

# CONTRIBUIÇÃO PARA O CONTROLE DE UM VEÍCULO SUBAQUÁTICO AUTÔNOMO COM SISTEMA DE PROCESSAMENTO DE IMAGEM EMBARCADO

Thiago Correia Haddad<sup>1</sup>; Dr. Wânderson de Oliveira Assis<sup>2</sup>

<sup>1</sup> Aluno de Iniciação Científica da Escola de Engenharia Mauá (EEM-CEUN-IMT);

<sup>2</sup> Professor da Escola de Engenharia Mauá (EEM-CEUN-IMT).

**RESUMO.** *Este projeto propõe a introdução de sistema de captura e processamento de imagem em um veículo subaquático autônomo (AUV – Autonomous Underwater Vehicle) que possa locomover-se dentro de um tanque, utilizando-se de tecnologias como uma câmera embarcada e um sistema de sensoramento. O projeto inclui a criação de um algoritmo que permita controlar a movimentação do veículo dentro de um tanque a partir do processamento das imagens capturadas em tempo real com o objetivo de efetuar a exploração e detecção de objetos pela cor e forma. Para introduzir o sistema de captura no AUV será necessário realizar modificações na estrutura mecânica, introduzindo um visor que permitirá obter imagens na parte inferior do veículo não descuidando da vedação do veículo.*

## Introdução

O estudo da robótica autônoma vem sendo efetuado há alguns anos no Instituto Mauá de Tecnologia – IMT principalmente no projeto de pesquisa “Desenvolvimento de Robôs Autônomos”. Este projeto foi implantado em 2003, inicialmente com o objetivo de incentivar os alunos de graduação a desenvolverem pesquisas acadêmicas relacionadas à mecatrônica e robótica e tendo como atrativo a participação em competições estudantis de robótica.

Uma área que vem sendo explorada recentemente na instituição refere-se a pesquisas utilizando UUVs (*Ummanned Underwater Vehicles*). Estes veículos podem ser operados remotamente ou ainda se movimentarem de forma autônoma para desenvolver ações como inspeção, perfuração, exploração de profundidades, enterramento de cabos submarinos, entre outros (Valavanis *et al.*, 1997). Veículos aquáticos operados remotamente (*Remotely Operated Vehicles*), geralmente são utilizados quando o ambiente de navegação na água é desconhecido ou as condições do ambiente não são propícias para uma missão tripulada (Centeno, 2008), (Bohm e Jensen, 1997). Para a navegação utilizam um cabo que permite a conexão entre o veículo e uma embarcação, na superfície. Veículos subaquáticos autônomos, os AUVs (*Autonomous Underwater Vehicles*), são veículos não tripulados, com fonte de energia própria e processamento interno que executam ações de forma autônoma a partir de informações obtidas por sensores (Batistella, 2011). Apresentam custo operacional mais reduzido, uma vez que o operador humano não é necessário. Além disso, como as ações são executadas de forma autônoma utilizando processamento embarcado, não é necessário efetuar conexão via cabo podendo por isto atingir distâncias significativas a partir de um navio de apoio ou plataforma. No entanto, seu projeto é relativamente mais complexo e mais limitado em termos de controle e capacidade de processamento, bem como, fica dependente da capacidade da fonte de energia embarcada. Neste sentido, muitas pesquisas estão sendo realizadas em todo o mundo tendo como ênfase a autonomia, navegação, detecção de objetos, fontes de energia e sistemas de informação para veículos autônomos subaquáticos (Gonzales and Woods, 2004), (Goheen e Jefferys, 1990), (Yuh, 2000), (Yoerger *et al.*, 2007).

Em 2011 e 2012 em projetos de iniciação foram desenvolvidos os AUV's (*Autonomous Underwater Vehicles*) ilustrados na Figura 1. Embora a estrutura mecânica dos dois protótipos tenha sido construída, o segundo protótipo foi escolhido por apresentar maior estanqueidade e simetria além de menor empuxo (Forni, 2011).



