seus microfones. Os pares de microfones analisados pelo algoritmo foram apenas os pares adjacentes, de forma que um total de 127 pares foram processados. A busca foi realizada em um volume de 3 x 3 x 1 metros cúbicos, discretizado em um *grid* com resolução de 10 cm^3. A janela de tempo considerada pelo algoritmo foi de 25 ms. O sinal emitido foi uma gravação de voz humana com duração de 3 segundos, sendo que utilizados apenas 160 quadros desta gravação. O maior índice de sinal-ruído presente no sinal emitido foi de 24 dB, sendo os quadros com valor inferior a 8 dB excluídos do experimento. A métrica de precisão adotada foi a distância euclidiana entre a posição estimada e real do emissor, conforme também foi realizado neste trabalho e pode ser visto na Seção 4.2. Todas as estimativas realizadas exibiram erro inferior a 10 cm de distância.

4 COMBINAÇÃO SRP-PHAT E KINECT

Em conformidade com os objetivos descritos no Capítulo 1, este trabalho se propõe a avaliar a qualidade, em termos de solução de Localização de Fonte Sonora, do algoritmo SRP-PHAT em ação conjunta com o Kinect em lugar de arranjos de microfones convencionais. Para realizar esta avaliação foi necessário implementar um protótipo capaz de se comunicar com o Kinect, ler as informações de áudio obtidas por seus microfones, executar o SRP-PHAT sobre as mesmas e exibir as posições estimadas para o usuário. Após implementado o *software* se iniciou a fase de testes, com o propósito de quantificar a qualidade das estimativas. Este capítulo se dedica a descrever a metodologia e as particularidades envolvidas tanto na implementação quanto na avaliação da solução.

4.1 Implementação

Como já mencionado, o programa desenvolvido possui duas funcionalidades principais: a comunicação com os microfones do Kinect e a localização de fonte sonora por meio do SRP-PHAT. Além destas capacidades fundamentais o *software* foi expandido com implementações paralelas do algoritmo de localização e o suporte a um segundo Kinect para realizar buscas em espaço tridimensional. A linguagem de programação escolhida foi C++, sendo o gerenciamento de janelas e componentes gráficos realizado através da biblioteca Qt (Digia, 2013). Também foi utilizado o Kinect SDK fornecido pela Microsoft (Microsoft Corporation, 2010), viabilizando o fácil acesso aos microfones do Kinect, e a biblioteca FFTW (MATTEO FRIGO, 1997), que disponibiliza implementações eficientes da Transformada rápida de Fourier. Ainda foram utilizadas Pthreads (IEEE Standards Association, 2008) e OpenCL (Khronos Group, 2013) para a paralelização do SRP-PHAT. Nas subseções a seguir são apresentadas peculiaridades pertinentes a certas etapas da implementação.

4.1.1 Comunicação com o Kinect

O Kinect se comunica com o computador por um cabo USB. O programador tem acesso facilitado aos seus recursos de *hardware* por meio do *Kinect for Windows SDK*, disponibilizado gratuitamente pela Microsoft. O kit de desenvolvimento requer os sistemas operacionais *Windows 7* ou *Windows 8*, juntamente com a instalação do ambiente de desenvolvimento *Visual Studio 2010* ou superior. Entre as facilidades introduzidas pelo *Kinect for Windows SDK* está a *Natural User Interface* (NUI), uma interface de programação que permite o acesso de alto nível às funcionalidades do dispositivo. No entanto, como será visto a seguir, a leitura de sinais de áudio em estado bruto requer a utilização de outras interfaces como a Windows Multimedia Device (MMDevice).

Para encapsular a utilização das bibliotecas do *Windows* e abstrair a comunicação com o

Kinect foi criada a classe KinectAudioCapture. Dentre os atributos privados da classe cabe mencionar:

```
1  INuiSensor* nuiSensor_;
2  IMMDevice* device_;
3
4  IAudioClient* audioClient_;
5  IAudioCaptureClient* captureClient_;
6  WAVEFORMATEX* mixFormat_;
```

O atributo nuiSensor_ permite a contagem do número de Kinects conectados ao computador por meio da função NuiGetSensorCount(). Uma vez verificado o índice do dispositivo que se deseja utilizar, pode-se instanciá-lo invocando a função NuiCreateSensorByIndex().

Os microfones são identificados genericamente pelo sistema como um dispositivo de captura de áudio. A classe MMDeviceEnumerator, por meio do método EnumAudioEndpoints(), oferece uma lista dos dispositivos de captura de áudio disponíveis. Encontrados os microfones, seu endereço em memória passa a ser armazenado no atributo device_. Sua inicialização se dá pela chamada device_->Activate(), onde é instanciado um objeto IAudioClient, apontado por audioClient_.

No aplicativo desenvolvido, a classe IAudioClient é utilizada com dois propósitos: fornecer informações a respeito do formato de áudio adotado e instanciar um objeto IAudioCaptureClient. As informações de áudio são retornadas pelo método GetMixFormat() e apontadas por mixFormat_. Já o objeto IAudioCaptureClient, retornado pela chamada audioClient_->GetService() e apontado por captureClient_, possui o papel mais importante: prover acesso ao *buffer* dos microfones. Seu método GetBuffer() retorna um ponteiro para a região de memória onde os sinais capturados são armazenados e o volume de dados já disponíveis.

Todos os atributos até então apresentados são obtidos durante a criação e inicialização das instâncias da classe KinectAudioCapture. Sua organização interna será explicada em conjunto com seus principais métodos públicos:

```
1  bool start();
2  bool stop();
3  void readBuffer(unsigned char** buff, int size);
```

Os métodos start() e stop() refletem a natureza do dispositivo encapsulado, cujo início e término do ciclo de captura são caracterizados pela invocação dos métodos Start() e Stop() de IAudioClient. O método start() é responsável pela criação de uma *thread* local de captura de áudio e por acionar o dispositivo chamando audioClient_->Start(). Analogamente stop() requisita o fim da execução da *thread* e invoca audioClient_->Stop().

Enquanto os microfones trabalham de forma assíncrona, provendo acesso não bloqueante ao seu *buffer*, KinectAudioCapture oferece leitura síncrona e bloqueante, provendo uma interface mais amigável para o programador. A classe mantém uma cópia local do *buffer* dos microfones, sendo atualizada na medida em que os dados são disponibilizados pelo dispositivo. Esta atualização constante é efetuada pela *thread* de captura, que desempenha o papel de produtor

em uma relação produtor-consumidor. A função de consumidor é desempenhada pelo método readBuffer(), que realiza uma espera bloqueante até que todos os dados requisitados estejam disponíveis no *buffer* local.

Antes de detalhar o procedimento de readBuffer() e da *thread* de captura é necessário introduzir ainda dois atributos da classe:

```
1  CRITICAL_SECTION buffLock_;
2  CONDITION_VARIABLE buffNewContentCondition_;
```

O primeiro mecanismo de sincronização, buffLock_, delimita seções críticas, isto é, trechos de código que não devem ser executados em paralelo. Seu objetivo é impedir que múltiplos fluxos de execução realizem operações de leitura e escrita sobre o *buffer* local simultaneamente. Caso o acesso ao *buffer* não fosse tratado com tal exclusividade não haveria como garantir a consistência de seus dados. Já buffNewContentCondition_ constitui uma variável de condição, acionada pela *thread* cada vez que novos dados são copiados para o *buffer* local. Sua função é permitir a hibernação de fluxos de execução até que uma dada condição seja preenchida. A utilidade desta abordagem ficará claro a seguir, no detalhamento do método readBuffer().

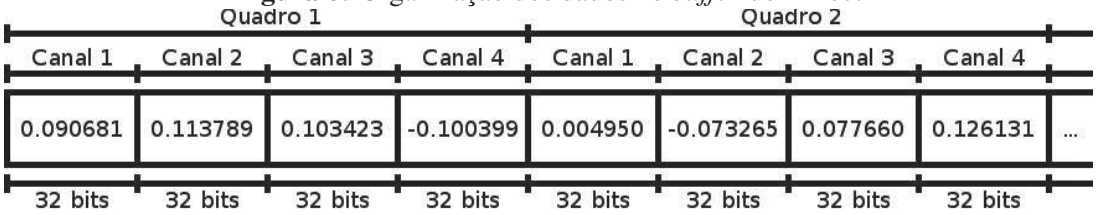
Ao chamar readBuffer() o programador especifica em size o quanto dos sinais deseja ler e uma área de memória buff onde serão disponibilizados. O método declara uma seção crítica protegida por buffLock_, copia os dados do *buffer* local para buff até ler a quota requerida e libera buffLock_, abandonando a seção crítica. Caso o *buffer* não possua dados o suficiente o método libera buffLock_ e entra em hibernação até que a variável de condição buffNewContentCondition_ seja acionada. Quando despertado readBuffer() automaticamente recebe acesso exclusivo à seção crítica e prossegue com a cópia dos dados. Esta abordagem visa evitar a espera ocupada, ou *busy waiting*, onde o processo bloqueado continua a ocupar recursos de processamento durante períodos de espera.

O dever de despertar o fluxo de readBuffer() e atuar como o produtor na relação produtor-consumidor de KinectAudioCapture recai sobre sua *thread* de captura. Durante sua execução, a *thread* invocará periodicamente captureClient_->GetBuffer() para verificar a disponibilidade de novos dados por parte do dispositivo. Em caso positivo é declarada uma seção crítica, novamente protegida por buffLock_, onde será feita a cópia dos sinais do *buffer* dos microfones para o *buffer* local. Ao fim da cópia, já fora da seção crítica, é acionado buffNewContentCondition_, indicando a disponibilidade de novos dados e despertando o fluxo de readBuffer() caso esteja em hibernação.

Com respeito à representação discreta dos sinais transferidos de um *buffer* para outro, seus valores variam entre -1,0 e 1,0, sendo armazenados em conjuntos de 32 bits. Sua escrita se dá de forma intercalada por microfone. A Figura 6 exemplifica o conteúdo de um *buffer* retornado pelo Kinect. A primeira amostra capturada pelo primeiro microfone está escrita nos 32 bits iniciais, identificado como canal 1. Os valores seguintes seguem a mesma organização, sempre distando entre si 32 bits, de forma que o segundo, terceiro e quarto valores correspondem às primeiras amostras do segundo, terceiro e quarto microfones (canais 2, 3 e 4) respectiva-

mente. Estes quatro valores contendo a primeira amostra de cada microfone correspondem a uma mesma janela de tempo, sendo agrupados e identificados como quadro 1. Da mesma forma os quadros seguintes são formados pelos valores de seus respectivos espaços de tempo, ocupando sempre um espaço de 128 bits.

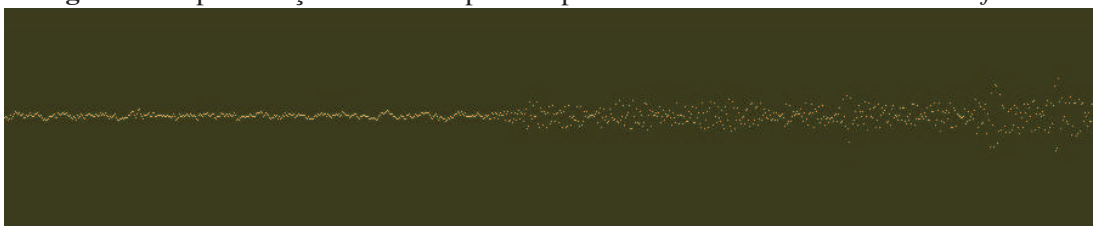
Figura 6: Organização dos dados no *buffer* do Kinect



Fonte: Elaborada pelo autor

A interface gráfica do aplicativo foi desenvolvida com o auxílio da biblioteca Qt (Digia, 2013). Na medida em que a *thread* de captura atualiza o *buffer* local, o programa separa os 4 canais de áudio e os desenha em sua janela. A Figura 7 retrata o sinal conforme obtido por um dos microfones e exibido pelo aplicativo.

Figura 7: Representação do sinal capturado por um microfone na interface do *software*



Fonte: Elaborada pelo autor

Uma variável que possui impacto significativo na fase de captura é o tamanho do *buffer* local. Conforme afirmado na Seção 2.3, ao tratar das especificações do Kinect, o aparelho trabalha com uma taxa de amostragem de 16 kHz. Em outras palavras são obtidos 16000 valores de 32 bits por segundo para cada microfone, totalizando 64000 valores representados em 256000 bits a cada segundo. O tamanho do *buffer* é proporcional ao intervalo de tempo analisado em uma iteração do SPR-PHAT. Este *buffer* será processado pelo algoritmo, sendo o seu resultado impactado pelo volume de dados enviados ao mesmo. Se o intervalo de tempo contemplado for muito curto os resultados provavelmente não serão confiáveis. Ao considerar a questão em relação ao GCC, DiBiase (2000) constatou que intervalos inferiores a 25 milissegundos eram insuficientes para gerar bons resultados. Por outro lado, amostras muito longas, além de exigir maior poder de processamento, podem não ser adequadas a cenários onde a fonte sonora (ou dispositivo de captura) possa se mover. É o caso, por exemplo, quando se procura atribuir uma posição única em um intervalo de 3 segundos a uma pessoa a caminhar. Neste trabalho foi adotado o tamanho empírico de 65536 bits, ou seja, um intervalo de 0,256 segundo, aproximadamente dez vezes maior do que o mínimo apontado por DiBiase (2000).

Além do acesso direto aos microfones, também foi implementada a alternativa de escrita e leitura dos sinais em arquivos WAVE. Sendo assim, é possível armazená-los em disco e reutilizá-los como entrada para o programa, de forma que ele possa executar sobre as gravações em lugar do Kinect. Embora este documento não detalhe sua implementação, tal funcionalidade foi fundamental para a realização dos experimentos da Seção 4.2, pois garantiu sua reprodutibilidade e determinismo.

4.1.2 SRP-PHAT

Adquiridos os dados dos microfones, cabe definir a área de busca e encaminhá-los ao SRP-PHAT. Como o funcionamento do algoritmo em si foi abordado na Seção 2.2, esta subseção tratará dos aspectos menos enfocados como a definição da área de busca e suas implementações paralelas. A adaptação da técnica para a linguagem de programação se encontra pormenorizada no trabalho de Minotto (2010), de modo que o processo não será reproduzido aqui novamente. Cabe destacar que tanto neste quanto naquele trabalho foi utilizada a otimização introduzida por Zhang, Zhang e Florencio (2007), apresentada na Equação 2.36. Por questão de eficiência e praticidade, sinais de áudio são convertidos para o domínio da freqüência. Esta conversão se dá por meio da biblioteca FFTW (MATTEO FRIGO, 1997).

Conforme afirmado na Subseção 2.2.3.4, o desempenho e a precisão do algoritmo são diretamente influenciados pelo número de microfones e tamanho do *grid*. Ao se trabalhar com um único Kinect, o total de microfones (previamente representado pelo símbolo M) passa a ser 4. Permanece ainda variável o número de posições que formam o *grid* da área de busca. Por questão de simplicidade de implementação o programa realiza uma busca exaustiva do *grid*.

