

# Detecção de formação: Uma ferramenta para detecção de esquemas táticos de times de futebol durante um jogo

Jullyana Fialho Pinheiro<sup>1</sup>, Jordan Boaz Rodrigues<sup>1</sup>

<sup>1</sup> Curso de Ciência da Computação – Universidade Federal do Maranhão (UFMA)  
65.080-805 – São Luís – MA – Brazil

jullyanafialho92@gmail.com, jordanboaz14@gmail.com

**Abstract.** Soccer is one of the most played sport worldwide and its matches has audience in all continents, which makes popular research aimed to understand what happens during the game. A fundamental piece of this sport is the tactics. Tactics is defined by great number of features such as the function performed by each player, their movement and positioning. The methodology proposed in this paper proposes the recognition of tactical formation, meaning the position taken by each player during the game. The methodology created in this work proposes the tactics recognition of the teams during the games. The presented methodology resulted in an accuracy of 61.6 % in the tests.

**Resumo.** O futebol é um dos esportes mais praticados no mundo e suas competições possuem espectadores em todos os continentes, tornando popular pesquisas que visam entender o que ocorre em campo durante seus jogos. Um dos fundamentos mais importante deste esporte é a tática. Tática compreende desde a função exercida por cada jogador, a sua movimentação e o seu posicionamento. A metodologia apresentada neste trabalho propõe o reconhecimento da formação tática, ou seja, o posicionamento adotado por cada jogador durante o jogo. A metodologia apresentada obteve acurácia na detecção de formação de 61.6% nos testes realizados.

## 1. Introdução

Criado na Inglaterra em 1863, o futebol era composto por um conjunto de jogadores, 1 goleiro e 10 jogadores de linha, estes desempenhavam a mesma função, atacar. Com o tempo se percebeu que esses atletas sem atribuições acabavam atrapalhando seus próprios companheiros de equipe. Com isso, funções começaram a ser atribuídas aos jogadores, criando assim os esquemas táticos [Parreira 2005].

Hoje, o futebol tem os esquemas táticos como um dos seus principais alicerces. Tática pode ser entendida como um grupo de regras e comportamentos utilizados para realizar a utilização de forma ótima dos meios condicionais, motores e psíquicos [Rocha 2010]. No futebol a tática é feita tanto para jogadores de forma individual, como coletiva. Boa parte dela é aplicada com ou sem a posse de bola por parte de um time [Greco et al. 2009].

Entender a técnica utilizada por adversários é considerada de extrema importância para se adquirir respostas positivas dentro de um jogo, conseguir observar o esquema usado pelo oponente em partidas anteriores e até mesmo em um tempo anterior da partida atual pode ajudar na definição da tática de um time ou na alteração da mesma [Carvalho et al. 2013].

Um dos sistemas utilizados atualmente para se ter um melhor entendimento do comportamento de jogadores em campo é GPS PRO SOCCER<sup>1</sup>, uma ferramenta que utiliza da tecnologia GPS para permitir a análise dos movimentos dos jogadores. Porém o estudo realizado por essa tecnologia tem como objetivo maior obter dados referentes a capacidade física de cada atleta, de forma a melhorar os treinamentos físicos individuais [Camarão 2009]. O problema dessa tecnologia é a falta de viabilidade que ela apresenta para times de pequeno porte. O uso de imagem pode então provê um meio de estudo mais viável do que o utilizado hoje.

O objetivo desse trabalho é detectar a formação tática usada por equipes de futebol durante uma partida, utilizando de técnicas de processamento de imagem e aprendizado de máquina, com o intuito de fornecer um meio de análise tática para profissionais e aficionados da área.

## **2. Referencial Teórico**

Para o desenvolvimento da metodologia de reconhecimento de formação tática, utilizamos técnicas de processamento de imagem; técnicas de aprendizagem de máquina e análise dos esquemas táticos de futebol.

### **2.1. Técnicas de Processamento de Imagem**

São técnicas utilizadas para transformar uma imagem de forma sucessiva com o objetivo de extrair mais facilmente as informações nela presente [de Albuquerque; Marcelo Portes de Albuquerque 2000].

#### **2.1.1. HSV**

HSV é a abreviatura utilizada para o sistema de cores formado pelas componentes matiz(Hue), saturação ( Saturation) e valor (value). Matiz, ou tonalidade, corresponde as cores, indo do vermelho ao azul em uma volta de 360°. Saturação, ou pureza, indica a quantidade de cinza presente na imagem. Valor, ou brilho, define, como o nome indica, o grau de brilho da imagem. Seus valores variam de 0 à 100%.

#### **2.1.2. Thresholding**

*Thresholding* é uma técnica de segmentação utilizada para extrair pixels de uma imagem que representam certos objetos. Esses objetos normalmente são identificados por características semelhantes, uma dessas características é a intensidade. A saída desse procedimento é uma imagem binária que separa *pixels* que estão fora da intensidade desejada dos que estão dentro dos limites [Morse and University 2000].

