

***NAVision* – Sistema de Visão Subaquática para Navegação e Montagem de Mosaicos em AUVs**

André P. Vargas¹, Carlos A. B. C. W. Madsen² & Silvia S. C. Botelho³

¹*Engenharia de Computação – FURG, Rio Grande, RS – ec6apv@furg.br*

²*Engenharia de Computação – FURG, Rio Grande, RS – ec6mad@furg.br*

³*Departamento de Física - FURG, Rio Grande, RS – silviacb@ee.furg.br*

RESUMO:

O presente trabalho trata da construção de um módulo de visão subaquática para AUVs. Por ter como principal tarefa a inspeção de ambientes subaquáticos esses veículos são dependentes de seus sistemas de visão, que deve prever alternância de luminosidade, presença de ruídos nas imagens e etc. Neste trabalho serão abordadas técnicas para inspeção de regiões de tamanho arbitrário, maiores que o campo de visão de veículo, com o objetivo final de executar um mapeamento subaquático, formando um mosaico das imagens capturadas pelo AUV. Para tanto serão discutidas técnicas de tratamento e correlação de imagens capturadas, visando a geração do mosaico. Para que a missão seja realizada com menor gasto de energia e tempo, também são estudadas técnicas de geração de trajetórias. Tais técnicas culminam na criação do *NAVision*, inserido no módulo de visão do robô *lambdaRI*.

PALAVRAS-CHAVE: AUV, mosaico, visão computacional, mapeamento subaquático, navegação autônoma.

1. INTRODUÇÃO

Por situar-se numa região intimamente ligada aos aspectos portuários e pesqueiros, a Fundação Universidade Federal do Rio Grande (FURG) é uma universidade voltada para o ecossistema costeiro e marinho. Assim surge como desafio o desenvolvimento de tecnologia oceanográfica ligada a exploração da flora, fauna e recursos subaquáticos. Nesta linha, o domínio da tecnologia de fabricação de veículos subaquáticos autônomos (AUV – *Autonomous Undersea Vehicles*) que desoneram o ser humano em tarefas inóspitas e algumas vezes inexecutáveis torna-se estratégico. Estes apresentam controle e fonte de alimentação embarcados, não necessitando desta forma, de operador (tanto embarcado como remoto) ou cabos de alimentação. No entanto devido a insalubridade associada a questão submersa, bem como a dificuldade de modelagem dos aspectos hidrodinâmicos relacionados, poucos e caros são as ferramentas e instrumentos existentes para monitoramento, inspeção e atuação em ambientes subaquáticos. Tais soluções precisam tratar de uma série multidisciplinar de aspectos. No início desta década algumas dezenas de veículos subaquáticos com autonomia se encontram no mercado acessíveis financeiramente a poucos setores, como por exemplo, indústrias petrolíferas e de extração de gás natural [1,2]. No entanto, o desenvolvimento atual da tecnologia eletroeletrônica acena para o desenvolvimento de sistemas autônomos embarcados, que encapsulados poderiam tornar factível a custo mais baixo a monitoração e a

realização de tarefas no ambiente submerso. Nesta linha, estudos recentes propõem arquiteturas embarcadas para controle e supervisão de AUVs [3,4,5,6,7]. O grupo de Robótica e Inteligência Artificial (LRIA-FURG) atualmente agrega pesquisadores de diferentes áreas que estão engajados no objetivo de, a longo prazo, desenvolver *knowhow* no que tange a construção de veículos subaquáticos com funcionalidades para diversos tipos de tarefas e para operação em diferentes ambientes subaquáticos. Recentemente, obteve-se os primeiros resultados relativos ao software e hardware necessários para proporcionar autonomia ao veículo. As implementações em hardware e software vêm sendo desenvolvidas de acordo com uma arquitetura modular, que engloba tanto a supervisão de sinais sensorizados visando à modelagem do ambiente e estado do veículo, bem como fornecem subsídios para o controle adequado de seus atuadores [2]. Esta arquitetura é instanciada e implementada no projeto preliminar de um veículo autônomo em miniatura, *lambdaRI*, para inspeção subaquática. O veículo proposto deverá navegar de forma autônoma, visando a inspeção de determinadas áreas submersas fornecida pelo usuário de maneira remota, dentro do tanque de testes instalado no LRIA-FURG.

Para realização da tarefa de monitoramento dos ambientes submersos, dois módulos são essenciais na arquitetura deste tipo de veículo: o sistema de tratamento das imagens capturadas e o de controle navegacional. Neste trabalho será descrito o desenvolvimento do módulo de tratamento de imagens e gerenciamento de navegação, tendo como objetivo principal mapear ambientes que, na maioria das vezes, são muito maiores que o campo de visão do veículo. A partir da concatenação de imagens consecutivas, tal módulo fornecerá mosaicos da área a ser inspecionada. Secundariamente, tais mosaicos gerados auxiliarão o módulo responsável pelo controle, repassando possíveis correções constatadas nas imagens capturadas em relação ao posicionamento dado pelo controle dinâmico. Além disso, no decorrer do processamento das imagens também serão geradas trajetórias de navegação a fim de realizar a exploração do ambiente da forma mais rápida e com maior economia de energia.

1.2. Domínio do Problema

É complexa a dinâmica entre AUVs e o ambiente submerso. Na proposta do grupo, o submergível apresenta 6 graus de liberdade onde devem ser levados em conta os efeitos hidrodinâmicos [8]. Entretanto a garantia de estabilidade do controle do AUV, conjugando translação, giro e variação de profundidade (z), é ainda uma questão em aberto, tais questões não fazem parte deste trabalho, cujo principal objetivo se concentra na tarefa de geração de mosaicos, obtidos através da justaposição adequada das sucessivas imagens capturadas pelo veículo. O movimento do veículo, do ponto de vista do sistema de visão, se limitará a translação em um plano onde o z é fixo. Neste caso o movimento de translação, pode ser obtido de uma forma simplista através de uma bússola, instalada entre a estrutura do AUV e sua câmera, garantindo assim que o raio de visão nunca irá rotacionar, como visto em [9]. No entanto a implementação de diretivas no sistema de visão que tratam rotações e mudanças em z , são amplamente discutida na bibliografia [10,11], a decisão de estabelecer o domínio do problema sem abordar tais questões deve-se ao fato de que atualmente o LRIA-FURG não possui sensores inerciais, como giroscópios e profundímetros, capazes retornarem ângulo de rotação, distância do veículo em relação ao fundo, dentre outras medidas. É importante frisar que o veículo *lambdaRI* se movimentará em todas as direções acima descritas, no entanto para

o módulo e visão computacional, a imagens chegarão como se o AUV apenas transladasse neste plano, ficando por conta de outro módulo do submersível tratar isso.

