

Desenvolvimento de sistema de visão e do módulo de estratégia de navegação para robôs móveis

Development of vision system and navigation strategy module for mobile robots

● Antonio Valerio Netto¹

RESUMO

O sistema de visão computacional utilizado pelos robôs autônomos diferencia-se, entre outros aspectos, pela localização da câmera e por consequência, os algoritmos utilizados para realizar sua navegação. No caso, da aplicação para futebol de robôs além da navegação autônoma, as tomadas de decisões em tempo real são gargalhos de processamento principalmente devido aos múltiplos rastreamentos de objetos. Neste contexto, o sistema de visão computacional deve ser extremamente otimizado e englobar desde a marcação e os rastreamentos dos objetos, a correção da distorção causada pelo conjunto câmera-lente, até a geração do correto posicionamento dos objetivos móveis no campo de jogo. Neste artigo é apresentado o desenvolvimento de um sistema de visão que permite automatizar o controle de distorção da imagem digital em tempo real e a inclusão de um módulo de estratégia de navegação para permitir, de forma automática, sua inserção para ser utilizado durante os jogos de futebol de robôs.

Palavras-chave: Controle de distorção da imagem. Futebol de robôs. Módulo de estratégia. Robô móvel. Imagem em tempo real.

¹ DT CNPq | avnetto@hotmail.com

Desenvolvimento de sistema de visão e do módulo de estratégia de navegação para robôs móveis

Development of vision system and navigation strategy module for mobile robots

ABSTRACT

The system of computer vision used by autonomous robots differs, among other aspects, by the location of the camera and consequently, the algorithms used to carry out its navigation. In the case of the robot soccer application beyond autonomous navigation, real-time decision making is a processing issue mainly due to multiple object traces. In this context, the computer vision system must be extremely optimized and encompass everything from marking and tracking of objects, correction of distortion caused by the camera-lens assembly, to generation of correct positioning of moving targets on the playing field. In this paper we present the development of a vision system that allows the automation of digital image distortion control in real time and the inclusion of a navigation strategy module to automatically allow its insertion to be used during games of robot soccer.

Keywords:Control of image distortion. Robots soccer. Strategy module. Mobile robot. Real time image.

1 Introdução

Este artigo tem por objetivo apresentar o desenvolvimento de um sistema de visão computacional que permite automatizar o controle de distorção da imagem digital em tempo real para realizar corretamente o tracking dos robôs e da bola durante um jogo de futebol de robôs. Além disso, descreve a criação de um módulo de estratégia de navegação para permitir, de forma automática, a inserção de uma estratégia durante a partida sem a necessidade de nenhum tipo de recompilação de código fonte do sistema computacional.

Inicialmente a preocupação maior foi dada ao estudo e a implementação do algoritmo de rastreamento dos objetos, de tal forma que fosse utilizada uma abordagem simples, com alto desempenho e que pudesse ser executada em tempo real. Desse modo, o módulo de rastreamento ficou responsável por realizar as atividades de reconhecimento e o acompanhamento da dinâmica dos objetos, a sua implementação foi incorporada a uma interface que irá executar o jogo de futebol de robôs, e que integra o sistema de visão com a comunicação e a estratégia. As atividades descritas neste artigo baseiam-se na inclusão da calibração e na correção da distorção, procedimentos ligados à definição correta do posicionamento dos robôs no campo, na definição da orientação, e na integração entre o software de controle geral e as novas rotinas de comunicação desenvolvidas.

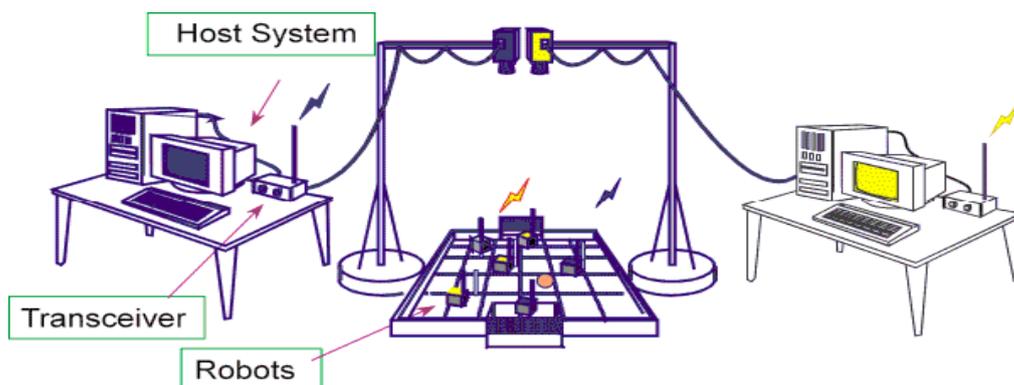
O artigo inicia com a descrição do problema relacionado à distorção das imagens digitais. Posteriormente, é discutido o tratamento do efeito barril presente nas imagens digitais, que provoca modificações nas posições relativas do robô em relação ao campo. O objetivo é realizar a correspondência entre as coordenadas de pontos 3D não distorcidos do mundo real (em milímetros), para as coordenadas 2D nos pixels da imagem distorcida. Dentre os algoritmos de distorção descritos na literatura a solução adotada foi à proposta definida por Tsai (1987), útil em várias aplicações e capaz de lidar com pontos coplanares e não coplanares. São descritos detalhes a respeito da implementação da integração entre a interface e as novas rotinas de comunicação que foram implementadas. Além disso, foi tratada a definição da orientação do robô por meio dos marcadores coloridos.

