Faculdade de Engenharia da Universidade do Porto

FEUP

EXTRACÇÃO DE INFORMAÇÕES RÍTMICAS DE MOVIMENTO EM DANÇA ATRAVÉS DE UM SINAL DE VÍDEO

André Miguel Passos Baltazar

Dissertação/Relatório de Projecto realizada(o) no âmbito do

Mestrado Integrado em Engenharia Electrotécnica e de Computadores

Major Telecomunicações

Orientador: Prof. Dr. Jaime S. Cardoso

Co-orientador: Prof. Dr. Carlos Guedes

Fevereiro de 2009

Resumo

O movimento corporal é um meio importante de percepção, expressão e interacção no mundo. Se o capturarmos numa sequência de vídeo e extrairmos as suas características, podemos encontrar o ritmo desse movimento. Ao usar esse ritmo para gerar sons desencadeia-se um tipo de interactividade entre o Homem e a Máquina e novas formas de arte são criadas.

O estado da arte actual continua longe de promover uma solução satisfatória para a interpretação corporal aplicada ao contexto musical. Para ajudar a preencher esta lacuna, foi criado o RitMoVídeo.

Este projecto tem como principais objectivos investigar e desenvolver uma aplicação, a funcionar em tempo real, capaz de estimar e executar ritmo musical pela análise de movimentos humanos captados em vídeo.

Foram também analisados e desenvolvidos algoritmos de segmentação e seguimento de objectos, assim como metodologias capazes de estimar ritmo através do seguimento efectuado.

Os principais procedimentos foram comparados, os resultados obtidos e as conclusões retiradas. Por fim os algoritmos mais adequados foram integrados sob uma plataforma comum e foi criada a aplicação, cumprindo assim os objectivos definidos.

Este relatório expõe todos os passos tomados no decorrer deste projecto, as qualidades e limitações actuais do sistema, assim como o trabalho futuro que se considera relevante.

Abstract

Bodily movement is an important mean to perceive, express and interact with the world. By capturing it in a video sequence and extracting its characteristics, one can find the rhythm of that movement. By using that rhythm to generate sounds one triggers one type of interactivity between Men and Machine and new art forms are created.

The current state of the art is still far from promoting a satisfying solution for the corporal interpretation applied to the musical context. To help filling this gap, application RitMoVídeo was created.

The main objectives of this project is to investigate and develop an application that works in real time and is able to estimate and execute musical rhythm by the analysis of human movements captured in video.

Tracking and developed segmentation algorithms were investigated, as well as methodologies that are able to estimate the rhythm through the performed tracking.

The main procedures were compared, the results obtained and the conclusions drawn. In the end, the best algorithms were integrated under a common platform and the application was created fulfilling all the defined objectives.

This report describes all the steps taken in de course of this project, the qualities and limitations of the system, as well as any future work considered relevant.

Agradecimentos

Gostaria de agradecer principalmente aos meus orientadores Prof. Doutor Carlos Guedes e Prof. Doutor Jaime S. Cardoso, pela paciência e disponibilidade demonstrada no decorrer do desenvolvimento deste projecto.

Agradeço também aos restantes investigadores do INESC Porto, com quem a convivência e discussão de várias ideias ao longo destes três meses permitiu a realização de um projecto mais consistente.

A nível mais pessoal, aqui deixo o meu agradecimento aos meus amigos de sempre e à Ana que me faz sorrir todos os dias.

Por fim, mas não menos importante, quero aproveitar para agradecer à minha família pelo amor e apoio demonstrado em todos os momentos da minha vida.

Índice de Conteúdos

Lista de Figuras

Lista de Tabelas

Abreviaturas

Lista de abreviaturas (por ordem alfabética):

AE	Sequência na auto-estrada
CO	Sequência no corredor
DC	*Direct Current*
DFT	Transformada Discreta de Fourier
EX	Sequência exterior
FEUP	Faculdade de Engenharia da Universidade do Porto
FFT	*Fast Fourier Transform*
FT	Sequência na fonte
HSV	*Hue and Saturation Value*
I&D	Investigação e Desenvolvimento
INESC Porto	Instituto de Engenharia de Sistemas e Computadores do Porto
KDE	*Kernel Density Estimation*
LAB	*Luminance A-B* (A e B são componentes cromáticos)
LUV	*Luminance, U (red vs green), e V(blue vs yellow)*
MEMS	*Micro-Electro-Mechanical Systems*
MIDI	*Musical Instrument Digital Interface*
MIEEC	Mestrado Integrado em Engenharia Electrotécnica e de Computadores
MoG	Mistura de Gaussianos
MSP	*MAX Signal Processing*
OpenCV	*Open Source Computer Vision Library*
PD	Pure Data
PF	*Principal Features*
Ravg	*Running Average*
RE	Sequência no restaurante
RF	Rádio frequência
RGB	*Red Green blue*
UTM	Unidade de Telecomunicações e Multimédia

Capítulo 1

Introdução

Esta dissertação insere-se no projecto de Mestrado Integrado em Engenharia Electrotécnica e de Computadores (MIEEC) da Faculdade de Engenharia da Universidade do Porto (FEUP). O projecto intitula-se Extracção de Informações Rítmicas de Movimento em Dança Através de um Sinal de Vídeo (RitMoVideo).

O trabalho decorreu nas instalações do Instituto de Engenharia de Sistemas e Computadores do Porto (INESC Porto), na Unidade de Telecomunicações e Multimédia (UTM).

Este capítulo inicia-se com uma breve descrição do INESC Porto e da UTM. Em seguida apresenta-se a motivação, o projecto e os objectivos do mesmo.

1.1 Apresentação do Instituto de Engenharia de Sistemas e Computadores do Porto (INESC Porto)

O INESC Porto – Instituto de Engenharia de Sistemas e Computadores do Porto é uma associação privada sem fins lucrativos reconhecida como instituição de utilidade pública, tendo adquirido em 2002 o estatuto de Laboratório Associado. Desenvolve actividades de investigação e desenvolvimento, consultoria, formação avançada e transferência de tecnologia nas áreas de Telecomunicações e Multimédia, Sistemas de Energia, Sistemas de Produção e Optoelectrónica.

O INESC Porto é uma instituição criada para constituir uma interface entre o mundo académico e o mundo empresarial da indústria e dos serviços, bem como a administração

pública, no âmbito das Tecnologias de Informação, Telecomunicações e Electrónica, dedicando-se a actividades de investigação científica e desenvolvimento tecnológico, transferência de tecnologia, consultoria e formação avançada. Procura pautar a sua acção por critérios de inovação, de internacionalização e de impacto no tecido económico e social, sobretudo pelo estabelecimento de um conjunto de parcerias estratégicas que garantam a sua estabilidade institucional e sustentabilidade económica.

A Unidade de Telecomunicações e Multimédia

A Unidade de Telecomunicações e Multimédia actua em áreas chave no âmbito das modernas redes e serviços de comunicação, em especial arquitecturas de redes, serviços de telecomunicações, processamento de sinal e imagem, microelectrónica, TV digital e multimédia.

Através da organização de grupos de Investigação e Desenvolvimento, realiza investigação e promove a formação avançada de recursos humanos, explorando nomeadamente financiamentos de programas de I&D europeus e nacionais. Participa em projectos europeus que permitem a cooperação científica e técnica com empresas e centros de I&D de vanguarda, a actualização tecnológica permanente e o acompanhamento da actividade de organismos de normalização. As actividades da UTM têm sido realizadas nomeadamente em parceria com operadores de redes e fornecedores de serviços de Telecomunicações, operadores de Televisão e fabricantes de sistemas de comunicação e de equipamento de teste.

1.2 Motivação

A motivação para integrar tecnologia na nossa maneira de viver já existe há muito, primeiro por questões de eficiência, depois para tarefas funcionais. Mais recentemente, também o mundo artístico começa a fazer uso desta mais-valia para melhorar o seu processo criativo.

O movimento corporal é um meio importante de percepção, expressão e interacção no mundo. Podemos mesmo encará-lo como uma forma de comunicação, um reflexo do nosso estado de espírito. Muitas vezes basta olhar para alguém e constatamos na forma como se move e gesticula que é uma pessoa alegre, triste, tímida, sexy, etc.

É através desse movimento que os bailarinos criam e comunicam a sua arte. Tradicionalmente, o coreógrafo cria a coreografia para uma música já existente, ou trabalha com um músico para compor uma música original para uma dança. Os bailarinos sincronizam os seus movimentos com a música e a coreografia é ensaiada repetidamente até estar consolidada com os sons e imagens visuais que a acompanham.

Num ambiente digital interactivo baseado no movimento, o coreógrafo e os bailarinos, através da execução dos seus movimentos, podem manipular a música e o estilo visual presente, alterando assim a aparência e sentimento da coreografia, de exibição em exibição. Facultar aos bailarinos a interacção com computadores, através de um vasto "vocabulário" de movimentos expressivos, permite-lhes executar a sua coreografia e exibição, das mais diversas formas e controlar vários aspectos da mesma.

Actualmente existem alguns sistemas que permitem a interacção Homem-Máquina através de movimentos capturados em vídeo. No entanto, o estado da arte actual continua longe de promover uma solução satisfatória para a interpretação corporal aplicada ao contexto musical. É neste sentido que surge o projecto RitMoVídeo.

1.3 Apresentação do Projecto

O RitMoVídeo é um projecto interactivo que funciona em tempo real. Neste pretende-se desenvolver uma aplicação, em que o computador analisa um sinal vídeo de entrada e estima o ritmo dos movimentos dos objectos presentes na cena.

Para tal, o desenvolvimento deste projecto passará, numa primeira fase, pela correcta segmentação *background*[1]/*foreground*[2]. Concluída esta fase com sucesso, a aplicação deverá separar as diversas componentes do objecto presente em *foreground* e para cada uma fazer a análise das suas características de movimento, de forma a estimar o ritmo a que se movimentam. Obtido o ritmo, este é usado para estabelecer o tempo musical de uma sequência em reprodução.

No exemplo da Figura 1, o objecto em análise será uma bailarina. Na sua representação em vídeo, o seu corpo será dividido em cabeça, tronco e membros e serão estas componentes que permitirão estimar o ritmo.

O projecto foi então abordado como uma cadeia composta por quatro blocos (Figura 1):

- Segmentação *background*/*foreground*
- Divisão dos componentes presentes (*tracking*[3])
- Extracção e análise de características
- Bloco de estimação de ritmo

Figura 1 - A imagem original à esquerda seguida da cadeia (quatro blocos) de análise.

[1] Região exterior aos objectos de interesse na imagem.

[2] Região ou objecto de interesse na imagem, neste caso a bailarina.

[3] Estimar a trajectória de um objecto presente num plano de imagem conforme este se move pelo cenário.

O método de desenvolvimento baseou-se no seguinte:

- Estudo das diferentes técnicas que permitem executar as tarefas propostas em cada bloco;
- Análise comparativa, dessas mesmas técnicas;
- Desenvolvimento de cada bloco, de acordo com a análise efectuada
- Melhoria de cada bloco da cadeia progressivamente;
- Integração dos módulos num sistema completo, em tempo real;

1.4 Objectivos

Os objectivos propostos para esta dissertação focaram essencialmente a problemática da automatização da extracção de características de movimento dos corpos presentes em *foreground* e a sua correcta análise ao nível rítmico. Neste sentido, o trabalho desenvolvido visou o cumprimento dos seguintes objectivos:

- Estudo comparativo e crítico de diferentes metodologias para segmentação foreground/background de uma sequência de vídeo em tempo real;
- Desenvolvimento de algoritmos automáticos de decomposição da silhueta de um corpo nos seus principais constituintes;
- Estudo de diferentes metodologias para a análise de características de movimento em tempo real;
- Desenvolvimento de algoritmos automáticos de análise das características de movimento;
- Desenvolvimento de algoritmos automáticos de geração e manipulação de ritmo e tempo musical, respectivamente;

1.5 Organização e temas abordados no presente relatório

Este relatório é composto por seis capítulos. O primeiro capítulo é dedicado ao enquadramento do projecto, contexto global do problema e objectivos a alcançar. Termina com uma pequena descrição das contribuições alcançadas.