Figura 1 – Protótipos de AUV's desenvolvidos na Mauá

Neste protótipo foi incluído todo o sistema eletrônico incluindo sensores de distância para detecção de obstáculos e motopropulsores para produzir a movimentação em todas as direções. Um sistema microcontrolado foi incorporado para permitir o monitoramento dos sinais dos sensores bem como o controle do acionamento dos motopropulsores. Contudo, embora a construção completa do veículo tenha sido realizada com sucesso, a etapa de programação não havia sido finalizada.

Diante do exposto, este trabalho propõe a continuação do projeto de pesquisa com o veículo aquático com foco na introdução de um sistema de captura e processamento de imagens utilizando câmera CMUCam e a programação da mesma para fazer a identificação de objetos pela cor.

O sistema poderá ser utilizado em várias aplicações onde se deseja fazer a identificação por reconhecimento de cor, tais como ocorre em algumas competições estudantis de robótica como o SeaPerch (MIT, 2010) (SeaPerch, 2011).

## Material e Métodos

O trabalho visa a programação de um veículo subaquático autônomo que seja capaz de localizar objetos a partir de uma cores pré-definidas, além da adaptação do veículo para que todos os componentes necessários sejam introduzidos.

Nesta seção pretende-se demonstrar as características, funcionalidades e aplicações dos dispositivos e softwares utilizados, bem como os materiais utilizados na adaptação e programação do projeto.

### a) Softwares

#### Solidworks

O Solidworks baseia-se em computação paramétrica, criando formas tridimensionais a partir de formas geométricas elementares. No ambiente do programa, a criação de um sólido ou superfície tipicamente começa com a definição de topologia em um esboço 2D ou 3D. A topologia define a conectividade e certos relacionamentos geométricos entre vértices e curvas, no esboço e externos ao esboço.

Com tal recurso foi possível analisar os dimensionamentos do projeto anteriormente realizado e projetar um molde para o acoplamento do visor da câmera, o qual, além de permitir maior visibilidade fornece também um maior espaço no interior do AUV.

## Cmucam2.Gui

Para os testes iniciais e configurações da CMUcam2 utilizou-se o software CMUcam2.gui. Este programa fornece uma visão geral da CMUcam2. Ele é um programa em Java concebido para permitir a exploração de muitas das funções da CMUcam2 que normalmente seriam difíceis de se visualizar. Por exemplo, através deste software é possível analisar a variação de cor de um determinado objeto. As imagens geralmente são capturadas em cores, ou seja, o sensor de captura absorve os fótons e monta três matrizes de *pixels* ao invés de apenas uma. Cada uma dessas matrizes representa uma das cores primárias: o Vermelho (*Red*), o Verde (*Green*) e o Azul (*Blue*), o conjunto dessas matrizes forma o que ficou conhecido como uma imagem RGB, ou uma imagem em cores. Assim, o padrão RGB (*Red*, *Green* e *Blue*) apresenta valores entre 0 e 255 para cada unidade de cor medida em cada pixel da imagem. Tais dados são fundamentais para posterior programação, já que é necessário ter pré-definido os valores das cores que serão rastreadas através da função TC. Este é o principal comando e mais usado no projeto. É ele que realiza ou utilizando a CMUCam.

## Arduino.IDE

O Arduino IDE é uma aplicação multiplataforma escrita em Java derivada dos projetos *Processing* e *Wiring*. É esquematizado para introduzir a programação até mesmo a pessoas não familiarizadas com o desenvolvimento de software. Inclui um editor de código com recursos de realce de sintaxe, parênteses correspondentes e identificação automática, sendo capaz de compilar e carregar programas para a placa com um único clique. Com isso não há a necessidade de editar *Makefiles* ou rodar programas em ambientes de linha de comando.

Tendo uma biblioteca chamada "Wiring", ele possui a capacidade de programar em C/C<sup>++</sup>. Isto permite criar com facilidade muitas operações de entrada e saída, tendo que definir apenas duas funções no pedido para fazer um programa funcional.