#### **2.1.3. Filtros Lineares**

Um filtro é chamado de linear quando sua aplicação resulta na substituição de um pixel pela soma ponderada de seus *pixels* vizinhos [Laganière 2011]. Matematicamente, essa

---

<sup>1</sup><http://gpsprosoccer.com.br/>

operação é chamada de correlação, e é definida pela Equação 1 [Szeliski 2011].

$$g(i, j) = \sum_{k, l} f(i + k, j + l)h(k, l) \quad (1)$$

onde  $f$  é a matriz da imagem e  $h$  é a máscara ou kernel, as entradas nessa máscara são comumente chamadas de coeficientes do filtro [Szeliski 2011]. Existem diversos tipos de filtros lineares, com funções diferentes, neste trabalho foi utilizado o filtro linear da Gaussiana.

#### 2.1.4. Median Blur

O *Median Blur* é uma técnica que tem como objetivo desfocar a imagem com o uso de um filtro de média. O filtro de média utiliza uma matriz de tamanho  $ksize \times ksize$ , onde  $ksize$  deve ser um número ímpar [OPENCV.ORG 2014]. Essa técnica foi usada na extração da máscara de detecção de jogadores, para a retirada das linhas do gramado.

#### 2.1.5. Operação Morfológica de Fechamento

Operações morfológicas são utilizadas no processamento de imagem executando operações não lineares [Dawson-Howe 2014]. Fazendo uso de um elemento estruturante<sup>2</sup>, essas operações executam teste lógicos em todas as posições possíveis da imagem entre esse elemento e uma porção da imagem correspondente, o resultado é então armazenado em uma imagem de saída [Dawson-Howe 2014].

#### 2.1.6. A transformada de Hough

A transformada de Hough foi desenvolvida para detectar formas que são podem parametrizadas de forma simples, como linhas, círculos e elipses, em imagens binárias. Essa técnica é aplicada após a imagem sofrer um pré-processamento, normalmente a detecção de bordas [Jamundá 2000].

### 2.2. Técnicas de Aprendizado de Máquina

Aprendizado de máquina é a capacidade de desenvolver em computadores a habilidade de aprender sem precisar de uma programação explícita. Existem muitos tipos de técnicas nesta área, neste trabalho será utilizado um classificador. Um sistema que recebe características discretas ou contínuas e retorna um único valor discreto, chamado de classe [Domingos 2012].

#### 2.2.1. K-Nearest-Neighbor

*K-Nearest-Neighbor* (KNN) é um classificador de objetos que fundamenta-se em uma base de dados previamente treinada [Harrington 2012]. Esse treinamento é realizado com

---

<sup>2</sup>Elementos estruturantes são pequenas conjuntos de pontos definidos em volta de sua origem

a atribuição de rótulos (classes) para um conjunto de objetos que possuem determinadas características. Para realizar a classificação, sempre que o KNN receber um novo dado, ele o categoriza com uma das classes configuradas. Essa classificação se baseia na distância entre os dados do novo objeto e os dados na base de treino. A classe que estiver mais presente entre as  $k$  menores distâncias encontradas, será a classe do novo objeto [Harrington 2012].

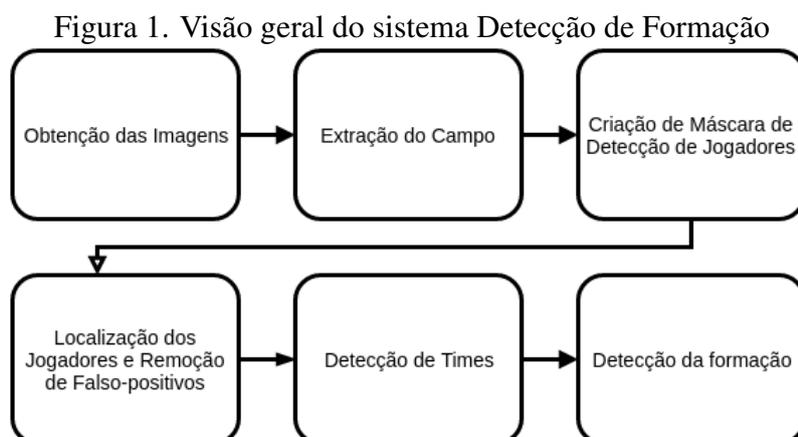
### 2.3. Esquemas Táticos de Futebol

Os esquemas táticos são formações que definem a posição dos atletas em campo durante uma partida de forma a se obter o melhor resultado possível. Eles são distribuídos de forma a organizar os jogadores em 3 regiões: defesa, meio-campo, ataque. Para dar nome a um sistema, são colocados os números de atletas que compõem cada uma dessas áreas. Um time com 3 defensores, 5 jogadores de defesa e 2 de ataque, teria seu sistema definido como 3-5-2. O número de esquemas táticos padrões é pequeno, mas deve se levar em conta que um esquema pode sofrer alterações.

Entender a técnica utilizada por adversários é considerada de extrema importância para se adquirir respostas positivas dentro de um jogo, conseguir observar o esquema usado pelo oponente em partidas anteriores e até mesmo em um tempo anterior da partida atual pode ajudar na definição da tática de um time ou na alteração da mesma [Carvalho et al. 2013].