2. ROBÓTICA SUBAQUÁTICA PARA INSPEÇÃO

2.1. Robôs Submersíveis

O desenvolvimento de veículos submersíveis não é novo. Em 1775 foi construído o primeiro submarino conhecido como *Turtle*, esse era tripulado e sua submersão se dava por tanques. Depois disso vários veículos foram desenvolvidos. De acordo com [12] estes podem ser divididos em duas grandes categorias: tripulados (submarinos) e não tripulados UUV (*Unmanned Underwater Vehicle*). A última pode ser dividida em algumas subclasses: Os tipos mais simples são os equipamentos, sem manobrabilidade, presos a navios através de cabos, por onde essa estrutura recebe energia e transmite os sinais de seus sensores. Os ROVs (*Remote Operated Systems*), por sua vez, são manobrados por um operador remotamente, localizado na superfície, sendo sua comunicação e alimentação também se dá através de cabos. Finalmente tem-se os AUVs que possuem a fonte de alimentação e lógica de controle embarcado, sendo este, devido a fatores já relatados, o tipo escolhido para ser desenvolvido neste trabalho.

Na última década foram desenvolvidos aproximadamente 66 AUVs em 12 países diferentes, principalmente em instituições acadêmicas e militares, sendo sistemas não comerciais [13]. Mas os AUVs estão no início de uma curva na área comercial, que vem crescendo de forma exponencial[1]. Atualmente, veículos como *Hugin 3000* (Noruega), *Maridan 600* (Dinamarca), *AQUA EXPLORER 2* (Japão), *SAILARS* (Canadá) e *OSIRIS, Sea Oracle* e *CETUS II* (USA), são exemplos de AUVs comerciais [1].

Tabela 1 - AUVs acadêmicos recentemente desenvolvidos e seus ambientes de testes [9]

AUV	Instituição	Ambiente de Testes
Theseus	ISE	Oceano
Odyssey	MIT	Oceano
ABE	WHOI	Oceano
REMUS	WHOI	Oceano
Ocean Voyager 2	FAU USF	Oceano
Phoenix	NSP	Tanque de Teste/Oceano
OTTER	ARL at Stanford	Tanque de Teste

A seguir são descritos os seguintes módulos que um AUV característico possui.

2.1.1 Modelagem Dinâmica e Controle

A dinâmica de interação do fluido com a estrutura é muito complexa e altamente dependente da profundidade de trabalho. No caso de portar manipuladores ou cabos de alimentação a modelagem se torna ainda mais complexa, não havendo consenso sobre a abordagem mais adequada para o tratamento deste problema[2]. A lei de controle tem de ser robusta e capaz de tratar as perturbações, normais neste tipo de ambiente, possibilitando um controle eficaz e possível de ser executado em tempo real.

Apesar de ser vasta a bibliografia que trata de modelagem dinâmica de submarinos convencionais, esta não se aplica aos ROVs/AUVs, isso se justifica ao se analisar as características do movimento que estes dois tipos de submersíveis realizam. Em geral, os submarinos se movimentam para frente, enquanto que estes veículos exploram todas as direções em conjunto no seu movimento[8]. Recentemente, técnicas avançadas de controle têm sido desenvolvida para esses veículos, ajudando e incrementando a capacidade de seguir pontos de referência e trajetórias. ROVs precisam ser controlados em 6 graus de liberdade (*degrees of freedom* - dofs). Mesmo que o problema do controle seja cinematicamente similar ao controle de um corpo rígido livre no espaço, que apresenta 6 dofs, tendo sido largamente estudado na literatura, a presença dos efeitos hidrodinâmicos fazem com que o controle de um ROV seja muito mais difícil, sendo necessário o conhecimento das características específicas de um corpo movendo-se num fluido [14].

2.1.2 Comunicação

A utilização de comunicação acústica é ainda a mais viável. Entretanto investiga-se outras tecnologias como laser, tratamento de ruído em RF (*radio-frequency*), bem como transmissão de grande quantidade de dados via comunicação acústica e com baixa taxa de perda.

2.1.3 Gerenciamento de Energia

Buscam-se por soluções tecnológicas ligadas ao abastecimento de energia que permitam aos veículos o desenvolvimento de longas missões, seja através de baterias mais eficientes, sistemas fechados de alimentação, fontes recarregáveis de energia, bem como sistemas que tirem proveito do ambiente oceânico. Os modelos de baterias mais utilizados são as de chumbo-ácido em formato tubular(3ET205) e as de *sódio-sulfúrico* (*Na-S*) [15]. Os sistemas de alimentação fechados mais conhecidos são as Células de Combustível de Polímeros Sólidos e o Ciclo Fechado de Combustão a Diesel[15]. Além destas fontes de energia, despontam alternativas mais adaptadas ao meio como a combustão de alumínio com água do mar, apresentadas em recentes pesquisas[16].

2.1.4 Localização e Navegação

Todas as missões que envolvem ROV/AUV necessitam de métodos de posicionamento do veículo, de acordo com uma navegação pré-estabelecida ou uma que vise manter o submersível em perseguição sobre algum alvo móvel. Vários sensores podem ser usados em ambientes subaquáticos para prover ao veículo o conhecimento de sua posição como sonares, redes acústicas baseadas em transponders, sensores inerciais, GPS e câmeras subaquáticas [33,13,20,15,9]. Para missões de exploração próximas ao fundo do oceano câmeras

embarcadas aos veículos são as escolhas naturais, tendo um alta precisão e um custo baixo para a realização deste tipo de tarefa.

2.2. Sistema de Sensoriamento e Processamento de Imagens Subaquáticas

Os sistemas de sensoriamento e processamento de imagens no que diz respeito aos AUVs são empregados principalmente para a realização das seguintes tarefas: i. perseguição de objetos de interesse, onde o veículo deve manter sempre um dado objeto dentro de seu campo de visão. ii. Permanecer em estado estacionário, neste caso o submersível deve manter sua posição constante, dada por uma imagem de referência, corrigindo erros causados pelos efeitos hidrodinâmicos. iii. Finalmente a geração de mosaicos, que pode ser considerada a principal tarefa de um AUV.

Na perseguição de objetos de interesse as câmeras são direcionadas para um objeto de interesse. Pode-se tomar como uma aplicação o acompanhamento da migração de certos animais marinhos, tarefa esta que não pode ser executada com muita eficiência pelos sonares pela sensibilidade que muitos animais possuem, sendo mais viável a utilização de câmeras subaquáticas. As principais técnicas utilizadas podem ser vistas em [31].