### b) Hardware

## Câmera

Para monitorar as imagens e localizar objetos utilizou-se a CMUCam2 que já havia sido escolhida em (Forni, 2011). A CMUCam2, ilustrada na Figura 2, apresenta as seguintes características:

- câmera colorida com uma resolução suficiente para o reconhecimento de objetos tanto pela forma quanto pela cor;
- processador capaz de capturar a imagem da câmera e fazer o reconhecimento dos objetos;
- meio de transmissão do sistema de captura para o veículo autônomo de forma rápida e eficiente.

Além dos fatores citados a CMUcam2 foi escolhida devido ao seu baixo custo, por já possuir toda a estrutura projetada e também a forma de comunicação utilizando o padrão RS-232 ou TTL, presente na maioria dos microcontroladores existentes no mercado, portanto facilitando a elaboração da placa de controle do robô (Rowe, 2002).

As principais características presentes na CMUcam2 são:

- programável utilizando microcontrolador SX52;
- inclui câmera OV6620;
- apresenta resoluções de 88x143 e 176x255;
- conexão serial RS232 ou TTL com taxa de transferência de 1.200 bps a 115.200 bps;

- alimentação: 6 a 15V, 200 mA;
- suporta o controle de até 5 servomotores..

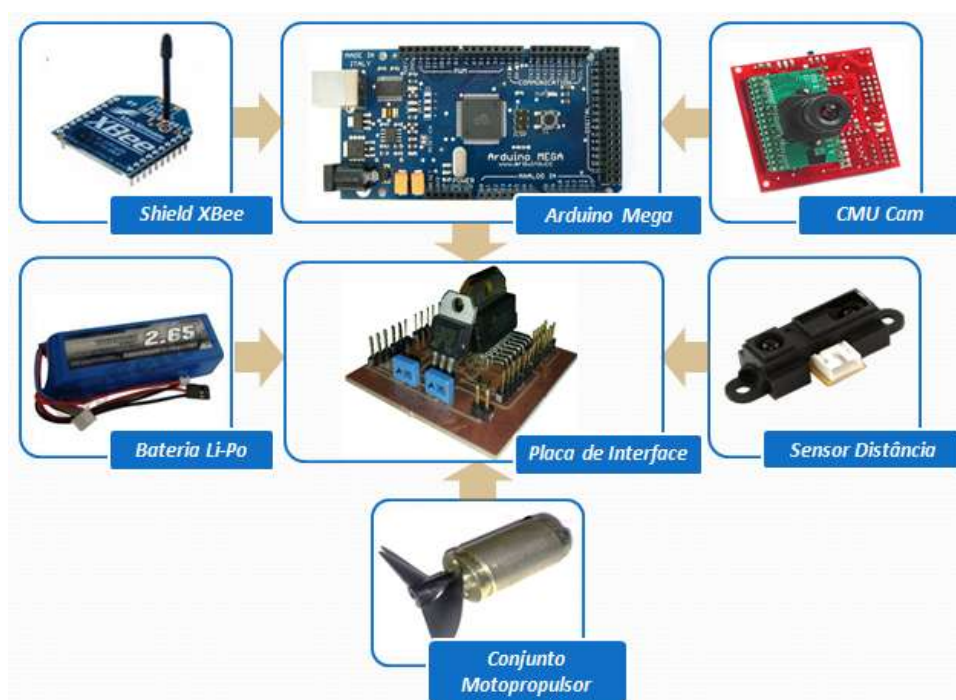


Figura 2 – Interação entre os dispositivos introduzidos no AUV

A Figura 3 ilustra o layout da placa CMUCam2 apresentando os pinos de entradas e saídas, *jumpers* de configurações e modo de comunicação serial.