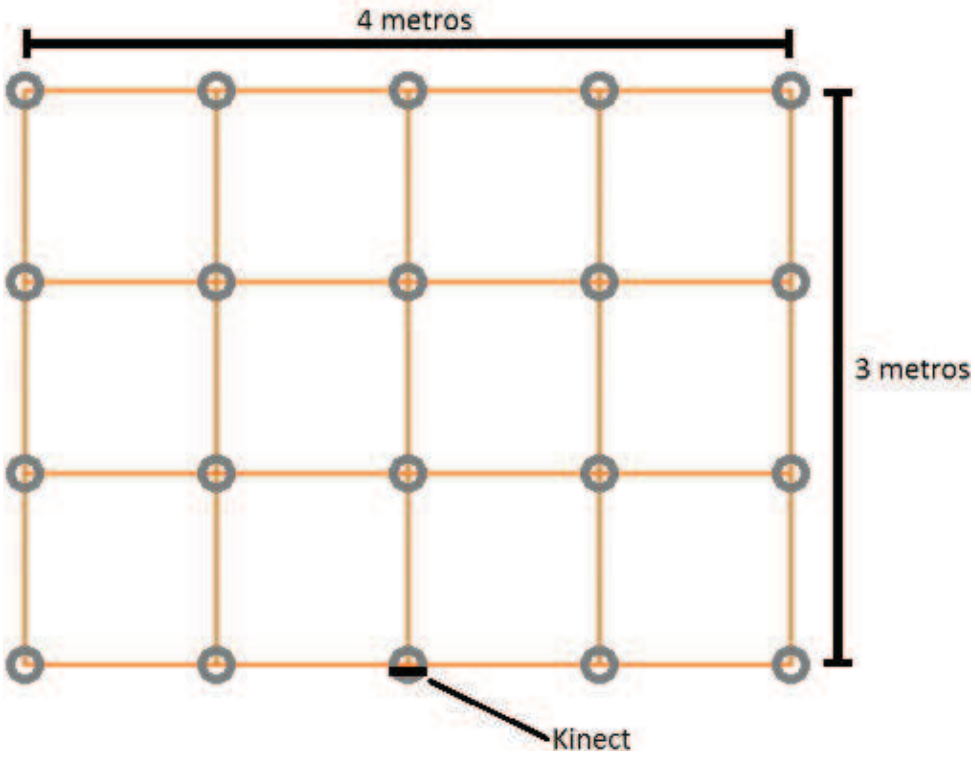
4.1.2.1 Área de busca

Para que o algoritmo possa efetuar uma busca espacial, primeiramente é necessário representar o espaço e sua discretização. Considerando o espaço bidimensional, pode-se descrevê-lo com os seguintes parâmetros: comprimento, largura, número de posições ao longo do comprimento e número de posições ao longo da largura. O comprimento e largura (trabalhados em metros) determinam a área de busca do algoritmo. Posições atrás do Kinect não são localizadas pela solução, então é importante que a área seja definida à frente do mesmo.

A Figura 8 exemplifica uma área de busca com 4 metros de comprimento e 3 metros de largura. Os parâmetros de número de posições ditam a discretização do espaço previamente definido. No exemplo anterior, se especificadas 5 posições ao longo da comprimento e 4 da largura, um *grid* de 20 posições será definido (representadas pelas circunferências da figura). Neste caso o *grid* descrito divide o espaço em áreas de 1 m^2.

A definição de um *grid* tridimensional é análogo ao bidimensional. Especifica-se juntamente com os demais parâmetros a altura e o número de pontos ao longo da mesma. Neste caso o *grid*

Figura 8: Área de busca.



Fonte: Elaborada pelo autor

estará dividindo o espaço em volumes.

4.1.2.2 Processamento paralelo

Como parte de um estudo de Processamento de Alto Desempenho, o programa foi desenvolvido com três implementações do SRP-PHAT: seqüencial, paralela em CPU e paralela em GPU. A primeira alternativa é uma implementação seqüencial comum, onde todas as posições do *grid* são analisadas iterativamente, de forma independente. As execuções paralelas permitem a análise simultânea de múltiplas posições, uma em cada fluxo de execução criado pelo programa.

O paralelismo em CPU é obtido pela criação de *threads* por meio da biblioteca Pthreads (IEEE Standards Association, 2008). Uma *thread* representa um fluxo de execução independente dentro do programa. É permitido ao usuário especificar o número de *threads* desejadas, de forma que ele tenha certo controle seu grau de paralelismo. Cada ponto no *grid* origina uma tarefa, estas, por sua vez, são mantidas em um *pool* de tarefas. Cabe às *threads* requisitar e executar tarefas até que o *pool* esteja vazio.

A implementação paralela em GPU foi escrita em OpenCL (Khronos Group, 2013), podendo na realidade ser executada tanto em CPU quanto GPU. A divisão de carga permanece a mesma: uma tarefa por ponto no *grid*. Porém agora as tarefas são mapeadas para os núcleos de processamento da GPU, que possui uma arquitetura massivamente paralela. Como o número

de unidades de processamento disponíveis nas GPUs costuma ser muito mais elevado do que o das CPUs, o nível de paralelismo atingido por esta implementação tende a ser muito maior.

4.2 Validação

Na etapa de validação da solução ela foi submetida a testes em cenário real e onde foi medida a qualidade de suas respostas. Nesta seção são descritos ambiente, procedimento dos testes e métricas escolhidos. Primeiramente, na subseção a seguir, serão descritos os aspectos comuns a todo o processo experimental, como o ambiente e o processo adotado para os testes. Os experimentos em si estão organizados em dois grupos, de acordo com a variável em estudo, cada qual explicado em sua respectiva subseção. O primeiro grupo observa oscilações na qualidade dos resultados sob diferentes fontes sonoras, enquanto o segundo varia apenas a posição da fonte para verificar a mesma questão.

4.2.1 Configuração geral dos testes

O ambiente escolhido para a avaliação da solução foi o laboratório de visualização da universidade, cujas dimensões são 4,77 x 5,90 x 3,44 metros. Uma fotografia do ambiente pode ser vista na Figura 9. Além de constituir um cenário típico de uso do Kinect (local fechado, com dimensões inferiores a 8 metros), o laboratório também é um exemplo adequado de cenário real, onde diversos sons estão presentes, mesmo que não sejam predominantes. Sons externos ao laboratório captados durante o período de experimentação incluem um ar condicionado, grilos, cigarras e movimentação humana em geral. Além do equipamento para a execução dos testes, as únicas fontes sonoras não de interesse dentro do próprio ambiente são os demais computadores funcionando no local.

Para avaliar a qualidade do conjunto Kinect e SRP-PHAT como solução de SSL, optou-se por realizar uma série de testes de precisão. Tais testes consistem em situar microfones e emissor sonoro em posições conhecidas, possibilitando a comparação de suas coordenadas com aquelas retornadas pelo algoritmo (*ground truth*). Foi adotado um sistema de coordenadas cujos eixos são alinhados às paredes do local e onde uma unidade corresponde a 1 metro, facilitando a representação espacial do laboratório. Sendo assim, em uma tupla (x, y, z) os eixos X, Y e Z denotam comprimento, largura e altura respectivamente. A Figura 11 representa o recinto visto de cima, com suas medidas de comprimento e largura. A origem do sistema se encontra identificado no canto inferior esquerdo da figura, sendo sua altura equivalente a do piso do laboratório. Tomando-se por exemplo a tupla $(4,05;\ \ 2,45;\ \ 0,88)$, destacada na figura, tal posição dista 4,05 metros da parede esquerda da figura, 2,45 metros da parede inferior e se encontra 88 centímetros acima do chão.

No papel de emissor sonoro foi utilizada uma caixa de som comum para computadores, exibida na Figura 10. Como a caixa de som está longe de ser um emissor pontual, foi esco-

Figura 9: Fotografia do laboratório de visualização.

Fonte: Elaborada pelo autor

lhido um ponto próximo ao alto-falante para representar sua posição, cerca de 14 cm acima de sua base. Já no caso dos microfones, suas posições no interior do Kinect são conhecidas (conforme visto na Seção 2.3), não requerendo estimativas. Por questão de comodidade, tanto em ilustrações como ao longo do texto a posição dos microfones será abstraída para a posição do Kinect, que corresponde à coordenada central do mesmo. Tomando a Figura 11 como exemplo novamente, a coordenada k aponta para o centro de um Kinect alinhado paralelamente ao eixo Y. Sabendo-se que o valor de k é $(0,58,\ \ 2,95,\ \ 0,78)$, deduz-se que os microfones se encontram aproximadamente nas coordenadas $(0,58,\ \ 2,84,\ \ 0,78)$, $(0,58,\ \ 2,99,\ \ 0,78)$, $(0,58,\ \ 3,03,\ \ 0,78)$ e $(0,58,\ \ 3,06,\ \ 0,78)$.

Durante todos os testes sinais capturados pelos Kinects foram gravados e armazenados em arquivos WAVE por meio do *software* implementado. Esta medida garante a reprodutibilidade dos testes, permitindo inclusive o pós-processamento dos arquivos de áudio com diferentes configurações da área de busca. Esta funcionalidade foi de grande importância na realização dos testes, como será visto nas subseções a seguir.

4.2.2 Testes de precisão com diferentes sinais sonoros

O primeiro conjunto de testes verifica o comportamento da solução quando submetida a diferentes sinais sonoros. Seu objetivo é determinar o sinal que origina uma localização mais precisa, para que este seja adotado nos testes subseqüentes. Para esta avaliação foram executa-

Figura 10: Fotografia do emissor sonoro.

Fonte: Elaborada pelo autor

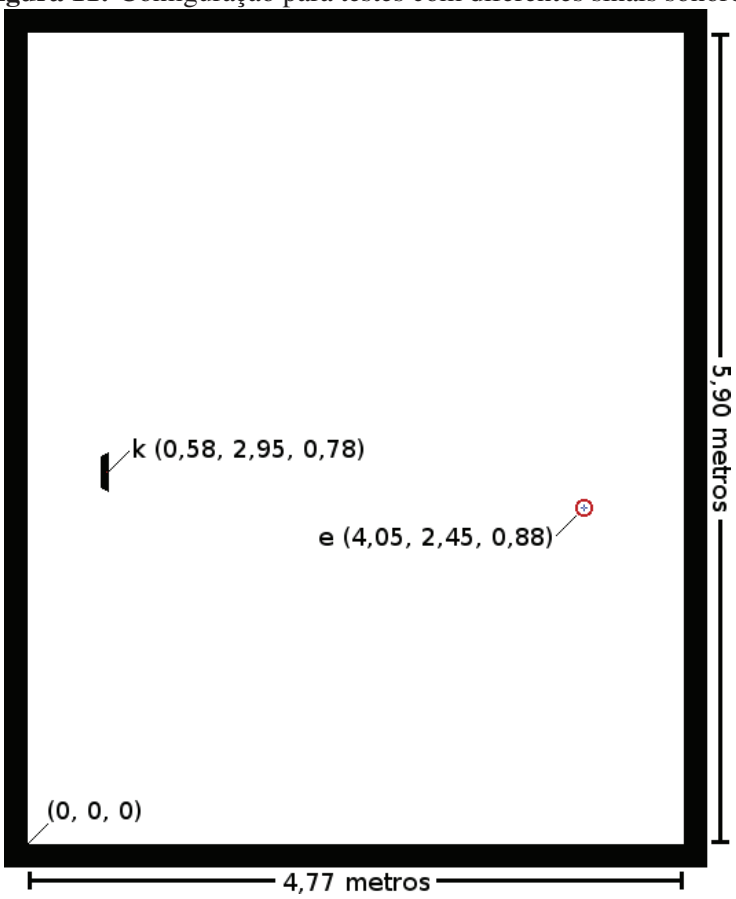
dos testes de precisão em um cenário básico representado na Figura 11. Compõem este cenário um único Kinect, identificado como k, situado na coordenada $(0,58,\ \ 2,95,\ \ 0,78)$ e o emissor e, posicionado logo à sua frente na coordenada $(4,05,\ \ 2,45,\ \ 0,88)$. Nesta série de experimentos foi executada a busca por comprimento e largura, que por praticidade será chamada localização 2D. Os únicos cuidados necessários em relação à posição de e é que este não se encontre atrás de k, sendo que o aparelho se encontra voltado para a direita na ilustração, e, para assegurar a confiabilidade dos resultados, que se evite diferenças muito grandes entre as alturas de k e e quando comparadas à área de busca.

A criação de sinais de teste foi facilitada pela ferramenta Matlab (MathWorks, 2014). Um total de 16 sinais foram fabricados, tendo suas características conhecidas e controladas. As principais características em questão são as freqüências presentes e a largura de seu espectro de freqüências. Por meio destes parâmetros é possível verificar se há uma variação de comportamento entre sinais munidos de freqüências mais altas e mais baixas, ou entre sinais que exibam poucas e muitas freqüências.

No que tange as freqüências presentes nos sinais foram selecionadas três freqüências puras, que servem de base para a maioria dos demais sinais e mantêm-se dentro do limite máximo de 8 kHz determinado pelo Kinect. Elas são 250 Hz, 1000 Hz e 4000 Hz, escolhidas com um fator 4, sendo o valor máximo de 4000 Hz representativo por ser a metade da taxa de amostragem do dispositivo.

Para alargar o espectro de freqüências dos sinais foram gerados novos sinais cuja magnitude de suas componentes de freqüência segue uma distribuição normal com média μ e desvio padrão σ. Os valores utilizados para μ foram 250 Hz, 1000 Hz e 4000 Hz. Já no caso de σ, como o ser humano percebe o intervalo de 1 Hz de forma muito diferente em freqüências altas ou baixas,

Figura 11: Configuração para testes com diferentes sinais sonoros



Fonte: Elaborada pelo autor

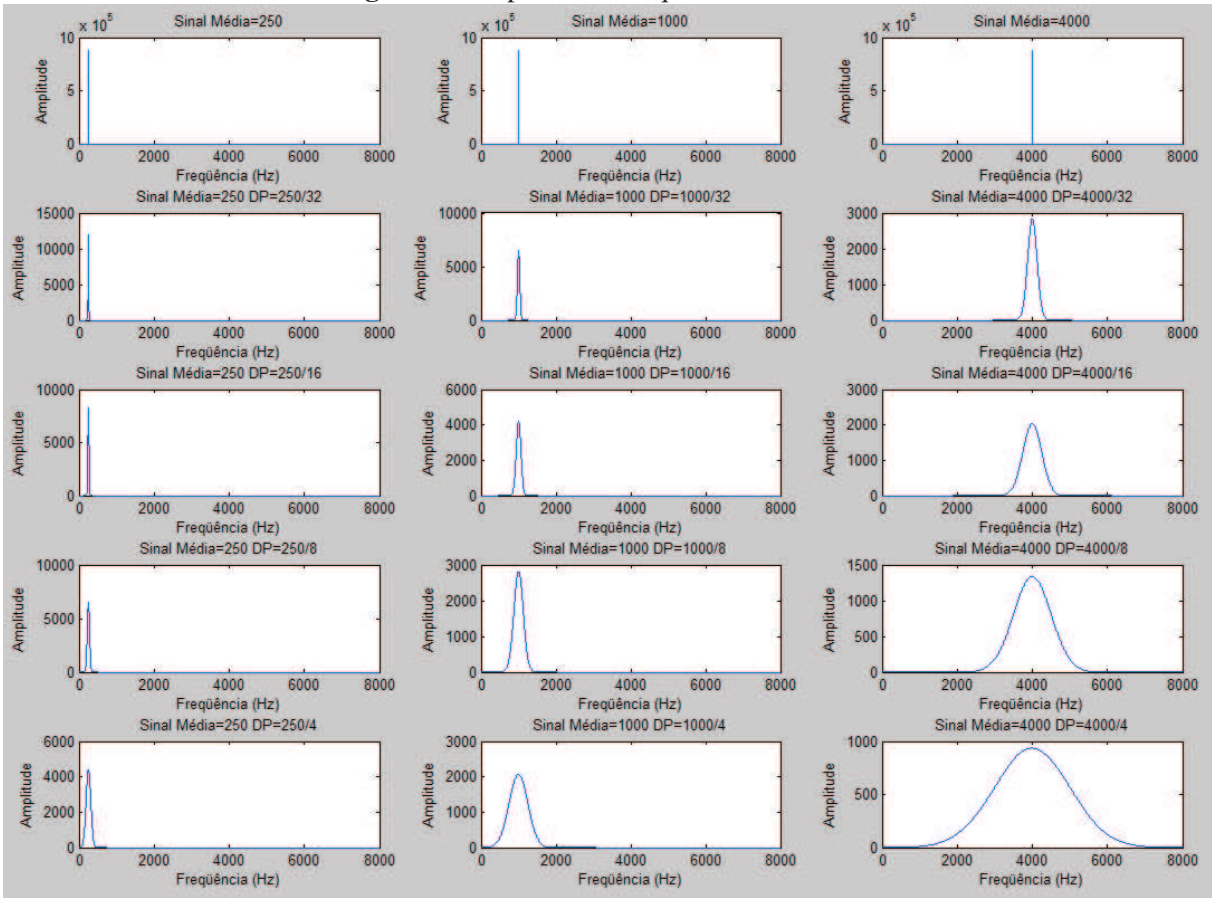
é interessante utilizar valores relativos à média. Por esta razão, para cada novo sinal os valores de σ equivalem a $1/32$, $1/16$, $1/8$ e $1/4$ do valor de seu μ. Tome-se por exemplo os sinais derivados da freqüência de 1000 Hz, σ assumiria os valores $31,25$ Hz, $62,5$ Hz, 125 Hz e 250 Hz. A fase dos sinais é sorteada a partir de uma distribuição uniforme com valores dentro do intervalo aberto de 0 a 2π.

