|  |  |  |
| --- | --- | --- |
|  | INSTITUTO TECNOLÓGICO DE AERONÁUTICA  CONSELHO NACIONAL DE DESENVOLVIMENTO CIENTÍFICO E TECNOLÓGICO - CNPq |  |

**PROGRAMA INSTITUCIONAL DE BOLSAS DE INICIAÇÃO CIENTÍFICA - PIBIC**

**Desenvolvimento de ferramentas de testes, depuração e calibração para robô humanoide**

**Igor Albuquerque Silva**

**RELATÓRIO FINAL DE ATIVIDADES**

Orientadores: Prof. Celso Massaki Hirata

M.C. Marcos Ricardo Omena de Albuquerque Máximo

**MARÇO / 2017**

|  |  |  |
| --- | --- | --- |
|  | INSTITUTO TECNOLÓGICO DE AERONÁUTICA  CONSELHO NACIONAL DE DESENVOLVIMENTO CIENTÍFICO E TECNOLÓGICO - CNPq |  |

**PROGRAMA INSTITUCIONAL DE BOLSAS DE INICIAÇÃO CIENTÍFICA - PIBIC**

**Relatório Final de Atividades**

**DESENVOLVIMENTO DE FERRAMENTAS DE TESTES, DEPURAÇÃO E CALIBRAÇÃO PARA ROBÔ HUMANOIDE**

São José dos Campos, \_\_ / \_\_ / \_\_\_\_\_

|  |  |
| --- | --- |
| Nome do aluno | Igor Albuquerque Silva |
| Assinatura do aluno |  |

|  |  |
| --- | --- |
| Nome do orientador | Celso Massaki Hirata |
| Assinatura do orientador |  |

|  |  |
| --- | --- |
| Nome do orientador | Marcos Ricardo Omena de Albuquerque Máximo |
| Assinatura do orientador |  |

INSTITUTO TECNOLÓGICO DE AERONÁUTICA

### PROGRAMA INSTITUCIONAL DE BOLSAS DE INICIAÇÃO CIENTÍFICA - PIBIC

Formulário de Aprovação de Relatório pelo Orientador

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| **Relatório:** | **X** | Rel. Parcial |  | Rel. Final |

|  |
| --- |
| **1- CONSIDERO O RELATÓRIO APROVADO COM BASE NOS SEGUINTES ASPECTOS** |
|  |
| **2-** **APRECIAÇÕES DO ORIENTADOR SOBRE O DESEMPENHO DO BOLSISTA NA EXECUÇÃO DO TRABALHO DE INICIAÇÃO CIENTÍFICA** |
|  |

**Local e data:**

**Assinatura dos Orientadores:**

**ÍNDICE**

|  |  |
| --- | --- |
| **1 Resumo do plano inicial** | **5** |
| **2 Resumo das atividades realizadas** | **6** |
| **3 Descrição do problema**  3.1. Ferramenta para a visão computacional  3.2. Ferramenta de calibração da tabela de cores  3.3. Ferramenta para a percepção e as ações | **7**  7  8  8 |
| **4 Resultados obtidos**  4.1. Ferramenta para a visão computacional  4.2. Ferramenta de calibração da tabela de cores | **9**  9  12 |
| **5 Conclusões** | **15** |
| **6 Agradecimentos** | **16** |
| **7 Bibliografia** | **17** |

1. **RESUMO DO PLANO INICIAL**

O plano inicial consistia em desenvolver ferramentas de visualização e de calibração de alguns algoritmos presentes no código do robô humanoide da ITAndroids que é a equipe de robótica do ITA. Os softwares de visualização envolveriam a visão computacional, as percepções e as reações do robô. Os parâmetros que necessitam de calibração envolvem a tabela de cores e a câmera do robô.

A finalidade do software de teste da visão computacional é visualizar os objetos que o robô identifica nas imagens fornecidas pela câmera, que envolvem componentes presentes no futebol de robôs como bola, traves e linhas do campo. As percepções do campo e as reações do robô necessitam de outro software, que seja capaz de representar o campo de futebol e a percepção do robô sobre ele, isto é, a localização dele mesmo, da bola e de outros robôs.

A calibração da câmera e dos modelos de cor do robô é uma ação necessária para a visão computacional e que depende de cada ambiente e de fatores como iluminação e tom de cor dos objetos. Por isso, é necessário o desenvolvimento de ferramentas que sejam rápidas e de fácil uso para a calibração destes componentes.