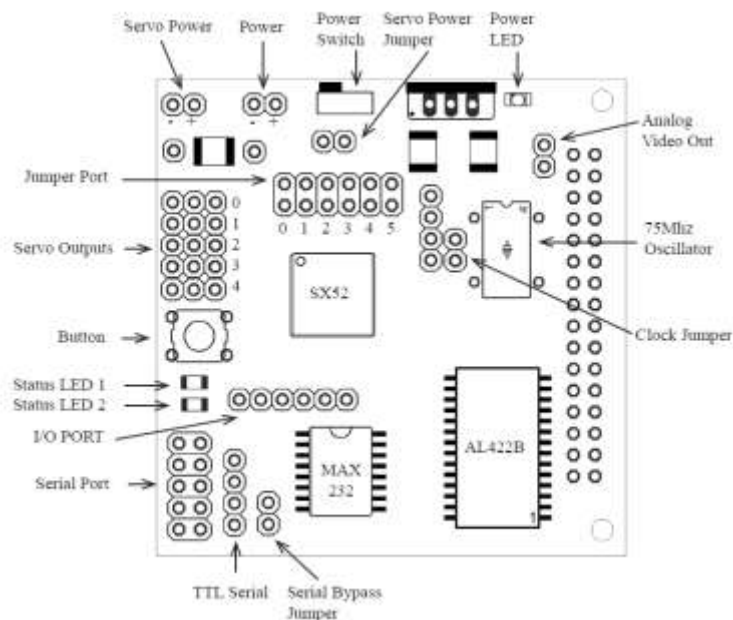


Figura 3 – Layout da CMUCam2. FONTE: GUEDES et al, 2009

A comunicação entre a CMUcam2 e o computador, para configuração de parâmetros e cores, é feita utilizando o padrão serial RS-232 sendo que a conexão deve ser realizada como mostrado na Figura 4.

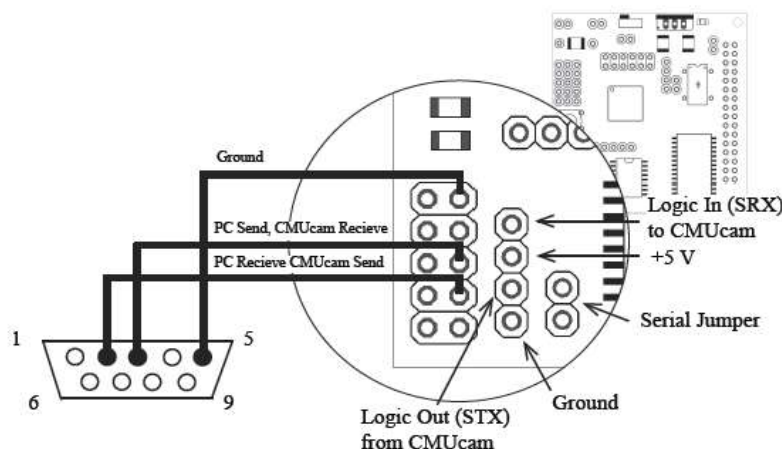


Figura 4 – Comunicação entre computador e CMUcam2 usando padrão RS232.  
FONTE: GUEDES et al, 2009

### Microcontrolador

Para o projeto, foi utilizado a placa Arduino<sup>TM</sup> Mega (Figura 2). O Arduino<sup>TM</sup> Mega é uma placa microcontroladora baseada no ATmega1280. Possui 54 pinos digitais de entrada e saída (dos quais 14 podem ser utilizados como saídas PWM), 16 entradas analógicas, 4 UARTs (portas seriais de hardware), um cristal com oscilação de 16 MHz, uma conexão USB, uma entrada para fonte externa, um ICSP Header, e um botão de *reset*. Contém todos os elementos necessários para suportar o microcontrolador, bastando conectar a placa a um computador através de um cabo USB, ou a uma fonte externa com um adaptador AC/DC ou bateria para iniciar. O Mega é compatível com a maioria dos *shields* desenvolvidos para Arduino<sup>TM</sup> Duemilanove ou Diecimila (outros modelos de placas microcontroladoras).

A escolha deste microcontrolador baseou-se no fato de haver a necessidade de utilizar duas vias de comunicação serial (uma via para a CMUcam2 e outra para comunicação com módulo *wireless* XBee<sup>TM</sup> e computador).

### Outros dispositivos

O AUV integra vários outros componentes em sua estrutura, tais como: sensor de distância, motores, bateria, LEDs, etc., os quais não são relevantes nessa etapa do projeto.