No segundo capítulo “Estado da Arte” são descritos projectos e ferramentas existentes que permitem perceber as melhores estratégias de abordagem ao problema tendo em conta as virtudes e limitações de cada solução.

Após realizar a avaliação científica e tecnológica do estado da arte, no capítulo três “Comparação experimental de algoritmos de segmentação e *tracking*” são apresentados detalhadamente e comparados os principais algoritmos de segmentação e seguimento de objectos. A análise crítica evidencia os resultados e justifica qual o algoritmo a implementar.

O capítulo quatro “Estimação de Ritmo” revela a análise desenvolvida relativamente à extracção de características fundamentais do movimento captado. Detalha e compara as equações mais relevantes, assim como os principais algoritmos que permitem estimar o ritmo em tempo real. Tal como na secção anterior termina com uma análise crítica sobre os resultados e determina os algoritmos a utilizar.

O capítulo “Integração” inicia com a visão global da arquitectura do sistema. Depois, os algoritmos implementados para cada bloco são descritos por ordem cronológica de funcionamento. São descritas as principais limitações e dificuldades do sistema.

Por fim, o sexto e último capítulo resume as principais conclusões obtidas. É também sugerido o trabalho futuro.

1.6 Contribuições relevantes

O desenvolvimento do projecto RitMoVídeo gerou várias questões interessantes. De certa forma, as soluções encontradas para os problemas que surgiram apresentam-se como contribuições relevantes para as mais diversas áreas de investigação e desenvolvimento.

Na área da segmentação de imagens, é de destacar o estudo comparativo de qualidade *versus* tempo de execução dos principais algoritmos de segmentação e a respectiva implementação do que foi considerado mais adequado.

Ao nível do *tracking*, foi realizada uma análise que demonstra a melhor forma de representar os movimentos humanos. Foi desenvolvido um algoritmo que permite o *tracking* e divisão do corpo humano nos seus diversos componentes.

Os estudos e algoritmos desenvolvidos ao nível de extracção de características e estimação de ritmo apresentam uma contribuição significativa para aplicações, não só do mesmo contexto, como para outras, por exemplo, didácticas ou médicas.

Por fim, ao nível de aplicações vocacionadas para o mesmo propósito esta apresenta-se como uma boa alternativa relativamente às criadas previamente, pois permite a escolha da componente do corpo humano sobre a qual se pretende calcular o ritmo.

Devido à sua interactividade o RitMoVídeo foi escolhido para representar o INESC Porto, no Dia da Universidade do Porto a decorrer em Março de 2009.

Capítulo 2

Caracterização do Problema

Tendo em conta os objectivos delineados para o trabalho, neste capítulo descreve-se a investigação sobre projectos existentes cujos resultados possam ser aplicados no sistema proposto. A abordagem adoptada para a análise do estado da arte não se cingiu à análise de projectos de segmentação background/foreground. Na verdade, o contexto foi alargado às principais técnicas de *tracking* de objectos em vídeo e a projectos com contributos relevantes na análise de ritmo.

O esclarecimento dos conceitos básicos subjacentes a estas técnicas permite compreender a sua aplicação em situações concretas, as quais serão introduzidas posteriormente.

Serão expostas também as principais plataformas de programação de projectos deste género.

2.1 Estado da Arte

Nas últimas duas décadas, várias técnicas têm sido desenvolvidas e aperfeiçoadas para suportar os requisitos de software de sistemas combinados de música e animação criadas e executadas em tempo real por um computador. As técnicas incluem novos ambientes de programação que suportam a integração de várias funções e processos que se podem executar em tempo real.

Existem vários métodos para traduzir os movimentos físicos em dados digitais que possibilitam controlar parâmetros musicais. Axel Mulder [1], na década de 90, distinguiu três técnicas que ainda hoje são uma referência para o *tracking* dos movimentos humanos: Inside-In (de dentro para dentro), Inside-Out (de dentro para fora) e Outside-In (de fora para dentro).

Nas subsecções seguintes estas três técnicas serão detalhadas e exemplificadas.

2.1.1 Inside-In

As técnicas *Inside-in* recorrem a sensores de forma a mapear os movimentos dos membros do corpo humano, por exemplo braços ou dedos. Estes sensores podem ser luvas, sensores flexíveis *piezo-eléctricos* (medem os ângulos das articulações), acelerómetros, giroscópios[4] e inclinómetros[5]. Em geral, não permitem medir a rotação e não são dependentes de um ponto de referência. Uma vez que os sensores são posicionados no corpo, esta técnica é considerada intrusiva e invasiva. A necessidade de cabos de alimentação e comunicação nestes sensores pode interferir com a liberdade de movimentos.

[4] Dispositivo que consiste de um rotor suspenso por um suporte formado por dois triângulos articulados. O seu funcionamento baseia-se no princípio da inércia. Serve como referência de direcção, mas não de posição, ou seja, permite medir com precisão qualquer mudança na sua orientação, excepto rotações que ocorram no plano de giro dos discos do giroscópio.

[5] Dispositivo referente à inclinação do objecto relativamente ao eixo horizontal

Exemplo: Inside-In

A empresa Measurand [2] utiliza fibras ópticas flexíveis e sensores geotécnicos, recorrendo à tecnologia MEMS (Micro-Electro-Mechanical Systems), que consiste na integração de elementos mecânicos, sensores, actuadores e electrónica num substrato de silicone com micro-tecnologia para mapear os movimentos. As fibras podem ser aplicadas a coletes, luvas (Figura 2) e outras peças de vestuário.

Figura 2 - Luva Measurand (de [2]).

2.1.2 Inside-Out

As técnicas *inside-out* empregam sensores no corpo que permitem efectuar o *tracking* a partir de fontes externas. Podem ser usados para medir o movimento dos grandes membros do corpo humano (braços e pernas), assim como fornecer informações acerca da posição da pessoa relativamente ao cenário. Estes sensores podem ser acelerómetros, giroscópios, ou leds infravermelhos colocados no corpo, o que retorna ao mesmo problema de obstrução e invasão relatado anteriormente.

Exemplos: Inside-Out

Existem vários tipos de sensores que permitem mapear os movimentos executados por uma pessoa.

1) Em 2002, Mark Feldmeier [3], desenvolveu transmissores de frequências rádio (RF) (Figura 3) que usam acelerómetros e transmitem dados para um computador que executa a *Fast Fourier Transform*[6] (FFT) e outros algoritmos de processamento de sinal. Assim, extrai o ritmo musical e, consoante as mudanças de tempo dos bailarinos, o sistema adapta a música.

O funcionamento do sistema é bastante básico, os sensores acelerómetros estão configurados para uma determinada aceleração, quando esta é ultrapassada um sinal RF é enviado para uma "base" que é actualizada de 2ms em 2ms. Esta informação é enviada por uma porta MIDI (*Musical Instrument Digital Interface*) para um computador onde é processada no ambiente de programação *Max*[7]. O programa executa a FFT do sinal (de 10 em 10 segundos, por exemplo) e retorna a frequência fundamental. Através desta frequência obtém-se o tempo musical.

O *Max* é usado diversas vezes para programar este tipo de sistemas e os sensores pesam apenas 5gr, têm uma duração de bateria elevada e o preço é de apenas 8 dólares por cada 10 unidades.

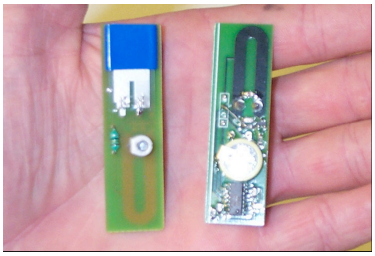


Figura 3 - Protótipo dos transmissores RF com acelerómetros embutidos (de [3]).

2) Em 2006, Urs Enke [4], apresentou o DanSense. Este projecto consiste em analisar o ritmo de movimentos humanos em tempo real, com recurso a sensores acelerómetros aplicados no corpo. Os dados extraídos com os acelerómetros permitem distinguir as magnitudes dos movimentos e a sua distribuição no tempo. Assim, utilizando uma

[6] Transformada rápida de Fourier - é um algoritmo eficiente para se calcular a Transformada Discreta de Fourier (DFT) e a sua inversa.

[7] Max – programa desenvolvido por Miller S. Puckette em meados da década de 80, é altamente modular e a maior parte das rotinas existem sobre a forma de bibliotecas partilhadas (open source). Explicado na pag. 19.

combinação de análise espacial e espectral (Transfomada de Fourier) desses dados, Enke consegue extrair os padrões de ritmo. O seu projecto foi testado com movimentos simples e complexos, atingindo resultados satisfatórios. No entanto, conforme se exemplifica na Figura 4, esta metodologia pode ser um pouco intrusiva do ponto de vista da sua utilização em dança.

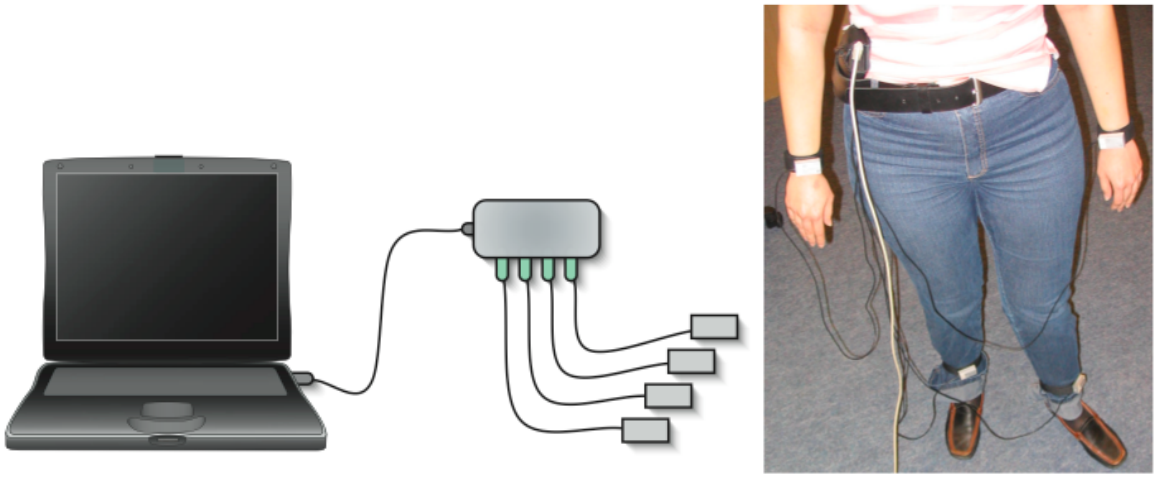


Figura 4 – O hardware de DanSense e a sua aplicação no corpo humano (de [4]).

3) O sistema "MCM: Motion Capture Music" [5] foi desenvolvido capturar movimentos a partir do sistema Vicon8 [6] e usar essa captura para controlar dados, por exemplo MIDI.

O sistema Vicon8 captura informação tridimensional acerca da localização exacta de pontos num corpo, que é rodeado por oito câmaras apoiadas por luzes estroboscópicas (vulgarmente conhecidas por flash). No corpo são colocados marcadores especiais que reflectem a luz e as imagens obtidas pelas oito câmaras são usadas para calcular as coordenadas cartesianas X, Y, Z de cada marcador. Uma configuração típica utiliza trinta ou mais marcadores que são detectados a uma taxa de trinta, sessenta, ou cento e vinte *frames*[8] por segundo. Os dados podem ser armazenados num ficheiro para uso futuro, ou com o sistema Vicon RT (funciona em tempo real) podem ser enviados directamente para software de edição gráfica para controlar uma animação.

A Figura 5 ilustra a arquitectura geral do sistema.

[8] Uma imagem de uma sequência de vídeo.