Posteriormente são apresentadas as informações relacionadas à montagem do campo e aos problemas e soluções encontrados no caso do campo real. Em seguida os resultados encontrados para a correção da distorção, com um exemplo de como é montado o procedimento de calibração da distorção. Também são apresentados os testes no campo real com o algoritmo de rastreamento de objetos para o robô e a bola, com um exemplo do software em movimento, e o funcionamento da comunicação entre o robô e o sistema de rádio-base. Por fim, são apresentados os resultados e as discussões a respeito dos tópicos envolvidos no projeto procurando descrever os problemas encontrados e as soluções adotadas. Em seguida são descritas as considerações finais a respeito do trabalho.

2 Processo de distorção das imagens

O sistema de futebol de robôs funciona em um campo especialmente desenvolvido para tal e que deve possuir uma estrutura análoga à ilustração apresentada na Figura 1. Tal montagem inclui o campo, a estrutura de iluminação, a câmera situada sobre o campo, e o computador que irá processar as imagens e a estratégia do jogo.

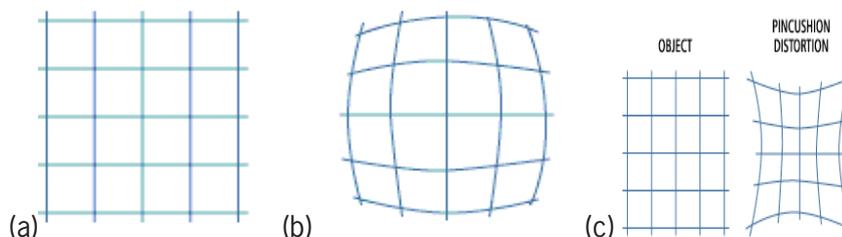
Figura 1 – Ilustração do sistema adotado em sistemas de futebol de robôs



Fonte: CELIBERTO JUNIOR, 2017

A câmera deve estar situada a aproximadamente 4m de altura e deve captar imagens digitais de todo a extensão do campo. A câmera utilizada nessas condições necessita de uma lente que seja capaz de focar e incluir todo o campo, normalmente uma lente com características *wide-angle*. As imagens obtidas pelo conjunto lente-câmera nesta situação sofrem um efeito denominado distorção do tipo barril (Figura 2b). O efeito é resultante da magnificação da lente ser maior no centro do que na periferia. Outro tipo comum de distorção é denominado *pincushion* (Figura 2c) cujo efeito é diminuir a magnificação no centro e aumentar na periferia da lente. As lentes da câmera digital são normalmente construídas com vários elementos visando reduzir o número de aberrações que ela pode gerar, estas aberrações são divididas em aberrações de borramento e de distorção. Dentre as aberrações de borramento é possível destacar, a esférica, coma e astigmatismo (HANSEN, 1999); com relação às de distorção é possível citar o campo de curvatura e as distorções (barril e pincushion).

Figura 2 - Imagem de grade sem distorção (a), distorção barril (b) e pincushion (c).



Fonte: elaborado pelo próprio autor

As distorções não causam borramento, mas sim um efeito pior que é o deslocamento de um ponto com relação ao eixo, em casos nos quais a localização correta é importante ela pode desempenhar um papel essencial. Ela é causada pelo aumento da magnificação e depende fortemente do ângulo com o eixo óptico. Se a lente causa aumento na magnificação conforme nos distanciamos do eixo óptico, tem-se a distorção pincushion, caso ela cause diminuição tem-se a distorção barril. Além disso, ela é das únicas aberrações que permite realizar uma calibração clássica da câmera para simular uma lente perfeita.

A Figura 3 apresenta a imagem real do campo sob a visão da câmera e da lente *wide-angle* presente no sistema de visão computacional, com a câmera situada a aproximadamente 4m de altura. Nesta situação, é visível a distorção (efeito barril) das linhas do campo na imagem obtida

pela câmera. O principal efeito da distorção barril é provocar a modificação das posições relativas do robô em relação ao campo, tornando necessária a aplicação da correção do efeito provocado pela lente, uma vez que os dados obtidos na imagem são dados de posições distorcidas do robô. Além disso, o algoritmo de rastreamento retorna as posições em *pixel* e estas devem ser convertidas para as coordenadas em milímetros do campo, sem a distorção. Logo, a correção da distorção irá atuar nas transformações das informações de posicionamento dos robôs que serão enviados ao módulo de estratégia. O efeito barril não provoca modificações na qualidade da imagem, apenas nas posições observadas na imagem.

Figura 3 - Imagem do software apresentando o campo de futebol real.



Fonte: elaborado pelo próprio autor

3 Desenvolvimento do módulo de controle de distorção

Existem diversos algoritmos que se propõem a solucionar o problema da distorção (HONTSCH; KARAM, 2002), desde as interpolações lineares, interpolações lineares como inicialização de métodos não lineares e métodos não lineares propriamente ditos (WINKLER, 1999). Na sua grande maioria as soluções são baseadas no modelo matemático de pinhole (ou modelo de perspectiva) para uma câmera perfeita (GREMBAN et al., 1988), seus parâmetros servem como fundamentos para a modelagem dos outros sistemas de câmera, ele é muito utilizado devido a sua simplicidade e similaridade a um conjunto lente-câmera.

A transformada linear direta (DLT) pode ser entendida como uma extensão do modelo pinhole que inclui uma movimentação da origem da imagem e uma distorção linear no plano da imagem (TOMMASELLI, 1999). A sua solução é baseada em decomposição de matrizes e pode ser um processo custoso computacionalmente e baseado em métodos não iterativos e rápidos (MELEN, 1984), conforme comentado anteriormente, seus resultados podem ser utilizados para estimação de parâmetros da câmera digital, que por sua vez podem ser utilizados como ponto inicial de métodos mais avançados.

Na fotogrametria clássica a distorção é normalmente modelada por duas componentes de distorção não lineares denominadas radiais e tangenciais ao invés da distorção linear dada pela DLT (TOMMASELLI, 1999). A distorção radial, simétrica ao redor do centro óptico, é não-linear e caracteriza-se como a maior componente da distorção e também como a mais modelada. A componente tangencial

normalmente é negligenciada, pois vários estudos comprovaram que o custo-benefício na precisão não é importante.