#### c) Métodos e Desenvolvimento

Para desenvolver a programação da CMUcam primeiro foi necessário entender o funcionamento do dispositivo e realizar testes usando o software CMUcam2.Guide para realizar o rastreamento de cores, onde o máximo desempenho é obtido com cores de alto contraste, exemplificado na Figura 5. O rastreamento da câmera, então, possibilita a análise da imagem e extração das informações solicitadas referentes a uma cor específica.

A captura é feita através do sensor de imagem CMOS. Após a luz passar pela lente da câmera e pelo sensor, são geradas 3 tensões proporcionais à própria luz (canais): vermelho, azul e verde. A tensão é convertida para um número de 8 bits, que define a cor.

Para especificar a cor que se deseja rastrear, é necessário definir valores mínimos e máximos para cada canal, através do *software* de comunicação CMUcam2.Gui (Figura 6). A

união dos limites para vermelho, azul e verde (6 valores) determina a cor desejada. Logo em seguida, a câmera começa a processar a imagem pelo canto superior esquerdo, varrendo todos os *pixels* da imagem naquele determinado quadro, buscando parâmetros dentro dos limites especificados. Os limites máximos e mínimos para cada canal são fornecidos pelo *software*, possibilitando a utilização das funções embarcadas para cada aplicação, nesse caso de rastreamento das cores específicas.



Figura 5 – Exemplo de reconhecimento de cor usando software CMUCam2.Gui.  
FONTE: GUEDES et al, 2009

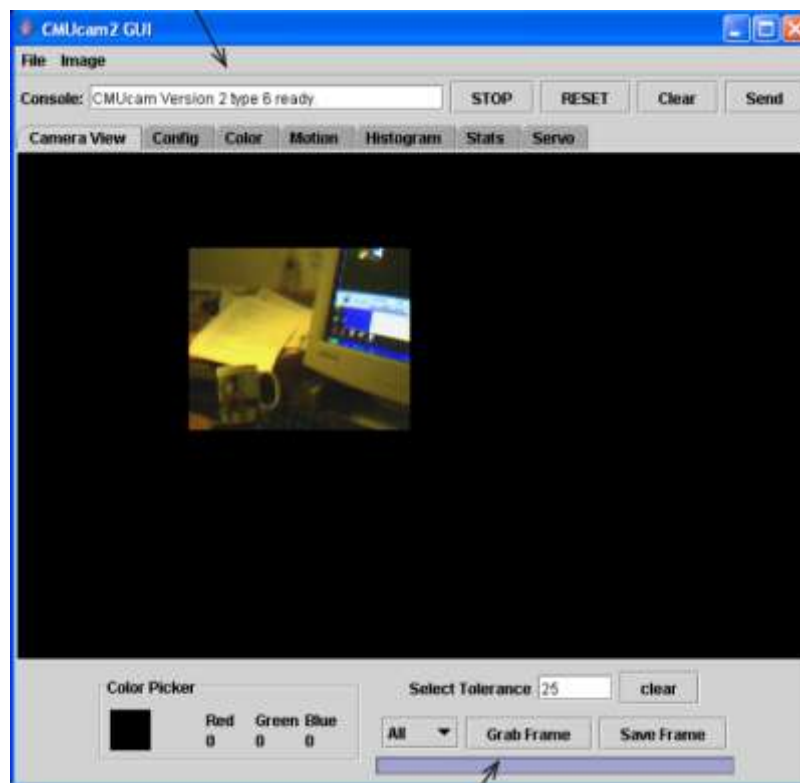


Figura 6 – Exemplo de captura de imagem no software CMUCam2.Gui.  
FONTE: GUEDES et al, 2009

Para determinar os limites dos canais RGB, é necessária a comunicação com o *software*. Nesta etapa, define-se uma tolerância para os valores de RGB de tal forma que o software irá buscar pixels que têm valores definidos dentro da faixa de tolerância especificada. Aciona-se o botão “*Grab Frame*” para a câmera gerar a imagem no *software*. Clicando na cor desejada e navegando até a aba “*Color*”, percebem-se 6 valores: vermelho mínimo, vermelho máximo, verde mínimo, verde máximo, azul mínimo e azul máximo. Esses são os valores que serão enviados como parâmetros para a câmera realizar as funções determinadas.