## 3. Metodologia

A proposta se fundamenta no uso de conhecimentos de visão computacional, processamento de imagem e análise estatística para realizar a obtenção de formação tática, propiciando assim uma ferramenta de pesquisa de para profissionais da área, bem como aficionados pelo esporte. A Figura 1 exemplifica o funcionamento geral do sistema. Cada uma das 6 etapas é descrita na subseções seguintes.



### 3.1. Obtenção das Imagens

As imagens usadas neste trabalho foram extraídas do jogo de videogame FIFA Soccer - 2014, elas foram obtidas pela modificação da câmera de jogo, para aumentar a sua altura e permitir uma melhor visualização do gramado. A altura e dados da câmera

estão informados na Tabela 1. Utilizar o jogo como base de dados para a execução da metodologia foi necessário pois as imagens geradas por emissoras de televisão são voltadas para a melhor visualização de jogadas específicas.

Tabela 1. Parâmetros de jogo modificados para obtenção de uma melhor visão do gramado

<b>Parâmetros</b>	<b>Valores</b>
CAMERA_ZOOM_MODIFIER_COOP	0.09
CAMERA_HEIGHT_MODIFIER_COOP	300.1

### 3.2. Preprocessamento

O preprocessamento é uma das maiores etapas da metodologia e desempenha um papel fundamental na construção do resultado final. A seguir, explicaremos suas etapas.

#### 3.2.1. Extração do Campo

A obtenção do gramado é a primeira parte da resolução do problema, pois nele estão contidos jogadores, bem como juízes, técnicos e a bola. Em [Choi et al. 1997] o campo é caracterizado como o objeto de maior extensão no vídeo e que mantém uma cor verde dominante. E sua extração é realizada com a conversão dos *frames* para HSV e a retirada de *pixels* em que os limites inferiores e superiores de matiz, citados por [Khatoonabadi and Rahmati 2009] e mostrados na Tabela 2, sejam ultrapassados, gerando assim uma imagem binária, mostrada na Figura 2.

Essa abordagem, porém, pode apresentar problemas em jogos onde a arquibancadas também apresenta cores dominantes de verde ou onde existam regiões de gramados não adjacentes ao campo.

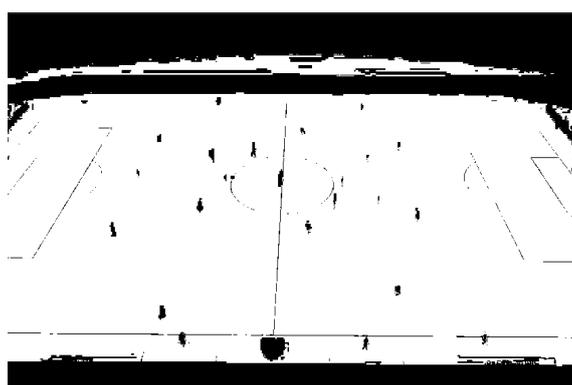


Figura 2. Retirada do verde com áreas verdes não adjacentes

<b>Limites</b>	<b>Tabela 2. Limites de Matiz</b>	
	<b>Valores</b>	
Limite Inferior	54	[Khatoonabadi and Rahmati 2009]
Limite Superior	90	

O problema apresentado foi resolvido com o uso do algoritmo *Connected Components* que tem como objetivo encontrar *pixels* adjacentes que possuem o mesmo valor, o retorno desse algoritmo nos trás o número de componentes conectados encontrados e suas localizações. Criamos, então, uma nova imagem que receberá apenas os *pixels* presentes no componente de maior área, já que entende-se que o campo ocupa a maior parte do vídeo, segundo [Choi et al. 1997]. A Figura 4 mostra o maior componente encontrado na Figura 3.

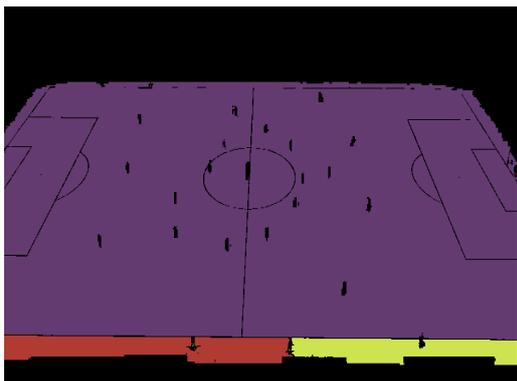


Figura 3. Todos os componentes encontrados

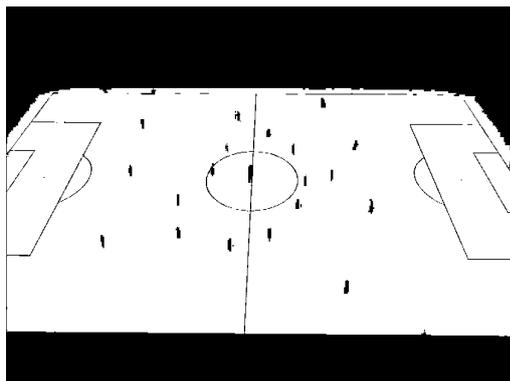
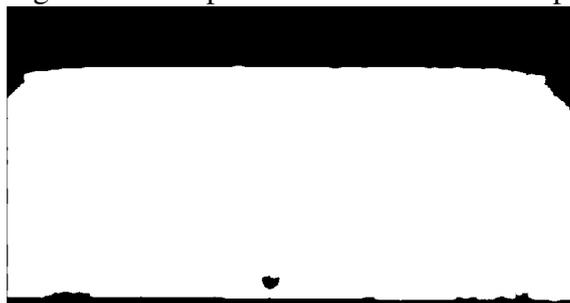