No estado estacionário os robôs devem se manter sobre um objeto presente no fundo do mar. Por exemplo, a verificação visual de uma formação rochosa em uma particular localização no fundo do mar obrigará o veículo a permanecer sobre o ponto de interesse mantendo este ponto no centro do campo de visão de sua câmera [17,18,19]. O resultado é que o veículo nunca irá sair totalmente de cima do objeto, não correndo assim o risco de se perder. No entanto, a magnitude do movimento do veículo fica limitada ao campo de visão da imagem de referência. Assim, correntes marinhas muito fortes podem empurrar o veículo para uma posição onde o objeto de interesse não se encontre mais no seu campo de visão, não sendo assim possível que ele corrija sua posição[30,33].

A técnica de mapeamento de ambientes subaquáticos através de mosaico é uma das principais tarefas dada aos AUVs. Tal técnica demonstra-se necessária pelo fato de que, na maioria das vezes, a câmera a bordo do veículo não tem um raio de visão suficientemente grande para englobar as estruturas que normalmente se desejam mapear, por exemplo corais, navios naufragados, fundo oceânico, dentre outras.

O mosaico de um ambiente submerso é basicamente a justaposição de sucessivas imagens capturadas pela câmera embarcada. Para tanto, é necessário que haja uma estipulação precisa do posicionamento desta imagem adquirida em relação a sua antecessora [20]. A geração de um mosaico pode ser considerada uma extensão do *estado estacionário* do veículo, como visto anteriormente, nesse estado o veículo captura uma imagem de referência e a compara com as sucessivas imagens estipulando o quanto ele está fora de sua posição. Já no processo de geração de mosaico, cada imagem capturada atualmente é alinhada com sua imagem de referência, quando a imagem atual estiver próxima de uma das extremidades da imagem de referência, uma nova imagem de referência é capturada. Além disso, a posição da imagem capturada em relação a imagem de referência não interfere na movimentação do veículo que segue sua trajetória, não tendo a preocupação de se posicionar no centro da imagem de referência como ocorre no estado estacionário. Esta técnica de mapeamento tem a vantagem de possibilitar ao veículo movimentos de translação de tamanho arbitrário. No entanto, existe uma desvantagem que é o acúmulo de erro de uma imagem em relação as suas subseqüentes,

que pode ser percebido no final da geração do mosaico [9]. A visão subaquática e a geração de mosaicos serão discutidas mais detalhadamente neste trabalho.

2.3. Visão Computacional para Geração de Mosaicos e Auxílio à Navegação

A visão é um processo perceptivo orientado por objetos, cujo propósito é fator determinante para análise dos mecanismos de seu funcionamento. A abordagem utilizada pelo sistema visual frente aos estímulos recebidos se baseia na economia de esforço, atuando de forma a extrair prioritariamente as informações necessárias à execução da tarefa [21]. Em vista de que o objetivo final deste trabalho é a geração automática de mosaicos, serão apresentados no decorrer desta seção os conceitos de alguns mecanismos que se prestam para esta tarefa. Primeiramente serão discutidos preceitos básicos, como imagem digital e as etapas que constituem a visão computacional. Em seguida técnicas de verificação de deslocamentos e mosaicos.

2.3.1 Imagem Digital

Uma imagem é definida como sendo uma função de intensidade de luz, uma vez que as imagens são captadas da energia radiante do ambiente por sensores. Matematicamente, uma imagem é uma função bi-dimensional contínua, ou seja, pode ser descrita por $f(x,y)$, onde x e y são variáveis contínuas que denotam as coordenadas espaciais do sistema de referência da imagem captada.

Para ser adequada ao processamento computacional, uma imagem representada pela função $f(x,y)$ precisa ser digitalizada tanto espacialmente como em amplitude. O resultado é uma amostragem discreta, formando uma matriz onde cada elemento representa uma intensidade de cinza em uma coordenada da imagem. [11].

2.3.2 Etapas do Processo de Visão Computacional

A visão computacional pode ser dividida em várias etapas. Apesar dessas tarefas estarem interligadas, a descrição de cada uma separadamente pode ser útil para a compreensão do processo [21].

As etapas mostradas abaixo freqüentemente compõem processos associados à visão computacional. Entretanto em função do domínio a ser visualizado pode ser que alguns procedimentos citados não sejam obrigatórios, bem como outros não apresentados se façam necessários.

Aquisição:

Nesta etapa, as informações visuais do meio são convertidas para sinais elétricos através de dispositivos ou sensores ópticos. A qualidade da imagem obtida é de fundamental importância para as demais etapas do processo. Os estudos das técnicas de iluminação, amostragem e estereoscopia constituem as principais áreas de pesquisa neste campo [21, 11].

Pré-Processamento Para Realce da Imagem

O principal objetivo das técnicas de realce é processar uma imagem de modo que o resultado final seja mais adequado que a imagem original para uma aplicação específica, no caso seja mais fácil de correlacionar pontos de duas imagens capturada em seqüência. Realce de imagens inclui expansão de contraste, suavização, realce de bordas e pseudocoloração. Estas técnicas envolvem duas categorias principais: métodos que operam no domínio espacial e métodos que operam no domínio da freqüência. O *domínio espacial* refere-se ao próprio plano da imagem, e as técnicas nesta categoria são baseadas na manipulação direta dos *pixels* de uma imagem. As técnicas de processamento no *domínio da freqüência* se baseiam na modificação da transformada de Fourier de uma imagem [22,11]. Entre as principais técnicas utilizadas estão a suavização e o realce de bordas.

A suavização busca uma homogeneização dos *pixels* presentes nas diversas regiões das imagens, alterando *pixels* com níveis de cinza pouco semelhantes aos da vizinhança e que podem representar um ponto ruidoso. No domínio da freqüência, suavização é obtida através de filtros passa-baixa. No domínio espacial, através de ações realizadas dentro dos limites de uma máscara, que se desloca sobre toda a imagem efetuando operações lineares e não-lineares baseadas em informações de uma vizinhança 3x3 do pixel atual. O nível de cinza do *pixel* central da janela de imagem definida pela máscara é substituído por um valor que é função do método empregado e dos níveis de cinza da vizinhança definida pela janela. Estas técnicas geralmente incorporam características do ruído, o conhecimento a priori sobre bordas e propriedades do sistema visual humano para obter o efeito desejado. Técnicas de suavização têm como objetivos principais a remoção de ruído e a uniformização dos níveis de cinza dos *pixels* na imagem [11]. Essa técnica é útil para o processamento de imagens submersas [9]. A suavização porém causa um efeito colateral indesejado. Ao retirar o ruído, este filtro acaba por tirar um pouco da nitidez das bordas dos objetos.