Os valores definidos nesta etapa de testes com o software CMUCam2.Guide serão posteriormente utilizados na programação do microcontrolador.

No Arduino, a inicialização da câmera é realizada pelo envio de algum comando, sempre verificando o retorno da câmera informando se o comando foi aceito e entendido ou se o comando falhou.

Os itens a seguir descrevem todas as funções internas que foram usadas na programação da placa Arduino por meio do software Arduino.IDE.

Um exemplo de código para comunicação entre Arduino e CMUCam2, é apresentado a seguir.

```
//Procedimento para inicialização da CMUcam 2;

void Iniciar_CMUcam() {

  Serial.println("Inicializando CMUcam 2...");
  delay(500);
  Serial1.print("RS \r");      //Reseta CMUcam;
  delay(500);
  Serial1.print("CR 18 44 \r"); //Configura CMUcam para AutoExposure;
  delay(500);
  Serial1.print("RM 0 \r");    //Configura CMUcam para Raw Serial Mode;
  delay(500);
  Serial1.print("LO 0 \r");    //Desliga LED 0;
  delay(500);
  Serial1.print("L1 2 \r");    //Configura LED 1 p/ acender Led ao detectar cor;
  delay(500);
  Serial1.print("PM 1 \r");
  delay(500);
  Serial1.print("TC 99 152 171 234 16 41 \r"); //Inicia o rastreamento de cor RGB;
  delay(500);
  Serial.println("CMUcam 2 pronta.");
  delay(3000);
}
```

Figura 7 – Exemplo de código para inicialização e identificação com em aplicação com Arduino e CMUCam2

Note que no rastreamento de cores, os limites mínimos e máximos RGB são inseridos no formato: TC [VermelhoMín VermelhoMáx VerdeMín VerdeMáx AzulMín AzulMáx]. Após o envio desse comando a CMUCam2 retorna um pacote tipo “T” informando o centro de massa em x e em y, os limites geométricos, o número de *pixels* rastreados e o nível de confiabilidade. Dessa maneira, o formato de saída da função é:

T [mx my x1 y1 x2 y2 *pixels* confiabilidade].

No projeto, utilizou-se somente as primeiras duas informações do pacote “T”, o centro de massa em x e em y. Com esta informação, é possível determinar o a movimentação do alvo com relação ao submarino.



Foram utilizados dispositivos específicos (LED's, a própria CMUcam2, entre outros) para testar a capacidade e precisão da programação, sem que o AUV estivesse envolvido, para assim poder evitar possíveis problemas de programação.

## Resultados e Discussão

O algoritmo para reconhecimento de cores foi desenvolvido em Arduino permitindo o monitoramento das imagens capturadas pela CMUcam2 sendo os resultados apresentados na tela do computador utilizando a interface serial.

Observou-se a eficiência do algoritmo no reconhecimento de cores, porém com algumas variações e imprecisões resultantes de fatores externos como iluminação, ângulo de captura da imagem, reflexão, interferência de outras cores que apresentam limiares próximos dos valores de busca definidos, etc. Assim, tornou-se muito importante realizar uma calibração para especificar corretamente os limites dos valores de RGB de forma a identificar com precisão a localização de objetos.

Dependendo do futuro uso da aplicação, caso haja necessário localizar determinados objetos é necessário analisar com cuidado o ambiente em que os mesmos estarão inseridos para evitar que esses interfiram no rastreamento da cor desejada.

Para permitir a introdução da câmera no veículo AUV, realizou-se o projeto em Solid Works de um molde visando construção de um visor de acrílico que permita a captura de imagens na parte inferior do AUV, ampliando seu campo de visão. A Figura 8 ilustra o molde projetado em Solid Works. O projeto foi concebido de forma a encaixar adequadamente na estrutura do AUV existente, mas sem interferir na disposição dos sensores já introduzidos no veículo.

