

# Desenvolvimento de métodos de análise de vídeo para avaliação do movimento de espermatozoides de touro

Autor: Felipe Batista Coelho

Orientador: Bruno Augusto Nassif Travençolo

Faculdade de Computação

Universidade Federal de Uberlândia – Uberlândia, MG – Brasil

[felipe@live.in](mailto:felipe@live.in), [travencolo@gmail.com](mailto:travencolo@gmail.com)

Nível: Iniciação Científica

Bolsa de Estudo: Voluntário

## Sumário

1.	Introdução .....	3
2.	Proposta .....	3
3.	Metodologia .....	3
4.	Amostras.....	4
	Amostra 1 .....	4
	Amostra 2 .....	5
	Análise dos Algoritmos .....	6
	1ª Proposta .....	6
	2ª Proposta .....	7
	.....	8
	O Algoritmo .....	8
5.	Resultados .....	18
	<b>Teste 1:</b> .....	19
	Teste 2: .....	20
	Teste 3: .....	21
	Teste 4: .....	22
	Teste 5: .....	23
	Teste 6: .....	24
5.	Conclusão e Trabalhos Futuros .....	25
6.	Referências .....	26

## 1. Introdução

A qualidade do sêmen é utilizada como variável importante na avaliação do potencial fértil de várias espécies e o espermograma tem sido o principal método de auxílio nesse tipo análise. Além de avaliação visual (volume testicular, volume e concentração do ejaculado e morfologia do espermatozoide), o espermograma busca quantificar aspectos importantes na avaliação do sêmen, como, motilidade, morfologia do espermatozoide e caracterização da compactação da cromatina [13].

Usualmente a motilidade dos espermatozoides é estimada de forma subjetiva por meio da avaliação visual das células sob microscopia óptica convencional dependendo esta, sobretudo, da experiência do técnico [8] [3].

## 2. Proposta

Dessa forma, objetivando uma avaliação mais precisa e livre de erros do examinador, é proposto a análise por computador utilizando técnicas de processamento de vídeo. Nesse sentido, o trabalho se baseia no estudo da proposta apresentada por [2], que, por sua vez, é adaptada de [6]. O algoritmo foi escolhido pelo fato de lidar com características adversas, inerentes ao problema proposto: conseguir rastrear múltiplos objetos, sobrepostos ou não uns aos outros, na presença de mudanças de luz/sombra e outros objetos indesejados que se movem na cena [6]. Além disso, o algoritmo apresenta parâmetros configuráveis, em suas duas fases, que permitem que ele se adapte ao dado problema.

## 3. Metodologia

O estudo deu início com a busca por referências bibliográficas [5] e trabalhos anteriores que pudessem direcionar o aprendizado referente ao processamento de vídeos. Duas abordagens, descritas em [1] e [2], foram testadas sendo a última a escolhida para direcionar o projeto.

Ao mesmo tempo em que o algoritmo da proposta foi estudado, um trabalho de aprimoramento das amostras, provenientes de diferentes técnicas de captura e processos de filtragem, também foi desenvolvido com auxílio de uma bolsista do curso de Veterinária, com o intuito de diminuir impurezas presentes na amostra e melhorar o desempenho do algoritmo na detecção dos espermatozoides. Três amostras foram comparadas e analisadas nesse projeto.

Por fim, a pesquisa bibliográfica foi atualizada na medida do necessário, tendo em vista o forte embasamento teórico de alguns processos vitais ao algoritmo.

#### 4. Amostras

A qualidade das amostras é determinante para o resultado final do processo, ou seja, para o rastreamento dos objetos *frame a frame*. Para o problema é fundamental que se obtê-las com o mínimo de ruído possível, facilitando o trabalho do algoritmo para distinguir objetos indesejados e espermatozoides. Foram utilizadas duas câmeras para realizar o processo de captura. As principais características dos vídeos obtidos por elas são apresentadas na Tabela 1.

TABELA 1 - CARACTERÍSTICAS DOS VÍDEOS COLETADOS POR DIFERENTES CÂMERAS.

	A1	A2
FORMATO:	WMV	AVI
COLETA:	Microscópio	Câmera Fotográfica
TAMANHO:	22MB	56MB
DURAÇÃO:	1min 54s	52s
RESOLUÇÃO:	704x480	640x480
TAXA DE DADOS:	102kbps	11911kbps
TAXA DE BITS TOTAL	110kbps	12000kbps
TAXA DE QUADROS:	30 quadros/s	29 quadros/s
AUDIO:	8kbps	88kbps
CANAIS:	1(mono)	1(mono)
TAXA DE AMOSTRAGEM:	8kHz	11kHz

##### Amostra 1

O principal problema encontrado na amostra **A1** foi a presença de resíduos que seriam um empecilho no processo de segmentação. Os mesmos estão presentes de duas formas: **estaticamente**, não se movendo durante todo o vídeo, e **dinâmicos** que se movem de acordo com o fluido. O primeiro se dá através da lente do microscópio, podendo ser ou sujeira ali presente, ou um dano na mesma ou até mesmo uma característica do material,

que por sua vez está sendo aumentado centenas de vezes. A origem do segundo resíduo não foi esclarecida, mas foi proposto a dissolução do sêmen com outra substância a fim de eliminá-lo.



**FIGURA 1 - AMBAS IMAGENS EXTRAÍDAS DA AMOSTRA A1. A PRIMEIRA AOS 21s E A SEGUNDA A 1MIN 48s. VERIFICA-SE A PRESENÇA DO RESÍDUO (OU FALHA) NA LENTE DO MICROSCÓPIO.**

Além dos resíduos é desejável que se elimine um ruído “fantasma” presente nos espermatozoides, e que é causado porque os espermatozoides se movem numa velocidade maior do que o Hardware consegue captar.