A Figura 12 exibe os espectros de freqüências dos 15 sinais descritos. Da primeira à terceira coluna são exibidos os sinais centralizados nas freqüências 250 Hz, 1000 Hz e 4000 Hz respectivamente. Na primeira linha se encontram as freqüências puras. A partir da segunda linha são exibidos os sinais cujas componentes de freqüência seguem uma distribuição normal. Na segunda linha o valor de σ equivale a $\mu/32$. Nas terceira linha σ assume o valore $\mu/16$, na quarta $\mu/8$ e na quinta $\mu/4$.

Por fim, o último sinal criado é constituído por ruído gaussiano branco, ou apenas ruído branco. Tal sinal se caracteriza por possuir todas as suas componentes de freqüência uniformemente distribuídas, conforme pode ser observado na Figura 13.

É importante salientar que os gráficos das Figuras 12 e 13 possuem a mesma escala em relação à freqüência, porém não em relação à amplitude. Todos os sinais tiveram suas amplitudes normalizadas no domínio do tempo, de forma que possuam o mesmo valor máximo. Seu

Figura 12: Espectro de freqüências dos sinais



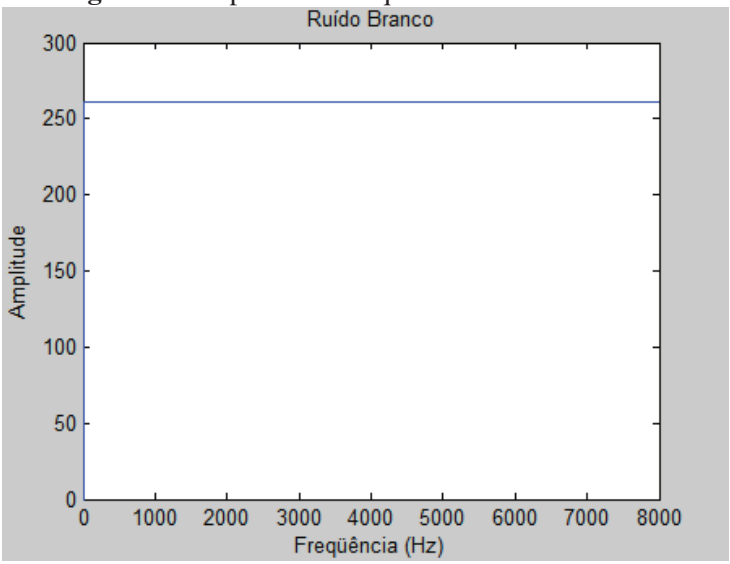
Fonte: Elaborada pelo autor

armazenamento se deu pela escrita de arquivos WAVE com 40 segundos de duração. As figuras se concentram no intervalo de interesse de até 8000 Hz, porém os sinais foram gerados com qualidade superior, possuindo taxa de amostragem de 44100 Hz.

Finalmente a execução dos experimentos propriamente dita ocorreu da seguinte forma. Primeiramente foram posicionados o kinect e o emissor, conforme já descrito e ilustrado na Figura 11. Em seguida foi escolhido um dos sinais, iniciada sua reprodução por parte do emissor e captura pelo Kinect. Nesta etapa ainda não é efetuada a localização, o *software* se limita à registrar os sinais tais quais retornados pelos microfones do dispositivo. O período de captura foi de 33 segundos, sendo realizadas 5 iterações de cada sinal sonoro. Ao todo foram realizadas 80 gravações, resultando em um total de 80 arquivos contendo 4 canais de áudio e taxa de amostragem de 16 kHz.

A área de busca utilizada nos experimentos foi definida em 4,19 x 5,90 metros, cobrindo toda a largura do laboratório e maior parte do seu comprimento. Visto que a solução não contempla posições atrás do Kinect, os 58 cm de comprimento (eixo X) não abrangidos pela área de busca se devem à posição de k, voltado para a parede direita e afastado exatos 58 cm da parede esquerda da Figura 11. A resolução escolhida foi de aproximadamente 5 cm^2, sendo portanto o *grid* formado por 84 x 118 pontos, totalizando 9912 pontos.

Figura 13: Espectro de freqüências do ruído branco



Fonte: Elaborada pelo autor

Durante os 33 segundos de execução de uma iteração dos testes o SRP-PHAT é executado 130 vezes. Os dados registrados a cada passada do algoritmo são o erro de posição e direção das posições estimadas. O erro de posição é obtido pelo cálculo da distância euclidiana entre a posição medida e e a estimada retornada pelo programa, chame-se $\hat{e}$. A Equação 4.1 dá a distância euclidiana entre dois pontos $p = (x_p, y_p)$ e $q = (x_q, y_q)$ no espaço bidimensional.

$$dist(p, q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2} \tag{4.1}$$

Já o erro de direção estimada é o ângulo horizontal formado pelas projeções dos vetores $\vec{ke}$ e $\vec{k\hat{e}}$ sobre o plano XY. O ângulo θ entre dois vetores $\vec{a}$ e $\vec{b}$ é dado em graus pela Equação 4.2:

$$\theta(\vec{a}, \vec{b}) = \arctan(\frac{x_{\vec{a}} - x_{\vec{b}}}{y_{\vec{a}} - y_{\vec{b}}}) * \frac{180}{\pi} \tag{4.2}$$

Ao fim de uma iteração são calculados a média e desvio padrão de seus erros de posição e direção.

Além dos sinais artificiais, também foram gravados 5 períodos de ruído ambiente ao longo dos testes. Isto é, em 5 momentos ao longo da execução dos testes foram registrados intervalos de 33 segundos sem a atuação do emissor. O propósito desta captura adicional é realizar uma estimativa da relação sinal-ruído, ou SNR do inglês *Signal-to-Noise Ratio*, de cada microfone durante os testes. Seguindo a metodologia descrita em (DIBIASE, 2000), os sinais discretos capturados pelos 4 microfones do Kinect são descritos por $x_1[n]...x_4[n]$, cada um possuindo o total de N amostras. Chame-se, então, $x_m^s[n]$ ao sinal capturado pelo m-ésimo microfone durante a emissão do sinal s. Da mesma forma o sinal discreto contendo apenas ruído ambiente é rotulado $x_m^r[n]$. O valor estimado da relação sinal-ruído do sinal s quando capturado pelo

microfone m é dada por 4.3:

$$SNR_m^s = 10\log_{10}\left\{\frac{\sum_{n=1}^{N}(x_m^s[n])^2}{\sum_{n=1}^{N}(x_m^r[n])^2}\right\} \tag{4.3}$$

Como os sons produzidos pelo emissor possuem muito maior amplitude, predominando sobre os demais, é aceitável assumir que o ruído ambiente oferece uma contribuição desprezível à potência de resposta dos sinais $x_m^s[n]$.

Os 85 arquivos de áudio (80 testes e 5 ruídos de fundo) tiveram suas potências calculadas para os 4 microfones. Em seguida foram feitas as médias de seus valores pelas 5 amostras de cada situação. Estas médias foram então aplicadas à Equação 4.3 gerando os índices de sinal-ruído que serão apresentados e discutidos na Seção 5.1.

4.2.3 Testes de precisão para localização 2D e 3D

Dando prosseguimento aos testes de precisão e à avaliação da solução implementada, o segundo conjunto de testes utiliza apenas um dos sinais cuja criação foi descrita na subseção anterior. Em lugar de testar diferentes sinais sonoros, são testadas diferentes posições do emissor sonoro. Nesta etapa somente o sinal contendo ruído gaussiano branco foi utilizado. Os resultados dos testes anteriores que justificam sua escolha podem ser consultados no Capítulo 5. A motivação por trás do novo grupo de testes é avaliar efetivamente a qualidade da solução proposta.

Até então a solução foi utilizada somente para localização 2D, isto é, considerou apenas as coordenadas x e y do sistema adotado, que descrevem comprimento e largura. A inclusão da coordenada z, que descreve a altura, para a localização 3D requer um mínimo de 4 microfones (ALAMEDA-PINEDA; HORAUD; MOURRAIN, 2013), todavia estes não podem ocupar posições colineares. Como os microfones do Kinect estão alinhados sobre um mesmo eixo, faz-se necessário um segundo Kinect para introduzir microfones em coordenadas que variem em relação ao eixo Z. O primeiro Kinect, rotulado $k1$, permaneceu em sua disposição convencional, mantendo horizontal o alinhamento de seus microfones. Já o segundo, $k2$, foi disposto verticalmente, de forma que seus microfones conservem os mesmos valores em x e y, porém variem em z A Figura 14 registra a disposição dos dois dispositivos durante esta etapa de testes.

O cenário dos testes permanece o mesmo, sendo, portanto, usado o mesmo sistema de coordenadas. O centro de $k1$ ocupou a coordenada $(0,58,\ \ 2,68,\ \ 0,78)$ enquanto o centro de $k2$ esteve em $(0,58,\ \ 2,92,\ \ 0,92)$. Os dois aparelhos foram alinhados no eixo X, respeitando a mesma distância de 58 cm da parede esquerda observada nos testes anteriores, como pode ser visto na Figura 15. Ambos foram orientados de frente para a parede à direita da ilustração.

As posições escolhidas para o emissor são $e1$ em $(4,19,\ \ 1,05,\ \ 0,14)$, $e2$ em $(3,31,\ \ 4,43,\ \ 1,68)$ e $e3$ em $(1,60,\ \ 3,62,\ \ 0,88)$, também identificadas na Figura 15. O ponto $e1$ representa a posi-

Figura 14: Disposição dos Kinects para segunda etapa de testes

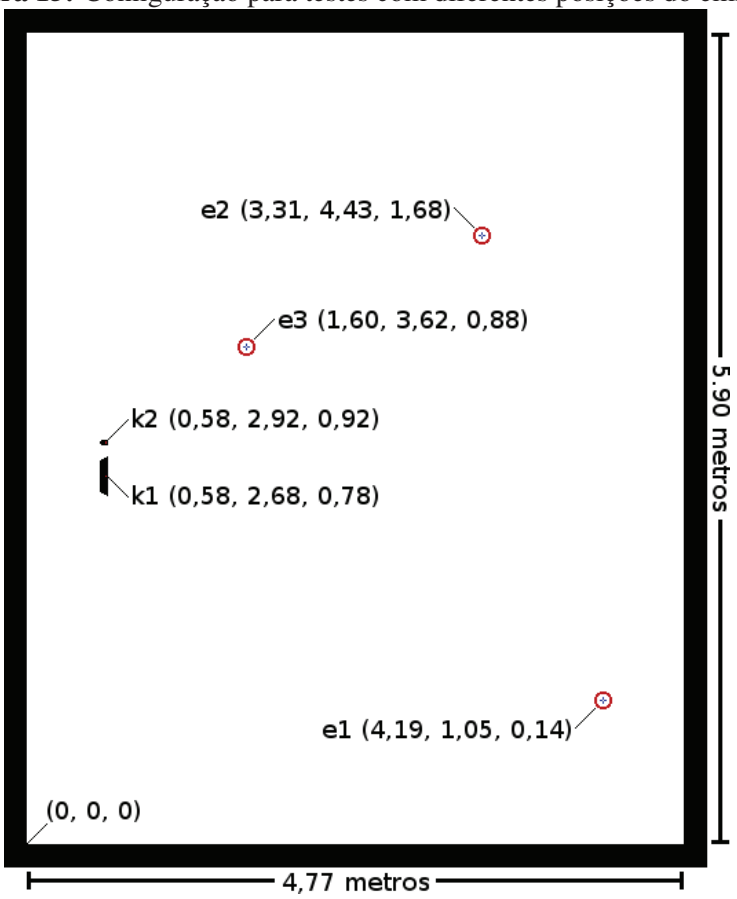
Fonte: Elaborada pelo autor

ção mais baixa do laboratório, sendo o emissor sonoro posicionado no chão para ocupá-la. Seus 14 cm correspondem à altura da própria caixa de som utilizada para a emissão. Além da altura, $e1$ também é caracterizado por estar bastante à direita da ótica dos aparelhos e ser o ponto mais afastado dos mesmos. A posição mais à esquerda dos Kinects corresponde a $e2$, com 1,68 metros de altura, constituindo-se, portanto, no ponto mais alto dos três. Por fim $e3$ se destaca por estar bem próximo dos microfones, possuindo também uma altura muito similar a destes.

A execução dos experimentos ocorreu de forma muito similar aos anteriores. Primeiramente os Kinects foram posicionados em suas respectivas coordenadas e a caixa de som em $e1$.Em seguida se iniciou a reprodução do ruído branco e execução do programa implementado para a captura de áudio através dos dois aparelhos. O período de captura permaneceu o mesmo, 33 segundos, porém o número de repetições foi aumentado de 5 para 10. A cada execução do *software* são gravados dois arquivos WAVE, um por Kinect. Após capturadas 10 amostras e gerados, portanto, 20 arquivos, o processo foi repetido para as demais posições do emissor, isto é, as coordenadas $e2$ e $e3$. No total foram realizadas 30 gravações de ruído branco, totalizando 60 arquivos de áudio de 4 canais amostrados em 16 kHz.

Findas as gravações cabe definir a área de busca e processar os arquivos de áudio. Neste experimento a solução foi avaliada tanto para a localização 2D quanto 3D. A localização 2D

Figura 15: Configuração para testes com diferentes posições do emissor



Fonte: Elaborada pelo autor

ocorre da mesma forma que nos testes com múltiplos sinais: consideram-se apenas os sinais capturados por $k1$ e define-se o *grid* sobre o espaço descrito pelos eixos X e Y, ignorando Z. Já a localização 3D contou com duas abordagens distintas, que serão descritas nos parágrafos a seguir.