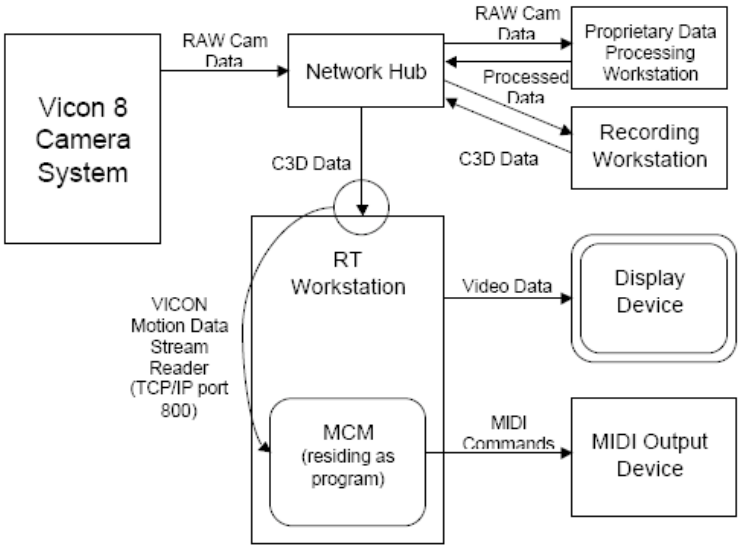


Figura 5 – Visão geral da arquitectura do sistema MCM (de [5]).

2.1.3 Outside –In

As técnicas *Outside-In* recorrem a sistemas electro-ópticos que conseguem efectuar o *tracking* de marcadores reflectores, ou fontes naturais no corpo. Um exemplo simples desses sistemas é uma câmara de vídeo externa que permite fazer o *tracking* do corpo. As técnicas *Outside-In* são as menos intrusivas das três e são indicadas para mapear grandes partes do corpo, mas apresentam vários problemas em fazer o mapeamento de partes mais pequenas, como dedos. A iluminação do cenário também pode influenciar o correcto funcionamento do sistema. Estas técnicas requerem, em geral, uma elevada capacidade de processamento computacional e estão limitadas pela oclusão.

Apesar de todas estas contrapartidas, a metodologia *Outside-In* apresenta-se como a melhor alternativa para o projecto em questão. Assim, foi efectuada uma investigação mais detalhada a nível de exemplos, que se expõem de seguida.

Exemplos: Outside-In

A aquisição sem sensores é efectuada recorrendo apenas às câmaras de vídeo e software de análise de imagem. Existem já aplicações desenvolvidas com propósitos

semelhantes ao proposto neste trabalho e outras, que foram desenvolvidas para outros fins, como terapêuticos ou armazenamento de dados digitais, mas que podem contribuir com métodos de análise e processamento de imagem para este trabalho.

1) Sistema desenvolvido para armazenamento de dados digitais

O projecto de Atsushi Nakazawa [7], refere técnicas para gerar um arquivo digital das tradições e heranças culturais do povo japonês. O objectivo principal é arquivar passos de dança, detectar passos primitivos (básicos).

Estes passos têm se estar sincronizados com o ritmo da música. Por isso, foi desenvolvido um método que segmenta o movimento de acordo com o ritmo musical para extrair os passos básicos. Divide a captação do sujeito em centro de massa, pés e mãos, e através da conjugação com o tempo da música consegue extrapolar se foi feita uma sequência de movimentos ou não (considera que uma sequência de movimentos é correcta se existe uma pausa que corresponde ao tempo musical). O método de extracção do ritmo, assim como o processo implementado na conjugação de ritmo e imagem, pode ser consultado na Figura 6.

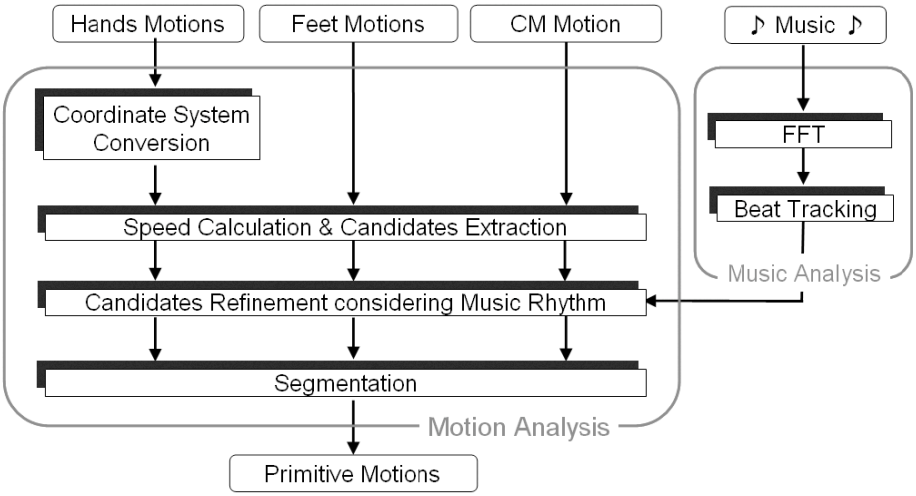


Figura 6 – Fluxograma de funcionamento do sistema de Nakazawa (de [7]).

2) Sistema desenvolvido para fins terapêuticos

O "Music Maker", desenvolvido por Mikhail Gorman [8], permite às pessoas com debilitações físicas que as impedem de tocar um instrumento convencional, fazerem música enquanto executam os seus exercícios terapêuticos. Usa técnicas de visão computacional para converter os movimentos dos membros das pessoas (dedos, mãos ou pés) em sons ou ambientes visuais. Pode ser ajustado às particularidades terapêuticas de cada paciente e fornece ferramentas quantitativas para controlar o processo de recuperação e estabelecer novos objectivos terapêuticos.

É utilizado o software *Eyesweb* [9] e a detecção e orientação geométrica do objecto são feitas por análise de cor de pele humana, pelas suas componentes de cor características: vermelha, verde e azul (Figura 7).

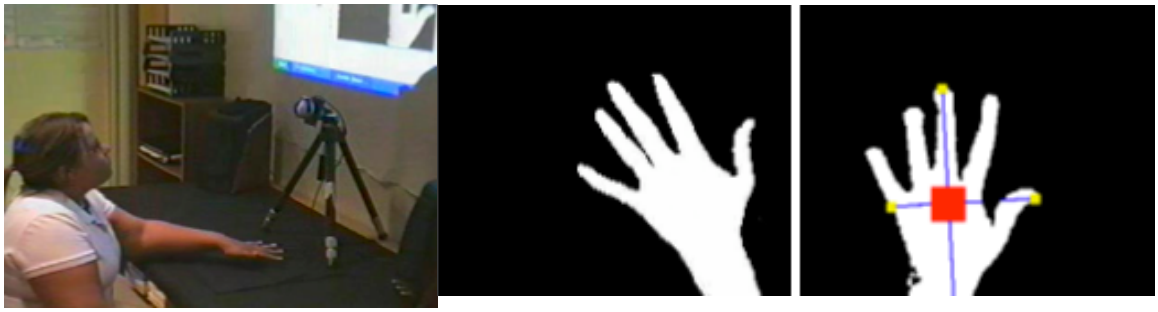


Figura 7 – A mão é captada pela câmara (esquerda), a imagem é segmentada (centro) e a orientação e centro do objecto calculado (direita) (de [8]).

Para obter melhores desempenhos, no início de cada terapia permite a programação das cores, visto existirem diversos tons de pele de pessoa para pessoa. A Figura 8 demonstra de uma forma geral como funciona o programa.

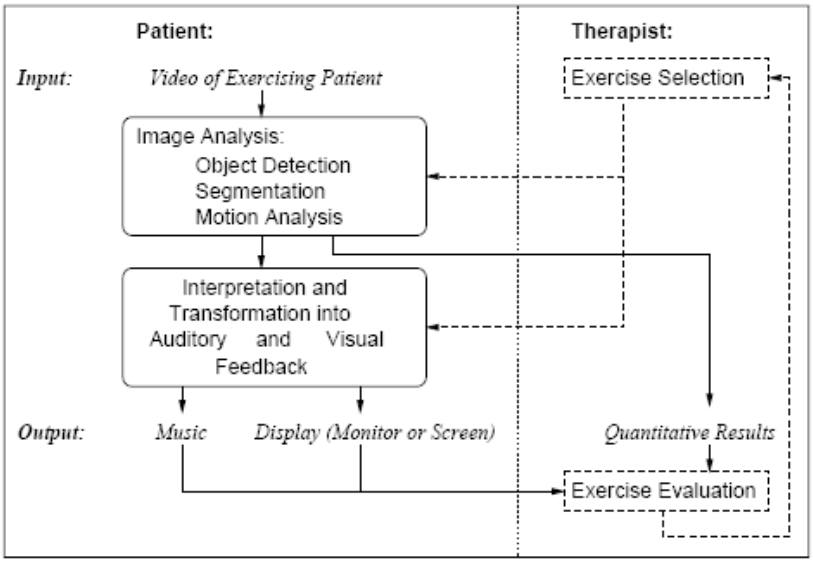


Figura 8 - Visão conceptual do Music Maker (de [8]).

3) Sistemas desenvolvidos com fins artísticos

a) Em 2004, António Camurri e colegas [10], desenvolveram um estudo que pretende analisar se os observadores de uma dança conseguem antecipar o próximo passo do bailarino. Usa métodos para detectar o centro de massa dos bailarinos, apenas por análise de imagem e cria a linha temporal do centro de massa do início ao fim da performance.

Na experiência discutida neste estudo, o *Eyesweb* foi responsável por extrair automaticamente as coordenadas 2D do centro de massa de uma pessoa presente num vídeo, apresentar o vídeo aos observadores e gravar as respectivas respostas.

A posição do centro de massa da silhueta do bailarino, presente no vídeo, é obtida utilizando dois algoritmos incluídos nas bibliotecas do *Eyesweb*. O primeiro aproxima a posição do centro de massa através da primeira ordem dos momentos da silhueta em 2D. Apesar desta medida ser precisa para muitas aplicações diferentes, muitas vezes sofre erros devidos ao ruído ou a uma segmentação errada dos membros do corpo humano. Uma medida mais robusta foi alcançada com uma técnica que emprega a projecção de padrões espácio-temporais. Embora não seja em tempo real, esta técnica permite a extracção de sub-regiões da silhueta. A sub-região maior é usualmente associada ao tronco e as outras aos membros. As sub-regiões ajudam, assim, a reduzir a área em que o cálculo do centro de massa tem de ser efectuado, reduzindo também o ruído ou eventuais erros introduzidos

pela detecção dos membros. A Figura 9 demonstra um exemplo da extracção das sub-regiões de um bailarino. Podemos verificar que o rectângulo azul é a fronteira da silhueta total. A sub-região dentro do rectângulo central verde inclui o tronco, cabeça e pernas, enquanto as duas sub-regiões a vermelho delimitam os braços. O ponto no centro do corpo demonstra o centro de massa calculado a partir do rectângulo verde e o ponto inferior, junto aos pés do bailarino, representa o centro de massa projectado no chão.

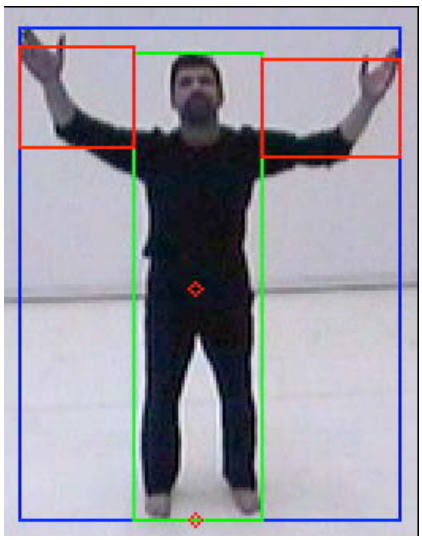


Figura 9 - Extracção das sub-regiões da silhueta e do centro de massa a partir das coordenadas 2D (de [10]).

b) Em 2005, Carlos Guedes [11] constata que se registar o número de *pixels* que mudaram a sua claridade em acções de movimento periódico ao longo do tempo, torna-se possível detectar periodicidades no sinal de análise de vídeo, que têm correspondência directa com as acções executadas. Removendo a componente DC (contínua) desses sinais, pode-se verificar uma elevada similaridade com sinais acústicos periódicos. Isto significa que se aplicar um algoritmo capaz de detectar as periodicidades deste sinal, como a Transformada Rápida de Fourier (*Fast-Fourier Transform* - FFT), por exemplo, pode-se detectar o ritmo presente nesse sinal, calcular a sua frequência fundamental e assim obter o tempo que pode ser usado para fazer algo relevante musicalmente, tal como a geração de ritmos a partir do movimento de dança ou possibilitar ao bailarino o controlo, em tempo real, do tempo musical.