**FIGURA 2 - AQUI SE NOTA CLARAMENTE A PRESENÇA DO EFEITO “FANTASMA” DE MODO QUE O ESPERMATOZOIDE SE APRESENTA COM DUAS CALDAS.**

## Amostra 2

Para resolver esse problema foi proposta a coleta usando uma câmera fotográfica

acoplada ao microscópio, com a qual foi gerada a amostra **A2**. Pela análise da amostra foi constatado que o problema foi resolvido. Com exceção deste último problema todos os outros são encontrados também nessa amostra.

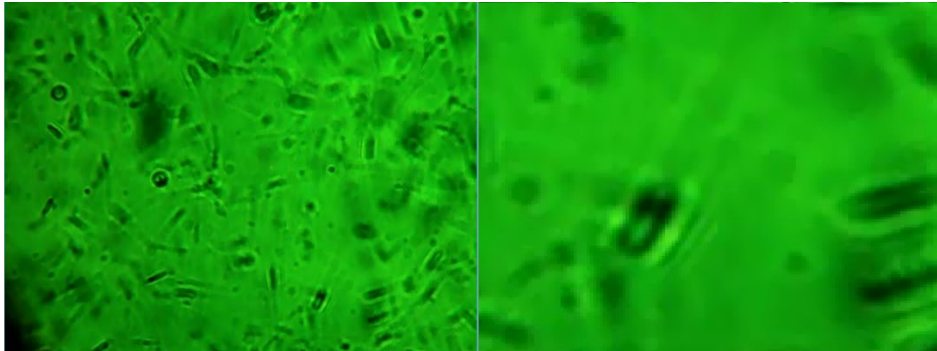


FIGURA 3 - A ESQUERDA UM FRAME RETIRADO DA AMOSTRA 2. A DIREITA UM ESPERMATOZOIDE EM FOCO, DEMONSTRANDO A CORREÇÃO DO EFEITO "FANTASMA".

Apesar de resolver o efeito indesejado, mostrado na Figura 2, a Amostra 2 apresenta um número muito grande de espermatozoides, em contrapartida a Amostra1. Desta forma, é preferível um número menor, tendo em vista que, nas fases de Detecção e Rastreamento os resultados são mais fáceis de serem analisados.

## Análise dos Algoritmos

A seguir serão analisadas as propostas, [1] e [2], estudadas durante a pesquisa. Uma vez que a 2ª Proposta foi escolhida para dar continuidade no projeto, o seu algoritmo será melhor detalhado e na seção 5 os resultados de detecção são apresentados.

### 1ª Proposta

A primeira tentativa para resolução do problema se deu através da implementação da proposta [1]. Um algoritmo de segmentação de vídeo baseado na diferenciação de *frames* consecutivos para registro de *background*. Apesar de propor um algoritmo rápido, se limita apenas a segmentação, não discorrendo em nada sobre a detecção de objetos.



FIGURA 4 - RESULTADO PARCIAL OBTIDO NA IMPLEMENTAÇÃO DA PROPOSTA [1]. CADA *PIXEL* DA IMAGEM, NO DADO INSTANTE, CONTA POR QUANTOS *FRAMES* O MESMO NÃO FOI ALTERADO.

Dado que na proposta até então estudada não se previa o rastreamento dos objetos segmentados, iniciou-se o estudo da proposta [2], que além de se propor a rastrear os objetos, se mostra mais criteriosa e adequada ao problema vigente, detectando, como já citado anteriormente, objetos sobrepostos uns aos outros.

## 2ª Proposta

A segunda proposta, tem, como uma de suas amostras-exemplo, algo mais próximo do problema atual:

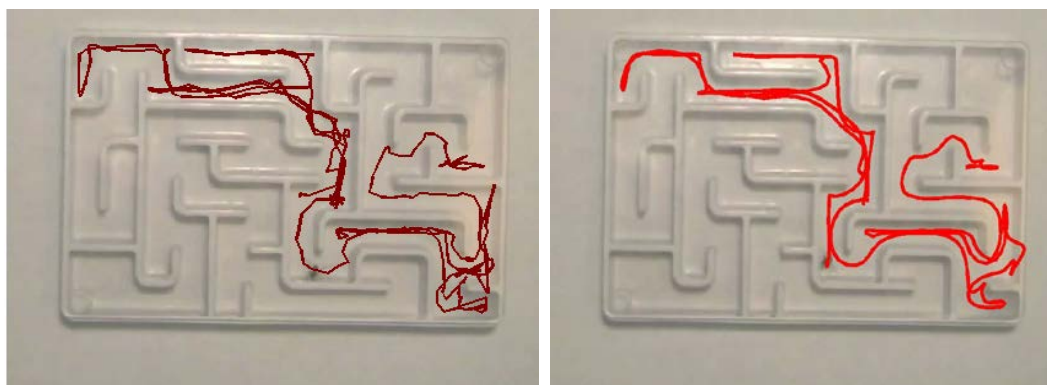


FIGURA 5 – A DIREITA A AMOSTRA-EXEMPLO DA SEGUNDA PROPOSTA. A ESQUERDA O PROCESSO REPLICADO.

A imagem acima mostra o trajeto de uma formiga em um labirinto. Uma tentativa de replica-lo foi feita antes do estudo da proposta ser iniciado.