A primeira abordagem de localização 3D, de natureza intuitiva e implementação trivial, será rotulada ”3D simples”. Sua estratégia consiste em atribuir uma área de busca para cada Kinect e executar o SRP-PHAT sobre os dados de cada um separadamente. Neste caso são definidas uma área de busca horizontal para $k1$ e outra vertical para $k2$. Enquanto a execução do algoritmo sobre os dados de $k1$ resultará estimativa das coordenadas x e y do emissor, os dados de $k2$ conduzirão à estimativa das coordenadas x e z. O resultado final é composto pela média dos valores de x obtidos e os valores de y e z gerados a partir dos sinais de $k1$ e $k2$ respectivamente. Na prática se tem duas localizações 2D simultâneas, cujos resultados são unificados em uma única coordenada do espaço tridimensional. Apesar de simplista, esta abordagem oferece a vantagem de possuir um custo relativamente próximo ao da localização 2D, dobrando o número de pontos na área de busca e de pares de microfones analisados pelo SRP-PHAT, que passam de 6 para 12.

Menos simplista, a segunda abordagem de localização 3D será denominada ”3D completa”.

Seu manuseio dos microfones de $k1$ e $k2$ os trata como componentes de um único arranjo formado por 8 microfones. A discretização espacial passa a contemplar os três eixos das coordenadas, gerando um "volume de busca" em lugar da tradicional "áreas de busca". Em conseqüência disto, o número de pontos do *grid* é aumentado em função da nova dimensão e o número de pares de microfones verificados pelo algoritmo passa a ser 28, em lugar dos 12 da abordagem anterior. Esta estratégia é, portanto, consideravelmente custosa se comparada às demais.

Para a localização 2D foi preservada a área de busca de 4,19 x 5,90 metros escolhida anteriormente. Embora a posição de $k1$ não seja a mesma do cenário anterior, seu deslocamento ocorre apenas em relação ao eixo Y, de forma que a área continua definida à frente do aparelho. O número de pontos e a resolução do *grid* também permanecem os mesmos: 84 x 118 pontos com aproximadamente 5 cm^2 de resolução. A mesma área de busca também é atribuída a $k1$ na localização 3D simples. No caso de $k2$ uma nova área de 4,19 x 3,44 metros é definida, delimitando um *grid* de 84 x 69 pontos, novamente com cerca de 5 cm^2 de resolução. O volume descrito durante a abordagem 3D completa, por sua vez, mede 4,19 x 5,90 x 3,44 metros. Para o *grid* foram selecionados 84 x 118 x 69 pontos, resultando em uma resolução de 5 cm^3. Em suma os totais de coordenadas analisadas na localização 2D, 3D simples e 3D completa respectivamente foram 9912, 15708 e 683928.

Como de costume, são registrados os erros de posição e direção das estimativas do *software* a cada execução do SRP-PHAT. As métricas de erro da localização 2D são as mesmas descritas na Subseção 4.2.2, recorrendo utilizando das Equações 4.1 e 4.2. O erro de posição das localizações 3D simples e completa leva também em conta a altura. Conseqüentemente, a distância euclidiana entre os pontos $p = (x_p, y_p, z_p)$ e $q = (x_q, y_q, z_q)$, agora em espaço tridimensional, é dada pela Equação 4.4.

$$dist(p, q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2 + (z_p - z_q)^2} \tag{4.4}$$

O erro de direção é medido verticalmente e horizontalmente. Portanto a Equação 4.2 é utilizada para calcular tanto o erro horizontal quanto vertical que diferencia os vetores $\vec{k1e}$ e $\vec{k1\hat{e}}$, onde $\hat{e}$ representa a coordenada estimada pelo algoritmo. O ângulo horizontal é formado entre as projeções dos dois vetores sobre o plano XY. Semelhantemente o ângulo vertical se encontra entre as projeções dos mesmos vetores sobre o plano XZ. Os dois ângulos têm seu desvio padrão computado após completadas suas 10 iterações.

Para a estimativa da relação sinal-ruído em cada um dos 8 microfones foram realizadas 10 gravações sem a presença do emissor, seguindo o mesmo processo exposto na Subseção 4.2.2, com base em (DIBIASE, 2000). As gravações ocorreram de forma distribuída ao longo do período de experimentação, sendo todas de 33 segundos de duração. Ao todo foram realizadas 40 gravações nesta etapa experimental (30 testes e 10 ruídos de fundo), originando 80 arquivos de áudio. A partir das 10 iterações de cada cenário foram calculadas suas médias, que por sua vez foram utilizadas no cálculo da relação sinal-ruído (vide Equação 4.3). Os valores da relação

serão exibidos juntamente com os resultados do experimento no Capítulo 5.

5 RESULTADOS

Neste trabalho foi proposta a utilização do Kinect como solução de SSL para a execução do algoritmo SRP-PHAT. Suas principais contribuições são um protótipo funcional da solução proposta e sua avaliação experimental. O programa suporta até dois Kinects atuando em conjunto, podendo ser adaptado para a execução de outros algoritmos de localização ou mesmo para a incorporação de outros tipos de microfones.

O algoritmo SRP-PHAT foi implementado de forma sequencial, *multithreaded* e em GPU (OpenCL). Foram realizados testes preliminares de desempenho executando sobre *grids* de 50 x 50 e 150 x 150 pontos. Os cenários considerados incluem a execução da implementação sequencial, *multithreaded* com 6 e 12 *threads* e 4 variações da implementação em OpenCL. Duas variações decorrem de sua execução em placas de vídeo diferentes, ambas da NVIDIA: uma Quadro 5000 e uma Tesla C2075, possuindo 352 e 448 núcleos de processamento respectivamente. As outras variações foram executadas em CPU, a primeira com o recurso de autovetorização habilitado e o segundo desabilitado. Este recurso busca maximizar o uso de instruções paralelas oferecidos pela CPU. Como a CPU em questão é um Core i7-3930K (3.20GHz, 6 núcleos físicos e 12 lógicos), fabricado pela Intel, e a implementação de OpenCL utilizada é a fornecida pela Intel, foi observado um impacto significativo desta otimização. A Tabela 1 exibe as médias e desvio padrão dos tempos de execução dos testes preliminares para um total de 200 execuções. Testes adicionais foram interrompidos devido a problemas que indisponibilizaram o equipamento.

Tabela 1: Tempos de execução (milissegundos)

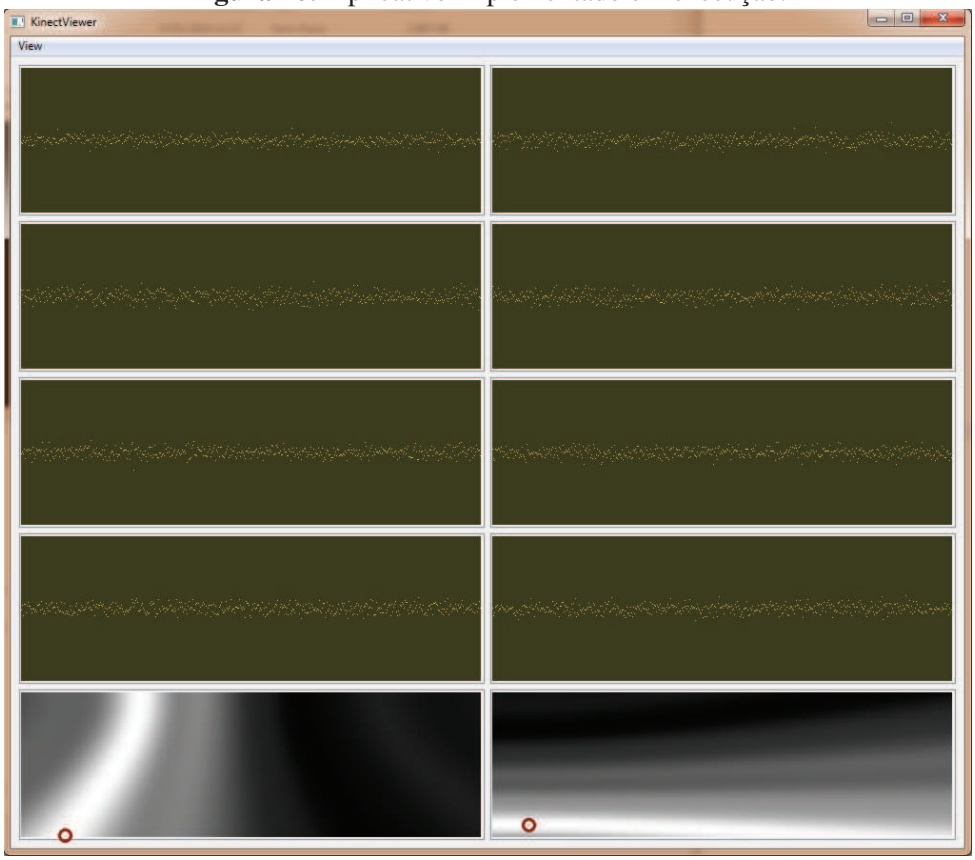
	50 X 50		150 X 150	
Cenários	Média	Desvio Padrão	Média	Desvio Padrão
Sequencial	2.181,65	7,79	18.665,90	54,66
6 *threads*	382,43	8,18	3.494,53	23,78
12 *threads*	241,32	7,81	2.162,98	10,13
OpenCL CPU (sem autovetorização)	144,95	7,26	1.256,91	14,82
OpenCL CPU (com autovetorização)	30,65	1,95	227,78	9,78
OpenCL GPU (Quadro)	33,95	4,50	145,29	6,70
OpenCL GPU (Tesla)	30,11	2,02	87,16	7,54

A Figura 16 exibe o aplicativo em funcionamento. Nesta imagem em particular estão sendo processados os dados de dois Kinects. Os oito painéis que iniciam na parte superior da janela exibem os sinais de cada microfone tal qual retornados pelos dispositivos. É possível alternar o modo de exibição para que sejam mostradas as Transformadas de Fourier dos sinais. À esquerda ficam os painéis do primeiro Kinect e à direita os do segundo. Os microfones são apresentados, de cima para baixo, partindo da extremidade do microfone mais afastado dos demais até a outra extremidade do aparelho.

Na parte inferior da janela estão representados os mapas de energia gerados pelo SRP-

PHAT. Nesta imagem o protótipo está realizando uma busca 3D simples, de modo que o painel esquerdo representa o mapa do plano horizontal e o direito o mapa do plano vertical. Em relação ao laboratório descrito na Subseção 4.2.1, o painel esquerdo exibe largura x comprimento com o Kinect na parte superior do mapa, enquanto o direito retrata comprimento x altura estando o segundo Kinect à esquerda do mapa. Quanto mais clara a cor de uma determinada posição maior sua energia. Foram acrescentados dois círculos à Figura 16 para destacar as posições estimadas pelo programa no momento em que a imagem foi registrada. A resolução das áreas de busca responsáveis pelos mapas de energia desta imagem é de 2 cm^2.

Figura 16: Aplicativo implementado em execução.



Fonte: Elaborada pelo autor

O protótipo passou por duas seções de experimentos cujos pormenores foram fornecidos no Capítulo 4. As duas seções a seguir apresentam os resultados dos experimentos e fazem algumas observações a respeito dos mesmos. A última seção traz à luz algumas questões que surgiram a partir dos resultados e os compara com os de outros autores.

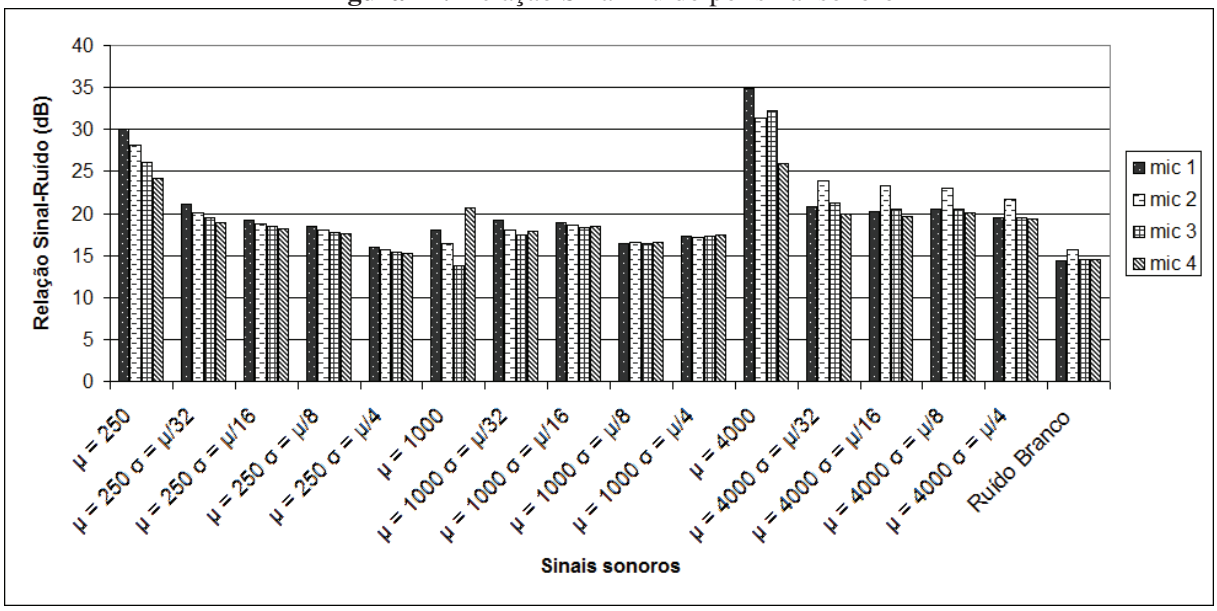
5.1 Testes de precisão com diferentes sinais sonoros

Neste teste inicial, descrito na Subseção 4.2.2, foram utilizados vários sinais sonoros para determinar qual seria de mais fácil localização para a solução. Relembrando que foram utilizados 16 sinais: ruído gaussiano branco, 3 senóides puras de 250 Hz, 1000 Hz e 4000 Hz e

4 variações de cada senóide, onde suas componentes de freqüência seguem distribuições normais com média μ igual à freqüência da senóide original e desvio padrão σ igual a $\mu/32$, $\mu/16$, $\mu/8$ ou $\mu/4$. Sendo assim, é examinada a influência de duas características do sinal emitido: sua freqüência e a largura de seu espectro de freqüências. Para cada sinal foram gravados 5 testes, estimadas as relações sinal-ruído dos microfones e verificada a precisão da estimativa gerada pelo programa. Antes de abordar a precisão da solução é importante informar a relação sinal-ruído observada nos testes.

Conforme já foi explicado, a relação é calculada com base nos arquivos de áudio gravados durante os testes juntamente com gravações contendo apenas ruído ambiente. Foram calculadas as potências médias para as 5 iterações de cada cenário (incluindo ruído ambiente), sendo a relação dada na Equação 4.3 já apresentada. Os valores estão representados na Figura 17, podendo ser consultados na Tabela 2.

Figura 17: Relação Sinal-Ruído por sinal sonoro



Fonte: Elaborada pelo autor

Na grande maioria dos casos as taxas de sinal ruído dos microfones se mantiveram entre 15 e 25 dB. O valor mínimo observado foi de 13,83 dB (1000 Hz, microfone 3) e o máximo 34,86 dB (4000 Hz, microfone 1). Baseado nestes valores, pode-se afirmar que os sinais de interesse possuem uma predominância considerável em relação ao ruído ambiente. Nos cenários que envolviam senóides puras os índices de potência calculados exibiram variações muito maiores que os demais, tanto entre microfones quanto entre iterações de um mesmo microfone. De forma geral, o aumento do espectro de freqüências dos sinais emitidos tende a reduzir as discrepâncias entre os microfones.