Figura 4. Maior componente

Como pode ser observado na Figura 4, a imagem ainda possui diversas lacunas, derivadas dos jogadores, linhas ou outros elementos que estavam no gramado. A máscara de campo é então encontrada ao realizarmos o preenchimento desses espaços, [Choi et al. 1997].

A máscara de campo deve ser uma imagem semelhante a Figura 5.

Figura 5. Exemplo de uma máscara de campo



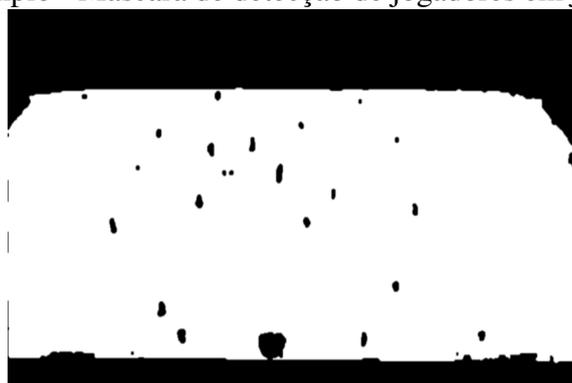
### 3.2.2. Criação de Máscara de Detecção de Jogadores

A máscara de detecção jogadores deve ser obtida para definir a área dos jogadores na imagem. Essa máscara não é exclusiva, pois nela podem estar incluídos quaisquer objeto que esteja, na área de gramado detectada pela máscara de campo. A sua obtenção é realizada a partir da máscara do campo antes do preenchimento, após utilizar sobre ela as seguintes operações: filtros lineares, para redução de ruídos e operações morfológicas de fechamento, para obter melhor separação entre jogadores. O conjunto dessas operações

também é responsável pela eliminação das linhas do campo na imagem, esse procedimento foi realizado por artigo1, com a utilização de um filtro, mostrado na Equação 2 na imagem em escala de cinza do *frame*. Exemplos de resultados desse procedimento são mostrados na Figura 6.

$$\begin{pmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{pmatrix} \quad (2)$$

Figura 6. Exemplo - Máscara de detecção de jogadores em jogos diferentes



### 3.2.3. Localização dos Jogadores e Remoção de Falso-positivos

Para encontrar os jogadores, localizamos primeiro as bordas dos objetos dentro da máscara de detecção. Isso é feito com a utilização do detector de bordas Canny aplicado sobre ela, Figura 7.

Para obtermos as de bordas utilizamos a função *findContours* que nos retorna uma lista de contornos, onde cada elemento da lista é um vetor de pontos que formam cada contorno. Usando essa lista, o menor retângulo que envolva cada contorno é obtido com o uso do *approxPolyDP*. A Figura 8 mostra os retângulos encontrados para a Figura 7.

Como discutido na seção anterior, a máscara de detecção de jogadores captura bem mais que apenas os jogadores, após encontrarmos todos os elementos dentro do campo, é necessário separá-los, para que possamos se possa obter apenas os atletas dos times que estejam em campo.

O tamanho esperado de um jogador com a câmera na distância configurada na seção de obtenção das imagens é de 40 *pixels* de altura e 20 de largura. Elementos que possuem valor de área desproporcional a esse tamanho são retirados. Para remover falso-positivos é feita a média truncada da área dos retângulos restantes. Essa média é calculada com a retirada dos extremos (maior e menor retângulo) e em seguida, computando-se a média simples. Qualquer área de um retângulo que é maior que a média é colocado em outra lista, já que poderemos utilizar esses retângulo posteriormente. Dessa forma, apenas pessoas no gramado serão detectadas pelo algoritmo.

Com essa aplicação, a detecção terá obtido apenas elementos no gramado que foram similar a uma jogador.