Os resultados das duas propostas também foram comparadas, usando os espermatozoides e claramente os resultados da última são mais satisfatórios. A Figura 5 ilustra o resultado das duas formas de captura

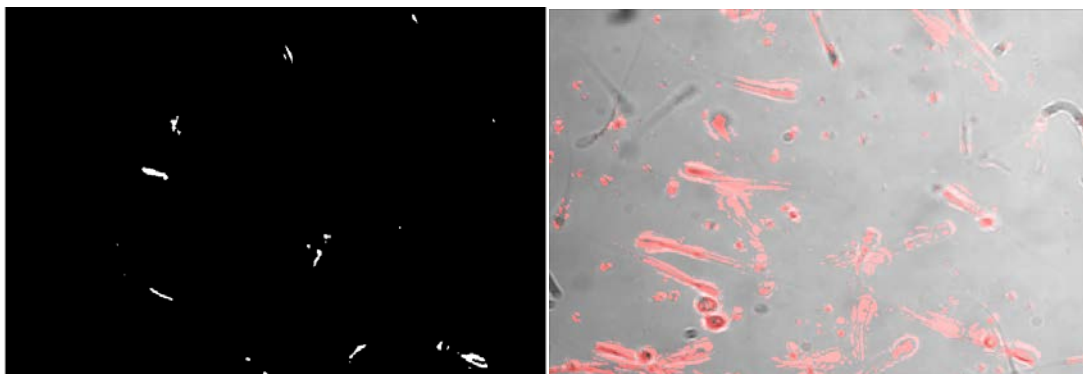


FIGURA 6 - A ESQUERDA A PROPOSTA ANTERIOR, A DIREITA A USADA DORAVANTE.

## O Algoritmo

O seu algoritmo é dividido em duas principais fases, que se decompõe em vários processos, cada uma delas com um componente essencial. São elas:

- Detecção, usando da Mistura de Gaussianas [7]
- Rastreamento, usando Filtros de Kalman [4]

### Passo 1: Detecção de objetos

O primeiro passo do algoritmo é a detecção dos objetos. Esse passo encaminhará os *frames* já segmentados pela fase de rastreamento. É importante que cada espermatozoide seja devidamente reconhecido, e que o máximo de ruído presente na imagem seja descartado para que se obtenha um melhor resultado na fase de rastreamento.

Ao lado da iluminação não uniforme (ver Figura 4), que mostrou pouco afetar na detecção, a presença de ruídos se põe como principal barreira para a detecção dos objetos de interesse, os espermatozoides. Os ruídos, em sua maioria, se apresentam na forma de resíduos, diluídos na amostra, que se movem numa velocidade mais lenta que a maioria dos espermatozoides, entretanto, podem aparecer também estaticamente como sujeira na lente ou simples característica do material, aumentada várias vezes, sobre o qual a amostra está a ser observada.



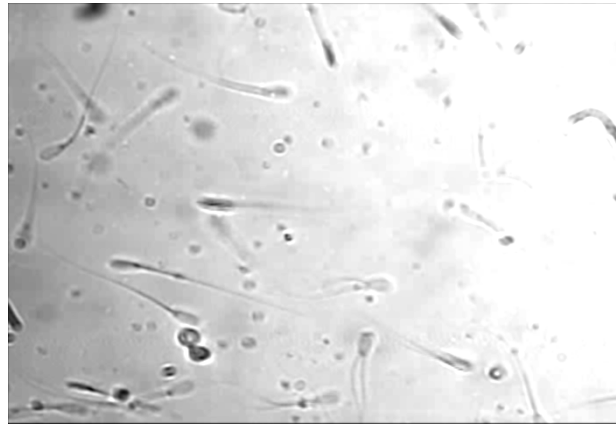


FIGURA 7 - IMAGEM 60% SATURADA. NOTE QUE A ILUMINAÇÃO NÃO É UNIFORME

Para lidar com esse problema, a proposta estudada implementa um algoritmo que detecta e rotula cada *pixel* de *background/foreground*. Ela faz isso identificando valores de *pixel* que são suficientemente incomuns comparados aos seus valores passados [6].

Essa tarefa é implementada usando um modelo de *background* que armazena a distribuição de valores prévios de cada *pixel*. Esse modelo é implementado usando uma Mistura de Gaussianas Isotrópicas.

Cada *pixel* ao longo do tempo é considerado como um “processo”, uma série temporal, que se comporta como uma K-Gaussiana (Stauffer e Grimson, 2000).

$$\{X_1, \dots, X_t\} = \{I(x_0, y_0, i) : 1 \leq i \leq t\}$$

Os *pixels* são modelados dessa maneira pois o algoritmo considera que mais de um *background*, K (geralmente, 3 ou 4), podem aparecer em uma determinada distribuição de *pixels*, assim, mais de uma gaussiana aparecerá na distribuição. Separando essas gaussianas em K, o algoritmo primeiro detecta em qual delas o *pixel* se encaixa e em seguida decide se ele faz parte do *background* daquela K gaussiana ou não. Dessa forma o sistema pode lidar com a presença de vários *backgrounds*.

Essa parte conta com alguns parâmetros configuráveis importantes que farão a manutenção desse modelo *frame a frame*:

1. K – Número de Componentes
2.  $\alpha$  – Constante de aprendizagem para os pesos
3.  $\rho$  (Rho) – Constante de aprendizagem para médias e covariâncias
4. Deviation\_SQ\_Thresh – Limiar usado para diferenciar em qual componente um *pixel* se encaixa

5. **Background\_Thresh** – Limiar que define o tamanho mínimo (em *pixels*) de um componente conexo
6. **Component\_Thresh** – Porcentagem do peso que precisa ser contado (na variável **Weight**) para que um *pixel* seja definido como de *background*

#### Implementação da Segmentação

O diretório do pacote contendo o algoritmo – seu *download* pode ser feito em [10] - tem um arquivo chamado *run\_tracker.m*, que inicia o algoritmo, em sua raiz e dois subdiretórios: *src* e *misc*. O primeiro contém o código propriamente dito, contendo funções como *mixture\_update* e *kalman\_update* que atualizam, respectivamente, os parâmetros da Mistura de Gaussianas e do Filtro de Kalman, citado mais adiante. O segundo, uma miscelânea de funções não necessárias ao algoritmo principal, contendo desde de funções para exibição das trajetórias (*plot\_trajectories.m*) calculadas no algoritmo principal até funções que calculam a trajetória mais próxima a um ponto (*closest\_trajectory.m*).