A distorção radial é pequena em direções que são próximas ao eixo óptico, crescendo com uma série de potência do ângulo, tendendo a ficar mais pronunciada com a utilização das lentes wide. A variação na posição devido à distorção radial pode ser modelada usando equações como as apresentadas na Eq. (1), no qual x é medido a partir do centro da distorção, e normalmente, são considerados o primeiro e o segundo termos da série.

$$\delta = x(k_1 r^2 + k_2 r^4 + \dots) \quad (1)$$

Muitos algoritmos têm sido sugeridos para a recuperação dos parâmetros da correção da distorção da lente, eles podem ser divididos grosseiramente em três grupos. Os métodos pumb-line, assumem a presença de linhas retas em uma cena, e os parâmetros de distorção são entendidos como linhas retas em uma imagem não distorcida (STRAND ; HAYMAN, 2000). Métodos iterativos que utilizam mínimos quadrados são utilizados para a estimação dos parâmetros que mapeiam curvas em imagens distorcidas para retas, como por exemplo, o algoritmo de Brown (SWAMINATHAN ; NAYAR, 1999).

O segundo grupo de métodos assume a correspondência entre pontos em uma cena cujas coordenadas 3D e suas projeções são conhecidas, de tal modo que um grande número de pontos aumenta a precisão da técnica, tais como as propostas de Tsai (1987) Goshtasby e Weng (SWAMI-NATHAN; NAYAR, 1999). O último grupo de métodos procura por parâmetros de distorção por meio do matching de múltiplas visões de uma mesma cena, entretanto, apesar de ser um método flexível o esquema não é aplicável em câmeras estáticas.

Dentre os métodos de calibração, o algoritmo de Tsai destaca-se como um dos mais conhecidos, uma vez que é útil em várias aplicações, que pode lidar com pontos coplanares e não coplanares, e que oferece a possibilidade de calibrar parâmetros internos e externos separadamente. Além disso, muitos autores já realizaram testes com relação à comparação entre diversos métodos de distorção, e o algoritmo de Tsai ou suas modificações foram os que resultaram em menor erro de medida (BRUCE; VELOSO, 2003). Baseado nestas afirmações, ele foi escolhido para ser utilizado no software e minimizar o efeito da distorção barril.

A calibração da câmera consiste na estimação de um modelo para uma câmera não calibrada cujos valores dos parâmetros internos não são conhecidos. O objetivo é realizar a correspondência entre as coordenadas de pontos 3D do mundo real, em milímetros, para coordenadas 2D nos pixels da imagem, por meio dos parâmetros externos (posição e orientação relativos ao mundo real), e internos da câmera (ponto principal ou centro da imagem, distância focal e coeficientes de distorção). Os processos envolvidos no algoritmo são feitos em estágios, começando com um método de mínimos quadrados para estimar alguns dos parâmetros e terminando com uma otimização não-linear iterativa para todos os parâmetros simultaneamente, de tal modo que durante a otimização o erro no plano da imagem é minimizado.

Em muitas aplicações de visão computacional considera-se uma câmera já calibrada, isto significa que os modelos dos parâmetros internos da câmera já estão disponíveis. Estes valores podem ser fornecidos pelo fabricante ou obtidos por meio de um alvo conhecido. Esta situação é conhecida como estimação de pose, ou "pose estimation". Entretanto, neste caso considera-se que nenhum desses parâmetros é conhecido e que eles serão calculados por meio do algoritmo. O modelo de Tsai possui 11 parâmetros, sendo que cinco são internos e seis externos.

Internos:

f: distância focal da câmera;

k_1, k_2 : coeficiente de distorção radial de primeira e segunda ordem;

C_x, C_y : coordenadas do centro de distorção radial das lentes;

s_x : fator de escala para lidar com qualquer incerteza devido a amostragem da placa de aquisição.

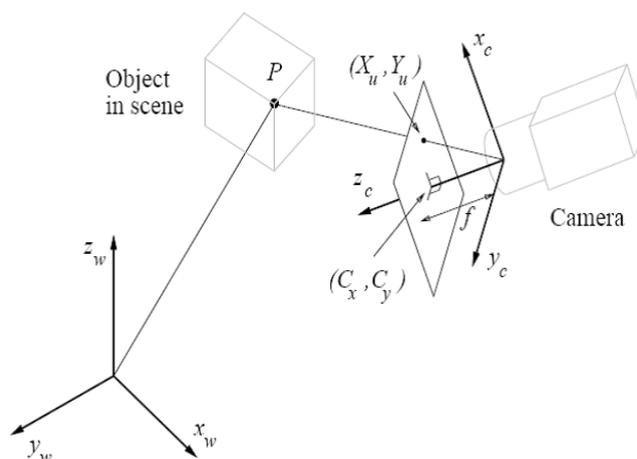
Externos:

R_x, R_y, R_z : ângulos de rotação para as transformações entre as coordenadas do mundo e da câmera;

T_x, T_y, T_z : componentes de translação para a transformação entre as coordenadas do mundo e da câmera.

Na Figura 4 é ilustrado o modelo de Tsai, a origem do sistema de coordenadas da câmera (x_c, y_c, z_c) coincide com o ponto da frente da câmera, e o eixo z_c coincide com o eixo óptico da câmera. O plano da imagem é assumido como sendo paralelo ao plano (x_c, y_c) e a uma distância f da origem, sendo que f é a distância focal efetiva da lente. A relação entre a posição de um ponto nas coordenadas do mundo (x_w, y_w, z_w), e a imagem do ponto no *frame buffer* da câmera (x_u, y_u) é definida por uma seqüência de transformações de coordenadas.

Figura 4 - Modelo geométrico de perspectiva-projeção para uma câmera fixa (WILSON, 1994).