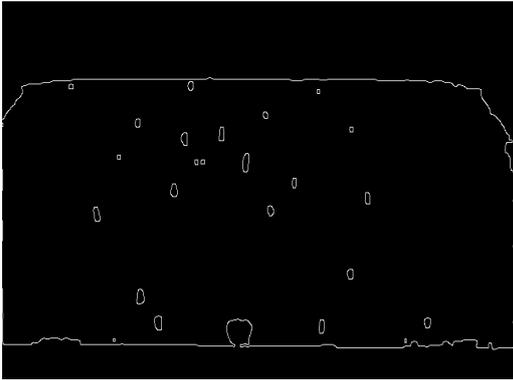


Figura 7. Exemplo - Saída do Canny

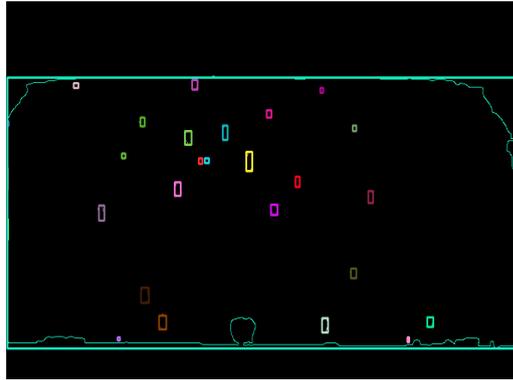


Figura 8. Exemplo - Contornos Encontrados

Como no gramado podem se encontrar técnicos e auxiliares as linhas de delimitação lateral são localizadas com o uso da Transformada de Hough, um exemplo dessas linhas está na Figura 9. A linhas encontradas são expandidas de forma que elas possuam a mesma largura da imagem. A Figura 10 mostra a expansão das linhas da Figura 9. Pessoas que estejam acima da linha superior e abaixo da linha inferior são eliminadas. Com isso, manteremos o rastro apenas dos jogadores reais.



Figura 9. Exemplo - Linhas laterais encontradas



Figura 10. Exemplo - Linhas laterais estendidas

### 3.3. Extração de Características

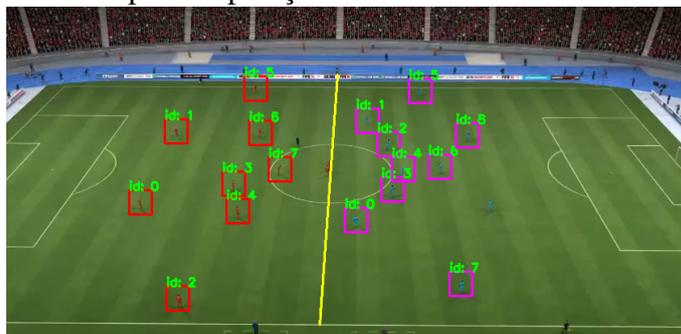
A detecção de times foi realizada com a execução do algoritmo KNN. O algoritmo utiliza uma base de dados treinada para classificar objetos em determinados rótulos. Para treinar a base de dados foram utilizados modelos de cada time criando assim duas classe, uma para o primeiro time e outra para o segundo. A criação do modelo foi realizada localizando-se os atletas no primeiro *frame* de jogo, onde a maioria dos jogadores está incluído, de forma a ter o melhor base possível.

#### 3.3.1. Criação dos Modelos de Times

A obtenção desses modelos é feita a partir com a localização da linha central. Para adquirir essa linha, a transformada Hough foi utilizada na imagem em escala de cinza do primeiro

*frame*. Aproveitando-se do fato que um jogo de futebol tem no seu início seus times postados em lados opostos no campo e que a exibição de um jogo inicia com foco da linha central, todos os jogadores encontrado à esquerda dessa linha são alocados no time 1 e os jogadores à direita, no time 2. A Figura 11 mostra essa divisão.

Figura 11. Exemplo - Separação dos times com base na linha central.



Com os jogadores alocados em suas respectivas equipes, as características das mesmas foram obtidas com a divisão de cada retângulo corresponde a um jogador pela metade, de forma a se criar 2 partes, a superior e a inferior. As Figuras 14 e 15 mostram as partes, superior e inferior, da Figura 13. Em seguida realiza-se o cálculo do histograma dessas partes para os três canais presentes em imagens com o formato de cores HSV: uma para a matiz, a saturação e o valor. No final desse procedimento, obtemos 6 histogramas para cada jogador. Esse conjunto é então classificado com o rótulo correspondente à equipe do jogador. Finalizando assim, o treinamento do KNN.

Figura 12. Exemplo - Divisão do jogador em duas partes. A Figura 13 mostra o jogador completo e as Figuras 14 e 15 mostram, respectivamente, a parte superior e inferior do jogador.



Acervo

Figura 13. Jogador Completo do autor.

Figura 14. Parte Superior

Figura 15. Parte Inferior

### 3.4. Reconhecimento

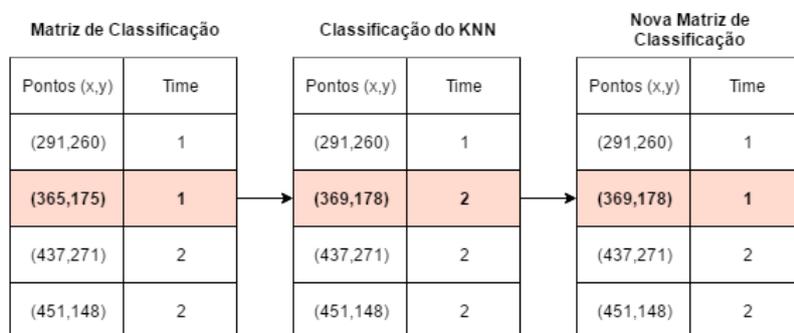
A cada *frame*, o resultado da detecção dos jogadores é adicionado ao KNN, que classifica-os com base em seus modelos treinados. Dessa forma, todos os atletas que forem detectados serão atribuídos a um dos times.