As principais variáveis que farão a manutenção do modelo de *background* são inicializadas da maneira como se segue:

Data (D, C, T)			MUS (D, K, C)			WEI (D, K, C)			SIG (D, K, C)		
C1	C2	C3	K1	K2	K3	K1	K2	K3	K1	K2	K3
X1	X1	X1	0	R1	R4	1	0	0	3	3	3
X2	X2	X2	0	R2	R5	1	0	0	3	3	3
X3	X3	X3	0	R3	R6	1	0	0	3	3	3
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
xD	xD	xD	0	R9	R10	1	0	0	3	3	3

Onde:

D – é o resultado da multiplicação da altura pela largura do *frame* da amostra.

Xi – são os valores de uma determinado *pixel* numa posição i.

C – É o número de componentes de cor. Por exemplo: se os *frames* forem (R, G, B), C será igual a 3.

K – É o números de componentes gaussianas

Ri – É um número aleatório entre 0 e 255

As variáveis são mais bem descritas nas funções a seguir.

### ***run\_tracker.m***

Esta função cuida de inicializar a variável *data*, responsável pelo armazenamento da sequência de *frames*, e os parâmetros da Mistura de Gaussianas e do Filtro de Kalman, doravante referidos como MG e FK. Uma vez que os dados necessários foram carregados a função chama *track.m*, que se encontra dentro de *src*.

*data(D, C, T):*

Os *pixels* de um *frame* T são armazenados em uma única dimensão, de tamanho D, que representa a altura vezes a largura do *frame* (HIGHT x WIDTH). C representa a quantidade de cores que cada *pixel* tem. Se o *frame* for representado em RGB o valor de C será “3”.

### ***track.m***

Essa função é o coração do algoritmo. Suas primeiras linhas permitem configurar alguns parâmetros de *output*, como *SAVE\_IMAGE\_FRAMES* e *SAVE\_OBJECT\_HISTORY*, que, se *verdadeiros* farão, respectivamente, com que, ao final da execução desta função, se obtenha a sequência de *frames* juntos em uma variável e o histórico dos objetos detectados, que posteriormente pode ser usado para extrair a trajetória dos espermatozoides.

Três variáveis importantes para manutenção do modelo de Gaussianas são aqui configuradas:

*Mus (D, K, C):*

O valor médio do *pixel* D numa Gaussianas K para cada componente de cor C.

*Sigma (D, K, C):*

É a covariância de um *pixel* D numa Gaussiana K para cada componente de cor C. Por questões de memória esta covariância é da forma  $\Sigma_{k,t} = \sigma_k^2 I$ , onde I é a identidade.

*Weight (D, K):*

É a estimativa do peso da K-ésima Gaussiana (a porção de dados contados por aquela Gaussiana) num *pixel* D.

*matched\_gaussian (D, K):*

Define se um *pixel*, no instante atual, da componente K, está dentro do desvio padrão.

Dois *loops* tem papel importante no algoritmo: um externo, de 1 até T, e outro interno, de 1 até K. O *loop* interno cuida de atualizar os desvios quadrados, baseado na variável *Sigma*, em *deviations\_squared*.

É no *loop* externo que acontecem o restante dos processos principais do algoritmo. Ele começa por atualizar a variável *matched\_gaussian*, baseado no limiar previamente definido *DEVIATION\_SQ\_THRESH*, na função *mixture\_params*. Em seguida dois processos são evocados: *update\_mixture\_params*, que atualizará os parâmetros da MG, e *segment\_image* que irá segmentar os *frames*, que por sua vez seguirão para o processo de Rastreamento.

### ***update\_mixture\_params.m***

Aqui a MG será atualizada pela leitura dos novos valores de *pixel* do *frame* corrente e atualização das componentes da mistura que estão abaixo do desvio (*DEVIATION\_SQ\_THRESH*) – que foram marcadas em *matched\_gaussian*. Essa função implementa o algoritmo de [6].

## **segment\_image.m**

Retorna uma máscara de *foreground* contendo todos os *pixels* que são considerados objetos. Para isso ela utiliza-se de um método que detecta valores de *pixels* muito discrepantes dos demais (*outliers*), definido também por [6]. A variável BACKGROUND\_TRESH define uma porcentagem para esse nível de discrepância.

## **foreground\_objects.m**

Após segmentar cada um dos *frames* essa função irá encontrar as componentes conexas menores do que o limiar definido em COMPONENT\_THRESH. O algoritmo usado para conectar as componentes pode ser *4-connected* usando como entrada “4” ou *8-connected* usando como entrada “8”. A função retorna o mapa, número, tamanho e posição (do centroide) dos objetos.

### Passo 2: Rastreamento de objetos

Esta fase rastreará os objetos, segmentados na fase anterior, no eixo temporal. Por *rastrear* entende-se obter a informação da posição de cada objeto (espermatozoide) ao longo dos T frames.

Para obter uma previsão de onde o objeto, detectado num dado momento  $t$ , estará no próximo *frame*  $t+1$  – e posteriormente decidir qual, dentre os tantos objetos detectados em  $t+1$  é o da previsão feita em  $t$  – é usada a ferramenta conhecida como Filtro de Kalman [4]. O Filtro opera em dois passos: Predição e Atualização. A Predição usa a estimativa do estado no passo anterior para obter uma estimativa do estado no tempo atual. Esta é chamada de estimativa *a priori*, pois não inclui a informação vinda da observação do estado atual.

No passo de Atualização, a predição *a priori* é combinada com a observação atual para refinar a estimativa do estado. A estimativa refinada é chamada de estimativa *a posteriori*.