A precisão é medida em termos de erro de posição e direção, obtidos através das equações 4.1 e 4.2 respectivamente, também apresentadas na Subseção 4.2.2. Cada iteração de um cenário de testes dura 33 segundos, o que implica em 130 execuções do SRP-PHAT. Ao fim

Tabela 2: Relação Sinal-Ruído por sinal sonoro

Sinais	mic 1	mic 2	mic 3	mic 4
$\mu = 250$	30,0446	28,0772	26,0948	24,2399
$\mu = 250\ \sigma = \mu/32$	21,0612	20,1233	19,4625	18,8823
$\mu = 250\ \sigma = \mu/16$	19,1768	18,8108	18,4107	18,1180
$\mu = 250\ \sigma = \mu/8$	18,4612	18,0776	17,7569	17,5269
$\mu = 250\ \sigma = \mu/4$	16,0029	15,6154	15,3701	15,1981
$\mu = 1000$	18,0778	16,3417	13,8345	20,7055
$\mu = 1000\ \sigma = \mu/32$	19,2141	18,0808	17,4027	17,9107
$\mu = 1000\ \sigma = \mu/16$	18,9579	18,6565	18,2742	18,4362
$\mu = 1000\ \sigma = \mu/8$	16,4448	16,5731	16,3743	16,5112
$\mu = 1000\ \sigma = \mu/4$	17,2169	17,1378	17,3340	17,4451
$\mu = 4000$	34,8628	31,3525	32,1769	25,9625
$\mu = 4000\ \sigma = \mu/32$	20,8030	23,9083	21,2894	19,9738
$\mu = 4000\ \sigma = \mu/16$	20,2296	23,3579	20,5571	19,6029
$\mu = 4000\ \sigma = \mu/8$	20,5021	23,0590	20,5366	20,0780
$\mu = 4000\ \sigma = \mu/4$	19,5445	21,7323	19,5206	19,3186
Ruído Branco	14,3103	15,7171	14,5698	14,5657

deste período, são registrados a média e desvio padrão dos erros de posição e direção das estimativas do algoritmo. Os valores apresentados na Tabela 3 correspondem ao valor médio das médias obtidas, assim como dos desvios padrões. Em outras palavras, seus valores representam média e desvio padrão de todas as 650 execuções do algoritmo para um dado sinal do emissor.

A maioria dos sinais testados apresentou erros de posição superiores a 0,40 cm e de direção inferiores a 8º. Um índice de erro bastante elevado, pelo menos no que tange à posição, considerado que foi aplicado um *grid* cuja resolução é de 5cm^2. Os sinais centrados na freqüência de 250 Hz não demonstraram mudança de comportamento com o aumento de seu espectro de freqüências, errando sempre em torno de 51 cm a posição (com 7 cm de desvio padrão) e 8º a direção (com 3º de desvio padrão). Já os sinais com μ igual a 1000 Hz e 4000 Hz exibem, na maioria dos casos, uma modesta melhora de estimativa quanto mais amplo o seu espectro. Esta melhora se observa nos erros de posição, especialmente em seu desvio padrão, que mostra que as respostas do algoritmo foram se tornando mais consistentes, ainda que indicando a posição errada. No caso do sinal $\mu = 4000\ \sigma = \mu/4$ em particular, se observou uma melhora drástica, onde a posição foi estimada com 12 cm de erro com 6 cm de desvio padrão e a direção com 8º de erro, porém 14º de desvio padrão.

De uma forma geral a solução demonstrou maior firmeza na determinação da direção do que da posição. É importante mencionar que foi possível observar a influência do ruído ambiente sobre todos os testes com exceção do ruído branco, que se mostrou mais robusto. O ruído ambiente em questão foi o som do próprio computador onde estava conectado o Kinect e foi executado o aplicativo.

Para concluir esta análise, constatou-se que os sinais com espectro de freqüências mais amplo geraram as duas melhores estimativas. Estes são $\mu = 4000\ \sigma = \mu/4$ e o ruído gaussi-

Tabela 3: Erros de posição e direção

	Posição (metros)		Direção horizontal (graus)	
Sinais	Média	Desvio Padrão	Média	Desvio Padrão
$\mu = 250$	0,5097	0,0705	6,8905	3,7215
$\mu = 250\ \sigma = \mu/32$	0,5169	0,0839	7,5618	2,7615
$\mu = 250\ \sigma = \mu/16$	0,5173	0,0758	8,1396	2,4210
$\mu = 250\ \sigma = \mu/8$	0,5165	0,0653	7,9161	2,0832
$\mu = 250\ \sigma = \mu/4$	0,5246	0,0891	7,5775	2,3508
$\mu = 1000$	0,6271	0,2537	1,2461	22,2621
$\mu = 1000\ \sigma = \mu/32$	0,5515	0,2056	2,2698	19,6714
$\mu = 1000\ \sigma = \mu/16$	0,4984	0,1401	2,7480	22,2091
$\mu = 1000\ \sigma = \mu/8$	0,4782	0,1201	6,6157	24,0249
$\mu = 1000\ \sigma = \mu/4$	0,4934	0,1162	6,3532	12,6095
$\mu = 4000$	0,7577	0,5636	12,3127	22,5508
$\mu = 4000\ \sigma = \mu/32$	0,8306	0,6680	4,0144	36,7456
$\mu = 4000\ \sigma = \mu/16$	0,7358	0,6593	1,6528	34,0808
$\mu = 4000\ \sigma = \mu/8$	0,4056	0,3361	11,9281	29,1533
$\mu = 4000\ \sigma = \mu/4$	0,1205	0,0689	8,5405	14,1245
Ruído Branco	0,0891	0,0149	0,2051	0,1365

ano branco, porém o segundo apresentou resultados muito superiores, oferecendo os menores índices de erro em todas as medidas. Seu erro médio foi de aproximadamente 9 cm (com desvio padrão de 1 cm) da posição correta, e apenas 0,2º (com desvio padrão de 0,1º) da direção. Novamente levando em conta a resolução de 5cm^2 do *grid*, tais índices se mostram bem mais toleráveis. Baseado nestes resultados, os demais experimentos recorreram apenas ao ruído branco como sinal emitido pela fonte sonora.

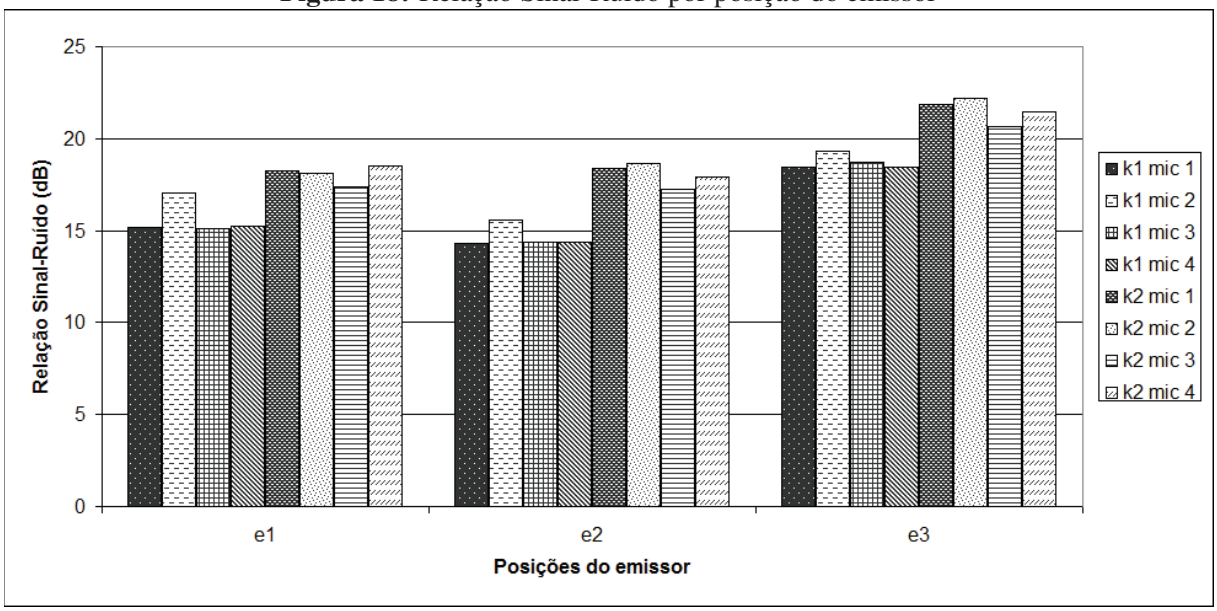
5.2 Testes de precisão para localização 2D e 3D

O segundo conjunto de experimentos utilizou dois Kinects, denominados $k1$ e $k2$, conforme consta na Subseção 4.2.3. Desta vez um único sinal foi reproduzido pelo emissor, o ruído gaussiano branco, porém o mesmo foi posicionado em três coordenadas diferentes: $e1$, $e2$ e $e3$. Para cada posição do emissor foram realizadas 10 capturas de áudio. Da mesma forma foram gravadas também 10 amostras de ruído ambiente, novamente utilizadas para o cálculo da relação sinal-ruído. Novamente foi adotado o período de captura de 33 segundos, o que se traduz em 130 execuções do SRP-PHAT por arquivo gravado.

Assim como nos testes anteriores, as novas taxas de sinal-ruído foram calculados através da Equação 4.3. Neste caso, entretanto, as potências aplicadas à equação são médias de 10 gravações (ao invés de 5). Na Figura 18 exibe um gráfico dos valores obtidos para os microfones de $k1$ e $k2$ nas três posições do emissor. Os valores exatos, em decibéis, estão disponíveis para consulta na Tabela 4.

Os índices de sinal-ruído variaram em torno de 14 dB a 22 dB, o que também se observou

Figura 18: Relação Sinal-Ruído por posição do emissor



Fonte: Elaborada pelo autor

Tabela 4: Relação Sinal-Ruído por posição do emissor

	Kinect horizontal (k1)				Kinect vertical (k2)			
Posições	mic 1	mic 2	mic 3	mic 4	mic 1	mic 2	mic 3	mic 4
e1	15,1769	17,0525	15,1183	15,2175	18,2533	18,1462	17,3703	18,5249
e2	14,3113	15,5860	14,3688	14,3461	18,3753	18,6504	17,2553	17,8972
e3	18,4159	19,2858	18,6912	18,4707	21,8713	22,1769	20,6350	21,4650

nos testes anteriores com respeito ao ruído branco. Como esperado, $e3$, a posição mais próxima dos Kinects, resultou em índices maiores que as demais. Na maioria dos microfones ocorre uma pequena redução do valor de $e1$ para $e2$, apesar de $e2$ estar mais próximo dos aparelhos.Esta diferença, observada principalmente nos microfones de $k1$, provavelmente se deve ao fato de $e1$ estar mais próximo de $k1$ enquanto $e2$ está mais próximo de $k2$. Em todos os casos testados os microfones de $k2$ se mostraram mais sensíveis do que os de $k1$.

No capítulo anterior, quando explicados os procedimentos dos testes, foram descritas três abordagens distintas de discretização espacial, rotuladas busca 2D, 3D simples e 3D completa. A busca 2D se concentra apenas em $k1$, ignorando a altura e estimando a posição da fonte em relação ao comprimento e largura do ambiente de teste. Os erros de posição e direção obtidos podem ser vistos na Tabela 5. Seus valores correspondem às médias e desvio padrão dos erros referentes a todas as execuções do SRP-PHAT de uma dada posição do emissor. Em outras palavras, são as médias e desvio padrão de 1300 estimativas.

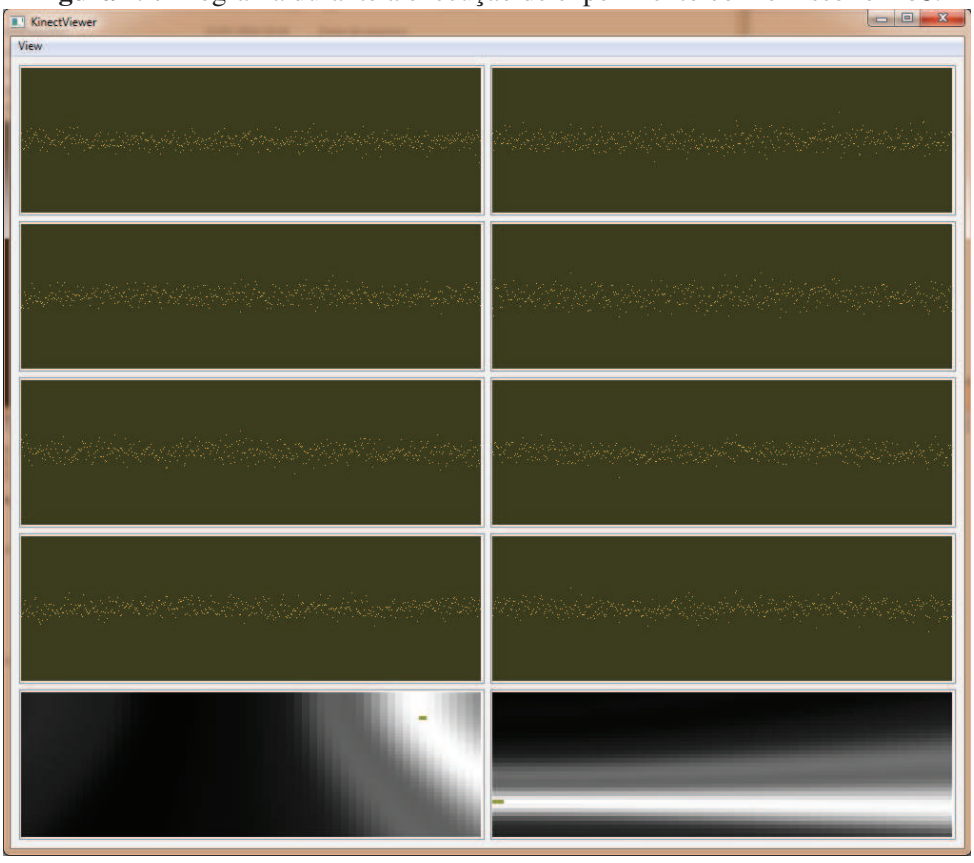
Os índices de erro obtidos neste teste foram bastante elevados, especialmente na posição $e3$. Na grande maioria dos testes o que se observou no mapa de energia criado pelo SRP-PHAT é um feixe de energia passando próximo às posições dos Kinects e do emissor. Valores como os de $e1$ tipicamente refletem dificuldades por parte do protótipo para determinar a posição do

Tabela 5: Erros de posição e direção da busca 2D

	Posição (metros)		Direção horizontal (graus)	
Posições	Média	Desvio Padrão	Média	Desvio Padrão
e1	0,3864	0,0798	4,1652	2,7277
e2	0,1191	0,0231	47,6000	6,6051
e3	1,5255	0,0481	38,9445	5,3989

emissor dentro deste feixe. A conseqüência visível é um erro de posição elevado acompanhado de um erro de direção muito baixo. No entanto, nos casos de $e2$ e $e3$ o feixe em $k1$ exibiu uma curvatura acentuada, como pode ser visto na Figura 19. Relembrando que $k1$ se encontra na parte superior do painel inferior esquerdo e $k2$ na parte esquerda do painel inferior direito, conforme visto no início do capítulo. Como estas posições estão mais próximas de $k1$, especialmente $e3$, mesmo um pequeno deslocamento da estimativa sobre o feixe pode significar um grande erro de direção. Esta questão será abordada novamente no final do capítulo.

Figura 19: Programa durante a execução de experimento com emissor em $e3$.



Fonte: Elaborada pelo autor

As buscas 3D simples e completa introduzem um ângulo vertical às medidas de erro de direção. O ângulo expressa o erro de estimativa em relação à altura do emissor. Assim como no ângulo horizontal, este erro é expresso na forma de média e desvio padrão da discrepância entre a direção real e a estimada.