Para o desenvolvimento destes softwares, primeiramente ocorreria uma etapa de análise dos algoritmos de visão, percepção e de localização da equipe, com foco no entendimento das estruturas do código. Também seria realizado um estudo de desenvolvimento de softwares com interfaces gráficas e de ferramentas com finalidades similares àquelas que iriam ser desenvolvidas. Então ocorreria um planejamento do projeto e definição de cada software, para em seguida eles serem implementados e testados.

1. **RESUMO DAS ATIVIDADES REALIZADAS**

Primeiramente o código da equipe do humanoide foi introduzido ao aluno, com as ferramentas de testes já existentes, para ele se familiarizar ao problema.

Em seguida, pela aproximação da Competição Latino Americana e Brasileira de Robótica (CBR-LARC), uma ferramenta simples de visualização da visão computacional foi desenvolvida pelo aluno com auxílio da ITAndroids, pois era urgente que o robô fosse capaz de detectar os objetos no campo com sucesso na competição. Foi então necessário um período para depuração dos problemas da visão computacional do robô para a competição, que ocorreu em outubro de 2016.

Decidiu-se que seria utilizada a plataforma Qt para o desenvolvimento das interfaces gráficas do projeto pela sua facilidade de uso e pela sua fácil integração com o código já existente.

Em seguida foi desenvolvida a ferramenta de calibração da tabela de cores utilizando a plataforma Qt, que teve como base uma ferramenta já existente no código que deixou de ser utilizada por dois motivos: o longo tempo de execução associado a ela e a linguagem que foi escrita, Java, que não é a padrão da ITAndroids.

Após isso, foram analisadas algumas ferramentas gráficas utilizadas na robótica e foi decidido utilizar o framework rqt [1] para desenvolver outras ferramentas de telemetria por ele ser voltado para a robótica e utilizar a plataforma Qt.

Então, uma ferramenta completa de teste, calibração e depuração da visão computacional foi implementada por meio deste framework. Com isso, uma arquitetura simples para gerar novas ferramentas foi criada.

Por último, um controle remoto foi desenvolvido utilizando o mesmo framework, para facilitar os testes a serem feitos no robô.

1. **DESCRIÇÃO DO PROBLEMA**

A ITAndroids é uma equipe de alunos do ITA, supervisionada por um professor, que participa de diversas competições de robótica nacionais e internacionais. Uma das categorias em que a ITAndroids participa é a do robô humanoide, que consiste em desenvolver um time de robôs capazes de jogar futebol. Esta tarefa envolve uma série de desafios complexos que variam desde a construção do robô até a sua tomada de decisões.

Neste contexto, é fundamental a presença algoritmos robustos para a realização das diversas ações do robô, tornando essenciais as ferramentas de testes, calibração e depuração.

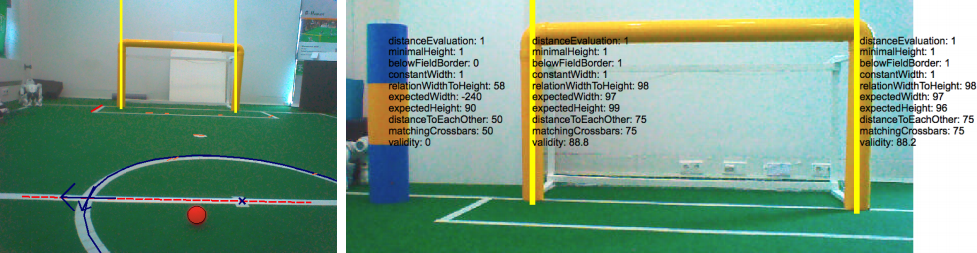
* 1. **Ferramenta para a visão computacional**

A complexidade envolvida no problema da detecção dos objetos do campo exige um software capaz de exibir de maneira clara a saída do algoritmo, isto é, as detecções em cada imagem.

O código da visão computacional passa por diversas etapas para classificar os elementos do campo. Uma delas consiste em ter várias possibilidades de elementos que passam por diversos filtros até que somente alguns são classificados como verdadeiros. Estes elementos possíveis ficam armazenados dentro de vetores internos da parte da visão do código do humanóide, enquanto os objetos verdadeiros são armazenados em estruturas chamadas “VisibleObjects”, que são passadas para as outras partes do código.