Fonte: WILSON, 1994.

A primeira transformação diz respeito aos parâmetros externos e descreve a relação entre o sistema de coordenadas do mundo (x_w, y_w, z_w) para o sistema da câmera (x_c, y_c, z_c) por meio das transformações de corpos rígidos (rotação e translação) no sistema da câmera. A segunda transformação é a projeção da perspectiva usando o modelo de câmera ideal de um ponto no sistema de coordenadas da câmera (x_c, y_c, z_c) para a posição de sua imagem em um sistema de coordenadas distorcido do sensor (x_u, y_u), dadas pelas Eq. (2).

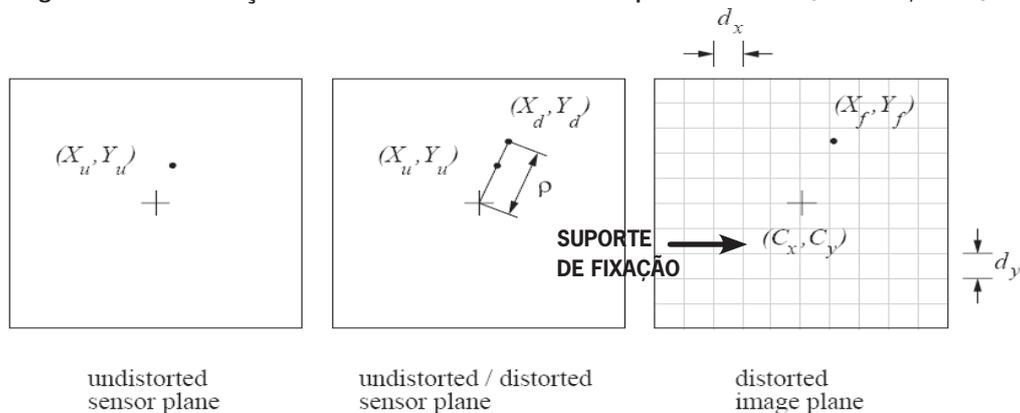
$$x_u = f \frac{x_c}{z_c} \quad \text{e} \quad y_u = f \frac{y_c}{z_c} \quad (2)$$

A terceira transformação, ilustrada na Figura 5, diz respeito à posição não distorcida da imagem do ponto (x_u, y_u) no plano do sensor para a verdadeira posição do ponto na imagem não distorcida (x_d, y_d), equacionado por meio das Eqs. (3).

$$x_u = x_d (1 + k_1 \rho^2) \quad \text{e} \quad y_u = y_d (1 + k_1 \rho^2) \quad (3)$$

Sendo, $\rho = \sqrt{x_d^2 + y_d^2}$ e k_1 o coeficiente de distorção radial das lentes.

Figura 5 - Transformações de coordenadas não distorcidas para distorcidas (WILSON, 1994).



Fonte: WILSON, 1994.

A transformação final é entre a verdadeira posição de um ponto na imagem no plano do sensor e suas coordenadas com relação ao frame buffer da câmera, descrita pelas equações apresentadas em (4).

$$x_f = d_x^{-1} x_d s_x + C_x \quad \text{e} \quad y_f = d_y^{-1} y_d + C_y \quad (4)$$

nos quais C_x e C_y são as coordenadas (em pixel) da intersecção do eixo Z_c e o plano do sensor da câmera, d_x e d_y são as distâncias efetivas entre os elementos dos sensores nas direções x e y, s_x é o fator de escala para compensar qualquer incerteza na razão entre o número de elementos de sensores no CCD da câmera e o número de pixel no *frame buffer* da câmera na direção x.

Durante o processo de calibração de Tsai a ideia é encontrar o valor do coeficiente de distorção da série de potência. A estratégia adotada é utilizar um alvo com geometria conhecida e determinar as correspondências entre pontos no alvo e pontos em suas imagens, este conjunto de dados é utilizado para estimar tantos parâmetros quanto possíveis utilizando matrizes pseudo-inversas. Nesta primeira situação não existe a minimização do erro no plano da imagem, mas sim o tratamento linear das equações. Na segunda etapa, o resto dos parâmetros são obtidos usando o método de otimização não-linear que encontra o melhor resultado entre os pontos da imagem observada e os pontos preditos pelo modelo alvo.

Após amplo estudo, observou-se que a implementação do algoritmo proposto por Tsai incluía diversos detalhes matemáticos e de programação que poderia levar meses para total implementação e validação do algoritmo, desse modo, optou-se por utilizar um código pronto, disponível na internet e fortemente validado e testado (WILLSON; THORHALLSSON, 1993). A completa inclusão do código C no paradigma de classes em C++ acabou demandando mais tempo que o esperado, onde várias estratégias foram tentadas sem sucesso inclusive a modificação do makefile do código. Finalmente, foi possível incluir o código capaz de gerar as transformações de coordenadas necessárias.

Desse modo, a correção da distorção no software foi implementada e deverá ser efetuada antes do início do jogo, ou toda vez que o sistema for inicializado. O campo e a câmera precisam estar montados e colocados corretamente em suas posições. O processo de correção possui quatro estágios: a definição dos parâmetros da câmera, a definição das coordenadas padrões, a determinação dos pontos distorcidos, e finalizando com a calibração do sistema propriamente dito.

4 Calibração do sistema de visão

Na fase de calibração do sistema, o usuário deverá escolher um padrão (quadriculado, pontos, ou outro) de posições conhecidas e posicioná-las fisicamente sobre o campo. Este padrão deve ser capaz de cobrir regiões centrais e periféricas do campo, com uma boa amostragem do campo todo e com um mínimo de 10 posições, quanto maior o número de pontos, menor é o erro da calibração. As informações sobre as posições físicas desses pontos serão consideradas como o padrão do sistema de calibração e deverão ser armazenadas em um arquivo de duas colunas no qual cada linha corresponde às posições x e y de cada ponto em milímetros, respectivamente. Como existe a necessidade da definição de alguns valores a respeito da câmera, também foi criada uma classe TFrameParameters responsável por gerenciar a modificação de parâmetros da câmera conforme apresentado na Figura 6.