O realce de bordas consiste em realçar principalmente as descontinuidades locais de iluminação de uma imagem. Essas regiões de contraste são os contornos das estruturas presentes na imagem.

Segmentação

O objetivo do processo de segmentação é particionar a imagem em regiões, as quais são as representações visuais das estruturas a serem visualizadas. A segmentação esconde um problema conceitual ao tratar de objetos, razão pela qual não se conhece solução geral para este paradigma [23]. Uma estrutura pode ser parte de uma região, ser constituído de regiões desconectadas ou possuir furos. A identificação de uma estrutura na imagem pressupõe consciência de quais pontos pertencem a ele e quais pertencem ao fundo ou a outras estruturas.

Quando deseja-se separar um grupo de elementos de um conjunto, pode-se utilizar duas estratégias: agrupar elementos que se assemelham ou separar elementos que diferem do grupo, segundo alguma característica [21]. De uma forma mais prática, pode-se dizer que as estruturas são separadas do fundo através dos pontos semelhantes ou do contraste de seu contorno com o fundo [23].

Além de separar as estruturas do fundo da imagem, é necessária uma forma de representação dessas estruturas. Uma boa representação deve considerar aspectos como eficiência de acesso e economia de memória, além de apresentar um formato que facilite as etapas subseqüentes do processo de Visão computacional, em especial a etapa de

reconhecimento. A representação das estruturas pode ser feita por matriz de pontos, vetor de características, por região ou por contorno [40,35,34,41].

Reconhecimento, Interpretação e Descrição

Nestas etapas o padrão de cada objetos é comparado com padrões já conhecidos a fim de se decidir a qual grupo ele pertence. O padrão torna-se um conjunto de características que são usadas conforme o objetivo da aplicação.

2.3.3 Verificação do Deslocamento

Como o escopo deste trabalho passa pela montagem de mapas visuais associados a frames sucessivamente capturados, o estudo de técnicas de verificação de deslocamento e correlação entre cenas (imagens) se faz importante para a criação de mosaicos. A correspondência entre imagens sucessivas fornecerá métricas de deslocamento capazes de permitir a montagem de mosaicos com considerado grau de exatidão.

A correspondência de imagens se baseia no fato de que duas imagens possuem partes em comum. As imagens subaquáticas possuem características peculiares, que devem ser levadas em consideração na hora de se escolher a melhor técnica. Um dos problemas é a presença de ruídos na imagens, vindos de pequenas partículas de areia e etc, os quais são chamados de "neve marinha" [9]. Além de saber lidar com tais ruídos, o sistema deve ser tolerante a diferenças de luminosidade. As imagens utilizadas nesse trabalho possuem bordas bem definidas e as estruturas e o fundo possuem textura, ou seja, não são homogêneas ou monótonas. As técnicas mais utilizadas são a correspondência de pontos, de intensidade e de textura.

A correspondência de pontos é utilizada em imagens em estéreo e em navegação robótica de dirigíveis[36,37]. O algoritmo de correspondência de pontos busca primeiramente pontos relevantes numa imagem. Para isso existe uma heurística pré-determinada, com o propósito de buscar pontos que possam ser facilmente encontrados na outra imagem. Tais pontos podem ser linhas retas, quinas, zonas de bom contraste, etc. O próximo passo é buscar na segunda imagem tais pontos. Pode-se então estimar o deslocamento do veículo através do deslocamento relativo dos pontos relevantes nas imagens sucessivas. Apesar desta abordagem ser bastante eficiente, ela necessita de ambientes com muitos pontos relevantes, o que não acontece no ambiente submarino [9]. Tal técnica é mais utilizada em ambientes artificiais, que contenham muitos pontos de contrastes, linhas, etc.

Na correspondência de intensidade uma região da primeira imagem é comparada com uma região de mesmo tamanho da segunda imagem. A comparação é feita pela intensidade de cada *pixel*, totalizando a comparação para toda a região examinada. Uma das técnicas mais utilizadas é a "soma dos quadrados das diferenças". A soma dos quadrados das diferenças é mostrada conforme a equação abaixo, $I_0(i,j)$ e $I_1(i,j)$ são as imagens a serem comparadas. As imagens têm dimensões $m \times n$ e (x,y) são a diferença de localização entre as imagens comparadas. Dessa forma é testada as imagens em cada (x,y) possível (de 0,0 até m,n). O (x,y) que tiver menor diferença (zero no melhor caso) é escolhido para a correspondência.[9]

$$SSD(\Delta x, \Delta y) \equiv \sum_{i=1}^m \sum_{j=1}^n [I_0(i, j) - I_1(i - \Delta x, j - \Delta y)]^2$$

Equação 1 – Correspondência de Intensidade

A correspondência de textura combina elementos da correspondência de pontos e da correspondência de intensidade. Ela consiste na execução de um *XOR* entre cada *pixel* das imagens que estão sendo comparadas, podendo devolver o número de *pixeis* que não casaram (erro). As imagens a serem comparadas passam por um processo de filtragem e binarização SloG, que será discutido no próximo capítulo.

2.3.4 Mosaicos

Seja o frame F a tupla (x,y,I) , onde x,y é a coordenada horizontal e vertical desse frame, o mosaico final de um dado objeto é montado a partir de n frames onde $M = \{F(0), F(1), \dots, F(n-1)\}$. Sendo que $F(0)$ é o primeiro frame capturado [9,24]. O mosaico tem por objetivo criar uma imagem de uma estrutura ou de um ambiente maior que o campo de visão do veículo, a partir de imagens menores ou iguais a este campo de visão [9,25,20]. Para a criação de um mosaico é necessário que se conheça as coordenadas dos frames em relação aos seus vizinhos, discutido na seção anterior, para que os este possa ser "montado" de forma eficaz.

3.NAVISION - UM MÓDULO PARA INSPEÇÃO E MAPEAMENTO SUBAQUÁTICO

O presente capítulo mostra a criação do *NAVision*, que é a implementação das técnicas escolhidas no capítulo anterior. Neste serão mostrados as adaptações que os algoritmos sofreram bem como a criação de novos detalhes teóricos, próprios da área de implementação.

1. Captura

Para que a imagem seja utilizada é necessário que esta seja capturada e disposta no formato adequado. Esta tarefa é feita pela classe *auv_camera*, que contém as chamadas às rotinas do driver.

A implementação desta classe foi feita com base em trabalhos já realizados para esta câmera e para o driver utilizado.

Foi criada uma segunda implementação desta classe, a qual não utiliza qualquer driver mas chama uma seqüência pré-gravada de imagens, bastando apenas informar o local em que estas imagens estão. Esta foi desenvolvida para que os testes pudessem ser realizados na ausência da câmera e em máquinas que não dispusessem de placas de captura. Para que o sistema pudesse obter as imagens de forma adequada, foi desenvolvido um módulo externo ao sistema que gera a partir de imagens em formatos conhecidos (gifs,jpeg) imagens no formato aceito pelo *NAVision*.