De forma a executar as tarefas descritas anteriormente, Guedes expõe a implementação de cinco objectos externos (m.bandit, m.peak, m.weights, m.clock e m.sample) para o programa *Max/MSP/Jitter*. Os objectos extraem informação do

movimento a partir de uma análise prévia do vídeo por *frame-differencing*[9] e permitem, ao bailarino, a geração de ritmos ou o controlo do tempo de uma sequência musical gerada em tempo real. Estes objectos podem ser divididos em objectos de análise (m.bandit, m.peak, m.weights) e objectos de processamento (m.clock e m.lp). De seguida apresenta-se um pequeno resumo acerca de cada objecto para compreender o seu funcionamento:

Objectos de análise:

m.bandit –É um dos objectos fulcrais desta biblioteca. Consiste num banco de 150 filtros IIR de 2ª ordem onde, em cada, é aplicado o algoritmo de Goertzel [12]. Este objecto tem como entrada a representação temporal do sinal extraído por *frame-differencing* e é capaz, entre outras, de estimar e extrair a frequência fundamental do sinal (Figura 10).

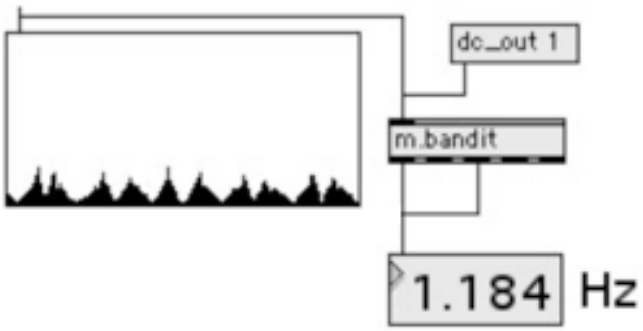


Figura 10 - O m.bandit retorna a frequência fundamental do sinal analisado (de [11]).

m.peak – tem, como entrada, a representação das variações de tempo resultantes do *frame-differencing* do sinal e, como saída, uma mensagem de execução, quando é atingido um pico significativo no sinal. Este objecto é especialmente útil para executar eventos (sons ou mensagens para outros objectos) quando ocorrem movimentos bastante acentuados.

m.weights – tem como entrada, o valor da frequência fundamental, calculado pelo m.bandit através do *frame-differencing* do sinal. Tem como saída, o valor que foi mais vezes transmitido pelo m.bandit nos últimos 60 *frames*. Uma das funções do m.weights é

[9] Diferenciação de frames, técnica de segmentação que consiste em subtrair os frames actuais a um anterior, pré-marcado como fundo.

fornecer uma "memória curta" ao m.clock, que pode ser utilizada para estabelecer o tempo de uma sequência de movimentos sem uma estimativa prévia.

Objectos de processamento:

m.clock – outro dos objectos fulcrais desta biblioteca. É um relógio adaptativo que permite ao bailarino controlar o tempo de uma sequência musical. Este objecto considera a frequência fundamental transmitida pelo m.bandit (convertida em milisegundos) como um candidato ao tempo musical e permite fazer a adaptação temporal da sequência se esse candidato está dentro de limites estabelecidos, a partir da estimação de tempo anterior.

m.sample – actua como um amostrador da diferença de luminosidade calculada pelo algoritmo que executa o *frame-differencing*. Este objecto foi elaborado para estabilizar a taxa de imagens captadas pelas câmaras USB. As taxas de captação de imagens das câmaras podem oscilar entre 23 e 34 *fps* (*frames* por segundo). Uma vez que o processamento efectuado pelo m.bandit depende da taxa de amostragem, é crucial que essa amostragem seja estável. Este objecto é, assim, responsável por garantir o fornecimento dos valores de diferença de luminosidade com um fluxo estável ao m.bandit.

c) Em 2006, Luiz Naveda [13] propõe a representação digital dos movimentos de dança do Samba. Este género musical é entendido pelos músicos, bailarinos e ouvintes como um fenómeno no qual a música e a dança estão intrinsecamente relacionadas, no entanto não existe um conhecimento aprofundado acerca da estrutura de ambos os domínios. Assim, Naveda desenvolveu ferramentas que permitem relacionar, ao nível métrico, a música e dança. Para tal, escolheu três excertos musicais de Samba para os quais gravou sequências de vídeo de bailarinos a dançarem.

Apesar de recorrer à segmentação manual dos vídeos, Naveda descreve algoritmos e métodos heurísticos relevantes para a conjugação da música com a dança. Como exemplo dos resultados obtidos pode-se visualizar na Figura 11 os movimentos periódicos detectados em várias métricas musicais.

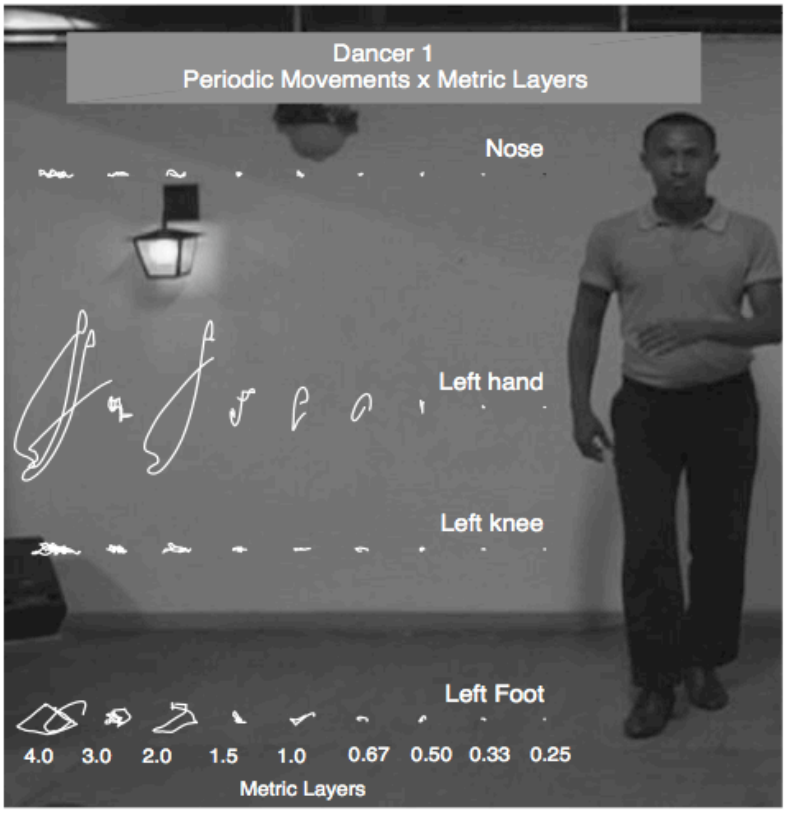


Figura 11 – As várias componentes do bailarino e o movimento efectuado ao longo de várias divisões métricas do tempo musical (de [13]).

2.2 Plataformas de programação

Existem inúmeras plataformas de programação para os mais diversos propósitos. No contexto deste projecto são descritas, em seguida, as mais proeminentes.

2.2.1 Max/MSP/Jitter

O ambiente de programação *Max* [14] foi desenvolvido por Miller S. Puckette em meados da década de 80, é altamente modular e a maior parte das rotinas existem sobre a forma de bibliotecas partilhadas (*open source*). O componente MSP ("*Max Signal Processing*") do *Max/MSP/Jitter* surgiu, em 1997, como uma biblioteca poderosa que permite a manipulação de sinais áudio-digitais em tempo real, possibilitando aos utilizadores criar o seu próprio sintetizador e processador de efeitos. Em 2003 surgiu o *Jitter*, a componente que veio permitir o processamento em tempo real de vídeo, matrizes e imagens em três dimensões.

Actualmente, o *Max/MSP/Jitter* possui mais de cento e cinquenta objectos que cobrem os elementos básicos de síntese, amostragem e processamento de sinal. Permite que o software seja criado através da junção desses diversos objectos existentes na sua biblioteca, ou através de objectos externos projectados pelo programador em linguagem C ou C++. Além disso, possibilita controlar facilmente, com o computador, diverso hardware e vice-versa. Devido à sua capacidade extensível e interface gráfica, é vastamente reconhecido como a "*língua franca*" para desenvolver software de música e performance interactivas. Este ambiente de programação gráfico é desenvolvido e comercializado pela empresa *Cycling '74*. É usado por compositores, performers, programadores, investigadores e artistas interessados em criar software interactivo.

2.2.2 Pure Data

Em 1996, Miller S. Puckette, disponibilizou um programa, de uso livre, denominado "*Pure Data*" (PD) [15]. O PD é muito similar ao *Max*, tem uma base modular de objectos externos que são usados como blocos de construção para os programas. Isto torna o programa altamente extensível através de uma interface de programação e encoraja os programadores a adicionarem as suas próprias rotinas, seja em linguagem C, Python, Ruby ou outras linguagens. Com a adição do ambiente gráfico para multimédia, é possível criar e manipular: imagens, vídeo e até gráficos *OpenGL*[10] em tempo real e conjugá-los com uma infinidade de possibilidades de interactividade, nomeadamente, com áudio digital ou sensores externos, por exemplo.

2.2.3 Eyesweb

A plataforma aberta *Eyesweb* foi originalmente concebida para o desenvolvimento de aplicações em tempo real de dança, música e multimédia. É baseada na biblioteca *OpenCv* da *Intel* (*Open Source Computer Vision Library*). A biblioteca *OpenCv* da Intel é uma colecção de códigos (funções em C, C++, classes de programação e algoritmos populares) desenvolvidos e disponibilizados por investigadores de todo o Mundo. O *Eyesweb* permite

[10] Open Graphics Library, é uma linguagem gráfica 3D desenvolvida pela Silicon Graphics.

ao utilizador experimentar modelos computacionais e mapear gestos de diferentes modalidades (desporto, dança, etc.) que depois podem ser transfigurados para eventos multimédia (sons, música e efeitos visuais). Permite um desenvolvimento célere de *performances* interactivas, através da inclusão de um ambiente de programação visual, que permite a transfiguração dos gestos a vários níveis, desde movimentos em sons, até movimentos em música integrada, ambientes visuais ou alteração de cenários.

2.2.4 Isadora

O Isadora [16], criado por Mark Coniglio, é um ambiente de programação gráfico para Macintosh (existe uma versão beta para Windows). É comercializado pela TroikaTronix e fornece controlo interactivo sobre dados digitais, com especial ênfase pela manipulação de vídeo em tempo real.

Uma vez que qualquer exibição ou instalação é única, o *Isadora* foi desenvolvido, não para ser um programa "plug and play", mas para fornecer blocos de construção que podem ser ligados entre si de forma flexível, permitindo aos utilizadores seguirem os seus impulsos artísticos. De entre as suas características, são de destacar, entre outras, os módulos de processamento de vídeo em tempo real e a aceleração de *hardware* permitida.

2.2.5 Processing

O Processing [17] surgiu em 2001 e é uma plataforma de programação centrada nas artes visuais. Inicialmente foi criado para servir como uma ferramenta de aprendizagem de programação básica, mas rapidamente se transformou numa ferramenta para projectos mais elaborados. Visto que é uma plataforma livre, os programadores que o utilizam partilham, normalmente, os seus programas e códigos. Com as contribuições da comunidade de utilizadores foi possível estabelecer várias bibliotecas, das quais se destacam, no contexto deste projecto, as de visão computacional e música.

2.3 Resumo e Análise Crítica

Este capítulo apresentou a revisão do estado da arte. Adquirir conhecimento nestes campos foi um passo necessário de forma a fornecer as bases para o desenvolvimento de uma aplicação que consiga analisar em tempo real as periodicidades presentes em movimentos de dança.

Pela análise dos sistemas e métodos descritos anteriormente, pode-se concluir que existem vários sistemas de detecção e mapeamento de movimentos capazes de gerar as mais diversas saídas. No entanto, existe ainda espaço para a continuação do desenvolvimento destas tecnologias de forma a implementar novos algoritmos de processamento que permitam uma optimização dos resultados e aumentar as funcionalidades de suporte à criatividade dos artistas.