Figura 6 - Tela de preenchimento dos parâmetros da câmera.



Fonte: elaborado pelo próprio autor

Após a calibração do sistema, os parâmetros são armazenados em uma estrutura de dados da interface, de tal modo que durante a execução do software eles serão utilizados automaticamente após o rastreamento executado pelo algoritmo Mean-Shift, transformando os valores encontrados na imagem distorcida, em pixel, para valores das coordenadas do mundo, não distorcidas e em milímetros (BIANCHI, 2006). Além da correção da distorção, outra medida que também foi adicionada ao software, foi à calibração em milímetros do valor de cada pixel. Tal medida é importante apenas para a visualização dos dados na tela. Durante o cálculo da distorção, a calibração do pixel já está implícita, de tal modo que a transformação dos valores das posições dos robôs em pixel, coordenadas da imagem distorcida, para as coordenadas em milímetros, coordenadas do mundo não distorcidas, são efetuadas automaticamente.

De acordo com a definição da interface, no início do jogo, o sistema de rastreamento de objetos atua concorrentemente em todos os robôs, até que o tempo do jogo termine. Portanto, o sistema de visão captura todas as posições (como em uma fotografia) e as alimenta no processo de execução da estratégia. Este, por sua vez envia comandos aos robôs de acordo com as ações a serem executadas. Os sistemas do software foram desenvolvidos independentemente: sistema de visão, de comunicação e de estratégia, e, logo depois, integrados. Este procedimento de integração foi realizado

com sucesso, entretanto, durante os testes práticos com os robôs foram sendo descobertos erros durante as execuções. Foram também encontrados problemas no encoder das rodas dos robôs, nas próprias rotinas de comunicação entre o robô e o rádio-base, com a perda de mensagens ao longo da comunicação, e posteriormente, erros na implementação das rotinas do rádio-base.

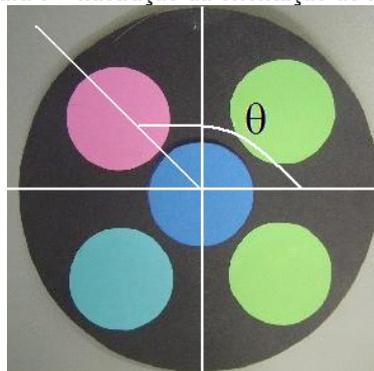
Todos esses fatores e outros culminaram com a reestruturação das rotinas de comunicação do robô e do rádio-base, sendo totalmente reformuladas. Ocorreu a adição da biblioteca “comscisoccer.h” na classe “Robot”. Neste processo, a criação do objeto “Robot” utiliza a informação do endereço físico do robô, como por exemplo, o comando descrito abaixo:

```
Robot *r1 = new Robot(OPONENTE, sist_com, “05 01 0c 0e 01”);
```

5 Módulo de estratégica

O módulo de estratégia foi definido como uma biblioteca dinâmica com um nome e assinaturas fixas, de tal modo que a interface possa utilizar diferentes estratégias de acordo com cada jogador. Durante o desenvolvimento da interface, além das informações a respeito das posições dos robôs, outra informação possui igual importância, a orientação do robô. Durante o jogo de futebol, a orientação do robô diz respeito ao ângulo que, a frente do robô móvel, faz com relação ao eixo x da circunferência trigonométrica, conforme ilustrado na Figura 7, onde θ representa a orientação. A marcação adotada determina que a frente do robô seja determinada por uma bola pink.

Figura 7 - Ilustração da orientação do robô.



Fonte: elaborado pelo próprio autor

O método de determinação deste ângulo é análogo a modificação adotada para o TrackerSys, determina-se o ponto médio do círculo cor de rosa e do centro do robô, encontrado pelo Mean-Shift. Seguindo o teorema de Pitágoras, estes dois pontos são utilizados para calcular o ângulo por meio do seu arco tangente. Entretanto, no caso anterior o ponto médio do círculo rosa era obtido porque a imagem toda sofria uma segmentação por limiares que isolava as cores dos marcadores, que eram utilizados também durante o sistema de rastreamento. Na situação atual é realizada uma limiarização apenas da cor rosa, e na região onde o rastreamento pelo Mean-Shift já atuou e já encontrou o robô, de tal modo que o sistema é capaz de fornecer a orientação e o posicionamento do robô em tempo real.

6 Resultados

A seguir é apresentada a descrição das atividades e os resultados obtidos com os testes de campo referentes à montagem do campo, ao sistema de correção da distorção e ao módulo de visão computacional com o rastreamento de objetos em tempo real.

Um campo real foi montado seguindo as seguintes especificações: 3,20m de comprimento e 2,40m de largura (Figura 8). A Figura 9 apresenta o suporte da câmera preso à estrutura. O suporte foi construído de tal modo que possibilitasse uma pequena variação de alguns dos parâmetros na câmera, tais como: altura, posição ao longo da horizontal e ângulo da câmera. A câmera situa-se aproximadamente 4m de altura do campo. Nessa situação vale destacar que a distância é muito diferente da utilizada para realizar os testes preliminares na bancada do laboratório. Parâmetros como iluminação, abertura e taxa de sincronização da câmera digital tiveram que ser revisados. Estes parâmetros podem atuar de modo efetivo sobre o sistema de visão, tanto positivamente quanto negativamente. Desse modo, o sistema lente-câmera-campo demandou um ajuste refinados dos parâmetros relacionados à câmera, principalmente, no que diz respeito à iluminação que nestas condições de altura teve influência direta sobre os resultados.