Independente da implementação da classe *auv_camera*, esta deve poder devolver um objeto *auv_imagem*, que será utilizado nas próximas etapas do programa. A classe

avv_imagem é uma matriz dinâmica de bytes, sendo cada byte uma representação de um *pixel* da imagem capturada.

3.2.Preprocessamento para realce da Imagem

Devido a facilidade de implementação e as características da aquisição da imagem, discutida nas seções a seguir, optou-se pelo método baseado no *domínio espacial*. Este método se baseiam nas operações de convolução, que podem ser vistas em [11].

O operador utilizado neste trabalho é o SloG, que é a junção do processo de binarização (*signum*) com o filtro LoG (Laplaciano do Gaussiano), também conhecido como operador *Marr-Hildreth* [9]. O filtro LoG é usado para remoção de ruídos da cena, bem como o realce de bordas.

Este filtro pode ser visto como uma cadeia de operadores que podem ser aplicados em seqüência. O primeiro deles é o Gaussiano, que tem por objetivo fazer com que cada *pixel* seja a média dele mesmo com seus vizinhos, minimizando assim os ruídos da cena.

O próximo filtro é o Laplaciano, que opera através da diferença de intensidade entre um ponto e seus vizinhos. Os pontos que apresentarem maior diferença de intensidade (contraste) são destacados, sendo estas as bordas da figura. A matriz abaixo apresenta a máscara de convolução adotada neste trabalho.

0	-1	0
-1	5	-1
0	-1	0

Finalmente a imagem é binarizada (operador *signum*) com base em um limiar estabelecido. Trabalhando com imagens binarizadas, a correlação de imagens e a segmentação tornam-se mais robustas e tolerantes à variações de luminosidade [9].

3.3.Segmentação e Geração da Trajetória

Para que o AUV possa navegar adequadamente, inspecionando todo o contorno da estrutura, o sistema de visão serve de guia para que este mantenha sempre a borda da estrutura no centro do campo de visão do veículo. Porém é necessário que primeiramente o veículo encontre uma estrutura que seja "interessante" de ser inspecionada.

A imagem já binarizada é segmentada através do algoritmo de Rosenfeld [26,21]. A representação de uma estrutura na imagem é feita por uma lista dos pontos que formam o contorno do objeto. Foi decidido que serão apenas consideradas estruturas grandes, que sejam maior que o campo de visão do veículo. Portanto, a primeira tarefa é rejeitar os objetos que caibam no campo de visão, que podem ser pequenas estruturas ou ruídos.

Foi adicionado ao algoritmo de Rosenfeld um novo critério de parada, que ocorre quando o contorno excede os limites do frame, o que é esperado ao se inspecionar esses tipos de objetos. Utilizando como referência os últimos pontos desse processo, o veículo responde de forma com que continue a percorrer o contorno nos próximos frames. O último ponto é chamado de "ponto de ruptura" (figura 1).

Para geração da trajetória o sistema divide a imagem em setores conforme figura 2. Dependendo de onde está dentro da imagem o ponto de ruptura, a direção de movimento é tomada.

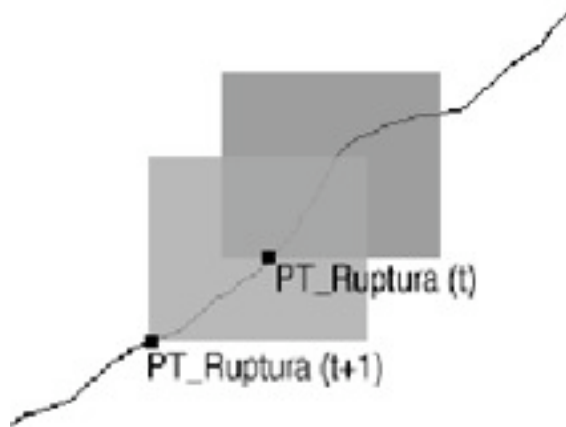


Figura 1 – Pontos de Ruptura dos Frames $F(t)$ e $F(t+1)$

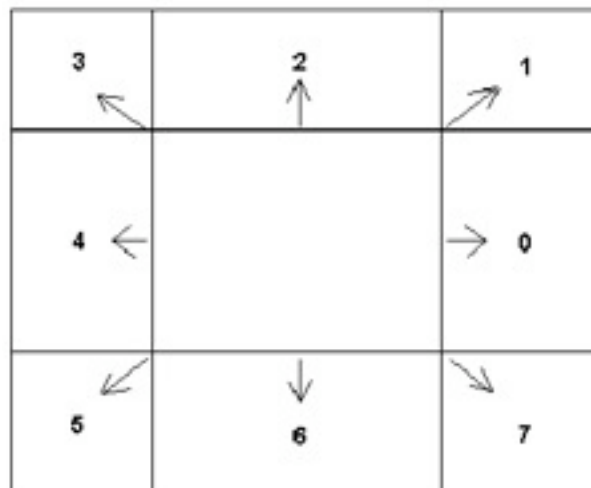


Figura 2 – Divisão dos Frames feito para trajetória.

3.4. Verificação do Deslocamento

Para tornar mais preciso deslocamento (e a localização) do veículo foi criado o frame de localização baseado em visão. Este sistema atua juntamente com um módulo de controle que pode dar uma estimativa do deslocamento (chamado neste trabalho de PID). Como a captura de frames é mais rápida que o deslocamento do veículo, as imagens capturadas na maioria das vezes se sobrepõem. Por isso, se todos os frames fossem enviados ao mosaico, haveriam

muitas imagens sobrepostas, desperdiçando memória. Por isso os frames de localização não são enviados ao mosaico, apenas os frames de mosaico.

Para otimização do cálculo da localização, foram criadas as janelas de textura e de busca.

3.4.1. Janela de Textura



Figura 3 – Geração da Janela de Textura (área demarcada) a partir da borda da estrutura.

O conceito de Janela de Textura foi originalmente criada neste projeto: sendo composta de uma sub-área gerada a partir da imagem binarizada do último frame. Tal imagem é utilizada para a correlação de imagens. O objetivo da criação de JT é utilizar, ao invés de todo o frame, apenas uma sub-área que possa ser relevante o suficiente para possibilitar a correspondência de textura e pequena o suficiente para otimizar seu processamento.