Relembrando a Subseção 4.2.3, na busca 3D simples são realizadas duas buscas 2D, uma por Kinect. Enquanto os sinais de $k1$ são utilizados para efetuar a busca em uma área horizontal, os sinais de $k2$ têm a si associados uma área de busca vertical. A posição em relação à largura é dada por $k1$, à altura por $k2$ e ao comprimento é formado pela média de ambos. Os mesmos sinais utilizados para os testes 2D acima foram processados para os testes 3D. Os resultados da abordagem 3D simples podem ser consultados na Tabela 6.

Tabela 6: Erros de posição e direção da busca 3D simples

	Posição (metros)		Direção horizontal (graus)		Direção vertical (graus)	
Posições	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
e1	1,6954	0,2446	21,3659	2,8067	4,9480	3,5684
e2	2,3198	0,2224	43,3169	7,2809	45,2357	12,3892
e3	1,7030	0,0790	39,8136	5,4783	16,1293	25,4285

Todas as estimativas de posição exibiram erros muito elevados para a configuração em questão, com mais de 1,5 metros de distância da posição correta. Mesmo a direção das estimativas sofreu bastante, exibindo erros médios de até 45º. A posição mais distante, $e1$, obteve a estimativa mais favorável, com um erro de direção vertical de apenas 4º. Em geral os valores elevados de desvio padrão denunciam uma grande flutuação por parte das estimativas ao longo dos testes.

Diferente da abordagem anterior, a busca 3D completa discretiza todo o ambiente em um volume de busca. O algoritmo percorre um único *grid* contemplando as três dimensões, enquanto manuseia os dois Kinects como partes de um único arranjo de microfones. Os resultados estão disponíveis na Tabela 7, que exibiu resultados muito mais favoráveis do que as abordagens anteriores.

Tabela 7: Erros de posição e direção da busca 3D completa

	Posição (metros)		Direção horizontal (graus)		Direção vertical (graus)	
Posições	Média	Desvio Padrão	Média	Desvio Padrão	Média	Desvio Padrão
e1	1,9021	0,7445	0,7830	0,4880	2,1028	1,5793
e2	2,3605	0,1901	1,0503	0,5291	3,8127	1,4409
e3	0,1282	0,0645	0,6038	0,2819	1,3711	0,2618

Como pode ser observado, os erros de posição continuam elevados nas posições $e1$ e $e2$, com médias de 1,90 m e 2,36 m cada. Seu desvio padrão também demonstra certo grau de inconstância nas estimativas. Já $e3$ foi estimada com apenas 12 cm de erro e 6 cm de desvio padrão. Em contraste com as posições, as direções estimadas, tanto horizontais quanto verticais, se mostraram muito promissoras. A maior média de erro registrada foi inferior a 4º, sendo os desvios padrões inferiores a 2º.

Em suma a solução se revelou fraca para a estimativa de posições, mas evidenciou potencial para a estimativa da direção de fontes sonoras. As abordagens 2D e 3D simples apresentaram margens de erro muito grandes, não sendo interessantes para estimar posições nem direções. Por sua vez, a busca 3D completa se mostrou robusta e precisa para apontar na direção da fonte,

o que indica um grande impacto do aumento do tamanho do arranjo de 4 microfones para 8. Relembrando que a busca 3D simples trabalha com dois arranjos de 4 microfones de forma independente, enquanto a 3D completa considera os dois Kinects parte de um único arranjo de 8 microfones.

5.3 Observações finais

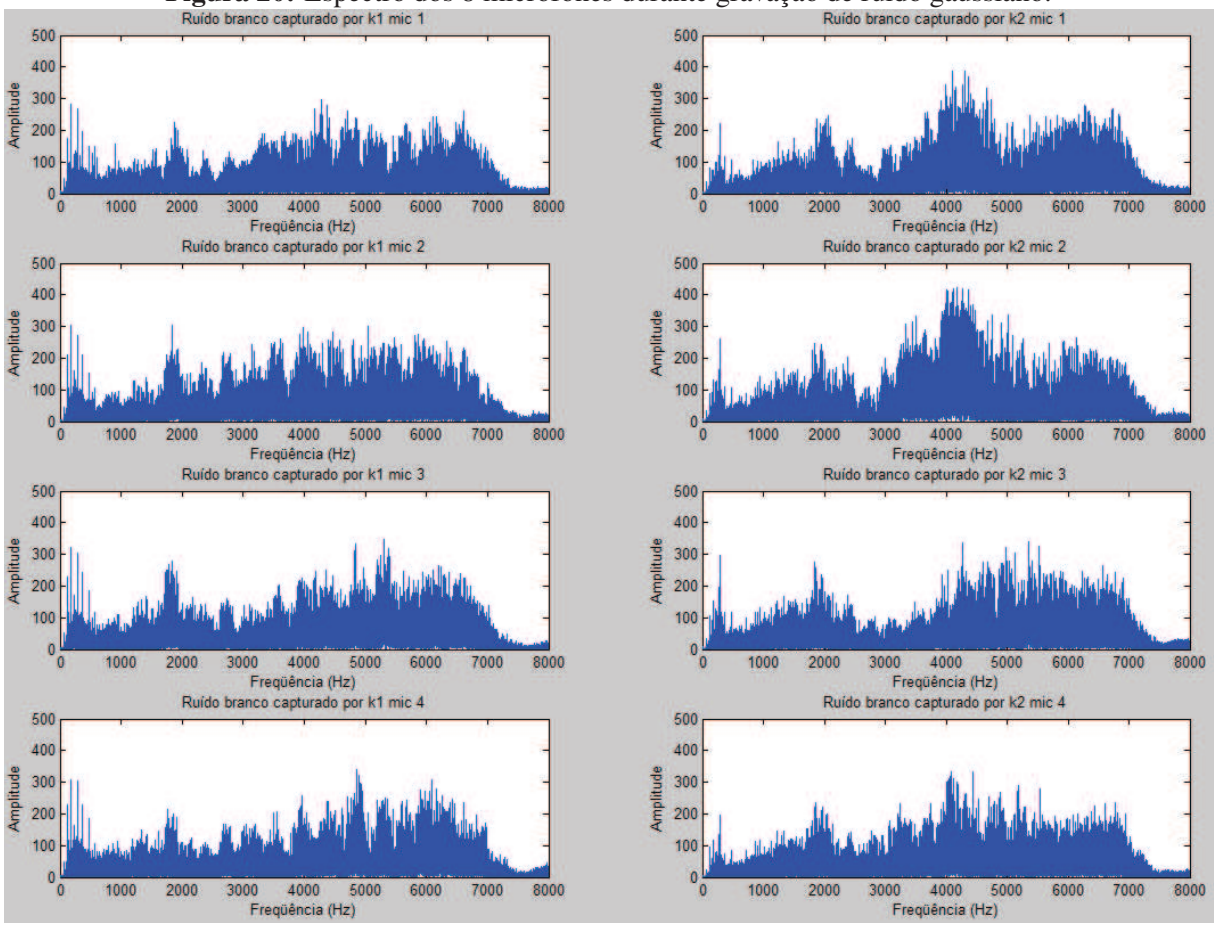
As seções anteriores apresentaram os resultados experimentais, discorrendo brevemente sobre seus comportamentos. A presente seção levanta algumas questões e considerações a respeito dos mesmos, bem como das limitações da solução proposta. Buscando compreender, sobretudo, as resultados desfavoráveis das buscas 2D e 3D simples, são apresentadas algumas hipóteses. Por fim são também comparados os resultados deste trabalho com os testes de precisão do SRP-PHAT disponíveis na literatura, fomentando nova discussão.

Além da diferença de sensibilidade entre os microfones de $k1$ e $k2$, observada nas relações sinal-ruído presentes na Figura 17, também há uma diferença audível entre as gravações de ruído gaussiano branco realizadas por um e por outro. Na esperança de compreender melhor esta questão e suas conseqüências, os canais das gravações realizadas durante os testes foram isolados em arquivos individuais para visualização de seus espectros de freqüências. A Figura 20 exibe os espectros dos 8 microfones durante uma das gravações de ruído branco com o emissor posicionado em $e3$ realizada para os testes. Em geral os sinais são bastante semelhantes entre si, especialmente aqueles capturados por $k1$. No entanto alguns sinais exibem diferenças marcantes, como é o caso dos microfones 1 e 2 de $k2$. Para facilitar a comparação, a Figura 21 aproxima os sinais do microfone 4 de $k1$ e 2 de $k2$. Naturalmente certo grau de diferença entre microfones é esperado, sobretudo considerando que os próprios microfones introduzem certo grau de ruído individualmente. Todavia as diferenças observadas são bastante significativas, principalmente na faixa de 3000 Hz a 5000 Hz.

Como é difícil verificar a influência dos microfones sobre um sinal com espectro tão amplo quanto o ruído gaussiano em questão, foram gravadas novas amostras das freqüências 250 Hz, 1000 Hz e 4000 Hz. Desta vez os dois Kinects foram posicionados horizontalmente, lado a lado, à mesma distância de 94 centímetros do emissor, posicionado logo a sua frente. As Figuras 22, 23 e 24 exibem os espectros dos três sinais, da menor à maior freqüência, conforme capturados pelos 8 microfones. Para facilitar a comparação, as mesmas escalas de freqüência e amplitude foram mantidas para todos os microfones, sendo mostradas apenas as faixas de freqüências onde há amplitude significativa.

Os gráficos não denunciam nenhuma distorção das freqüências, ou qualquer outro efeito que remeta ao observado na Figura 21. Muito pelo contrário, mostram que gravações realizadas pelos Kinects são de boa qualidade, havendo apenas flutuações perceptíveis nas amplitudes dos sinais. Da mesma forma quando isolados e reproduzidos os sinais de cada microfone, nenhuma diferença audível foi percebida entre eles. Baseado nestas constatações se descartou

Figura 20: Espectro dos 8 microfones durante gravação de ruído gaussiano.

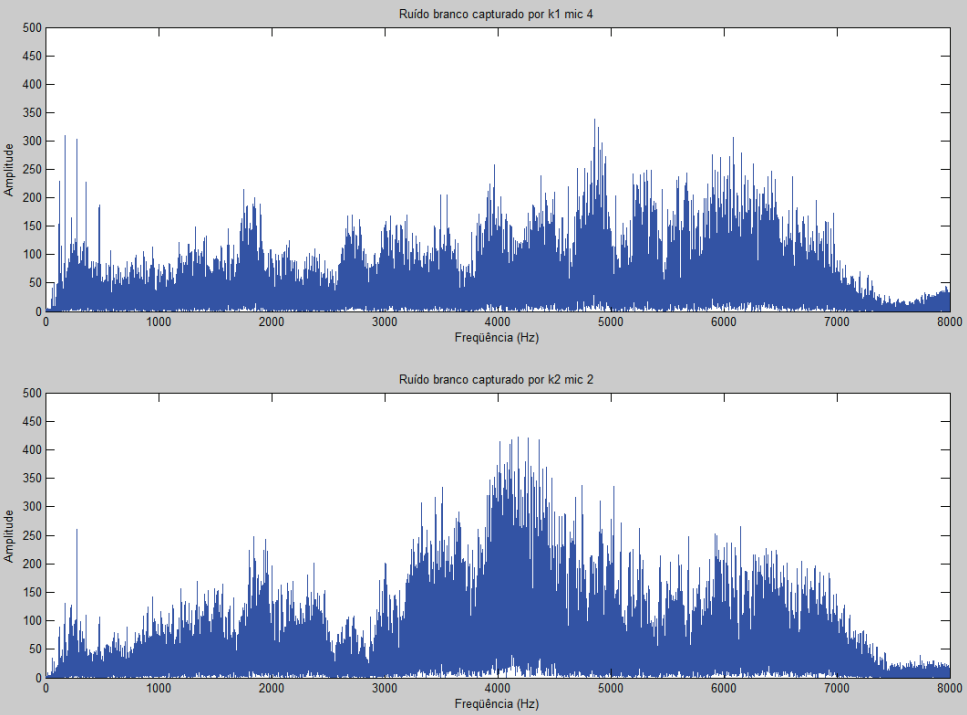


Fonte: Elaborada pelo autor

a hipótese de que o problema esteja relacionado à captura dos sinais. É, no entanto, possível que as diferenças sejam introduzidas durante a amostragem dos sinais. Lembrando que o ruído branco foi gerado com uma taxa de 44100 Hz e os Kinects o amostram em 16000 Hz.

Como foi mencionado anteriormente, o mapa de energia criado pela solução durante os testes freqüentemente exibiu feixes com altos índices de energia próximos às posições dos Kinects e do emissor. Em muitos casos os feixes eram retilíneos, de forma que dificuldades por parte do protótipo para escolher uma posição dentro de um feixe prejudicassem apenas estimativas de posição, e não de direção. Todavia, esta instabilidade quanto à distância do emissor em relação aos Kinects agravou severamente a busca 3D simples. Como não ocorre comunicação entre $k1$ e $k2$, antes suas estimativas são feitas independentemente, é comum haver discordâncias fortes quanto à distância do emissor por parte dos dois aparelhos. Um exemplo de pode ser visto na Figura 19, onde o mapa esquerdo afasta um pouco o emissor (para baixo) enquanto o mapa direito o coloca imediatamente à frente dos Kinects (esquerda). Pelo fato desta abordagem formar sua resposta pela composição de duas estimativas, e não de uma verificação da coordenada final estimada, mesmo a direção retornada pode destoar significativamente das estimativas originais. O mesmo não ocorre com a busca 3D completa, pois a estratégia considera a correlação entre

Figura 21: Comparação entre espectros de 2 microfones.



Fonte: Elaborada pelo autor

microfones de $k1$ e $k2$ e escolhe sua coordenada com base na energia verificada na mesma.

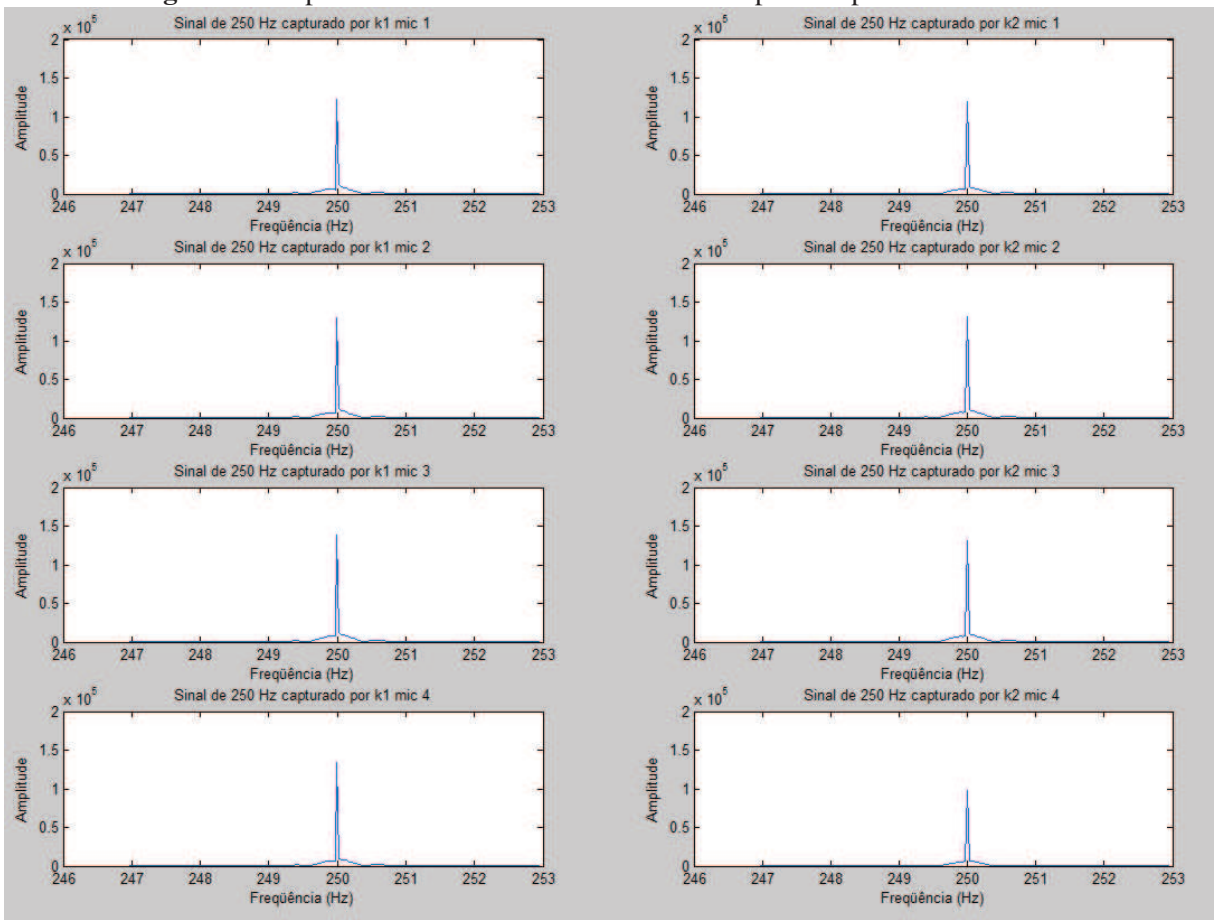
Dentre os experimentos realizados, apenas a busca 3D completa originou resultados comparáveis aos que se encontram na literatura. Na Seção 3.3 foram comentados alguns resultados de autores que avaliaram ou utilizaram como base de comparação o algoritmo SRP-PHAT. Todos os trabalhos em questão se concentraram na voz humana como fonte sonora de interesse, embora (DIBIASE, 2000) tenha utilizado ruído gaussiano ao testar o GCC.