As técnicas *Inside-In* e *Inside-Out*, pelas suas características intrusivas, foram preteridas em relação à Outside-In.

No capítulo seguinte apresenta-se um estudo acerca das diferentes técnicas de segmentação e tracking.

Capítulo 3

Comparação Experimental de Algoritmos de Segmentação e *Tracking*

Neste capítulo será efectuada uma análise comparativa dos algoritmos de segmentação mais proeminentes assim como das principais técnicas de *tracking*. São apresentados resultados para cada subsecção.

3.1 Segmentação de imagens

A segmentação de uma imagem consiste na sua decomposição nos seus elementos constituintes. Um exemplo simples deste processo seria a segmentação de uma imagem panorâmica de uma praia decompondo-a em zonas correspondentes às classes "mar", "céu" e "areia". Tendo em conta que a segmentação automática de imagens é um problema comum em aplicações que envolvem processamento de imagem e/ou vídeo, foram desenvolvidos vários procedimentos que se encontram descritos na literatura da especialidade [18].

Existem diversas estratégias que podem ser seguidas na resolução de problemas de segmentação, sendo que a eficiência de cada algoritmo está intrinsecamente relacionada com as características das imagens processadas.

Provavelmente devido à sua simplicidade, o método mais comum de distinguir um objecto que se move relativamente a um fundo, será a subtracção de fundos (*background subtraction*). O método consiste em subtrair à imagem actual, uma imagem de referência, obtida previamente. Os segmentos de imagem que não se alteraram são considerados *background*, enquanto os restantes segmentos (que sofreram alterações) são considerados

como objectos (foreground). No entanto, se o modelo de referência do background não for actualizado adequadamente, esta técnica fica gravemente susceptível às condições de ambiente em que se insere, como mudanças de luz, por exemplo.

Para alcançar uma modelação de *background* robusta, são necessárias técnicas que melhor se adaptem a comportamentos dinâmicos. Idealmente, o desempenho não deve depender da posição da câmara, nem deve ser sensível ao que acontece no seu campo de visão ou aos efeitos de luz usados. A modelação deve ser capaz de lidar com a oclusão momentânea de objectos, sombras, mudanças de luz, objectos que se movem lentamente e mudanças de cenário.

É também importante ter em conta que, para a aplicação em causa, a análise terá de ser efectuada em tempo real. Uma modelação complexa poderá apresentar resultados mais exactos, mas o tempo de análise será demasiado para o que se pretende. Deveremos assim encontrar um compromisso entre qualidade do algoritmo e a sua complexidade computacional.

É neste sentido que foi realizada uma análise comparativa experimental de diversos métodos existentes na actualidade.

Com fundamento em [19], foi efectuada uma comparação das técnicas mais proeminentes de segmentação, partindo da mais básica até à mais complexa. Em seguida apresenta-se um pequeno resumo de cada, assim como os resultados comparativos obtidos.

3.1.1 Running average (RAvg)

O *background* pode ser modelado como uma média das imagens anteriores mas, de forma a evitar requisitos elevados de memória, esta média é aproximada por um filtro adaptativo com uma taxa de aprendizagem α. Cada *pixel* do fundo, na posição (i,j) no instante t é dado por:

$$B_{(i,j)}(t) = \alpha I_{(i,j)}(t) + (1-\alpha) B_{(i,j)}(t-1) \quad (1)$$

O *foreground* é assim estimado usando uma subtracção limiar da imagem actual com a estimada como *background*. Esta técnica é provavelmente a mais simples e de implementação mais rápida. Os resultados ficam longe dos ideais, em particular em fundos

complexos. Uma vez que apenas se considera uma representação estática do fundo para fazer a subtracção, quando ocorre algum tipo de mudança dinâmica no fundo, este é incorrectamente classificado como *foreground*. O método *Running Average* representa o algoritmo básico.

3.1.2 Mixture of Gaussians (MoG)

Em vez de estimar directamente a representação do *background*, um método mais eficaz é construir um modelo que consegue prever o comportamento de cada *pixel*, usando o seu histórico. Tal pode ser obtido calculando a função densidade probabilidade (f.d.p.) de cada pixel. Assumindo que cada mudança estrutural que afecta o valor do pixel é causada por vários processos, cada modelado por uma distribuição Gaussiana podemos definir a probabilidade de observar o seu valor como:

$$P(v_t) = \sum_{k=1}^{k} P(G_k)P(v_t \mid G_k) = \sum_{k=1}^{k} \omega_k \eta(v_t, \mu_k, \sigma_k) \qquad (2)$$

Onde G_k é o "K-ésimo" Gaussiano de K distribuições, η é a função densidade normal, ω_k, μ_k e σ_k são, respectivamente, uma estimativa do peso, o valor médio e a variância de G_k. Além disso, pode ser facilmente provado ([20]) que, dado o vector de cor v_t de um pixel, a probabilidade de este pertencer ao *background* é:

$$P(B \mid v_t) = \frac{\sum_{k=1}^{K} P(v_t \mid G_k)P(G_k)P(B \mid G_k)}{\sum_{k=1}^{K} P(v_t \mid G_k)P(G_k)} \qquad (3)$$

Se $P(B \mid v_t) > T_{MoG}$ (onde T_{MoG} é um *threshold*[11] estabelecido) então o pixel é considerado *background*.

[11] Limiar de decisão, condição fronteira.

3.1.3 Kernel Density Estimation (KDE)

É possível aproximar a f.d.p. do background de cada pixel pelo histograma dos valores mais recentes classificados como *background*. Este método tem, no entanto, alguns problemas, nomeadamente, o facto de o histograma ser uma função discreta, muitas vezes, conduz a uma f.d.p. errada. Um modelo não-paramétrico baseado no KDE é proposto em [21]. O KDE garante uma representação contínua do histograma. A f.d.p. do background é dada pela soma dos *kernels* Gaussianos centrados nos N mais recentes valores de *background*:

$$P(v_t) = \frac{1}{N}\sum_{k=1}^{N}\eta(v_t - v_k, \sum_k) \qquad (4)$$

O pixel é classificado como *background* se $P(v_t) > T_{KDE}$. Um problema importante deste algoritmo é o cálculo do $\sum_k$ - a largura de banda *kernel*. Em [20] é proposta uma matriz diagonal para o cálculo da largura de banda.

3.1.4 Principal Features (PF)

Em [22] o *background* é representado em cada pixel pelas suas características mais frequentes, ou características principais. A classificação é efectuada usando um critério "Bayesiano" e demonstra que um pixel representado por v é classificado como *background* se $2P(v \mid B)P(B) > P(v)$.

Caso contrário é foreground. No entanto é necessário calcular à priori os valores de $P(v \mid B)$, $P(B)$ e $P(v)$. Como mencionado anteriormente, uma forma de calcular estes valores será construir o histograma temporal de cada pixel.

3.2 Teste comparativo dos algoritmos de segmentação

Um modelo geral para a comparação de segmentação de imagens foi proposto recentemente na literatura [23 e 24]. Esse modelo foi usado, com uma ligeira alteração na métrica.

O critério baseia-se no grafo de intersecção entre duas segmentações. Dois nós são conectados por uma aresta pesada se e só se essas duas regiões se intersectam (Figura 12).

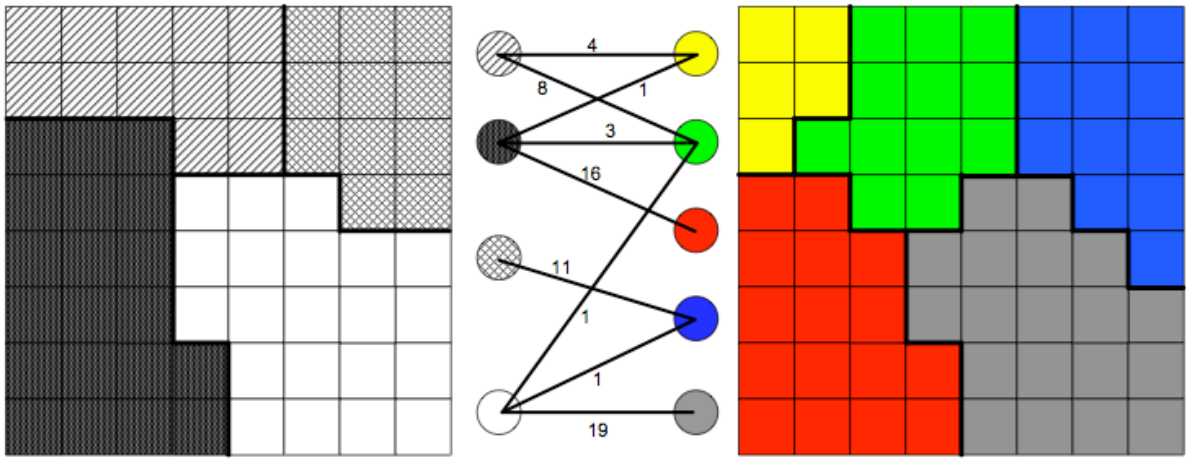


Figura 12 - Grafo de intersecção para duas segmentações (de [19]). Os pesos correspondem ao número de *pixels* na intersecção.

O grafo de intersecção associado a dois tipos de segmentação diferentes pode ser assim usado como um factor de similaridade entre ambos. Tendo em conta que os diferentes erros entre segmentações contribuem de formas diferentes para a percepção visual humana, os erros são considerados mais graves quanto mais longe os respectivos pixéis estão da fronteira a que deviam pertencer. Calculando a soma pesada desses erros obtemos assim uma distância pesada, d_{sym}^{c}, entre ambas as segmentações. Quanto maior a distância, pior foi o desempenho do respectivo algoritmo.

3.2.1 Resultados

Os quatro algoritmos foram usados em diferentes sequências de vídeo: AE – Auto-estrada, CO – Corredor, EX – Exterior, Re – Restaurante e FT – Fonte. Estas foram previamente segmentadas manualmente gerando assim o *ground-truth*[12]. Quando comparados, usando a métrica anterior, obteve-se a Tabela 1 que sumariza os resultados obtidos para cada algoritmo, em cada sequência. Para cada combinação algoritmo-

[12] Imagens de referência geradas manualmente para posterior comparação com outras obtidas por métodos automáticos.

sequência é apresentada a distância média pesada ao *ground-truth* e número de *frames* por segundo (fps) que o algoritmo consegue executar.

Tabela 1 - d^c_{sym} média e frames por segundo para cada método, nas diferentes sequências (de [19]).

		AE	CO	EX	RE	FT
RAvg	d^c_{sym} fps	0.185 134	0.302 152	0.347 156	0.357 571	0.336 526
MoG	d^c_{sym} fps	0.160 6.7	0.267 6.1	0.317 6.7	0.678 31.5	0.307 29.2
KDE	d^c_{sym} fps	0.109 1.3	0.267 1.1	0.261 1.4	0.285 5.6	0.093 7.5
PF	d^c_{sym} fps	0.150 2.9	0.318 2.2	0.276 2.9	0.362 13.7	0.126 14.7

A Tabela 1 demonstra que o RAvg é o algoritmo capaz de processar mais frames por segundo. Por outro lado, o KDE apresenta resultados mais precisos todos os outros. Uma vez que se pretende um compromisso entre qualidade e velocidade de processamento, o método MoG (Figura 13) apresenta-se como a melhor alternativa para o projecto em curso.

Figura 13 – Na fila de cima as sequências originais CO (Corredor), EX (exterior) e AE (Auto-estrada). Em baixo a respectiva segmentação efectuada pelo algoritmo MoG (de [19]).

Após a correcta segmentação dos objectos relativamente ao fundo, importa encontrar algoritmos e formas de mapear esses objectos espacialmente. Desta forma, apresenta-se em seguida, a análise aos diversos métodos de *tracking* conhecidos actualmente.