A Janela de Textura é criada a partir do entorno da borda da estrutura inspecionada no frame. Quando é terminada a pesquisa neste, é retornada a borda capturada. Os n últimos pontos pertencentes a essa borda farão parte de JT. Para isso verificamos entre esses pontos quais são os extremos valores de x e y , ou seja, qual o valor mais à esquerda, mais à direita, mais acima e mais abaixo desse trecho da borda (veja figura 3). Dessa maneira obtém-se as coordenadas dessa janela. É estipulado um valor mínimo para JT, para garantir que seja viável sua utilização na correspondência de textura. Se após o cálculo das dimensões a janela apresentar dimensões menores que o valor já arbitrado, será acrescido a ela a largura e altura mínimos.

Tal abordagem foi criada para que a sub-área escolhida seja a que proporcione melhor base para a correspondência de textura. A obtenção de JT a partir dos n últimos pontos faz com que esta fique disposta no limiar do frame. Como a trajetória será gerada tomando como base o ponto de ruptura (último ponto da borda) deste frame, isso garante que JT será encontrada nos próximos frame de localização. Além disso a região ao entorno da borda possui muitos pontos de contraste e geralmente não é monótona, o que facilita a correlação entre as imagens.

3.4.2. Janela de Busca

Cada frame de localização possui uma Janela de Busca, que faz efetivamente a correspondência de textura. Como esta técnica exige bastante tempo de processamento, JB é

gerada a partir de uma sub-área do frame, podendo ser menor ou igual ao frame. A sub-área de JB é criada a partir do PID e de seu erro. Quando é enviado ao JB a JT (janela de textura), existe uma estimativa, dado o seu deslocamento, de onde está a janela de textura dentro do frame de localização. Esse ponto é o centro da janela de busca. O sistema possui também uma estimativa do erro do controle, que é usada para criar a largura e altura de JB. Quanto maior o erro, maior será JB. No caso ideal (erro=0) JB terá o mesmo tamanho de JT, não necessitando fazer correlação entre as imagens. No pior caso JB terá o mesmo tamanho do frame.

3.4.3. Cálculo da Localização

A correlação é feita entre JB e JT para se ter o deslocamento da imagem em relação ao frame anterior. Esta pode ser executada em dois modos: o de melhor caso e o de tolerância. No modo de melhor caso JT é comparada em todas as localizações possíveis dentro de JB. Para cada posição é armazenado o número de pixels que não conseguiram combinar (*matching*). A posição de JT que possuir menor número (possivelmente zero) é a escolhida para a correlação. O segundo modo é o de tolerância. Nesse é dado ao programa um número máximo de pontos que não podem casar. O programa começa no centro da imagem (ponto estimado pelo PID) e começa a comparar as várias posições a partir desta. A ordem de busca segue uma espiral, começando do centro até as bordas de JB. A primeira posição que não extravasar a tolerância será a escolhida. O primeiro modo tem a vantagem de não necessitar de uma tolerância e de sempre achar uma posição, embora exija mais processamento, pois todas as posições são visitadas. O segundo caso possui a vantagem de não necessitar visitar todas as posições. Além disso, não é necessário também verificar todos os pontos de uma posição, pois quando se extrapola a tolerância, pode-se abandonar e seguir a espiral. Outra vantagem é a de começar a partir de uma estimativa, ou seja, se o controle estiver certo ou seu erro for pequeno, a correspondência rapidamente é encontrada. Sua desvantagem está na dificuldade de se arbitrar uma tolerância adequada.

Feito o algoritmo de correlação entre JT e JB, obtém-se a diferença de localização entre o fornecido pelo sistema de controle e o fornecido pelo sistema de visão. A partir destes é calculado o deslocamento entre este frame e o anterior. A localização do frame anterior é requerida e a partir deste é calculado o posicionamento em relação ao primeiro frame que forma o mosaico(*frame0*).

3.4. Geração do Mosaico

O objetivo final deste trabalho é a coleta das imagens de uma estrutura e gerar a partir delas um mosaico. Para isso, os objetos frames existentes são enviados à classe Mosaico, que os armazena e pode gerar a partir deles um objeto imagem, que será o mosaico para o usuário do sistema. Quando um frame é adicionado ao mosaico, este é armazenado no vetor de Frames do mosaico. Por ser um conjunto a ordenação dos diversos Frames não é levada em conta. O Frame enviado ao mosaico descarta sua imagem binarizada pelo SLoG pois essa não é mais interessante ao programa, ficando apenas com sua imagem original.

O Mosaico avalia o ponto de referência de cada frame recebido, para poder calcular as dimensões da Imagem final (em *pixels*). Cada frame possui um ponto de referência, que é a

distância em *pixels* do ponto x_0, y_0 (ponto mais acima e mais a esquerda do frame) deste frame ao ponto x_0, y_0 do *frame0*. Esse deslocamento pode ser negativo caso a imagem esteja acima ou a esquerda do *frame0*. Por exemplo, seja um frame J, disposto 100 *pixels* a direita e 200 *pixels* para baixo do *frame0*, este será expresso por J(100,200). O ponto usado para o cálculo do deslocamento é o x_0, y_0 (ponto mais acima e mais a esquerda do frame).

A cada inserção de um frame é calculado e armazenado os pontos PMin(Xmin,Ymin) e o PMax(XMax,YMax). XMin e XMax são respectivamente o maior deslocamento para esquerda e o maior deslocamento para direita, em relação ao *frame0* enquanto YMin e YMax são respectivamente o maior deslocamento para cima e para baixo, em relação ao *frame0*.

Seus valores são dados pelas seguintes equações:

$$XMin = \text{Min}(XMin, x_f)$$

$$YMin = \text{Min}(YMin, y_f)$$

$$XMax = \text{Max}(XMax, x_f)$$

$$YMax = \text{Max}(YMax, y_f)$$

onde Min retorna o menor valor entre o valor atual e o valor do frame atual e Max retorna o maior valor entre o valor atual e o valor do frame atual. A figura 4 mostra os extremos de um mosaico. Observe que x_f e y_f são os valores de x e y do frame inserido. Veja que para geração da imagem, é acrescido ao XMax o valor da largura do frame e YMax o valor da altura do frame.