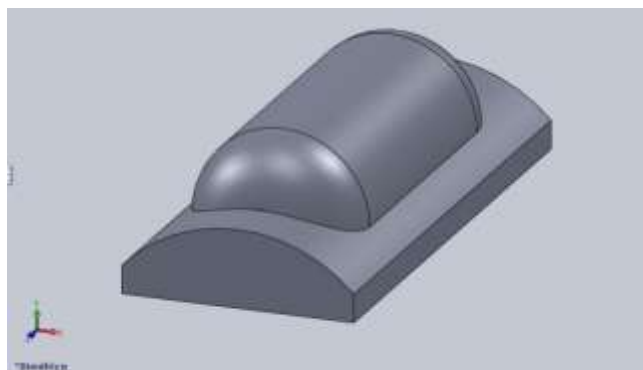


Figura 8 – Molde do domo de acrílico.

## Conclusões

Os estudos realizados mostraram a eficiência de localização de objetos a partir da especificação de faixa de valores de RGB, os quais podem ser realizados em computador, utilizando o software CmuCam2.gui bem como sistema microcontrolado. Contudo, observa-se claramente a variação da qualidade da imagem da câmera com a mudança de iluminação do ambiente, sendo necessário utilizar funções importantes para ajustes automáticos da iluminação tais como *Auto-gain* e *White Balance*. A etapa de programação do microcontrolador denotou muitos testes com funções diferentes para analisar o padrão de envio de informações da CmuCam2 tentando destacar o centro de massa da cor selecionada como variável a ser exibida. Através da função da CMUCam2 TC(*Tracking Color*) notou-se a necessidade de uma função capaz de armazenar e interpretar o pacote de dados enviado. Essa função deve discernir dentre os dados a localização de  $M_x$  e  $M_y$  para devolver a posição do



alvo. Assim tal valor pode ser usado para cálculo das potências dos propulsores, fazendo com que o veículo possa se mover até o destino.

O corpo do veículo previamente projetado não promovia espaço suficiente para a câmera, sendo necessário projetar um visor para que o campo de visão do AUV fosse ampliado, garantindo a estes maiores possibilidades de tipos de rastreamento.

Posteriormente sugere-se a introdução de um sistema *pan-tilt* para permitir a movimentação da câmera dentro do veículo, ampliando ainda mais o ângulo de visão.

Uma vez que o corpo está totalmente definido é possível em um próximo projeto iniciar a modelagem matemática do veículo, algo que depende diretamente das propriedades físicas do AUV e dos aspectos construtivos. Porém, um estudo detalhado do projeto construtivo do AUV poderá avaliar a viabilidade da realização e construção de um novo projeto mecânico, com um novo *design*, que proporcione um melhor desempenho dinâmico. Esse aspecto pode resultar na economia de energia necessária, por facilitar a movimentação subaquática, levando em conta as forças de arrasto, às quais o veículo é submetido.