### 3.4.1. Matriz de Classificação

Para melhorar a classificação do KNN, foi criada uma matriz de classificação. Essa matriz possui um número de linhas correspondente a quantidade de jogadores nesse *frame*, e duas colunas que contém, respectivamente, a posição central do retângulo de cada jogador e o time a qual ele foi atribuído. Antes de adicionar os jogadores nesta matriz, os atletas são organizados com base na coordenada  $x$  do ponto central do seu retângulo, em ordem crescente.

Em todos os *frames* do vídeo, a matriz será comparada com a classificação fornecida pelo KNN e sempre que a diferença entre a posição de um jogador encontrada no *frame* e um jogador presente na matriz for menor que 15 *pixels*, eles são considerados o mesmo jogador, e a classificação do KNN é substituída pela classificação da matriz e a posição do jogador na matriz é atualizada com a posição do jogador no novo *frame*. Dessa forma a matriz estará com as posições atualizada, minimizando sua taxa de erro. O diagrama da Figura 16 mostra como funciona esse sistema.

Figura 16. Exemplo - Funcionamento da matriz de classificação. O elemento em negrito é considerado o mesmo elemento na tabela da matriz de classificação e na tabela de classificação do KNN pois eles estão a uma diferença menor que 15 *pixels*. Logo, o time escolhido é o determinado pela matriz de classificação, mas a posição será a mesma da tabela do classificador KNN.



### 3.4.2. Detecção da Formação

A detecção de formação é dividida em 4 etapas: a leitura das formações, a criação das linhas de campo, o encontro dos separadores e a busca pelas formações possíveis.

A base de formações que o algoritmo tentará encontrar nos jogos é mostrada na Tabela 3. Elas foram armazenadas para que a comparação com formação encontrada possa ser realizada.

Para encontrar a formação utilizada por uma equipe, é necessário conhecer a distância em que os jogadores de um mesmo time estão um dos outros. Devido a perspectiva da filmagem, a simples distância entre os retângulos dos jogadores não nos daria o espaçamento real entre eles. Para solucionar este problema, nós utilizamos como referência a linha central, que será replicada entre o primeiro e o último jogador de um time. A Figura 3.4.2 demonstra essa replicação.

Tabela 3. Formações a serem encontradas

Formação	Defensores	Meio-campistas	Atacantes
5-3-2	5	3	2
5-4-1	5	4	1
4-5-1	4	5	1
4-4-2	4	4	2
4-3-3	4	3	3
4-2-4	4	2	4
3-5-2	3	5	2
3-4-3	3	4	3



Figura 17. Exemplo - Criação de replicações da linha central entre o primeiro e último jogador de um time. As linhas em verde representam a primeira e última linha. As linhas em lilás são chamadas de separadoras.

A replicação é realizada da seguinte maneira: com a lista de jogadores de um time ordenada pela coordenada  $x$ , é feito o cálculo da distância horizontal entre o primeiro e o último jogador, de forma que possamos obter o número de linhas que podem ser desenhadas entre eles. Considerando que a distância entre uma linha e outra foi configurada como 30 *pixels*, obtemos então a diferença entre a linha central (que é armazenada, ou seja, ela não precisa aparecer em todos os *frames*) e o primeiro jogador para descobrir o local de início das linhas.

Depois que as linhas foram criadas, armazena-se qual linha se encontra mais perto de cada jogador. Se o primeiro ou último jogador estiverem a uma distância superior a 4 linhas do seu vizinho mais próximo, esse jogador é descartado. Isso é feito para evitar que os goleiros possam aparecer nas formações. Um exemplo disso está na Figura 3.4.2. As distâncias calculadas serão usadas para a atribuição de jogadores a uma das partes da formação: ataque, meio-campo e defesa. As linhas de separação entre essas partes são chamadas de linhas separadoras.

### 3.4.3. Encontro das Linhas Separadoras

O descobrimento das linhas separadoras é feito com a definição das linhas que marcam o começo do ataque (separadora 0) o fim do ataque (separadora 1) e início da defesa (

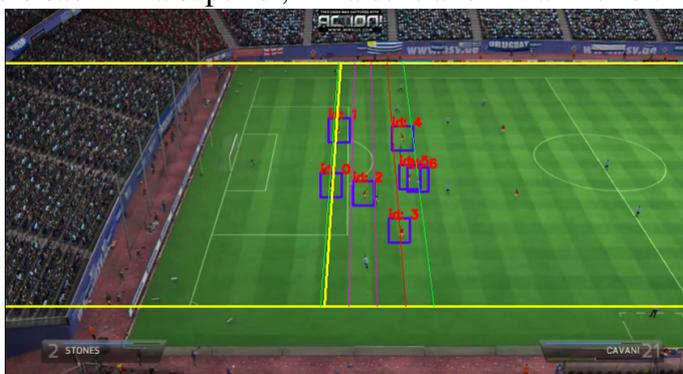


Figura 18. Exemplo - Uso das Linhas de Campo para Retirada dos Goleiros. O jogador de número 8 é retirado da contagem devido a distância dele para o jogador mais próximo (número 7)

separadora 2 ). Para o time dois esses separadores são invertidos. Todos os jogadores entre a separadora 1 e a separadora 2 são considerados meio-campistas. A Figura 3.4.2 mostra um exemplo dessas linhas em um *frame*.