É de interesse da equipe que tanto os objetos que foram desclassificados quanto os verdadeiros sejam visualizados na ferramenta da visão de forma clara e que suas variáveis internas também sejam disponibilizadas, para fins de depuração e calibração do algoritmo. A Figura 1 mostra uma ferramenta do time de robótica B-Human, apresentada no documento de descrição do time para a competição mundial de robótica, Robocup, em 2013 [2]. Note que na imagem à direita as variáveis de cada trave estão escritas na foto, para fins de depuração. A ferramenta a ser desenvolvida pelo aluno deverá ser similar à contida na Figura 1.

Outra necessidade da ferramenta é a possibilidade de alteração das variáveis da detecção dos objetos em tempo real para fins de calibração, por meio de barras que deslizam interativamente. Por meio destas barras, deve ser possível selecionar a melhor combinação de variáveis para a detecção dos objetos.



**Figura 1**: Ferramenta de visualização da visão computacional do time B-Human. Imagens retiradas do documento de descrição do time para a Robocup em 2013 [2].

* 1. **Ferramenta de calibração da tabela de cores**

Tabela de cores é uma tabela com todos os valores de um espaço de cores e um código simples para identificar aquela cor. Por exemplo, no espaço RGB o código (0, 0, 0) é da cor preta. Uma tabela de cores RGB possui todas as triplas e suas respectivas cores. Para a ITAndroids, as cores que são relevantes são aquelas contidas no jogo de futebol, portanto todas as triplas RGB são classificadas em verde, branco, preto, azul, laranja, rosa e amarelo.

Uma tabela de cores é essencial para o início do algoritmo da visão computacional para o futebol de robôs. Ao receber uma imagem, o software deve classificar cada pixel em uma cor para, em seguida, agrupar os pixels de cores iguais e criar formas que podem ser classificadas em objetos.

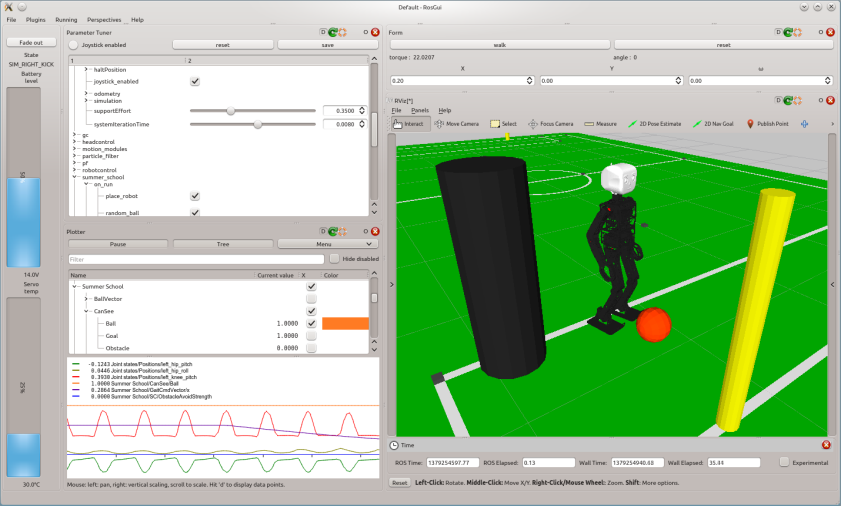
Todavia, dependendo do campo e da iluminação em cada momento, as cores sofrem alterações de tons e, portanto, a tabela de cores deve ser modificada para maximizar o desempenho da visão do robô. Por isso, uma ferramenta simples e rápida que gere uma tabela de cores é extremamente útil para a ITAndroids.

Outro requisito da ferramenta de calibração da tabela de cores é que o usuário selecione as cores que farão parte da classificação, pois esta ferramenta também será utilizada para outra categoria da ITAndroids, em que outras cores estarão presentes no contexto do problema.

* 1. **Ferramenta para as percepções e ações**

Um dos desafios contidos no futebol de robôs é determinar a posição do robô no campo após conhecer as posições dos objetos em relação a ele. Para testar a solução deste desafio é necessário ter uma visualização da percepção do robô sobre o seu ambiente. É a partir desta percepção que o robô decide o que fazer.