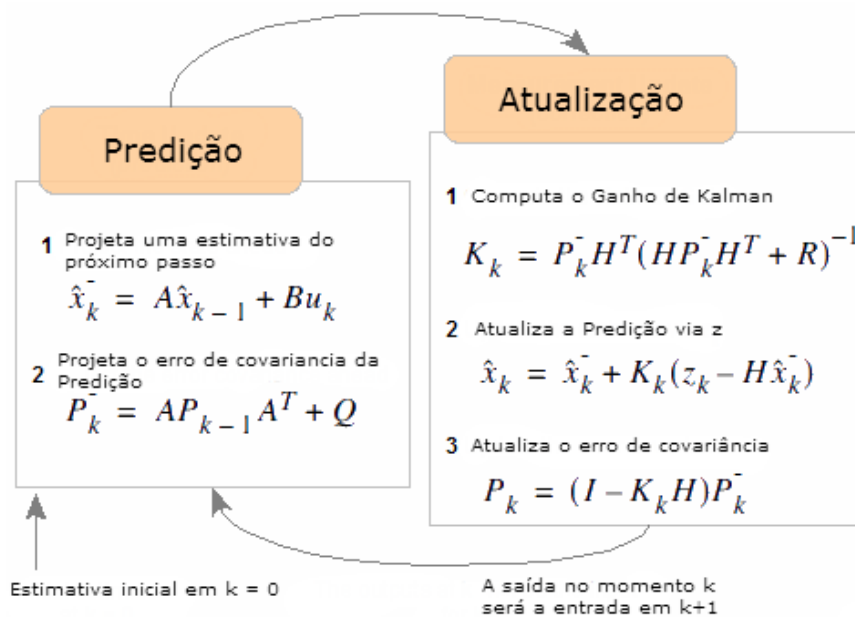


FIGURA 8 - REPRESENTAÇÃO MATEMÁTICA DO FILTRO DE KALMAN, OBTIDA DE [8].

Abaixo os parâmetros configuráveis do Filtro de Kalman:

#### 1. kalparam.ASSOC\_DUMMY\_COST

É usado no algoritmo *data\_association*, define um limiar de otimalidade estimada para distância

#### 2. kalparam.NEW\_HYP\_DUMMY\_COST

É usado no algoritmo *add\_new\_hypotheses* para construir os custos de associação.

#### 3. kalparam.A

É atribuída de acordo com o que foi definido em *KALMAN\_STATE\_TYPE*, e da mesma forma *KALMAN\_STATE\_SIZE*.

i) *KALMAN\_STATE\_TYPE* = 'pos'

*KALMAN\_STATE\_SIZE* = 2

kalparam.A = [1 0;

0 1]

ii) KALMAN\_STATE\_TYPE = 'pos+vel'

KALMAN\_STATE\_SIZE = 4

```
kalparam.A = [1 0 1 0;  
              0 1 0 1;  
              0 0 1 0;  
              0 0 0 1]
```

iii) KALMAN\_STATE\_TYPE = 'pos+vel+size'

KALMAN\_STATE\_SIZE = 5

```
kalparam.A = [1 0 1 0 0;  
              0 1 0 1 0;  
              0 0 1 0 0;  
              0 0 0 1 0;  
              0 0 0 0 1]
```

iv) KALMAN\_STATE\_TYPE = 'pos+vel+size+size\_vel'

KALMAN\_STATE\_SIZE = 6

```
kalparam.A = [1 0 1 0 0 0;  
              0 1 0 1 0 0;  
              0 0 1 0 0 0;  
              0 0 0 1 0 0;  
              0 0 0 0 1 1;  
              0 0 0 0 0 1]
```

5. kalparam.G, kalparam.B, kalparam.Q, kalparam.R

São matrizes constantes KALMAN\_STATE\_SIZE x KALMAN\_STATE\_SIZE usadas no cálculo do Filtro de Kalman.

6. kalparam.MAXPROPAGATE

Diz o número máximo de vezes que um objeto pode propagar sem ser atualizado até que pare de ser rastreado.

## 7. kalparam.ASSOC\_COST\_TYPE

Diz como o custo para associar objetos (em *data\_association*) será calculado. Pode ser pela *distância quadrada* entre pares candidatos ou pela *expectativa* (ou previsão) calculada pelo Filtro de Kalman.

## 8. kalparam.ASSOC\_ALG\_TYPE

Dita qual o algoritmo usado para associar os objetos *frame a frame*. Três opções são disponibilizadas: “SMP” [11], “MUNK”, uma implementação MATLAB do algoritmo Munkres (*Hungarian Method for Assignment Problem*) [12] ou “LAP”, uma implementação, em C++, do algoritmo publicado por R. Jonker and A. Volgenant [7].

### Associação de objetos

As circunstancias do problema exigem que o algoritmo seja efetivo não só na detecção de um, mas de múltiplos objetos, dessa forma, o Filtro de Kalman precisa associar, sem erros, um objeto B, no tempo T, ao seu correspondente em T+1.

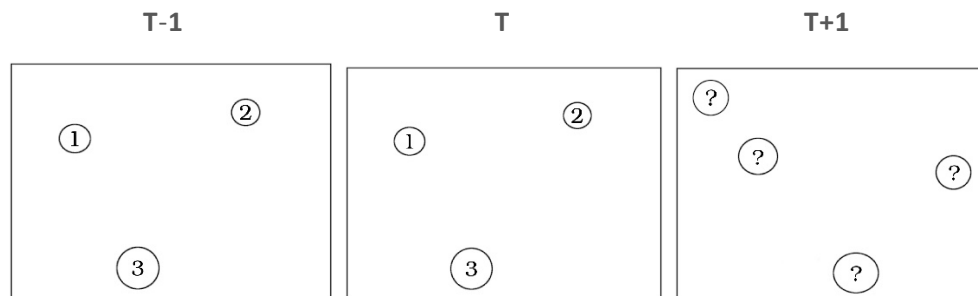


FIGURA 9: CADA CÍRCULO REPRESENTA UM OBJETO QUE FOI DETECTADO DA ESQUERDA PARA A DIREITA. NO FRAME T+1 O PRINCIPAL DESAFIO SE É ESTABELECEER UMA ASSOCIAÇÃO DOS NOVOS OBJETOS DETECTADOS E DOS FILTROS DE KALMAN GERADOS A PARTIR DOS OBJETOS DO FRAME T.