Existem 3 formas para definir a separação entre as linhas: a primeira ocorre caso o número de linhas seja menor que o número de jogadores, a segunda é aplicada quando o número de linhas é maior que o de jogadores e existe algum intervalo de linhas entre atletas, a terceira é selecionada sempre que o número de linha também é maior que o de jogadores, mas não existe intervalo entre essas linhas. A Tabela 4 mostra os valores usados para encontrar esses separadores, o separador 0 sempre será o primeiro jogador. Se a diferença entre o primeiro jogador e o último for menor que 5 linhas, a informação do *frame* é descartada, pois se considera um momento de jogo de indefinição, um exemplo desse momento é mostrado na Figura 19. Podemos observar que o número de jogadores detectados é pequeno e eles estão muito próximos, tornando a detecção de formação neste *frame* inviável.

Figura 19. Exemplo - *frame* descartado devido ao pequeno número de linhas. As linhas em amarelo são: linha superior, linha central e linha inferior.



Acervo do autor.

Tabela 4. Números utilizados para selecionar as linhas separadoras. Intervalos são linhas que não são as linhas mais próximas de nenhum jogador.

Quantidade de Linha	Separador 1	Separador 2
<Número de jogadores	separador 0 + 2 linhas	2º linha antes do último jogador
<20 com intervalos	separador 0 + 1º intervalo	1º intervalo antes do último jogador
<20 sem intervalos	separador 0 + 4 linhas	4º linha antes do último jogador

Acervo do autor.

### 3.4.4. Busca pelas Formações

A busca de formação é realizada após a análise de um conjunto de *frames* que representam determinado intervalo de tempo em um jogo. Ao final de cada intervalo, podemos ter ou não o número de jogadores encontrados em cada região, isso acontece pois os *frames* que possuem menos de 6 ou mais de 10 jogadores detectados em cada time são descartados. A resposta dessa análise nos fornece a formação encontrada, mas essa formação nem sempre está completa, já que jogadores podem não ter sido encontrados na detecção ou não atribuídos ao time correto. Então, todas as formações tem um peso atribuído. Esse peso corresponde ao módulo da diferença entre o número de jogadores na formação original e na encontrada. Um exemplo desse sistema é mostrado na Figura 20. Ao final do intervalo de *frames* estabelecido, a formação que obtiver a maior soma de pesos será a resposta apresentada.

Figura 20. Exemplo - Cálculo de Pesos. Os pesos são atribuídos por *frame* válido. No final do intervalo de *frames* configurados, a formação com maior peso será a formação utilizada naquele período

Formação Encontrada	Mudanças			Peso
	Defesa	Meio-Campo	Ataque	
3   2   3	$ 3 - 5  = 2$	$ 2 - 4  = 2$	$ 3 - 1  = 2$	$1/6 = 0,1667$
	$ 3 - 4  = 1$	$ 2 - 5  = 3$	$ 3 - 1  = 2$	$1/6 = 0.1667$
	$ 3 - 4  = 1$	$ 2 - 3  = 1$	$ 3 - 3  = 0$	$1/2 = 0.50$

Acervo do autor.

## 4. Resultados

Os vídeos utilizados para teste foram extraídos do jogo de videogame FIFA Soccer - 2014, para obtenção de uma melhor visualização do gramado. Dez jogos foram utilizados nos testes e seus resultados foram computados por períodos de jogo (1º e 2º tempo), obtendo assim 20 jogos.

A análise da formação foi feita baseada em intervalos de tempo. Como no videogame cada período de partida possui 4 minutos, 7200 *frames* foram necessários para transmissão de cada tempo do jogo. Para representar 15 minutos de jogo foram obtidos resultados a cada 2400 *frames*. Em uma partida normal seriam 81000 *frames* (45 minutos em vídeos de 30fps) por período e 27000 *frames* a cada 15 minutos. Como o resultado é dado a cada 15 minutos, 3 formações são encontradas a cada tempo de jogo.

Tabela 5. Porcentagens de acertos referentes a cada formação

Formação	Acurácia ( % )	
5 - 3 - 2	50.0	Acervo do autor.
4 - 5 - 1	25.0	
4 - 4 - 2	50.0	
4 - 3 - 3	90.0	
4 - 2 - 4	60.0	
3 - 5 - 2	50.0	
3 - 4 - 3	46.0	

Os testes foram realizados de acordo com a metodologia, para que cada parte possa ser avaliada de forma separada e os desempenhos possam ser analisados de forma mais precisa.