A Figura 2 mostra uma ferramenta deste tipo desenvolvida por membros do time NimbRO [3], que mostra uma modelagem tridimensional da percepção do robô.



**Figura 2**: Ferramenta desenvolvida por membros do time NimbRo [3], que mostra uma modelagem tridimensional da percepção do robô sobre o campo.

O software a ser desenvolvido possui três requisitos: um modelo tridimensional do robô utilizado pela ITAndroids (atualmente o ROBOTIS OP2 e em breve o Chape, que é um robô de desenvolvimento próprio da equipe), um modelo tridimensional do campo de futebol padrão da competição, incluindo os objetos e outros robôs de teste, e o conhecimento das ferramentas a serem utilizadas e do código da ITAndroids envolvendo a percepção e as ações.

* 1. **Controle remoto**

Devido às diversas situações possíveis em um jogo de futebol de robôs, é preciso uma forma de simular todas elas para o teste dos algoritmos desenvolvidos como o da visão, por exemplo. Assim, um controle remoto é fundamental para movimentar o robô pelo campo ao mesmo tempo que são testados estes agoritmos.

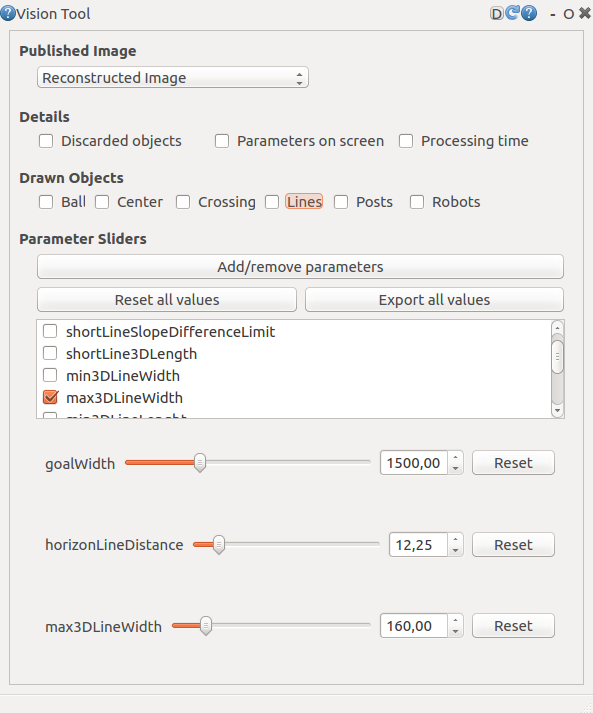
Além disto, um controle remoto permite visualizar facilmente os movimentos do robô em diferentes velocidades de caminhada para a análise deles.

1. **RESULTADOS OBTIDOS**

Todas as ferramentas com exceção da ferramenta para as percepções e as ações foram desenvolvidas com sucesso. Com elas, um sistema de telemetria do robô foi desenvolvido que alavanca o avanço da equipe.