Figura 8 - Foto do campo original montado.



Fonte: elaborado pelo próprio autor

Figura 9 - Foto da câmera com a estrutura do campo original.



Fonte: elaborado pelo próprio autor

Além dos problemas relacionados à iluminação e o tamanho dos objetos, a situação da câmera e da lente provocam o efeito barril na imagem adquirida, tal efeito deve ser anulado para garantir o correto posicionamento dos robôs e da bola pelo sistema de visão. Para a resolução do problema foi escolhida a aplicação do algoritmo de Tsai (HORN, 2000). A correção da distorção é efetuada antes do início do jogo, ou toda vez que o sistema for inicializado. O processo da calibração da correção possui quatro estágios: a definição dos parâmetros da câmera, a definição das coordenadas padrões,

a determinação dos pontos distorcidos, e a calibração do sistema propriamente dito.

A Figura 10 apresenta uma situação real inicial no qual o sistema é ligado com os pontos de marcação posicionados sobre o campo (pontos brancos) e no quais alguns dos parâmetros internos da câmera são preenchidos. Esta tela de parâmetros foi adicionada no software, pois seus valores são utilizados durante a calibração da distorção.

Figura 10 - Imagem real do campo e da tela de parâmetros, vistos pelo software.

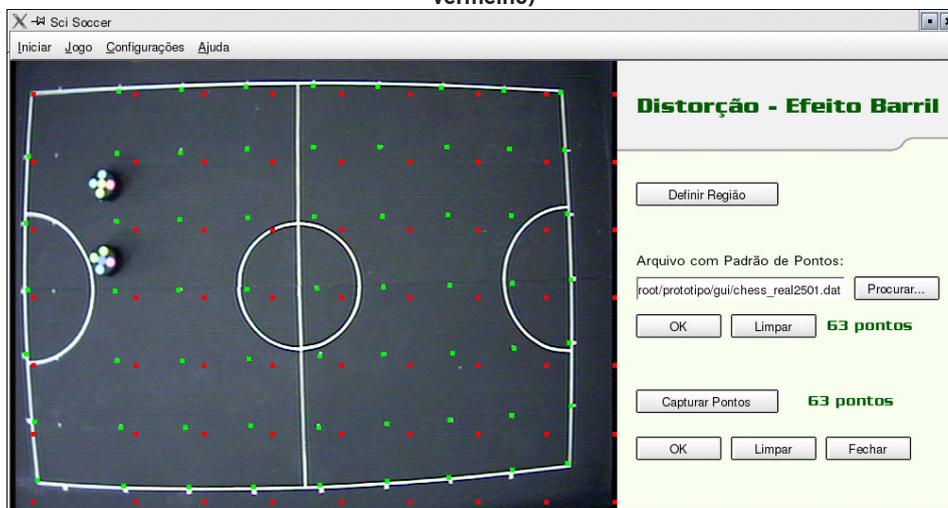


Fonte: elaborado pelo próprio autor

Na segunda etapa, a definição das coordenadas padrão, diz respeito à montagem de um arquivo com as posições originais dos pontos escolhidos situados sobre o campo, ou seja, o arquivo irá conter as coordenadas x e y dos pontos em milímetros e no mundo real, em um arquivo com duas colunas. O procedimento mais fácil foi escolher um padrão pré-definido (quadriculado, pontos ou outro) e posicioná-lo fisicamente sobre o campo, de tal modo que ele ocupasse regiões centrais e periféricas, com uma boa amostragem do campo todo e com um mínimo de 10 posições.

Quando a tela de correção da distorção for solicitada no menu CONFIGURAÇÕES, antes de abrir o arquivo de pontos, deve-se localizar na tela, dois pontos, restringindo a área onde estes pontos serão inseridos por meio do comando “Definir Região”, marcados na Figura 11 como os círculos em amarelo.

Figura 11 - Imagem real do campo com pontos originais distorcidos (em branco) e não distorcidos (em vermelho)



Fonte: elaborado pelo próprio autor

A próxima etapa consiste em carregar o arquivo de pontos reais, por meio do comando PROCURAR do sistema de arquivos. Estes pontos são apresentados na tela como os pontos em vermelho, totalmente distorcidos com relação a sua posição original. Nesse momento, o usuário deve calibrar o sistema pressionando o botão “Capturar Pontos” e pressionando o mouse sobre os pontos em branco da imagem distorcida apresentada na tela. Cada ponto pressionado aparecerá na tela como um ponto verde. Para cada ponto real não distorcido (em vermelho), deve-se ter um ponto distorcido (em verde) na tela, cuja correspondência será utilizada para a calibração da lente e a definição dos parâmetros do algoritmo de Tsai. Quando o usuário termina a captura e pressiona o botão “OK”, os parâmetros são calculados e armazenados em uma estrutura da classe TdataShare. A validação das medidas é realizada utilizando os parâmetros encontrados sobre os pontos em vermelho e verificando se consegue encontrar os pontos distorcidos na imagem, ou vice-versa.

Esse resultado é apresentado na Figura 12, na qual se tem os pontos distorcidos (verdes), não distorcidos (vermelhos) e os calculados (branco). É possível notar pela imagem que existe uma grande correspondência e semelhança entre os calculados (brancos) e os marcados pelo usuário (verdes), conforme o esperado.

Figura 12 - Imagem real do campo com pontos distorcidos (verdes), não distorcidos (vermelhos) e os calculados pelo software de correção da distorção (branco).



Fonte: elaborado pelo próprio autor

Durante a execução das baterias de testes foi possível calcular o erro de posicionamento dos pontos, uma vez que os valores originais não distorcidos são conhecidos durante a calibração. O valor médio do erro ficou em torno de aproximadamente 30mm para a situação escolhida de calibração utilizando aproximadamente 63 pontos.