Amostras dos *frames* classificados foram utilizados como base de dados para a verificação da detecção de times. A acurácia apresentada foi de 61.6%. As porcentagens de acertos referentes a cada formação estão presentes na Tabela 5. Foram analisados 20 jogos, com cada jogo fornecendo a formação utilizada a cada 15 minutos de partida. No total, 60 formações foram obtidas. A formação 5 - 4 - 1 não foi apresentada em nenhum jogo. A formação 4 - 5 - 1 apareceu 4 vezes, mas obteve um resultado negativo, tendo apenas 1 acerto. O sistema de classificação de times contribuiu para 31 % dos erros apresentados.

## 5. Conclusão

Este trabalho apresentou a proposta de metodologia que tem como objetivo o reconhecimento de formações táticas em um jogo de futebol, com o uso de processamento de imagem, aprendizado de máquina e fundamentos do esporte.

A etapa de extração de campo, apresentou resultados positivos, porém ela apresenta deficiência se a cor e a iluminação do gramado apresentar mudanças. A fase de reconhecimento de jogadores obteve bons resultados, com um número de falsos positivos aceitáveis.

A fase de classificação foi o período que mais afetou a aplicação da metodologia. A qualidade baixa dos vídeos obtidos influenciou significativamente os resultados nessa etapa. Uma melhora nesse sistema poderia representar um aumento considerável na acurácia dos resultados.

Com um resultado positivo, a metodologia abre espaço para a evolução dos trabalhos apresentados com o desenvolvimento de métodos que possam ser capaz de identificar o momento de jogo em que as formações foram detectadas. Dessa forma, uma equipe poderia perceber falhas ou mudanças em seu posicionamento que tenham afetado o resultado final do jogo. A análise tática de times adversários também pode ser feita com o uso dessa ferramenta. Em campeonatos continentais, equipes nem sempre tem conhecimento sobre o adversário, porém o estudo desse time pode ser realizado com o uso dessa ferramenta.

Os resultados apresentados também demonstram que o uso de vídeos com um melhor resolução podem gerar impactos significativos na qualidade da aplicação, pois facilita a separação de times fornecendo um número maior de informações.

A movimentação de câmera apresentada também gerou impactos na resolução do problema, devido a movimentação lateral que a mesma apresentou, que por diversas vezes removeu jogadores do campo de visão. A utilização de câmeras fixas e de qualidade superior seriam uma evolução a ser implementada na metodologia.

## Referências

- Camarão, B. (2009). A utilização do gps na prática de treinamentos e jogos de futebol. <http://universidadedofutebol.com.br/a-utilizacao-do-gps-na-pratica-de-treinamentos-e-jogos-de-futebol/>. Acesso em 22 de Março de 2016.
- Carvalho, F. M., Scaglia, A. J., and da Costa, I. T. (2013). Influência do desempenho tático sobre o resultado final em jogo reduzido de futebol. Acesso em 17 de Fevereiro de 2016.
- Choi, S., Seo, Y., Kim, H., and Hong, K.-S. (1997). Where are the ball and players? : Soccer game analysis with color-based tracking and image mosaick. <http://dl.acm.org/citation.cfm?id=686879>. Acesso em 12 de Dezembro de 2015.
- Dawson-Howe, K. (2014). A practical introduction to computer vision with opencv.
- de Albuquerque; Marcelo Portes de Albuquerque, M. P. (2000). Processamento de imagens: Métodos e análises. <http://www.cbpf.br/cat/pdsi/pdf/ProcessamentoImagens.PDF>. Acesso em 23 de Março de 2016.
- Domingos, P. (2012). A few useful things to know about machine learning.
- Greco, P., Mesquita, I., Müller, E., Silva, B., Garganta, D. C. J., and Costa, I. (2009). Análise do comportamento tático de jogadores de futebol através da aplicação do teste “gr3-3gr”, em dois períodos de jogo distintos. Acesso em 16 de Fevereiro de 2016.
- Harrington, P. (2012). Machine learning in action.
- Jamundá, T. (2000). Reconhecimento de formas: A transformada de hough. Acesso em 10 de Março de 2016.
- Khatoonabadi, S. H. and Rahmati, M. (2009). Automatic soccer players tracking in goal scenes by camera motion elimination. <http://www.emarketer.com/Article/2-Billion-Consumers-Worldwide-Smartphones-by-2016/1011694>. Acesso em 12 de Dezembro de 2015.
- Laganière, R. (2011). Opencv 2 computer vision application programming cookbook.
- Morse, B. S. and University, B. Y. (2000). Thresholding. Acesso em 12 de Março de 2016.
- OPENCV.ORG (2014). Image filtering. <http://docs.opencv.org/3.0-beta/modules/imgproc/doc/filtering.html>. Acesso em 03 de Março de 2016.
- Parreira, C. A. (2005). *Evolução Tática E Estratégias De Jogo*. Escola Brasileira de Futebol, São Paulo, Brasil.
- Rocha, R. A. (2010). Análise da evolução dos esquemas táticos do futebol brasileiro. Acesso em 16 de Fevereiro de 2016.
- Szeliski, R. (2011). Computer vision: Algorithms and applications.