* 1. **Ferramenta para a visão computacional**

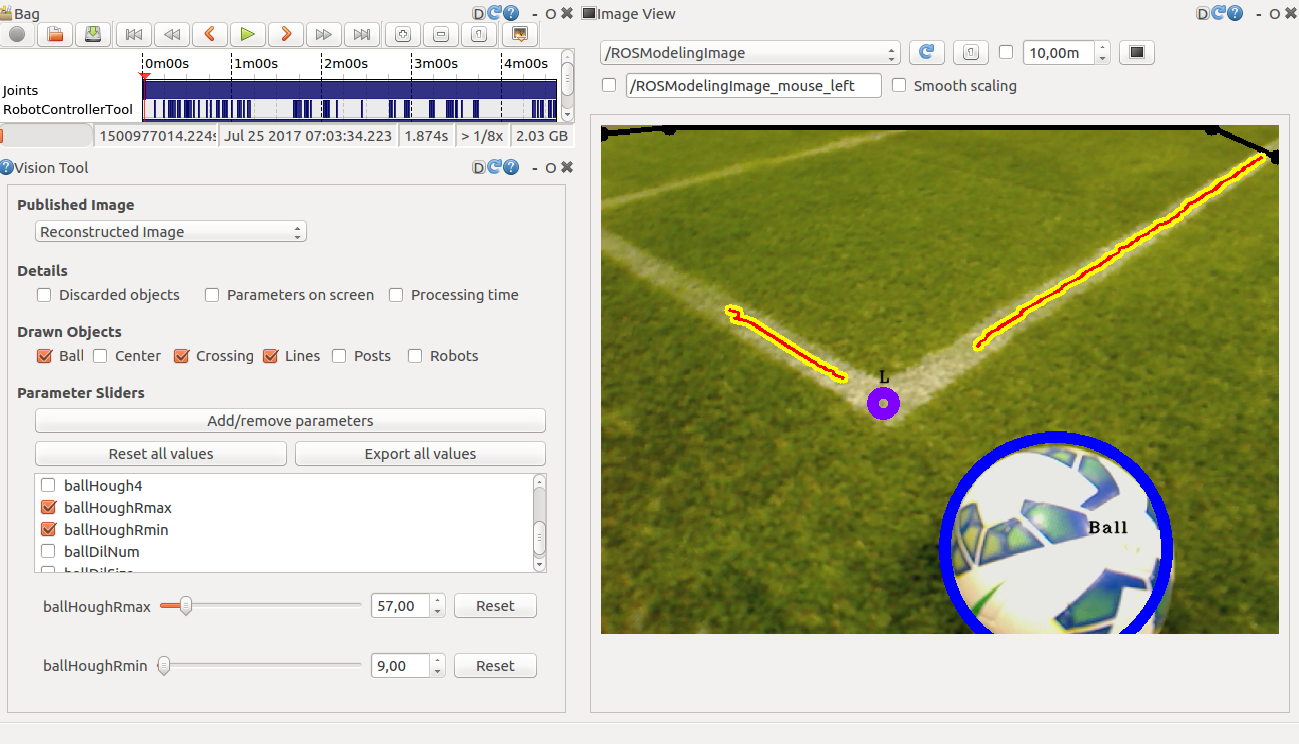
Utilizando o framwork rqt [1] e a biblioteca OpenCV [4], foi possível desenvolver uma interface gráfica que atende todos os requisitos da ferramenta para a visão. Esta ferramenta se comunica diretamente com o robô por meio do sistema ROS [1], sendo possível visualizar a saída do algoritmo da visão em tempo real, além de calibrá-lo. Assim, o usuário pode ver cada objeto detectado e também os não detectados. A Figura 3 mostra a interface da ferramenta.

**Figura 3:** Interface gráfica da ferramenta desenvolvida. Ela possui todas as funcionalidades necessárias para testar o algoritmo da visão computacional.

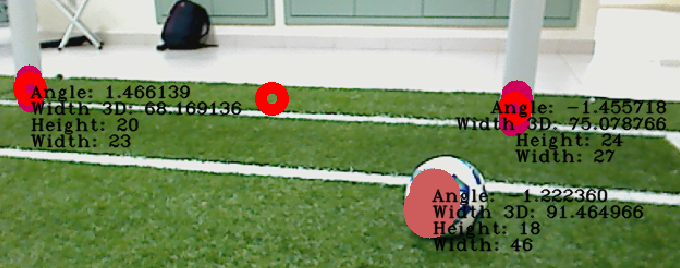
A interface foi separada em quatro blocos. O primeiro permite selecionar qual imagem será visualizada, isto é, a imagem original da câmera, a imagem segmentada em cores ou a imagem com os objetos detectados indicados. O segundo bloco permite algumas opções extras, isto é, visualizar os objetos descartados, visualizar os parâmetros internos do código para cada objeto e visualizar o tempo de processamento total do algoritmo da visão computacional. O terceiro bloco permite selecionar quais objetos serão visualizados. O último bloco permite calibrar os parâmetros do código por meio de barras iterativas.

Para gerar a imagem o sistema de publicação e subscrição do ROS [1] foi utilizado. Este sistema permite que o robô publique uma imagem pela rede para o computador do usuário visualizar esta imagem. Além disso, o usuário publica quais informações ele deseja na imagem a partir da ferramenta desenvolvida, e o robô altera a imagem de acordo com estas informações. O maior benefício deste sistema é a possibilidade de salvar estas imagens em um vídeo e poder publicá-las posteriormente, assim testando o algoritmo da visão sem a necessidade do robô.

A Figura 4 mostra a ferramenta em funcionamento no framework rqt [1].

**Figura 4:** Framework rqt [1] em funcionamento. Nele é possível unir a ferramenta desnvolvida com o visualizador de imagens na mesma janela.