O valor encontrado pode ser considerado bom se comparado ao comprimento e a largura do campo, apenas 1% e 1,3 % dos tamanhos destes. O erro da calibração envolve vários parâmetros desde o correto posicionamento dos pontos reais do padrão sobre o campo, imprecisão do usuário na definição dos pontos distorcidos (verdes), erros de arredondamentos, aumento da imprecisão na correção conforme se aumenta o raio, até o aumento do valor médio do erro quanto menor for o número de pontos amostrados na tela. Tais erros são previstos pelo algoritmo, e à utilização do termo de maior potência no cálculo da distorção, faz o algoritmo ainda assim caracterizar-se como um dos melhores relacionados à correção da distorção disponíveis na literatura (HORN, 2000).

Durante os testes preliminares pôde-se verificar que o procedimento de rastreamento de objetos utilizando o algoritmo Mean-Shift mostrou-se robusto com relação à perseguição dos objetos, mesmo quando foram realizados testes em uma montagem anterior do campo. Entretanto, como os sistemas mecânicos e eletrônicos ainda não estavam completamente implementados e o robô não se encontrava pronto, os testes do sistema de rastreamento foram efetuados com a variação manual dos marcadores posicionados sobre o robô. Posteriormente, os testes finais foram realizados com robôs móveis na estrutura do campo real e nas condições propostas pela Robocup (dimensões do campo e altura da câmera). Os mesmos mostraram que o método obteve bons resultados para o rastreamento dos robôs, incluindo inclusive o rastreamento da bola, em ambas as situações aconteceram poucos casos no qual o método perdeu o objeto por motivos de rapidez da movimentação do robô ou travamento do software.

Na Figura 13 é apresentado um exemplo do sistema de perseguição. O marcador do robô fica especificado por um quadrado em branco e o resultado do rastreamento está marcado pelo ponto em vermelho, representando o centro do objeto encontrado. Os testes foram realizados por meio de uma estratégia de movimentação dos robôs.

Figura 13 - Imagem apresentando o rastreamento do robô e da bola.



Fonte: elaborado pelo próprio autor

Foram realizados inicialmente testes simples para a verificação do funcionamento. Utilizou-se um marcador de papelão, simulando a presença do robô móvel e se executou o algoritmo de rastreamento, a chamada da estratégia e foi simulada por software a comunicação verificando o envio e o recebimento de comandos, uma vez que ainda não se dispunha de robôs.

Quando os robôs ficaram prontos novos testes foram realizados. Ao longo dos testes ficou claro que o protocolo de comunicação e as rotinas embarcadas no rádio-base precisariam ser completamente reestruturados. As rotinas perdiam muitos dos pacotes, gerando incerteza se o pacote havia sido transmitido e executado pelo robô. Diante disso, as rotinas de comunicação do robô e as rotinas embarcadas no rádio-base foram reimplementadas e corrigidas. Tal processo necessitou de uma nova integração e ajustes no software da interface, principalmente no que diz respeito à compilação e ao makefile da interface. A comunicação foi testada exaustivamente, inclusive se consegue controlar um time de cinco robôs conectado em um mesmo rádio-base permitindo enviar comandos para diferentes robôs de forma independentemente e simultaneamente. É importante ressaltar que os intervalos entre os comandos de movimentação são imperceptíveis.

7 Discussões

A correção da distorção precisa ser executada antes do início de cada jogo, ou toda vez que o sistema for reiniciado. Entretanto, em situações em que o campo e a câmera não se modificam os pontos marcados pelo usuário serão idênticos à situação anterior. Para facilitar esta operação foi criado um sub-menu “Carregar Valores da Distorção”, que aparece na Figura 12 e que carrega na tela a última configuração de pontos da correção da distorção utilizada pelo usuário. Quando for pressionado o botão “Calcular”, os parâmetros do algoritmo serão calculados e armazenados no vetor correspondente e na tela irão aparecer os pontos distorcidos (verdes), os originais de arquivo (vermelhos) e os calculados (branco).

Durante o desenvolvimento do projeto foi necessário à correção da distorção. Entretanto, o seu desenvolvimento foi negligenciada uma vez que um dos times da Robocup utilizava a correção linear (BRUCE & VELOSO, 2003) e acreditou-se que a situação em questão não seria diferente. Entretanto, quando se observou os detalhes da implementação, identificou-se que foi utilizada uma lente quase sem distorção, característica que não era observada na lente usada nos testes locais. Portanto, foi necessário desenvolver um algoritmo não linear de correção da distorção.

Apesar do sistema de rastreamento dos objetos já ter sido testado com várias condições de iluminação, a mesma ainda influenciou durante o decorrer dos testes. Em dias mais nublados, a iluminação gerada pelas lâmpadas não era suficiente para que o sistema de rastreamento atuasse de maneira satisfatória. Em locais muito escuros e com pouca iluminação houve a necessidade de aumentar a iluminação da estrutura montada. A presença de luzes com maior potência foi suficiente para sanar o problema do rastreamento. Inclusive em dias mais claros foi notado o bom desempenho do algoritmo de rastreabilidade. A Robocup exige que os sistemas de iluminação tenham de 700 a 1000 lux, o ambiente construído onde os testes de campo foram realizados esteve propositalmente no limite mínimo (700 lux em dias claros).

Ainda relacionado à iluminação, outro fator que atrapalhou o desempenho dos testes foram às linhas do campo. Os testes anteriores haviam sido efetuados no campo ainda sem o desenho de suas linhas brancas que definem um campo de futebol tradicional. A cor da linha branca ficou visivelmente destacada sobre o campo, ficando muito brilhante na imagem. Mesmo com pouca iluminação a linha ainda possuía um efeito muito forte. Em situações nas quais a iluminação era mais fraca, como os cantos do campo, o rastreamento se perdia e ficava estacionado sobre uma linha branca. Ficou visível durante os testes que o aumento da potência da iluminação diminuía este efeito. A pouca luminosidade faz com que as cores dos marcadores dos robôs se modifiquem perante a câmera digital, alterando a percepção do Red, Green, Blue em comparação com o modelo já determinado no rastreamento do Mean-Shift.