3.3 *Tracking* de objectos

O *tracking* de objectos é uma tarefa importante no campo da visão computacional [25]. A proliferação de computadores com elevada capacidade de processamento, a disponibilidade de câmaras de vídeo de elevada qualidade e a crescente necessidade de automatizar a análise de vídeo gerou um grande interesse em torno dos algoritmos de *tracking* de objectos. O uso desta técnica é pertinente para:

- Reconhecimento de movimentos, isto é, identificação baseada em características visuais, detecção automática de objectos;
- Vigilância automática, tal como monitorizar um cenário para detectar actividades suspeitas ou eventos pouco usuais;
- Indexar vídeo, por exemplo, gerar anotações automáticas e reconhecimento de vídeos em bases de dados multimédia;
- Interacção homem-computador, consistindo no reconhecimento de gestos e/ou movimentos para controlar aplicações;
- Navegação de veículos, ou seja, planeamento de caminho baseado em vídeo e capacidade de evitar obstáculos;

Na sua forma mais simples, o *tracking* pode ser definido como o problema de estimar a trajectória de um objecto presente num plano de imagem conforme este se move pelo cenário. O algoritmo deve, em cada frame ao longo de uma sequência de vídeo, detectar a posição do objecto em questão. Dependendo do objectivo, podemos adicionalmente extrair informações do objecto, tais como, a sua orientação, área ou forma. No entanto, o *tracking* pode-se tornar complexo devido a ruído nas imagens, movimentos ou objectos complexos, oclusão parcial ou total dos objectos, mudanças de luz e por fim, devido a requisitos de processamento em tempo-real.

3.3.1 Representação dos objectos

Num cenário de *tracking*, um objecto pode ser definido como algo que é interessante para analisar. Por exemplo, um barco no mar, peixes num aquário, veículos numa estrada, pessoas a dançar, são objectos que podem ser importantes em determinado domínio. Os objectos podem ser representados principalmente pela sua forma, mas também pela sua aparência. Serão descritas as principais representações baseadas na forma e em seguida as baseadas em aparência.

Representações baseadas na forma:

- ***Pontos*** – o objecto é representado por um ponto, o centróide (Figura 14 (a)) [26], ou por um conjunto de pontos (Figura 14 (b)) [27]. Em geral a representação baseada em pontos é adequada para o *tracking* de objectos que ocupam pequenas regiões numa imagem.

- ***Formas geométricas primitivas*** – o objecto é representado por um rectângulo, elipse ou outra forma geométrica simples (Figura 14 (c) (d)), [28]. Esta representação é normalmente utilizada para representar objectos rígidos, embora possa ser também usada para representar objectos mais flexíveis.

- ***Silhueta ou contorno*** – A representação por silhueta define o contorno do objecto (Figura 14 (g) (h)). É adequada para representar formas complexas [29].

- ***Modelo de formas articuladas*** – os objectos articulados são compostos por várias partes, unidas por juntas. Por exemplo, o corpo humano é um objecto articulado com tronco, braços, pernas, mãos, cabeça e pés. As partes são ligadas através de "juntas", neste caso, ligamentos. Para representar um objecto articulado, as suas partes constituintes são modeladas por cilindros ou elipses como se exemplifica na Figura 14 (e).

- ***Modelos de esqueleto*** – estes podem ser extraídos pela aplicação da projecção da silhueta do objecto no eixo médio do mesmo. O uso deste modelo é muito comum como representação de formas para reconhecimento de objectos [30]. Pode ser usado para modelar objectos articulados ou rígidos (Figura 14 (f)).

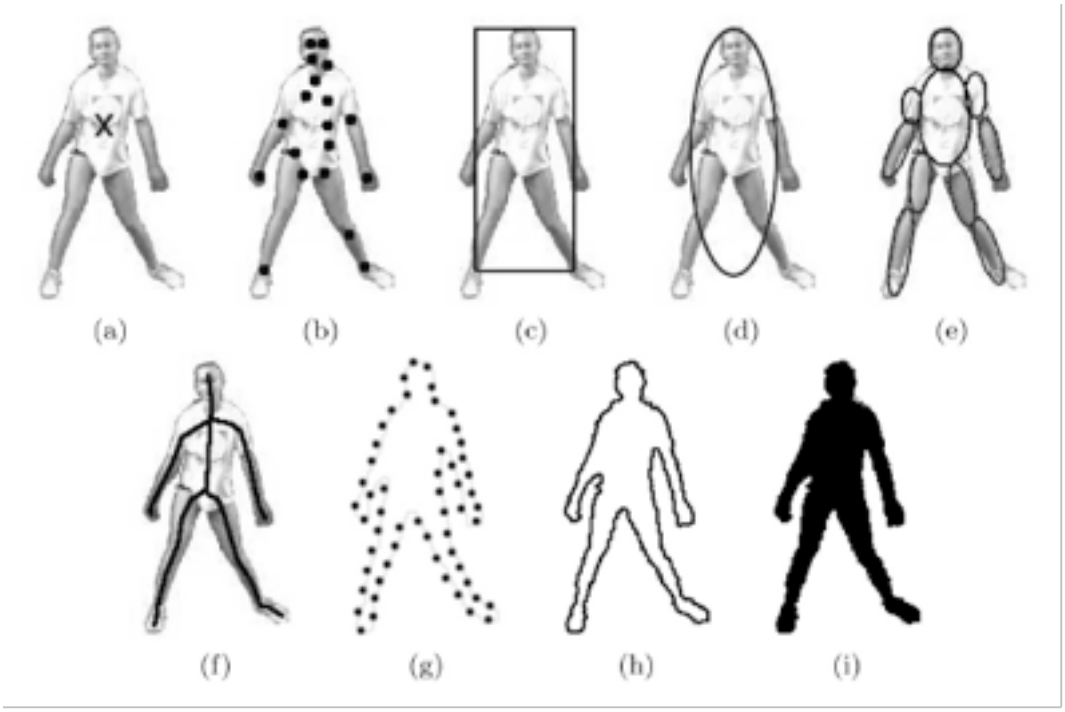


Figura 14 - Representações do objecto. (a) Centroide, (b) Pontos múltiplos, (c) rectângulo, (d) elipse, (e) múltiplas partes, (f) esqueleto, (g) contorno, (h) pontos de controlo no contorno, (i) silhueta (de [25]).

Representações baseadas na aparência:

Existem inúmeros modelos de representação através da aparência. É de salientar que no *tracking*, as representações baseadas na forma podem ser combinadas com as representações de aparência [31].

- ***Densidade de probabilidade da aparência do objecto (DPAO)***- A DPAO pode ser paramétrica (Gaussiana [32], ou mistura de Gaussianos [33]), ou não-paramétrica (janelas de Parzen [34] e histogramas [31]). A DPAO pode ser calculada a partir das regiões previamente definidas pelos modelos de formas (por exemplo, o interior de uma elipse ou contorno).

- ***Templates*** – são formados usando formas geométricas simples ou silhuetas [35]. A vantagem do template é permitir guardar ambas informações, espacial e aparência. No entanto, uma vez que apenas consegue codificar a aparência a partir de uma vista única, é apropriado para o *tracking* de objectos que não variam muito a sua posição.

- ***Modelos activos de aparência*** – São gerados pela representação simultânea da forma e aparência do objecto [36]. Em geral, a forma do objecto é definida por uma série de marcadores. É similar à representação por contornos, os marcadores podem estar no limite do objecto ou alternativamente, podem estar dentro da região do objecto. Para cada marcador, é armazenado um vector que contém a cor, textura ou magnitude do gradiente. Estes modelos requerem uma fase de treino onde as formas e aparências são "ensinadas" usando um conjunto de amostras que contêm, por exemplo, a principal componente de análise.

- **Modelos de aparência multi-vista** (MAM)– estes modelos codificam diferentes vistas do objecto. Um método para representar as diferentes vistas do objecto será gerar o sub-espaço das mesmas. Os métodos de sub-espaços, por exemplo, a análise de componentes principais (PCA) e a análise em componentes independentes (ICA), têm sido usados para a representação de formas e aparências [37] e [38]. Outro método para obter as diferentes vistas será treinando um conjunto de classificadores, por exemplo *Support Vector Machines* (SVM) [39] ou uma rede Bayesiana [40]. Uma limitação dos MAM é que as diferentes vistas do objecto são necessárias em avanço para a codificação.

Geralmente existe uma grande relação entre a representação do objecto e o algoritmo de *tracking*. As representações são escolhidas de acordo com o domínio que se pretende para a aplicação. Para o *tracking* de objectos que surgem como muito pequenos na imagem, a representação por pontos é normalmente a mais apropriada. por exemplo em [41] usam a representação por pontos para o *tracking* de pássaros distantes. Para os objectos cujas formas podem ser aproximadas por elipses ou rectângulos, a representação de formas geométricas primitivas são as mais indicadas. Em [27] é usada uma representação elíptica e a partir da mesma o histograma de cor do objecto é calculado para modelar a sua aparência. Para o *tracking* de objectos com formas complexas, por exemplo, humanos, a representação de contornos, silhuetas ou formas articuladas é a indicada. Em [42] o *tracking* é executado recorrendo às silhuetas numa aplicação de vigilância.

3.3.2 Selecção de características para o *tracking*

Seleccionar as características certas para o *tracking* é essencial. De modo geral, o mais importante na escolha de uma característica é a sua unicidade, para que os objectos principais possam ser facilmente distinguidos dos restantes. A escolha de características está relacionada com a representação do objecto. Por exemplo, a cor é usada como uma característica nas representações através de histogramas, enquanto para representações baseadas em contornos são usadas as arestas dos objectos. Normalmente os algoritmos de *tracking* utilizam uma combinação de características, sendo as mais comuns as seguintes:

- **Cor** – a cor aparente de um objecto é influenciada principalmente por dois factores físicos: a distribuição espectral da luminância e a capacidade de reflexão do objecto. Em processamento de imagem utiliza-se geralmente o espaço de cor RGB (red, green, blue) para representar cores. No entanto, o espaço RGB não é perceptível uniformemente, ou seja, as diferenças entre as cores no espaço RGB não correspondem às diferenças de cores percebidas pelos humanos [43]. Por outro lado os espaços de cor LUV (*Luminance, U e V*) LAB (*Luminance A-B*) e HSV (Hue, Saturation, Value) são uniformes ou aproximadamente uniformes mas são sensíveis a ruído [44]. Em suma, não existe um consenso acerca de qual o espaço de cor mais eficaz. Como tal encontra-se exemplos de aplicação de todos.

- **Edges (arestas)** – as fronteiras dos objectos geram fortes transições de intensidade luminosa na imagem. Assim, a detecção de arestas é usada para identificar estas mudanças. Uma propriedade importante das arestas é que são muito menos sensíveis a mudanças de luz que a cor. Os algoritmos que fazem o *tracking* baseado nas fronteiras dos objectos, normalmente usam as arestas como característica representativa. Devido à sua simplicidade e precisão, o método mais popular de detecção por arestas é o "Canny Edge Detector" [45]. Uma comparação de algoritmos de detecção de arestas pode ser encontrada em Bowyer [46].

- **Optical Flow (fluxo óptico)** – O fluxo óptico baseia-se em vectores de deslocamento que definem a translação de cada pixel numa região. São calculados usando o brilho de cada pixel, assumindo que esse brilho deve ser aproximadamente constante em *frames* consecutivos [47]. O fluxo óptico é usado normalmente como característica em segmentação e *tracking* baseados

em movimentos. As técnicas mais populares de cálculo de fluxo óptico são as de Horn e Schunck [47], Lucas e Kanade [48], Black e Anandan [49], e Szeliski e Couglan [50]. Foi feita uma avaliação dos vários métodos por Barron [51].

- **Textura** – é a medida de variação de intensidade de uma superfície que quantifica propriedades tais como suavidade e regularidade. Comparada à cor, a textura requer mais processamento, pois tem de gerar os descritores. Existem vários descritores de textura: Gray-Level Co-occurrence Matrices (GLCM's) [52] (um histograma em 2D que demonstra a co-ocorrência de intensidades numa direcção e distância específica), medição de texturas de Law [53], ou wavelets [54] (um banco de filtros ortogonais). Assim como as arestas, a textura também é menos sensível às mudanças de luz que a cor.

Tendo em conta as representações de *tracking* enunciadas e o objectivo da aplicação consideramos que a melhor forma de representar o corpo será pelo modelo de formas articuladas. Este permite a separação das diversas componentes do corpo como pretendido. Como característica, a mais relevante será a cor para permitir ao bloco de extracção de características distinguir entre os vários componentes.

Assim, foi necessário desenvolver um algoritmo de *tracking* capaz de tal representação.