Um dos requisitos da ferramenta implementado com sucesso foi a visualização dos parâmetros internos do código para cada objeto. A Figura 5 ilustra este requisito em funcionamento.

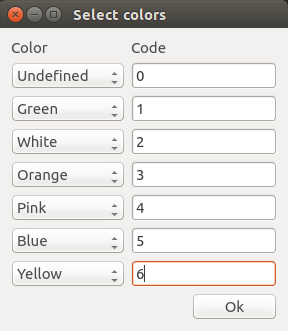
**Figura 5**: A ferramenta permite visualizar os parâmetros internos do código para cada objeto, seja ele descartado ou não. Aqui os parâmetros das traves estão sendo visualizados em comparação com uma trave descartada.

Os maiores desafios encontrado nesta etapa foram a compreensão do código da visão computacional da ITAndroids, o qual foi adaptado da liberação do código de 2012 da equipe Austin Villa [5] e a implementação do sistema de publicação e subscrição de imagens pelo ROS [1]. No primeiro foi necessário compreender as estruturas de dados nas quais os elementos da visão são armazenados para ser possível desenha-los na tela. No segundo foi necessário pensar na estrutura de códigos para ser possível ter uma comunicação entre o robô e o computador.

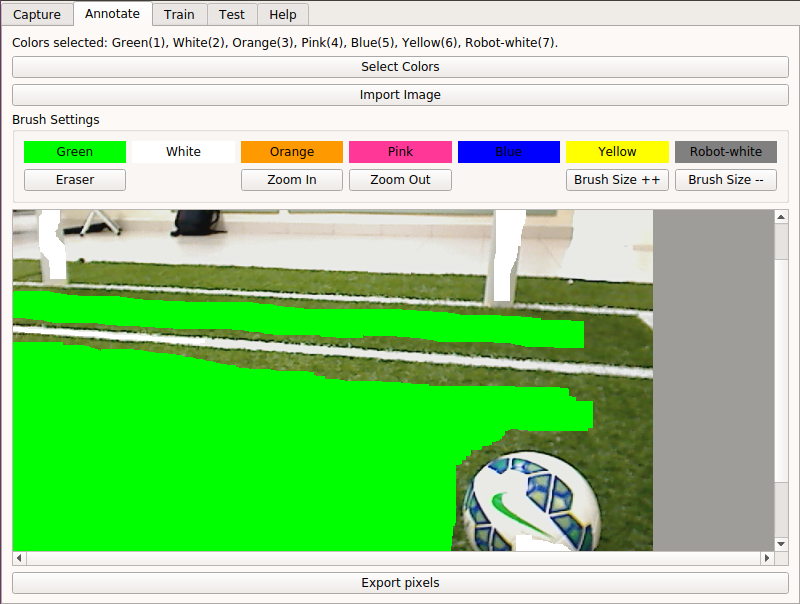
* 1. **Ferramenta de calibração da tabela de cores**

Com o uso da plataforma Qt, foi desenvolvida esta ferramenta. Ela utiliza o Matlab Pattern Recognition App [6], uma ferramenta do Matlab que possibilita o treinamento de redes neurais que se mostrou extremamente rápida, sendo aproximadamente 20 vezes mais rápido do que a ferramenta anterior da ITAndroids. A ferramenta possui cinco abas:

* Captura de câmera: possibilita tirar fotos utilizando a câmera do dispositivo para servir de amostra, caso a ferramenta seja utilizada diretamente do robô.
* Anotação: possibilita pintar as imagens que são amostras para fornecer a entrada para a rede neural. Esta aba possibilita ao usuário selecionar quais cores serão utilizadas, como representado na Figura 6. Ela está ilustrada na Figura 7.

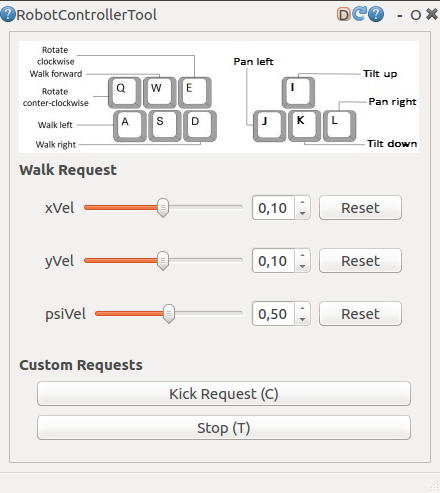


**Figura 6**: Janela que possibilita a escolha das cores presentes na tabela e seus respectivos códigos numéricos.

**Figura 7:** Aba do programa que possibilita ao usuário o fornecimento da entrada para a rede neural, pintando em algumas imagens as cores escolhidas.