Uma solução para esta situação seria alimentar o sistema de modelo e alvo com o padrão de cores HSL (do inglês: Tonalidade, Saturação e Luz), pois o parâmetro saturação da cor seria tratado de forma independente da matriz. Entretanto, essa conversão de cores não atua de forma tão efetiva na separação quanto se gostaria e também não existe a segmentação por limiares, como no caso do TrackerSys. Desse modo, é impossível prever se tal procedimento irá realmente gerar um resultado satisfatório, uma vez que tais implementações irão modificar o tempo de rastreamento, com a inclusão de conversões ao longo de todas as vezes que ocorrer a aquisição de uma imagem.

8 Considerações finais

Este artigo apresentou os refinamentos propostos para um sistema de visão computacional para ser utilizado em partidas de futebol de robôs. A maior preocupação foi propor um método adequado

de rastreamento de objetos em imagens adquiridos por câmera digital em tempo real e que atualizasse de forma automática o controle de distorção da imagem com um mínimo ou nenhuma perda. Neste contexto, a solução encontrada caracterizou-se como adequada, pois foi possível rodar em um computador com processador Intel core i5 com 4G de memória sendo capaz de realizar o rastreamento dos robôs (cinco) e da bola em tempo real sem que ocorressem perdas dessa rastreabilidade (tracking).

Conforme comentado na seção de resultados, alguns refinamentos ainda devem ser executados, principalmente no que diz respeito à tolerância a iluminação, de tal modo que ela permita uma maior homogeneidade dos resultados. Entretanto, isto não comprometeu a funcionalidade do protótipo do sistema quando testado dentro de um local fechado (galpão) com pouca entrada de luz ambiente externa. Os resultados dos testes de campo com o rastreamento utilizando os robôs podem ser considerados bons. Em uma primeira bateria de testes foi observado o rastreamento de apenas um robô, que mesmo com uma velocidade de deslocamento alta, o tracking manteve-se eficiente. Após várias baterias de testes preliminares foram colocados, no teste final, cinco robôs e uma bola, todos em movimento simultaneamente. Os resultados apresentados obtiveram êxito. Dessa forma, foi possível desenvolver um sistema para futebol de robôs incluindo um sistema de visão global capaz de rastrear múltiplos objetos de tamanhos e velocidades diferentes em tempo real, além da comunicação entre o software geral de controle e os robôs móveis com a inclusão de um módulo de estratégia de jogo de forma automatizada.

Referências

BIANCHI, A. G. C. et al. Sci-Soccer–Desenvolvimento de uma Plataforma de Robôs Móveis voltada ao Futebol de Robôs. In: **Anais do XXVI Congresso da SBC**. p. 203-212, 2006.

BRUCE, J., E VELOSO, M., Fast and Accurate Vision-Based Pattern Detection and Identification, Proceedings of ICRA-03, the 2003 **IEEE International Conference on Robotics and Automation (ICRA '03)**, 2003.

GREMBAN, K. D., THORPE, C. E., KANADE, T. Geometric Camera Calibration using System of Linear Equations, **IEEE Int. Proc. Conf. on Robotics and Automation**, p. 562-567, 1988.

HANSEN, J.F. **On Chromatic and Geometrical Calibration**, PhD Thesis on Technical University of Denmark, 1999.

HORN, B. K. P., **Tsai's Camera Calibration Method Revisited**, 2000. Disponível em: <<http://www.ai.mit.edu/people/bkph/papers/tsaiexplain.pdf>>. Acesso em: 2 mar. 2016.

HONTSCH, I.; KARAM, L. J. Adaptive image coding with perceptual distortion control. **IEEE transactions on image processing**, v. 11, n. 3, p. 213-222, 2002.

MELEN T., **Geometrical modeling and calibration of video cameras for underwater navigation**, Ph.D. thesis 94-103-W, Norges Tekniske Høgskole, Trondheim, Norway, 1994.

STRAND, R., HAYMAN, E., Correcting Radial Distortion by Circle Fitting, Computational Vision and Active Perception Laboratory (CVAP), **British Machine Vision Conference**, 2000.

SWAMINATHAN, R. E NAYAR, S. K., Non-Metric Calibration of Wide Angle Lenses and Polycameras, **IEEE Trans. on Pat. Analysis and Machine Intelligence**, 1999.

TSAI, R. Y., An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision, **Proceedings of IEEE Conference on Computer Vision and Pattern Recognition**, Miami Beach, FL, p. 364-374, 1986.

TSAI, R. Y., A versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the- Shelf TV Cameras and Lenses, **IEEE Journal of Robotics and Automation**, v. RA-3, n. 4, August, p. 323-344, 1987.

CELIBERTO JUNIOR, L. A. **Competições na área de robótica. Disponível em:** <<http://pre.univesp.br/competicoes-na-area-de-robotica#.WbbzrLKG000>>. Acesso em: 5 set. 2017.

WILSON, R., **Modeling and Calibration of Automated Zoom Lenses**, PhD dissertation, The Robotics Institute, Carnegie Mellon University, CMU-RI-TR-94, 1994.

WILLSON, R., THORHALLSSON, T., **Implementação do Algoritmo de Tsai**, 1993. Disponível em: <<http://www.cs.cmu.edu/~rgw/TsaiCode.html>>. Acesso em: 2 mar. 2016

WINKLER, S. Perceptual distortion metric for digital color video. In: **Human Vision and Electronic Imaging**. p. 175-184., 1999.

TOMMASELLI, A. M. G. et al. Fotogrametria: aplicações a curta distância. **FCT**, p. 147-59. 1999.