Para lidar com esse problema de atribuição linear a proposta estudada implementa um algoritmo em C++, publicado por R. Jonker and A. Volgenant [7]. O problema de atribuição linear é transformado em um problema de encontrar o conjunto de pontos, ou seja, a bolha, cujos valores maximizem a dada função de custos.



## Implementação da Detecção

A implementação dessa parte se dá basicamente em três funções: *kalman\_prediction*, *data\_association*, *update\_kalman\_filters*. Todas elas estão também contidas no diretório *src*.

O Filtro de Kalman é mantido em um *array* de estruturas *objects*. Cada estrutura contém os seguintes campos:

1. *new\_kalman*: lhe é atribuído “1” se necessita de inicialização. “0” do contrário.
2. *ntimes\_propagated*: quantas vezes o objeto foi propagado sem nenhuma atualização.
3. *label*: o rótulo do objeto.
4. *y\_t\_plus\_1*: um vetor de medição contendo qual dos objetos foi atualizado mais recentemente.
5. *x\_t\_given\_t*: A média do *estado oculto* dado observações *y* no estado *t*. As informações são estruturadas de acordo com o que foi definido em *KALMAN\_STATE\_TYPE*.
6. *x\_t\_plus\_1\_given\_t*: A média do *estado oculto* em *t+1* dado observações *y* no estado *t*. As informações são estruturadas de acordo com o que foi definido em *KALMAN\_STATE\_TYPE*.
7. *expected\_y\_t\_plus\_1\_given\_t*: A média esperada da próxima medição. As informações são estruturadas de acordo com o que foi definido em *KALMAN\_STATE\_TYPE*.
8. *P\_t\_given\_t*: a matriz de covariância do *estado oculto* dado observações *y* no estado *t*. As informações são estruturadas de acordo com o que foi definido em *KALMAN\_STATE\_TYPE*.
9. *P\_t\_plus\_1\_given\_t*: a matriz de covariância do *estado oculto* em *t+1* dado observações *y* no estado *t*. As informações são estruturadas de acordo com o que foi definido em *KALMAN\_STATE\_TYPE*.
10. *traj\_length*: é incrementado sempre que o objeto é propagado ou atualizado.

### ***kalman\_prediction.m***

Essa função implementa o passo de Predição do Filtro de Kalman. Nela irão ser computadas estatísticas (observações) que dirão onde possivelmente cada objeto estará no próximo *frame*. Essa predição só será de fato usada no próximo loop em *data\_association*. Os parâmetros aqui atualizados são: *x\_t\_plus\_1\_given\_t*, *P\_t\_plus\_1\_given\_t*, *expected\_y\_t\_plus\_1\_given\_t*, *expected\_sigma\_y\_t\_plus\_1\_given\_t* e *new\_kalman*, todos retornados dentro da estrutura *objects*.

### ***data\_association.m***

Aqui cada observação previamente feita (no *loop* anterior) vai ser associada a um objeto (componente) detectado no passo (*loop*) atual. Essa tarefa se configura como um problema de atribuição linear, e a proposta estudada implementa três algoritmos que podem cumpri-la, estes descritos anteriormente em *kalparam.ASSOC\_ALG\_TYPE*. O algoritmo se encontra no diretório *lap*, subdiretório de *src*, e sua chamada é feita dentro de *data\_association*. Tal algoritmo transforma o problema de atribuição linear no problema de encontrar o conjunto de pontos (o objeto), cujos valores maximizem a dada função de custos.

### ***update\_kalman\_filters***

Assim que as observações (o que se fez a previsão) e o que foi medido (as componentes conexas) forem associadas essa função se encarregará de executar o segundo passo do Filtro de Kalman, a Atualização. Atualizará o filtro dado que agora sabe onde cada objeto está.

## 5. Resultados

Alguns testes foram realizados sobre a amostra de espermatozoides para medir o impacto dos parâmetros da MG na qualidade do rastreamento sobre o esperma. O objetivo era conseguir uma configuração que obtivesse trajetórias que refletissem o comportamento e o número de espermatozoides da amostra usada (*amostra\_1*).

### Teste 1:

#### Configuração:

```
mixparam.K = 3;  
mixparam.ALPHA = 0.02;  
mixparam.RHO = 0.01;  
mixparam.DEVIATION_SQ_THRESH = 7^2;  
mixparam.INIT_VARIANCE = 3;  
mixparam.INIT_MIXPROP = 0.00001;  
mixparam.BACKGROUND_THRESH = 0.9;  
mixparam.COMPONENT_THRESH = 10;
```

#### Resultado:

Trajetórias: 525



O número elevado de trajetórias pode ser explicado pelo também elevada presença de ruídos na amostra.

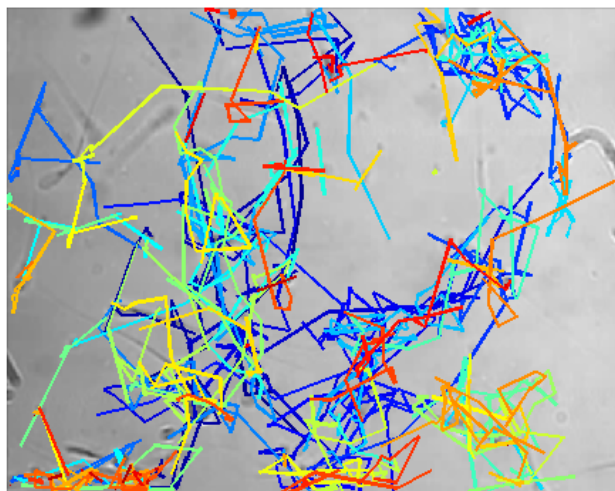
Teste 2:

Configuração:

```
mixparam.K = 5;  
mixparam.ALPHA = 0.02;  
mixparam.RHO = 0.01;  
mixparam.DEVIATION_SQ_THRESH = 7^2;  
mixparam.INIT_VARIANCE = 3;  
mixparam.INIT_MIXPROP = 0.00001;  
mixparam.BACKGROUND_THRESH = 0.9;  
mixparam.COMPONENT_THRESH = 10;
```

Resultado:

Trajetórias: 242



Nessa amostra o número de componentes da Gaussiana foi aumentado de 3 para 5. Segundo [6] normalmente se usam de três a quatro para que o desempenho do algoritmo não seja prejudicada.