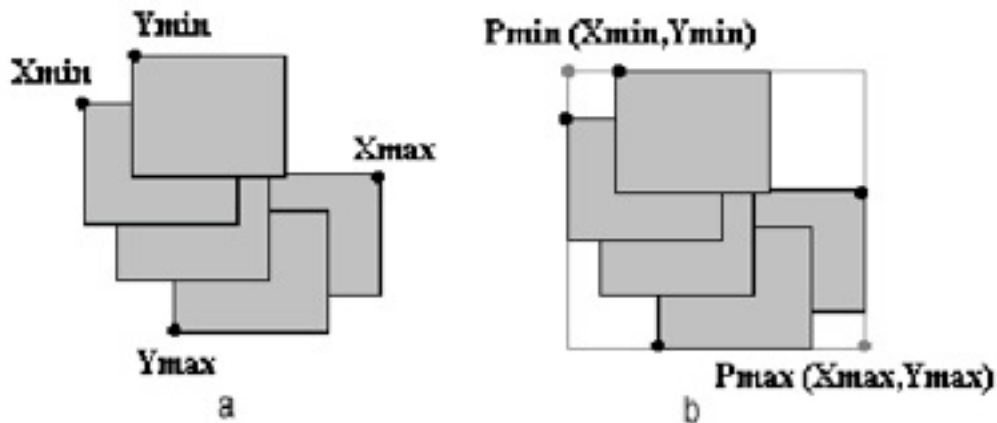


Figura 4 – a. Pontos extremos de um mosaico. b. Matriz demarcada de um mosaico e seus pontos extremos.

3.4.1 Geração da Imagem Final

Para gerar a imagem do mosaico que será vista pelo usuário o primeiro passo é verificar o tamanho da matriz (imagem) a ser criada. A diferença dos valores de x dos pontos extremos ($Xmax - XMin$) mostra sua largura, enquanto a diferença em y ($YMax - YMin$) mostra a altura desta imagem.

O próximo passo é colocar a imagem de cada frame no seu respectivo lugar na matriz do mosaico. Para isso subtrai-se de cada ponto de referência os valores do ponto de extremo de

menor valor. Dessa forma não se tem nenhuma coordenada negativa, mas continua-se mantendo o deslocamento original entre os frames. Na figura 4 é demarcado a matriz final gerada a partir dos frames do mosaico e seus pontos extremos.

Seja, por exemplo, um mosaico feito pelos frames A e B de largura 60 e altura 70, com deslocamentos em relação ao *frame0* de (10,-30) e (-40,20) respectivamente, além do *frame0* que é (0,0). Nesse caso tem-se como pontos de extremo (-40,-30) e (10,20).

Calcula-se primeiramente a altura e largura da imagem. Para ter-se valores positivos, subtrai-se o menor ponto extremo do maior ponto extremo e soma-se a este a largura e altura do frame.

$$(10,20) - (-40,-30) + (60,70) = (110,120)$$

Logo tem-se uma imagem de 110 pixels de largura por 120 pixels de altura. O próximo passo é calcular o lugar de "colagem" das imagens dos frames. Tem-se então os exemplos da tabela 2. Colada as imagens, um objeto imagem é retornado para o programa. A figura 5 mostra um exemplo de mosaico gerado pelo programa.

Tabela 2 – Exemplos de Frames e seus respectivos locais de colagem

Frame0	$(0,0) - (-40,-30) = (40,30)$
A	$(10,-30) - (-40,-30) = (50,0)$
B	$(-40,20) - (-40,-30) = (0,50)$

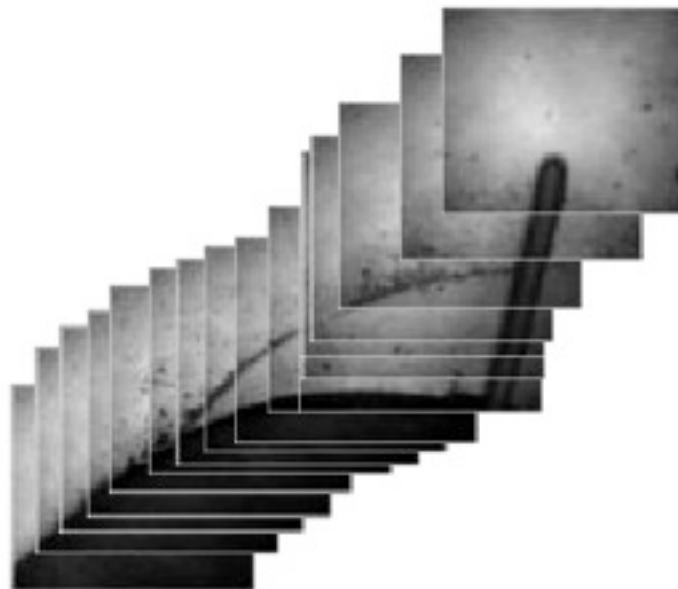


Figura 5 – Mosaico Gerado pelo Sistema

4.CONCLUSÃO

Esse trabalho apresentou uma discussão sobre aspectos relacionados a visão computacional subaquática e ao tratamento de imagens digitais, culminando no desenvolvimento do sistema de visão e auxílio a navegação NAVision.

O desenvolvimento desse tema se deve ao fato da FURG ser uma instituição voltada para o ecossistema costeiro e marinho, a qual tem grande interesse pelo estudo do desenvolvimento de tecnologia oceanográfica ligada a exploração da flora, fauna e recursos subaquáticos. Uma alternativa economicamente viável, para a o desenvolvimento dessa tecnologia, é a fabricação de AUVs. Pois a insalubridade associada ao ambiente submerso para o ser humano, encarece a maioria das ferramentas e instrumentos existentes para monitoramento, inspeção e atuação deste ambiente[2].

Mediante este desafio o grupo de Robótica e Inteligência Artificial da FURG, integrado por pesquisadores de diferentes áreas, esta engajado no objetivo de, a longo prazo, desenvolver *knowhow* no tocante a construção destes veículos com funcionalidades para diversos tipos de tarefas e para operação em diferentes ambientes subaquáticos. Devido aos recentes resultados obtidos relativos ao software e hardware necessários para proporcionar autonomia ao veículo foi construído um primeiro robô autônomo, inicialmente em miniatura, *lambdaRI* para inspeção subaquática[2].

Para tanto, temas como pré-processamento de imagens, correlação entre imagens e principalmente geração de mosaicos foram apresentados neste texto.

Alguns trabalhos ficam reservados para o futuro, principalmente a integração do sistema NAVision com os demais módulos do AUVs *lambdaRI*, de forma que possa se comprovar realmente a eficácia do sistema do ponto de vista de navegação do veículo (interação com os sistemas de controle). Outro fator a ser averiguado diz respeito a correção a serem realizadas no mosaico de forma a evitarem a propagação de erros de posicionamento cumulativos. Finalmente aponta-se também como continuação deste trabalho o tratamento de questões tempo dependentes.

5.BIBLIOGRAFIA

[1] WERNLI, R. *Auv's - a Technology Whose Times Has Come*. IEEE International Conference in Underwater Technologies 2002.

[2] BOTELHO S. *lambdaRI um Robô Subaquático Autonomo* SBAI Simpósio RBrasileiro de Automação Inteligente 2002.