3.4 Algoritmo de tracking

Este algoritmo recebe como entrada o resultado do algoritmo de segmentação *foreground*/*background*, que consiste numa sequência de imagens binárias, onde o branco representa o *foreground* e o preto o *background*. A sua execução consiste em três subrotinas, explicadas de seguida:

1) Frame-Differencing (FD) – Basicamente queremos identificar quais os *pixels* que passam de preto (P) para branco (B) em frames consecutivos. No entanto, isto não é suficiente para separar os componentes (braços), porque se o movimento for demasiado lento existem *pixels* que se vão manter brancos dum *frame* para o outro, o que também é importante detectar.

Existem quatro combinações de variação do pixel. Os pixels brancos apresentam o valor de 255, enquanto os pretos têm valor 0. Aplicando o FD obtém-se os seguintes valores (Tabela 2):

Tabela 2 – Transições entre frames consecutivas e respectivo valor do FD.

Branco – Branco (BB) Dif = 0	Preto - Preto (PP) Dif=0
Branco – Preto (BP) Dif= 255	Preto – Branco (PB) Dif= -255

A mudança PB é fácil de detectar, a dificuldade está em diferenciar a transição BB da PP, pois estas têm o mesmo valor (zero). Para resolver este problema foi introduzido um parâmetro de calibração (WB), definido pelo utilizador. Basicamente este parâmetro mede a quantidade de *pixels* brancos na imagem. Foi então imposta a seguinte condição:

Se $\frac{\sum pixels\_bran\cos}{\sum pixels\_imagem} > WB$ então está na transição BB.

Se, por outro lado, $\frac{\sum pixels\_bran\cos}{\sum pixels\_imagem} \leq WB$ significa que está na transição PP.

Na situação BB é considerado que existem suficientes *pixels* brancos para representar movimentos lentos. Neste caso o novo resultado obtido pelo FD é ignorado e considera-se o resultado anterior válido. A transição PP representa a situação em que nenhum objecto está presente.

2) Separação de componentes - Tem como função agrupar *pixels* que pertençam a um mesmo componente entre eles. Para isso é usado um algoritmo de seguimento. Começa-se por fazer uma pesquisa horizontal na imagem, caso seja encontrado algum pixel dum componente, é iniciado o algoritmo de seguimento. O algoritmo pesquisa para cada pixel do componente todos os seus vizinhos, tentando ver se existe algum que também pertença ao componente. Caso encontre algum pixel vizinho que lhe pertença, o ponto anterior é guardado numa *stack*[13], passando agora a fazer-se pesquisa no novo pixel encontrado. Caso

[13] Pilha de elementos – neste caso a pilha é LIFO (last in first out), ou seja o ultimo elemento colocado na pilha é o primeiro a ser retirado.

não se encontre nenhum pixel vizinho do mesmo componente, vai-se ao ponto anterior presente na stack e faz-se novamente a pesquisa pelos seus vizinhos, este algoritmo é efectuado até que a *stack* esteja vazia. Obviamente que se um pixel é considerado vizinho, já não pode ser pesquisado no ciclo seguinte. A Figura 15 permite visualizar o fluxograma de funcionamento do algoritmo.

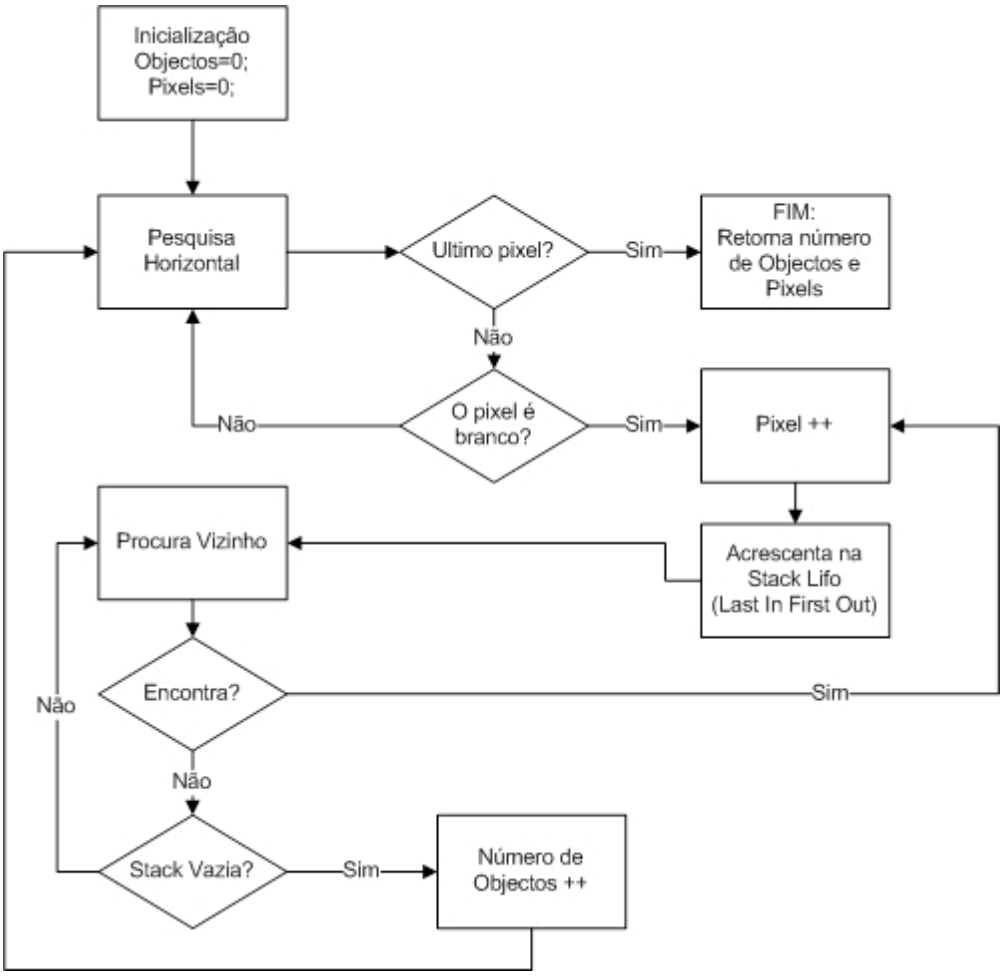


Figura 15 – Fluxograma de funcionamento do algoritmo de separação de componentes.

3) Coloração – Por fim, é necessário colorir cada objecto de uma cor. Para isso basta atribuir valores diferentes aos *pixels* (entre 0 e 255), consoante o objecto a que pertencem.

Na Figura 16 é possível visualizar a imagem de saída do algoritmo de segmentação que será a entrada do algoritmo de tracking e o resultado após a aplicação das subrotinas do mesmo.

a) b) c)

Figura 16 – a) Imagem de entrada para o algoritmo de tracking; b) Resultado da aplicação do FD e da separação de componentes; c) Coloração dos diferentes componentes

Desenvolvido o algoritmo, é necessário avaliar as suas capacidades. No ponto seguinte é explicado o método de avaliação e os respectivos resultados.

3.4.1 Resultados

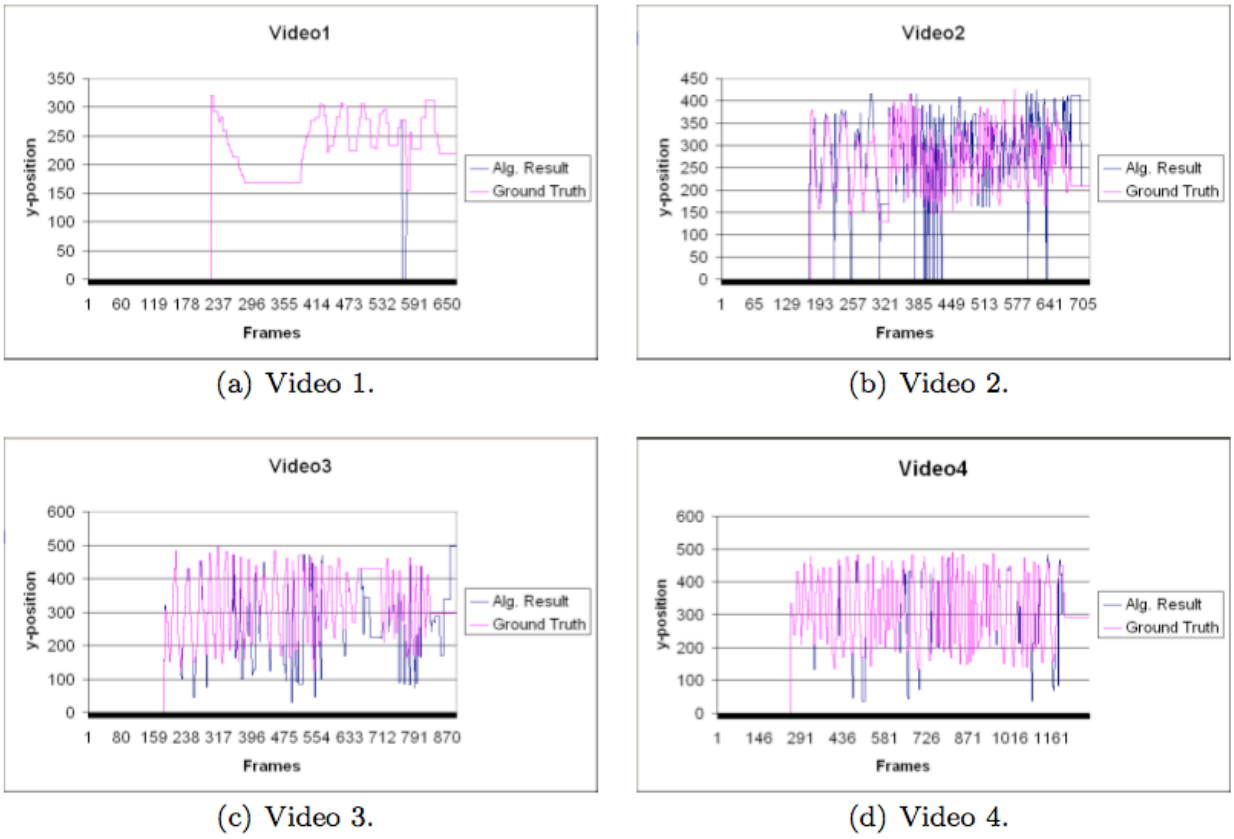
O desempenho do algoritmo foi testado com quatro vídeos de diferentes dificuldades de *tracking* (Tabela 3).

Tabela 3 – Sequências de vídeo de teste e respectiva dificuldade.

Nome do vídeo	Dificuldade de tracking
Vídeo 1	Fácil
Vídeo 2	Díficil
Vídeo 3	Média
Vídeo 4	Média

Nas sequências de vídeo, o movimento executado é principalmente no eixo do Y, no entanto o algoritmo está preparado para detectar movimento em qualquer direcção. Os vídeos 2, 3 e 4 exibem mais ruído que o vídeo 1. O primeiro vídeo consiste numa bailarina a mover ambos os braços ao mesmo ritmo, mantendo o resto do corpo imóvel. O segundo é

o mais difícil pois consiste no movimento de ambos os braços a ritmos diferentes e com movimento do corpo. Nos vídeos 3 e 4 apenas se move um braço, mas com alterações bruscas na velocidade do movimento. Para medir o desempenho do algoritmo foi criado manualmente o *ground-truth*. Na Figura 17 é apresentada a comparação entre os resultados obtidos pelo algoritmo e o que foi gerado pelo *ground-truth* para os quatro vídeos. Na Tabela 4 apresentam-se também os resultados em termos de percentagem.



(a) Video 1. (b) Video 2.

(c) Video 3. (d) Video 4.

Figura 17 – Resultados do algoritmo (a azul) e do *ground-truth* (a rosa).

Tabela 4 – Resultados de desempenho do algoritmo de *tracking* em percentagem.

Nome do vídeo	Desempenho de *tracking*
Vídeo 1	98,9%
Vídeo 2	50,8%
Vídeo 3	71,7%
Vídeo 4	90,7%

3.5 Resumo e Análise Crítica

Neste capítulo foram estudados e comparados os principais algoritmos de segmentação e *tracking*.