## Referências Bibliográficas

- ACCARINI, N. C.; ASSIS, W. O.; COELHO, A. D.; **Desenvolvimento de Sistema de Captura e Processamento de Imagens para Robôs Autônomos**. XVIII Simpósio Internacional de Iniciação Científica da USP, São Paulo, SP, 2010.
- ARDUINO.CC.; **Arduino**. Disponível em: <http://arduino.cc/>. Acesso em 20 de julho 2012.
- BATISTELLA, S.; QUEIROZ, M. H.; **Sistemas de Controle de Missão para Autonomous Underwater Vehicles – AUVs**, 2012. Disponível em < [http://www.das.ufsc.br/~raffo/SeminariosRobotica\\_DAS/06.Seminario\\_24.11.2011\\_S.Battistella/8.SCM%20de%20AUVs.11.2011.pdf](http://www.das.ufsc.br/~raffo/SeminariosRobotica_DAS/06.Seminario_24.11.2011_S.Battistella/8.SCM%20de%20AUVs.11.2011.pdf) > Acesso em 15 de Abril de 2012.
- BOHM, R.; JENSEN, V. **“Build Your Own Underwater Robot and Other Wet Projects”**, Massachusetts Institute of Technology, 1997.
- CENTENO, M. L.; GOMES, S.C.P.; PEREIRA, I.M.; **Controle em Profundidade de um Veículo Subaquático do Tipo ROV**, 2012. Disponível em < <http://www.semengo.furg.br/2008/13.pdf> > Acesso em 15 de Abril de 2012.
- COELHO, A. D., ASSIS, W. O., SILVA, J. G., **“The Scientific Initiation as an Instrument of Training for Future Researchers”**, Proceedings of 9th IFIP World Conference on Computers in Education, Bento Gonçalves, RS, Brasil, 2009.
- DENIS, C. **“Seleção de Tomates para Processamento Industrial por Meio de Redes Neurais Aplicadas em Sistema de Visão Computacional”**, Dissertação de Mestrado em Engenharia de Processos Químicos e Bioquímicos, Instituto Mauá de Tecnologia, 2009.
- FORNI, C. C.; ASSIS, W. O.; COELHO, A. D. **“Projeto de um Veículo Subaquático Autônomo”**, CONIC – SEMESP 2011, Santos, SP, 2011.
- GUEDES, A. L.; FRANÇA, M. B. M.; FRANÇA, J. A.; KOYAMA, M. H.; CANTERI, M. Instituto de Robótica.; **CMUcam2 Vision Sensor User Guide**, Carnegie Mellon University, 2009.
- GOHEEN, K. R., JEFFERYS, E. R. **“Multivariable Self-Tuning Autopilots for Autonomous and Remotely Operated Underwater Vehicles”**, IEEE Journal of Oceanic Engineering, v. 15, p. 144-151, July, 1990.
- GONZALES, R. C., e WOODS, R. E., **“Processamento de Imagens Digitais”**, Editora Edgard Blücher Ltda., 2000.
- GUIMARÃES, F. A. **“Desenvolvimento de Robô Móvel Utilizado para a Exploração de Ambientes Hostis”**, Dissertação de Mestrado em Engenharia de Processos Químicos e Bioquímicos, Instituto Mauá de Tecnologia, 2006.

- LIMA, B.; BELLINI, D.; ASSIS, W. O. ; FORNI, C. C.; COELHO, A. D. **“Controle de Movimentação de um Veículo Subaquático Autônomo”**, II Encontro de Iniciação Científica da UFABC, São Caetano do Sul, SP, 2012.
- MIT **“SeaPerch Construction Manual”**, National Naval Responsibility for Naval Engineering (NNRNE), Massachusetts Institute of Technology, August, 2010. Disponível em: <http://www.jedc.org/forms/SeaPerch%20ROV%20Construction%20Manual%20-%20Northwest%20Program%20Options%20-%202010-04NW%20-%20DS081810.pdf>. Acesso em 20 de dezembro de 2013.
- MOURA JÚNIOR, J. R.; MORA, J. V.; ENOMURA, L. T.; ANDRADE NETO, R. **“AEGIR - Veículo Subaquático Autônomo”**, Trabalho de Conclusão do Curso de Engenharia de Controle e Automação do Instituto Mauá de Tecnologia, São Caetano do Sul, SP, 2012.
- ROWE, A.; ROSENBERG, C.; NOURBAKHS, I. **A Low Cost Embedded Color Vision System**. *Proceedings of IROS 2002 – IEEE / RSJ International Conference on Intelligent Robots and Systems*, Switzerland, September / October, 2002.
- SEAPERCH **“Sea Perch Program”**, MIT Sea Grant College Program, Massachusetts Institute of Technology, 2011. Disponível em: <http://seaperch.mit.edu/>. Acesso em 20 de dezembro de 2013.
- VALAVANIS, K. P., GRACANIN, D., MATIJASEVIC, M., KOLLURU, R., DEMETRIOU, G. A. **“Control Architectures for Autonomous Underwater Vehicles”**, IEEE Controls Systems, p. 48 – 64, December, 1997.
- YOERGER, D. R., JAKUBA, M., BRADLEY, A. M., BINGHAM, B. **“Techniques for Deep Sea Near Bottom Survey Using an Autonomous Underwater Vehicle”** Robotics Research, Springer Berlin / Heidelberg, V. 28, p. 416 – 429, May, 2007.
- YUH, J. **“Design and Control of Autonomous Underwater Robots: A Survey”**, Journal Autonomous Robots, Springer Netherlands, v. 8, n. 1, p. 7-24, January, 2000.