[3] ROMAN, C., PIZARRO O., EUSTINE, R. AND SINGH H. *A New Autonomous underwater Vehicle for Imaging Research* IEEE International Conference in Underwater Technologies 2000.

[4] WICK, C. AND STILEWLL, D. *A miniature low-cost autonomous underwater vehicle*. IEEE Oceans 2001.

[5] WOOD, B., CONNELLY, P., BARSON, L. AND BOURNE, S. *An aoyonomous underwater vehicle as a student project*. IEEE International Conference in Autonomous Underwater Vehicles 1998.

- [6] CURTIS, T., PERRAUL, D., WILLAMS, C. AND BOSE, N. *C-scout: A general-purpose auv for systems research*. IEEE International Conference in Autonomous Underwater Vehicles 2000.
- [7] MILLER, D. *Design of a small, cheap uuv for under-ship inspection and salvage*. IEEE International Conference in Autonomous Underwater Vehicles 1996.
- [8] HSU, L., COSTA R. *Avaliação Experimental da Modelagem e Simulação Dinâmica de um Veículo Submarino de Operação Remota*. Revista Controle e Automação 2000.
- [9] FLEISHER S. *Bounded-Error Vision-Based Navigation of Autonomous Underwater Vehicle*. Phd Dissertation, Stanford University 2000.
- [10] DAN V., FEINER P., HOGHES C. *Computer Graphics: Principles and Practice*. 1998
- [11] GOZALES R.C., RICHARD E. *Digital Image Processing*. Addison-Wesley Publishing Company. New York 1993.
- [12] BLIDBERG, D AND ET AL. *Autonomous underwater vehicles: Current activities and research opportunities*. Robotics and Autonomous Systems 1991.
- [13] WERNLI, R. *AUV'S - The Maturity of the Technology*. OCEANS MST/IEEE, 1999.
- [14] ANTONELLI, G., CHIAVERITINI, S., SARKAR, N., WEST, M. *Adaptive Control of an Autonomous Underwater Vehicle. Experimental Results on ODIN*, CIRA 1999.
- [15] SCHUBAK G.E., SCOTT D.S. *A Techno-Economic Comparison of Power Systems For Autonomous Underwater Vehicles*. IEEE JOURNAL OF OCEANIC ENGINEERING 1995.
- [16] MILLER T.F., WALTER J.L., KIELY D.H. *A Next-Generation AUV Energy System Based On Aluminum-Seawater Combustion*. IEEE 2002.
- [17] NEGAHDARIPOUR, S. *Passive Optical Sensing For Near-Bottom Stationkeeping*. IEEE OCEANS 1990.
- [18] NEGAHDARIPOUR, S. *Undersea Optical Stationkeeping. Improved Methods*. Journal of Robotic Systems 1991.
- [19] YUH, J. *Control and Optical Sensing in Underwater Robotic Vehicles (URVs)*. IEEE OCEANS 1990.
- [20] XU, X., NEGAHDARIPOUR, S. *Vision-based Motion Sensing for Underwater Navigation and Mosaicing of Ocean Floor Images*. IEEE OCEANS 1997.
- [21] MACHADO A.M.C. *Metodologia para Reconhecimento de Padrões em Visão Computacional*. dissertação de mestrado submetida a Universidade Federal de Minas Gerais. dezembro de 1994.

- [22] VELOSO R.L. *Reconhecimento de Caracteres Numéricos Manuscritos*. dissertação de mestrado submetida a Coordenação dos Cursos de Pós-Graduação em Engenharia Elétrica da Universidade Federal da Paraíba. julho de 1998.
- [23] D. MARR. *Vision*. W. H. Freeman and Company, New York, 1982.
- [24] S. NEGAHDARIPOUR E X. XU. *3-D Motion and Depth Estimation from Sea-Floor Images for Mosaic-Based Station-keeping and Navigation of ROVs/AUVs and High-Resolution Sea-Floor Mapping*. IEEE 1998
- [25] PIO L. *Navegação Robótica Aérea Baseada em Visão - Requisitos de Processamento de Imagens para Projeto e Implementação*. III Workshop em Tratamento de Imagens. 2002
- [26] FREEMAN, H. *Computer processing of line drawing images*. Computer Surveys, March 1974.
- [27] GUO J., CHENG S., LIU T. *AUV Obstacle Avoidance and Navegation Using Image Sequences of A Sector Scanning Sonar*. IEEE 1998.
- [28] VICKERY K. *Acoustic Positioning Systems A Practical Overview Of Current Systems*. IEEE 1998.
- [29] HYLAN J.C., FOX S.R. *A Comparison of two Obstacle Avoidance path Planners for Autonomous Underwater Vehicles*. IEEE 1990.
- [30] WILLIAMS G.N., LAGACE G. E., WOODFIN A.J. *A collision Avoidance Controller for Autonomous Underwater Vehicles*. IEEE 1990
- [31] MARKS, R., ROCK, M. AND LEE, J. *Automatic object tracking for an unmanned underwater vehicle using real-time image filtering and correlation*. IEEE Systems, Man, and Cybernetics 1993.
- [32] JIN, L. *Real-Time Vision-Based Stationkeeping System For Underwater Robotics Applications*. IEEE OCEANS 1996.
- [33] MARKS L., WANG, H., LEE J. AND ROCK S. *Automatic visual station keeping of an underwater robot*. IEEE OCEANS 1994.
- [34] WANGENHEIM A.V. *Segmentação por Crescimento de Regiões*. disponível em <http://www.inf.ufsc.br/~visao/regioes.html> última visita 13/01/2004.
- [35] KENNETH R. CASTLEMAN. *Digital Image Processing*. Prentice Hall, 1979.
- [36] ALEXANDRE J. *Correspondência de pontos em imagens sem calibração*. <http://www.uel.br/pessoal/josealexandre/visao/match/index.html> .visitado 13/01/2004.
- [37] COELHO L. S. AND CAMPOS M. F. *Navegação de Dirigíveis Autônomos Baseados em Visão Computacional*. 1998 II Semana de Pós-Graduação em Ciência da Computação

[38] MILLER T.F., WALTER J.L., KIELY D.H. *A Next-Generation AUV Energy System Based On Aluminum-Seawater Combustion*. IEEE 2002.

[39] *Humminbird*. www.humminbird.com. última visita em 10/01/2004.

[40] THE KHOROS GROUPS. *Khoros User's Manual, Release 1.0.5*. Department of Electrical Engineering and Computer Engineering, University of New Mexico, Albuquerque, USA, 1993.

[41] TONGE, A. *Marlin, The UK Military UUV Programme - A Programme Overview*. UUV Conference, 2000.