* Treinamento: possibilita o acesso ao Matlab Pattern Recognition App [6], que treina a rede neural.
* Teste: possibilita o teste da rede neural criada, importando uma imagem do dispositivo e classificando seus pixels.
* Ajuda: aba descrevendo cada função da ferramenta e como utilizá-la.
  1. **Controle remoto**

A ferramenta de controle remoto do robô também foi desenvolvida utilizando o framework rqt [1] e o sistema de publicação e subscição do ROS [1]. A Figura 8 ilustra a interface gráfica desenvolvida.

**Figura 8:** Interface do controle remoto desenvolvido.

Nela, o computador do usuário publica a velocidade ou ação a ser adotada pelo robô e este executa a tarefa ao receber a mensagem. A direção da velocidade é controlada pelo teclado do computador, como está presente no diagrama presente na Figura 8. As ações a serem executadas, como chute e parada também podem ser controladas no teclado ou por meio de botões. O usuário também controla a velocidade dos movimentos por meio de barras presentes na ferramenta.

1. **CONCLUSÕES**

O framework rqt [1] é excelente para a criação rápida de ferramentas de interface com o usuário para a robótica. Estas ferramentas iniciais criadas agora serão modelo para a criação de mais ferramentas no futuro.

A ferramenta para a visão computacional já se mostrou extremamente útil para a equipe. Com suas funcionalidades, será possível maximizar o potencial do algoritmo de visão computacional da ITAndroids.

A ferramenta de calibração da tabela de cores se mostrou significativamente mais rápida do que aquela possuída anteriormente, além de ter uma interface gráfica mais completa. Com o auxílio desta ferramenta, é possível criar diversas tabelas de cores para cada ambiente sem muitas dificuldades.

O controle remoto agora permite controlar o robô com facilidade, permitindo o teste de outros algoritmos da equipe, como o da visão computacional e o da localização no campo.

1. **AGRADECIMENTOS**

Agradeço ao CNPq pela oportunidade e pelo financiamento do projeto.

Agradeço ao professor coorientador Marcos Máximo pelo constante apoio durante toda a execução do projeto, mostrando alternativas e apresentando ideias novas que foram fundamentais para o produto final.

Agradeço a toda a equipe da ITAndroids pelo suporte e acompanhamento do projeto, em especial ao Samuel Cerqueira pela contribuição às ferramentas desenvolvidas.

Agradeço ao professor orientador Celso Hirata pela oportunidade da realização do projeto.

1. **BIBLIOGRAFIA**

[1] QUIGLEY, MORGAN. et. al. ROS: an open-source Robot Operating System. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION 2009. Japão. 2009.

[2] ROFFER, T. et al. B-Human: Team Description Paper for RoboCup 2013. Bremen, Alemanha.

[3] ALLGEUER, P. et al. A ROS-based Software Framework for the NimbRo-OP Humanoide Open Platform. Atlanta, 2013.

[4] G. BRADSKI, Dr. Dobb’s Journal of Software Tools. 2000.

[5] [Samuel Barrett](http://www.cs.utexas.edu/~sbarrett), [Katie Genter](http://www.cs.utexas.edu/~katie), Yuchen He, [Todd Hester](http://www.cs.utexas.edu/~todd), [Piyush Khandelwal](http://www.cs.utexas.edu/~piyushk), Jacob Menashe, e [Peter Stone](http://www.cs.utexas.edu/~pstone). UT Austin Villa 2012: Standard Platform League World Champions. Em Xiaoping Chen, [Peter Stone](http://www.cs.utexas.edu/~pstone), Luis Enrique Sucar, e Tijn Van der Zant, editores, *RoboCup-2012: Robot Soccer World Cup XVI*, Lecture Notes in Artificial Intelligence, Springer Verlag, Berlin, 2013.

[6]MATLAB e Neural Network Toolbox, The MathWorks, Inc., Natick, Massachusetts, Estados Unidos.