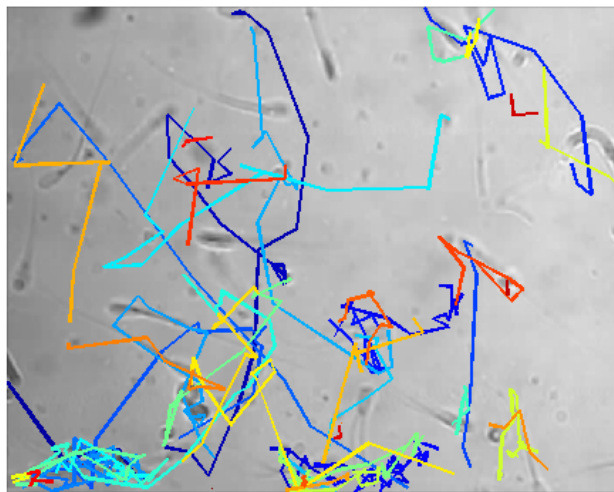
Teste 3:

Configuração:

```
mixparam.K = 3;  
mixparam.ALPHA = 0.02*30;  
mixparam.RHO = 0.01;  
mixparam.DEVIATION_SQ_THRESH = 7^2;  
mixparam.INIT_VARIANCE = 3;  
mixparam.INIT_MIXPROP = 0.00001;  
mixparam.BACKGROUND_THRESH = 0.9;  
mixparam.COMPONENT_THRESH = 10;
```

Resultado:

Trajetórias: 80



Aqui a constante de aprendizagem para os pesos ( $\alpha$ ) foi aumentada 30 vezes. Percebe-se também que o número de trajetórias caiu drasticamente.

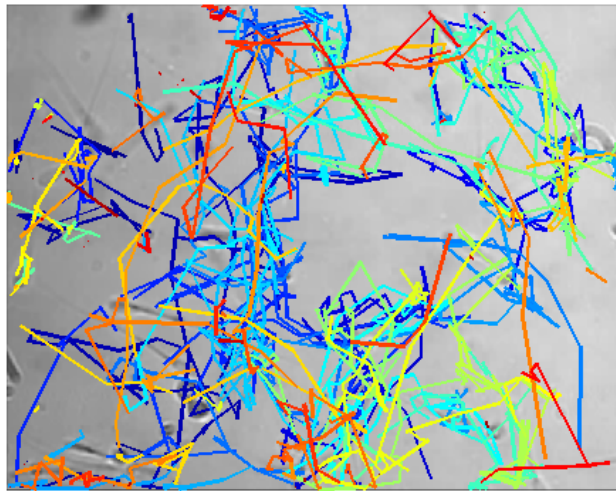
#### Teste 4:

##### Configuração:

```
mixparam.K = 3;  
mixparam.ALPHA = 0.02;  
mixparam.RHO = 0.01;  
mixparam.DEVIATION_SQ_THRESH = (7^2)/1.5;  
mixparam.INIT_VARIANCE = 3;  
mixparam.INIT_MIXPROP = 0.00001;  
mixparam.BACKGROUND_THRESH = 0.9;  
mixparam.COMPONENT_THRESH = 10;
```

##### Resultado:

Trajetórias: 308



Nessa amostra o limiar usado para diferenciar em qual componente um *pixel* se encaixa foi aumentado em 50% resultando em uma que de aproximadamente 40% do número de trajetórias, se comparado ao primeiro teste, sem nenhuma alteração.

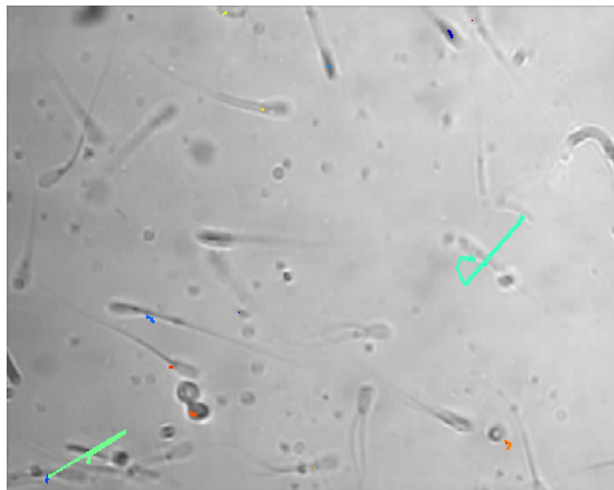
Teste 5:

Configuração:

```
mixparam.K = 3;  
mixparam.ALPHA = 0.02;  
mixparam.RHO = 0.01;  
mixparam.DEVIATION_SQ_THRESH = (7^2);  
mixparam.INIT_VARIANCE = 3;  
mixparam.INIT_MIXPROP = 0.00001;  
mixparam.BACKGROUND_THRESH = 0.9;  
mixparam.COMPONENT_THRESH = 10*25;
```

Resultado:

Trajetórias: 23



Nessa amostra o tamanho mínimo do limiar na detecção de objetos (COMPONENT\_THRESHOLD) foi aumentado em 25 vezes. O número de trajetórias caiu drasticamente, entretanto nenhuma trajetória relevante foi detectada. Isso pode querer dizer que o *threshold* está maior do que a área normal dos espermatozoides, impedindo que sejam detectados. Essa hipótese será confirmada no próximo teste.