Os resultados apresentados em (DO; SILVERMAN, 2007) e (DIBIASE, 2000) avaliam a estimativa de posição do algoritmo. Ambos os trabalhos compartilham o mesmo laboratório e equipamento. O primeiro utilizou 24 microfones do arranjo, tolerando erros de até 5 cm em relação ao comprimento e largura e até 10 cm à altura. Sua taxa de acertos foi de 100%, 96,6%, 87,8% e 67,3% para diferentes posições testadas. O segundo, por sua vez, empregou 128 microfones com tolerância de 10 cm^3 de erro e obteve 100% de acerto.

Com respeito à estimativa de posição, o protótipo deste trabalho apresentou margens de erro graves, mesmo durante a busca 3D completa. As estimativas das posições $e1$ e $e2$ sofreram erros médios de 1,90 m^3 e 2,36 m^3. No caso de $e3$ a média de 12 cm^3 de erro está bastante próxima do limite tolerado por DiBiase (2000). Entre as questões a se considerar sobre as diferenças de resultado dos três trabalhos está a impacto do número de microfones do arranjo. No entanto cabe mencionar que o arranjo de microfones da Universidade de Brown e um par de Kinects configuram dois extremos de arranjo de microfones, de forma que não constituem uma comparação razoável.

Em termos de *hardware* os estudos que mais se assemelham ao presente são (ZHANG;

Figura 22: Espectro do sinal de 250 Hz conforme capturado pelos 8 microfones.

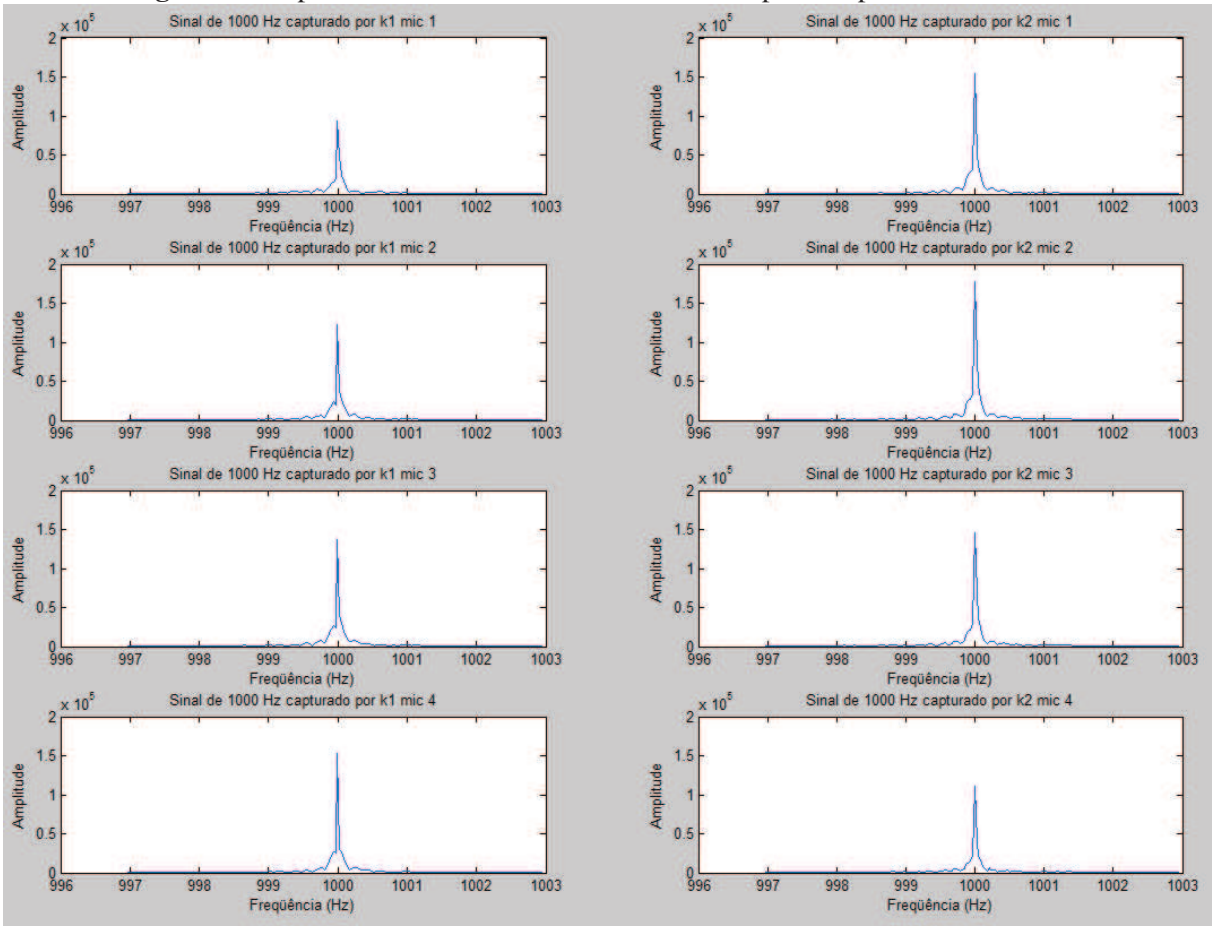


Fonte: Elaborada pelo autor

FLORENCIO; ZHANG, 2008) e (ZHANG; ZHANG; FLORENCIO, 2007). Ambos operam com um arranjo de 6 microfones distribuídos de forma circular. No primeiro os testes foram realizados em ambiente sintético, combinando gravações de sinais reais à simulação, enquanto o segundo foi totalmente executado em ambiente real. Nas execuções do SRP-PHAT realizadas em (ZHANG; FLORENCIO; ZHANG, 2008) foram medidas as taxas de erro de direção com tolerância de 2^o e 10^o para diferentes relações sinal-ruído e tempos de reverberação. Considerando uma relação sinal-ruído no intervalo entre 10 dB e 25 dB, que abrange os intervalos observados neste trabalho, os autores obtiveram de 58,8% a 97,6% das estimativas com erro inferior a 2^o. Estas taxas sobem para 77,0% e 98,9% quando tolerados até 10^o de erro. Ao experimentar em cenário real em (ZHANG; ZHANG; FLORENCIO, 2007) os autores utilizaram as tolerâncias de erro de 6^o e 14^o, obtendo os índices de acerto 81,73% e 88,13% respectivamente.

Dentre as seis médias de direção apresentadas pela busca 3D completa, apenas duas ultrapassaram os 2^o graus de erro, sendo a maior média inferior a 4^o. Como o maior desvio padrão é de aproximadamente $1{,}5^o$, houve pequena oscilação entre as estimativas, o que torna as médias bastante expressivas. Estes resultados demonstram potencial, visto não ultrapassarem os limites tolerados por (ZHANG; ZHANG; FLORENCIO, 2007) em cenário real. Mesmo quando com-

Figura 23: Espectro do sinal de 1000 Hz conforme capturado pelos 8 microfones.



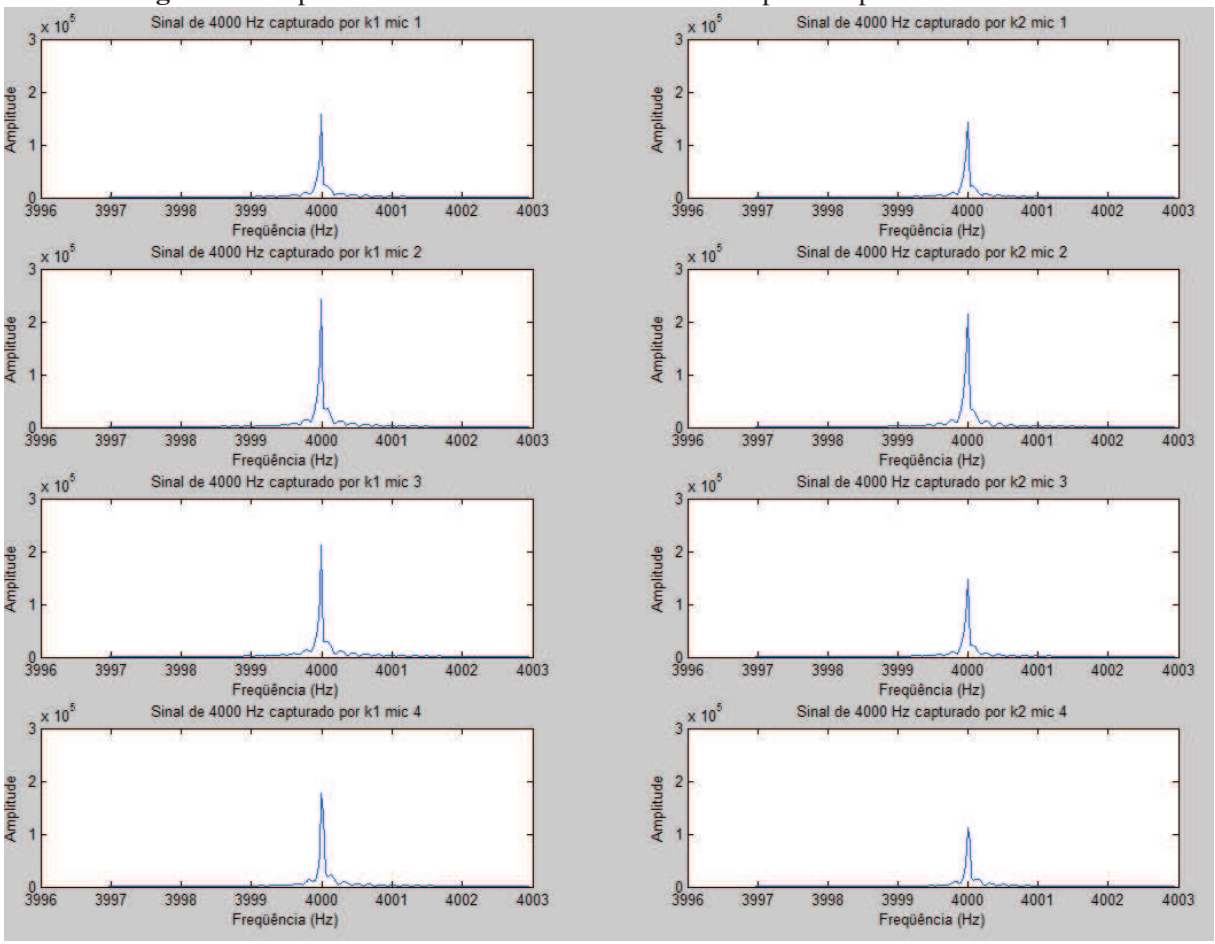
Fonte: Elaborada pelo autor

parado ao cenário virtual em (ZHANG; FLORENCIO; ZHANG, 2008), o limite extremo de 2º é desrespeitado na menor parte dos casos. Nesta comparação novamente o sistema com o maior número de microfones apresentou melhores resultados.

Além do número total de microfones, a relação sinal-ruído possui impacto significativo sobre as estimativas do SRP-PHAT, conforme evidenciado em (ZHANG; FLORENCIO; ZHANG, 2008). Quanto menor a relação, mais suscetível à reverberação serão as estimativas. Os melhores resultados obtidos em (ZHANG; ZHANG; FLORENCIO, 2007) foram gerados por sinais que chegam a 25 dB. Os índices utilizados nos experimentos deste trabalho foram, em sua maioria, inferiores a 20 dB, atingindo o valor máximo de 22 dB. É uma faixa de valores bastante próxima à dos autores, porém como o nível de reverberação do ambiente de testes não é conhecido, sua influência sobre os resultados pode ser expressiva.

Existem ainda outras variáveis cuja influência não foi avaliada e pode ser difícil de medir. Elas incluem o posicionamento dos Kinects, suas direções e a própria carcaça dos dispositivos. Por exemplo, como seriam os resultados se os Kinects estivessem mais afastados um do outro? E se estivessem voltados para direções diferentes? Em (ZHANG; ZHANG; FLORENCIO, 2007) foram usados microfones direcionais, o que levou os autores a constatar que a fase de

Figura 24: Espectro do sinal de 4000 Hz conforme capturado pelos 8 microfones.



Fonte: Elaborada pelo autor

sinais obtidos por microfones que não estivessem voltados para o emissor não era confiável. Os microfones do Kinect são supercardióides (TASHEV, 2012), o que significa que a recepção de sinais varia em função do ângulo de chegada dos mesmos. Lembrando que os microfones se encontram voltados para baixo no interior do aparelho, sendo talvez adequado virá-lo de acordo. Por fim, a própria carcaça do Kinect constitui uma obstrução entre a fonte sonora e os microfones, agredindo à premissa do SPR-PHAT que diz existir um caminho direto entre ambos. Seriam os resultados melhores se a carcaça fosse removida?

6 CONCLUSÕES E TRABALHOS FUTUROS

6.1 Conclusões

Neste trabalho foi sugerida e implementada uma solução econômica de Localização de Fonte Sonora que combina o dispositivo Kinect e o algoritmo SRP-PHAT. Além de uma versão seqüencial do algoritmo, o *software* foi enriquecido com duas implementações paralelas do mesmo, uma *multithreaded* e outra para execução em GPU. O protótipo é capaz de capturar e processar sinais em tempo real ou gravá-los em disco para realizar seu processamento posteriormente. Até dois Kinects são suportados pelo programa tendo sido implementada a localização 2D e 3D.

A solução teve suas estimativas avaliadas por uma série de testes de precisão. O protótipo revelou grande suscetibilidade a erros quando utilizado apenas um Kinect ou dois operando como arranjos de 4 microfones isolados. Mesmo ao se utilizar dois Kinects para formar uma arranjo de 8 microfones, as estimativas de posição ainda foram insatisfatória em sua maioria. No entanto as estimativas de direção se mostraram precisas, com erros inferiores a 4º, indicando um caminho a ser explorado.