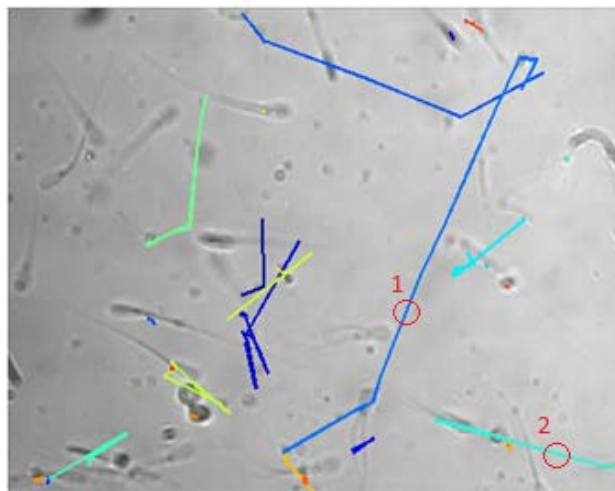
#### Teste 6:

##### Configuração:

```
mixparam.K = 3;  
mixparam.ALPHA = 0.02;  
mixparam.RHO = 0.01;  
mixparam.DEVIATION_SQ_THRESH = (7^2);  
mixparam.INIT_VARIANCE = 3;  
mixparam.INIT_MIXPROP = 0.00001;  
mixparam.BACKGROUND_THRESH = 0.9;  
mixparam.COMPONENT_THRESH = 5*25;
```

##### Resultados:

Trajetórias: 40



Um aumento e melhora na qualidade das amostras foi conseguido diminuindo o *threshold* do exemplo anterior pela metade. As trajetórias 1 e 2 apresentam o traçado similar ao apresentado durante a sequência de *frames* da amostra, colocando, desta forma, a configuração do Teste 6 como a melhor dentre as apresentadas.



## 5. Conclusão e Trabalhos Futuros

A motilidade espermática é determinante na avaliação da fertilidade animal. O conhecimento sobre as rotinas apresentadas foi aprofundado e uma vez que o significado das variáveis que as controlam foi melhor compreendido, testes foram realizados com diferentes configurações dos parâmetros da Mistura de Gaussianas e um melhor resultado foi alcançado no Teste 6. Apesar da melhor compreensão do funcionamento do algoritmo e, conseqüentemente, melhora nos resultados, nem todos objetos conseguiram ser rastreados. Para que se consiga superar essas deficiências, em trabalhos posteriores, dois pontos precisam ser melhor tratados: o melhoramento na qualidade das amostras, um aumento do contraste entre o espermatozoide e fundo da imagem (*background*) junto a eliminação de ruídos, visando uma melhor detecção dos espermatozoides no Passo 1; e além disso, os efeitos da alteração dos parâmetros do Filtro de Kalman também precisam ser testados, uma vez que influenciam fortemente na maneira, e qualidade, que os objetos serão associados *frame a frame*. Combinados estes dois pontos, espera-se conseguir um melhor resultado, próximo ao que se espera em uma situação real, e assim, com os espermatozoides devidamente rastreados, será possível mais facilmente colher informações estatísticas da amostra, como número e tipo de trajetória de um espermatozoide, e produzir um software que automatize um processo que é realizado de forma subjetiva.

## 6. Referências

- [1] CHIEN, S. et al. Efficient Moving Object Segmentation Algorithm.
- [2] Fabian Wauthier. Última atualização: 7 jun. 2012. Motion Tracker Project Synopsis. Disponível em: <<http://www.stats.ox.ac.uk/~wauthier/tracker/tracker-synopsis.pdf>>. Acesso em: 20 mar. 2014
- [3] Papa, F.O.; Crespilho, A.M.; Freitas Dell `Aqua, C.P.; Dell `Aqua Jr, J.A. Impacto do sêmen no sucesso dos programas de IATF: métodos básicos e avançados de avaliação.
- [4] KALMAN, R. E. A New Approach to Linear Filtering and Prediction Problems
- [5] BOVIK, A.C. Handbook of Image and Video Processing. 1ª Edição. 891.
- [6] STAUFFER, C.; GRIMSON, W.E. Learning Patterns of Activity Using Real-Time Tracking.
- [7] JONKER, R; VOLGENANT, A. A shortest augmenting path algorithm for dense and sparse linear assignment problems.
- [8] Bilgin Esme. Março, 2009. Kalman Filter For Dummies. Disponível em: <[bilgin.esme.org](http://bilgin.esme.org)>. Acesso em: 20 mar. 2014
- [9] BRAUNDMEIER, A.G.; MILLER, D.J. The search is on: finding accurate molecular markers of male fertility. Journal of Dairy Science, v.84, p.1915-1925, 2001.
- [10] Fabian Wauthier. Última atualização: 7 jun. 2012. Motion Tracking in Image Sequences. Disponível em: <<http://www.stats.ox.ac.uk/~wauthier/tracker>>. Acesso em: 20 mar. 2014.
- [11] GALE, D.; SHARPLEY, L.S. The American Mathematical Monthly. Vol. 69, No. 1.
- [12] R. A. Pilgrim. Munkres Assignment Algorithm. Disponível em: <<http://csclab.murraystate.edu/bob.pilgrim/445/munkres.html>>. Acesso em: 20 mar. 2014.
- [13] UNANIAN, M. M. Integridade da cromatina: método complementar para avaliação da qualidade do sêmen bovino. Brasília: Embrapa Recursos Genéticos e Biotecnologia, 2000. 21 p. Doc. 56.