Os resultados revelam que a solução proposta é adequada a contextos onde a direção da fonte sonora basta. Por exemplo para direcionamento automático de câmeras, seja durante discursos, videoconferências ou ambientes que requerem vigilância. Todavia a precisão pode flutuar em função da largura do espectro de freqüências do sinal de interesse. O contexto de Posicionamento *Indoor*, mencionado no Capítulo 1 como motivador inicial deste trabalho, não impõe restrições à escolha da fonte sonora de interesse. No entanto requer estimativas precisas da posição da fonte para realizar medidas espaciais. A solução não é adequada para este tipo de problema, embora talvez um conjunto de 3 ou 4 Kinects pudesse atender à questão. Porém tal cenário foge ao escopo deste trabalho e torna a solução menos econômica e, conseqüentemente, menos interessante.

6.2 Trabalhos futuros

Com base nos resultados obtidos e na discussão levantada no capítulo anterior, são sugeridos como trabalhos futuros as seguintes atividades:

- Extensão do protótipo para incluir mais Kinects;
- Implementação de otimizações sobre o SRP-PHAT;
- Experimentos com os Kinects em diferentes posições e direções;
- Experimentos com ruído branco usando taxa de amostragem de 16 kHz (em lugar de 44,1 kHz);

- Experimentos com Kinects com a carcaça removida;
- Experimentos com voz humana.

O primeiro item, inclusão de mais Kinects ao protótipo, é encorajado pelos resultados dos experimentos, sendo prioritário dentre os demais. A implementação de otimizações beneficiaria muito o protótipo, visto que a busca 3D por todo o *grid* é onerosa e inviabiliza execuções em tempo real. Todavia sua importância depende da natureza da aplicação à qual se deseja submeter o programa. Experimentos adicionais envolvendo ruído branco, posições, direções e a carcaça dos aparelhos objetivam responder a questões que ficaram em aberto e possivelmente atingir maiores níveis de precisão com a solução. No caso da experimentação com voz humana é esperado o contrário: menores índices de precisão. No entanto, conhecer o comportamento da solução em relação à voz é relevante por causa de sua aplicabilidade ao popular contexto das videoconferências, além de possibilitar uma comparação mais direta com outros trabalhos.

REFERÊNCIAS

AGRESTI, A. **Categorical Data Analysis**. [S.l.]: Wiley, 2002. (Wiley Series in Probability and Statistics - Applied Probability and Statistics Section Series).

ALAMEDA-PINEDA, X.; HORAUD, R.; MOURRAIN, B. The geometry of sound-source localization using non-coplanar microphone arrays. In: APPLICATIONS OF SIGNAL PROCESSING TO AUDIO AND ACOUSTICS (WASPAA), 2013 IEEE WORKSHOP ON, 2013. **Anais...** [S.l.: s.n.], 2013. p. 1–4.

ANZALONE, S. M.; IVALDI, S.; SIGAUD, O.; CHETOUANI, M. Multimodal People Engagement with iCub. In: BIOLOGICALLY INSPIRED COGNITIVE ARCHITECTURES 2012, 2013. **Anais...** [S.l.: s.n.], 2013. p. 59–64.

BENESTY, J. Adaptive eigenvalue decomposition algorithm for passive acoustic source localization. **Journal of the Acoustical Society of America**, [S.l.], v. 107, n. 1, p. 384–391, Jan. 2000.

BERGER, M.; SILVERMAN, H. Microphone array optimization by stochastic region contraction. **Signal Processing, IEEE Transactions on**, [S.l.], v. 39, n. 11, p. 2377–2386, 1991.

BRANDSTEIN, M.; ADCOCK, J.; SILVERMAN, H. A closed-form location estimator for use with room environment microphone arrays. **Speech and Audio Processing, IEEE Transactions on**, [S.l.], v. 5, n. 1, p. 45, jan 1997.

BRANDSTEIN, M. S. **A Framework for Speech Source Localization Using Sensor Arrays**. 1995. 181 p. Tese (Doutorado) — Brown University, 1995.

BRANDSTEIN, M. S.; ADCOCK, J. E.; SILVERMAN, H. F. **A Practical Time-Delay Estimator for Localizing Speech Sources with a Microphone Array**. 1995.

BRANDSTEIN, M.; SILVERMAN, H. A robust method for speech signal time-delay estimation in reverberant rooms. In: ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, 1997. ICASSP-97., 1997 IEEE INTERNATIONAL CONFERENCE ON, 1997. **Anais...** [S.l.: s.n.], 1997. v. 1, p. 375–378 vol.1.

CARTER, G. C.; NUTTALL, A. H.; CABLE, P. The smoothed coherence transform. **Proceedings of the IEEE**, [S.l.], v. 61, n. 10, p. 1497–1498, 1973.

ClearOne. **Products Ceiling Microphone Array**. Disponível em: <http://www.clearone.com/products_ceiling_microphone_array>. Acesso em: 02 abr. 2013.

COBOS, M.; MARTI, A.; LOPEZ, J. J. A Modified SRP-PHAT Functional for Robust Real-Time Sound Source Localization With Scalable Spatial Sampling. **Signal Processing Letters, IEEE**, [S.l.], v. 18, n. 1, p. 71–74, 2011.

DIBIASE, J. H. **A High-Accuracy, Low-Latency Technique for Talker Localization in Reverberant Environments Using Microphone Arrays**. 2000. 122 p. Tese (Doutorado) — Brown University, Providence, 2000.

Digia. **Qt**. Disponível em: <http://qt.digia.com/>. Acesso em: 10 out. 2013.

DO, H.; SILVERMAN, H. A Fast Microphone Array SRP-PHAT Source Location Implementation using Coarse-To-Fine Region Contraction(CFRC). In: APPLICATIONS OF SIGNAL PROCESSING TO AUDIO AND ACOUSTICS, 2007 IEEE WORKSHOP ON, 2007. **Anais...** [S.l.: s.n.], 2007. p. 295–298.

DO, H.; SILVERMAN, H. Stochastic particle filtering: a fast srp-phat single source localization algorithm. In: APPLICATIONS OF SIGNAL PROCESSING TO AUDIO AND ACOUSTICS, 2009. WASPAA '09. IEEE WORKSHOP ON, 2009. **Anais...** [S.l.: s.n.], 2009. p. 213–216.

DO, H.; SILVERMAN, H.; YU, Y. A Real-Time SRP-PHAT Source Location Implementation using Stochastic Region Contraction(SRC) on a Large-Aperture Microphone Array. In: ACOUSTICS, SPEECH AND SIGNAL PROCESSING, 2007. ICASSP 2007. IEEE INTERNATIONAL CONFERENCE ON, 2007. **Anais...** [S.l.: s.n.], 2007. v. 1, p. I–121–I–124.

DO, H. T. H. **REAL-TIME SRP-PHAT SOURCE LOCATION IMPLEMENTATIONS ON A LARGE-APERTURE MICROPHONE ARRAY**. 2010. 57 p. Dissertação (Mestrado) — Brown University, Providence, 2010.

DONOHUE, K. D.; HANNEMANN, J.; DIETZ, H. G. Performance of phase transform for detecting sound sources with microphone arrays in reverberant and noisy environments. **Signal Process.**, Amsterdam, The Netherlands, The Netherlands, v. 87, n. 7, p. 1677–1691, July 2007.

ECKART, C.; SHIPS. CONTRACT NOBSR-43356, U. S. N. D. B. of. **Optimal Rectifier Systems for the Detection of Steady Signals**. [S.l.]: Scripps Institution of Oceanography, 1952. (SIO reference).

GALATAS, G.; FERDOUS, S.; MAKEDON, F. Multi-modal Person Localization And Emergency Detection Using The Kinect. In: IJARAI) INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN ARTIFICIAL INTELLIGENCE, 2013. **Anais...** [S.l.: s.n.], 2013. v. 2, n. 1.

HAMON, B. V.; HANNAN, E. J. Spectral Estimation of Time Delay for Dispersive and Non-Dispersive Systems. **Journal of the Royal Statistical Society. Series C (Applied Statistics)**, [S.l.], v. 23, n. 2, p. pp. 134–142, 1974.

HWANG, D.; CHOI, J. TDOA map adaptation in sound source localization. In: EMERGING TECHNOLOGIES FACTORY AUTOMATION (ETFA), 2011 IEEE 16TH CONFERENCE ON, 2011. **Anais...** [S.l.: s.n.], 2011. p. 1–4.

IEEE Standards Association. **1003.1-2008 - IEEE Standard for Information Technology - Portable Operating System Interface (POSIX(R))**. Disponível em: <http://standards.ieee.org/findstds/standard/1003.1-2008.html>. Acesso em: 05 set. 2013.

JANA, A. **Kinect for Windows SDK Programming Guide**. [S.l.]: Packt Publishing, Limited, 2012. (Community experience distilled).

Khronos Group. **OpenCL**. Disponível em: <http://www.khronos.org/opencl>. Acesso em: 05 set. 2013.

Kinect Hacks. **Kinect Hacks**. Disponível em: <http://www.kinecthacks.com/>. Acesso em: 29 ago. 2013.

KNAPP, C.; CARTER, G. C. The generalized correlation method for estimation of time delay. **Acoustics, Speech and Signal Processing, IEEE Transactions on**, [S.l.], v. 24, n. 4, p. 320–327, 1976.

LEE, B.; KALKER, T. A Vectorized Method for Computationally Efficient SRP-PHAT Sound Source Localization. In: INTERNATIONAL WORKSHOP ON ACOUSTIC ECHO AND NOISE CONTROL (IWAENC 2010), 12., 2010. **Anais...** [S.l.: s.n.], 2010.

LIU, H.; DARABI, H.; BANERJEE, P.; LIU, J. Survey of Wireless Indoor Positioning Techniques and Systems. **Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on**, [S.l.], v. 37, n. 6, p. 1067–1080, 2007.

LOCH, C.; CORDINI, J. **Topografía contemporânea**: planimetria. [S.l.]: Universidade Federal de Santa Catarina, 1995. (Série didática).

MathWorks. **MATLAB The Language of Technical Computing**. Disponível em: <http://www.mathworks.com/products/matlab/>. Acesso em: 07 jan. 2014.

MATTEO FRIGO, S. G. J. **FFTW**. Disponível em: <http://letsmakerobots.com/node/31012>. Acesso em: 05 jun. 2013.

Microsoft. **INuiAudioBeam**::getbeam method. Disponível em: <http://msdn.microsoft.com/en-us/library/jj663734.aspx>. Acesso em: 17 mai. 2013.

Microsoft Corporation. **Kinect**. Disponível em: <http://www.xbox.com/pt-BR/Kinect/Home-new?xr=shellnav>. Acesso em: 02 abr. 2013.

Microsoft Corporation. **Microsoft Store Online**. Disponível em: <http://www.microsoftstore.com/store/msstore/en_US/pd/ThemeID.27509700/productID.246711100>. Acesso em: 02 abr. 2013.

MINOTTO, V. P. **Localização de fonte sonora em tempo real através de um arranjo de microfones**. 2010. 109 p. Trabalho de Conclusão de Curso (Bacharelado em Engenharia da Computação) — Curso de Engenharia da Computação, Universidade do Vale do Rio dos Sinos, São Leopoldo, 2010.

MONICO, J. **Posicionamento pelo GNSS**: descrição, fundamentos e aplicações. [S.l.]: Editora UNESP, 2007.

NAKADAI, K.; OKUNO, H. G.; TAKAHASHI, T.; NAKAMURA, K.; MIZUMOTO, T.; YOSHIDA, T.; OTSUKA, T.; INCE, G. Introduction to Open Source Robot Audition Software HARK. In: THE 29TH ANNUAL CONFERENCE OF THE ROBOTICS SOCIETY OF JAPAN, 2011. **Anais...** Robotics Society of Japan, 2011.

Polycom. **Polycom CX5000 HD Unified Conference Station for Microsoft**. Disponível em: <http://www.polycom.com/products-services/products-for-microsoft/lync-optimized/cx5000-unified-conference-station.html>. Acesso em: 010 out. 2013.

POURMOHAMMAD, A.; AHADI, S. Real Time High Accuracy 3-D PHAT-Based Sound Source Localization Using a Simple 4-Microphone Arrangement. **Systems Journal, IEEE**, [S.l.], v. 6, n. 3, p. 455–468, 2012.

PRIEMER, R. **Introductory signal processing**. [S.l.]: World Scientific Publishing Company, Incorporated, 1991. (Advanced series in electrical and computer engineering).

RUI, Y.; FLORENCIO, D. New direct approaches to robust sound source localization. In: MULTIMEDIA AND EXPO, 2003. ICME '03. PROCEEDINGS. 2003 INTERNATIONAL CONFERENCE ON, 2003. **Anais...** [S.l.: s.n.], 2003. v. 1, p. I–737–40 vol.1.

RUI, Y.; FLORENCIO, D. Time delay estimation in the presence of correlated noise and reverberation. In: ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, 2004. PROCEEDINGS. (ICASSP '04). IEEE INTERNATIONAL CONFERENCE ON, 2004. **Anais...** [S.l.: s.n.], 2004. v. 2, p. ii–133–6 vol.2.

SALVATI, D.; CANAZZA, S.; RODÀ, A. Sound spatialization control by means of acoustic source localization system. In: SOUND AND MUSIC COMPUTING CONFERENCE, 2011. **Proceedings...** [S.l.: s.n.], 2011. p. 284–289.

SILVEIRA JR., L. G. da; MINOTTO, V. P.; JUNG, C. R.; LEE, B. A GPU Implementation of the Srp-Phat Sound Source Localization Algorithm. In: INTERNATIONAL WORKSHOP ON ACOUSTIC ECHO AND NOISE CONTROL (IWAENC 2010), 12., 2010. **Anais...** [S.l.: s.n.], 2010.

SMITH, J.; ABEL, J. Closed-form least-squares source location estimation from range-difference measurements. **Acoustics, Speech and Signal Processing, IEEE Transactions on**, [S.l.], v. 35, n. 12, p. 1661–1669, 1987.

TASHEV, I. J. Audio for Kinect: pushing it to the limit. In: CREST SYMPOSIUM ON HUMAN-HARMONIZED INFORMATION TECHNOLOGY, 2012. **Anais...** [S.l.: s.n.], 2012.

TELLAKULA, A. K. **Acoustic Source Localization Using Time Delay Estimation**. 2007. 88 p. Dissertação (Mestrado) — Indian Institute of Science, Bangalore, 2007.

TóMASSON, H. **Speaker Localization and Identification**. 2012. 54 p. Dissertação (Mestrado) — Reykjavík University, Reykjavík, 2012.

WANG, H.; CHU, P. Voice source localization for automatic camera pointing system in videoconferencing. In: ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, 1997. ICASSP-97., 1997 IEEE INTERNATIONAL CONFERENCE ON, 1997. **Anais...** [S.l.: s.n.], 1997. v. 1, p. 187–190 vol.1.

ZHANG, C.; FLORENCIO, D.; ZHANG, Z. Why does PHAT work well in lownoise, reverberative environments? In: ACOUSTICS, SPEECH AND SIGNAL PROCESSING, 2008. ICASSP 2008. IEEE INTERNATIONAL CONFERENCE ON, 2008. **Anais...** [S.l.: s.n.], 2008. p. 2565–2568.

ZHANG, C.; ZHANG, Z.; FLORENCIO, D. Maximum Likelihood Sound Source Localization for Multiple Directional Microphones. In: ACOUSTICS, SPEECH AND SIGNAL PROCESSING, 2007. ICASSP 2007. IEEE INTERNATIONAL CONFERENCE ON, 2007. **Anais...** [S.l.: s.n.], 2007. v. 1, p. I–125–I–128.