Tendo em conta o estudo efectuado e o compromisso entre velocidade e qualidade de segmentação, concluiu-se que o melhor método de segmentação para a aplicação em questão será a Mistura de Gaussianos. Este foi implementado em C++, baseado em [55] e com recurso à biblioteca do OpenCV.

Para o *tracking* do corpo, tal como mencionado anteriormente, a representação adoptada foi o modelo de formas articuladas. A cor de cada componente permitirá a extracção de valores relevantes para o passo seguinte, a estimação de ritmo.

É importante salientar que a qualidade do *tracking* varia bastante com complexidade dos movimentos executados. Como se pôde verificar pela Tabela 4, o vídeo de complexidade difícil, fez com que o *tracking* tenha um desempenho de apenas 50,8%.

Capítulo 4

Estimação de ritmo

Neste capítulo é desenvolvido um estudo das várias características a extrair de um corpo em movimento, de forma a estimar o ritmo a que se move. São comparados os principais algoritmos que permitem essa estimação através da frequência fundamental de um sinal periódico, em tempo real. É apresentada também uma abordagem alternativa ao cálculo do ritmo.

4.1 Ritmo e o Corpo Humano

Existem muitas definições para ritmo, mas simplesmente pode interpretar-se como a alternância de sons no tempo.

Em [11] podemos verificar que, segundo Paul Fraisse (1982), a palavra "ritmo" vem do grego *rhythmos* (ritmo) e *rheo* (fluir). *Rhythmos* surge como uma palavra-chave na filosofia Ioniana com o significado de "forma", mas uma forma improvisada, momentânea e modificável; literalmente significa um "modo particular de fluir". Platão (347 A.C.) aplicou este termo a movimentos corporais, os quais, tais como os sons musicais, podem ser descritos numericamente. No seu livro, "*As Leis*", definiu ritmo como a *ordem no movimento* e Fraisse, partindo dessa definição definiu ritmo como "a percepção de uma ordem". Fraisse entende também que o ritmo tem um factor de previsibilidade que o distingue de arritmia, ou seja, na presença de uma sequência rítmica, é possível prever ou antecipar a sua evolução.

Pode-se assim afirmar que a associação entre o ritmo musical e o movimento humano data desde o início da civilização Grega. Faz sentido pois, parafraseando Guedes [11], é impossível dissociar o papel do corpo e movimento na percepção e execução do ritmo musical. Usamos movimentos corporais para produzir ritmos musicais (como tocar um instrumento) e normalmente respondemos (consciente ou inconscientemente) ao ritmo musical com movimentos, desde o mais simples, como balancear ligeiramente a cabeça, até ao dançar ao som da música.

Guedes constata, do mesmo modo, que é possível asseverar que a dança pode produzir ritmo. Quando se observa alguém a dançar, é fácil perceber na dança padrões de movimento organizados no tempo e espaço. Certas vezes, o grau de sincronização entre os movimentos corporais e a música é tal que dá a impressão que os movimentos estão a fazer um mapa espacial do ritmo musical. Estes aspectos sugerem que existem realizações rítmicas em ambas as formas de arte e motivam a investigação das similaridades entre o ritmo musical e o ritmo na dança.

Os movimentos periódicos duma dança podem ser detectados, capturando-os de uma forma não-invasiva com uma câmara de vídeo conectada a um computador. O computador analisa o sinal de vídeo e cria a comunicação necessária para uma interacção, em tempo real, entre músicos e bailarinos.

Nas próximas três secções serão explicadas as condições e algoritmos que permitem calcular o ritmo desses movimentos.

4.2 Características que permitem estimar o ritmo

Após o *tracking* do corpo humano e da divisão das suas componentes principais (braços, pernas, tronco, cabeça) é essencial encontrar medidas que permitem mapear no tempo os movimentos das mesmas. Características como as coordenadas do ponto central (x,y) do *blob* em cada *frame* são fundamentais, mas além dessas existem outras como o tamanho ou inclinação do *blob* que podem fornecer informações importantes acerca do seu movimento.

Tendo em conta os aspectos supracitados e visto que neste campo não existem muitas experiências documentadas, foi desenvolvido um estudo baseado numa análise das componentes x e y. Para esta investigação foram criadas sequências de teste com movimentos em arco: verticais, horizontais e diagonais. O movimento realizado foi

periódico e com frequências de 0,5 Hz, 1Hz e 2Hz ou seja, cerca de 30, 60 ou 120 batidas por minuto (bpm) respectivamente. Através da aplicação criada no âmbito deste projecto, explicada no capítulo seguinte, a área e as coordenadas do centro de massa do objecto foram extraídas e registadas (Figura 18).

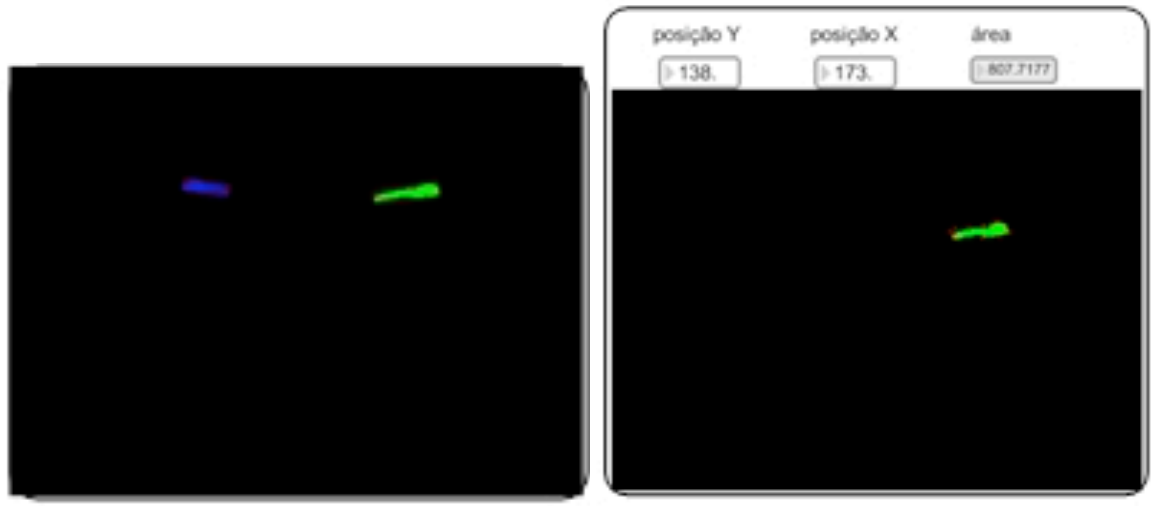


Figura 18 - Sequência de teste à esquerda e extracção das coordenadas do centro de massa e da área do objecto à direita.

Ao observar, por exemplo, a sequência de teste em arco vertical, verificamos que as coordenadas em x (horizontais) apresentam uma variação de amplitude demasiado baixa e inconstante para providenciar cálculos exactos. Pelo contrário, as coordenadas em y (verticais) têm uma amplitude suficiente para uma análise rigorosa (Figura 19).

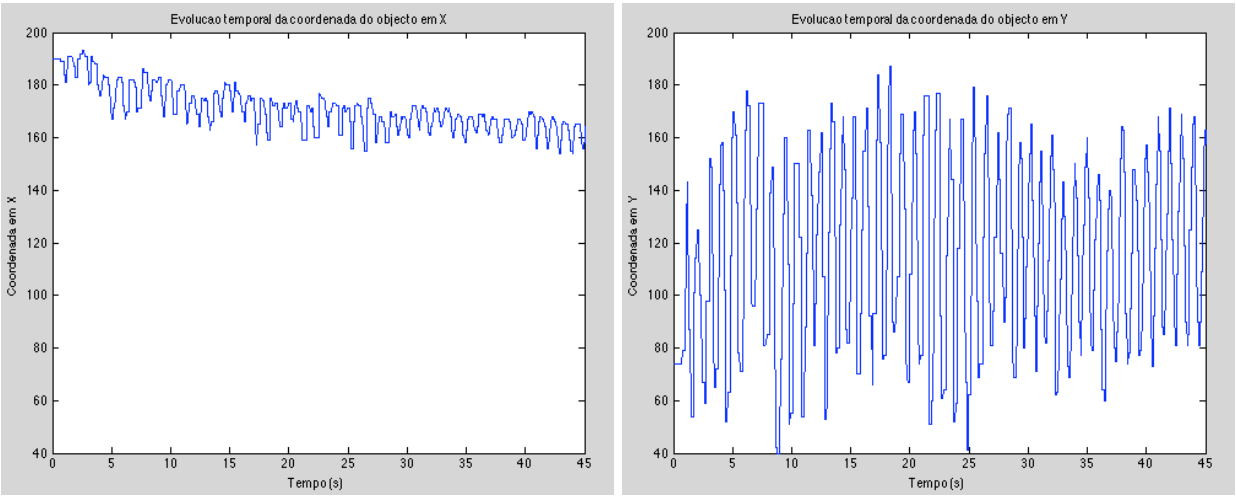
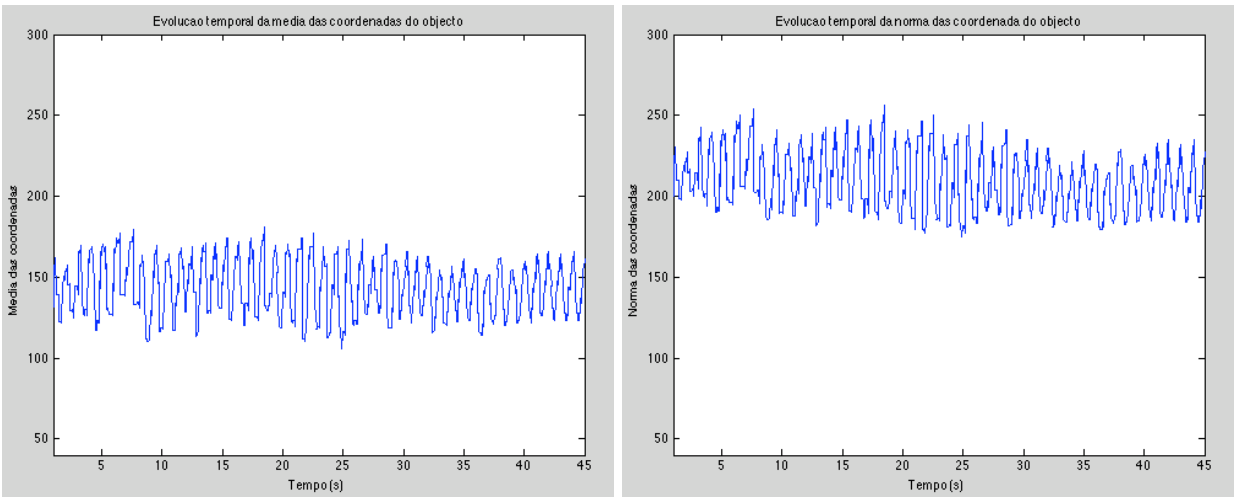


Figura 19 - Gráfico da variação das coordenadas em x e em y, respectivamente. Frequência de 1Hz.

Uma vez que os movimentos podem ocorrer em qualquer direcção, torna-se essencial correlacionar as coordenadas de tal modo que, qualquer que seja o movimento capturado, os dados providenciem cálculos pertinentes. Assim, foram testadas soluções que permitem essa relação, das quais se destacam a equação da média, $m = \frac{x + y}{2}$ (Figura 20 (a)) e a equação da norma de um vector $d = \sqrt{x^2 + y^2}$ (Figura 20 (b)).



(a) **(b)**

Figura 20 – Gráfico da equação da média (a) e da equação das normas (b).

Por consulta dos gráficos podemos constatar que a equação da norma apresenta um offset ligeiramente mais considerável. A ambas foram aplicados os algoritmos que nos permitem extrair a frequência fundamental do movimento (FFT e Algoritmo de Goertzel, descritos na secção 4.3). Verifica-se que originam resultados precisos.

4.2.1 Resultados

Novamente, devemos salientar que o vídeo de teste tem uma frequência de aproximadamente 1Hz e será este valor que devemos obter após o tratamento dos dados.

A equação da média providencia uma frequência fundamental de 0,9804Hz com amplitude de 14,43 (Figura 21).

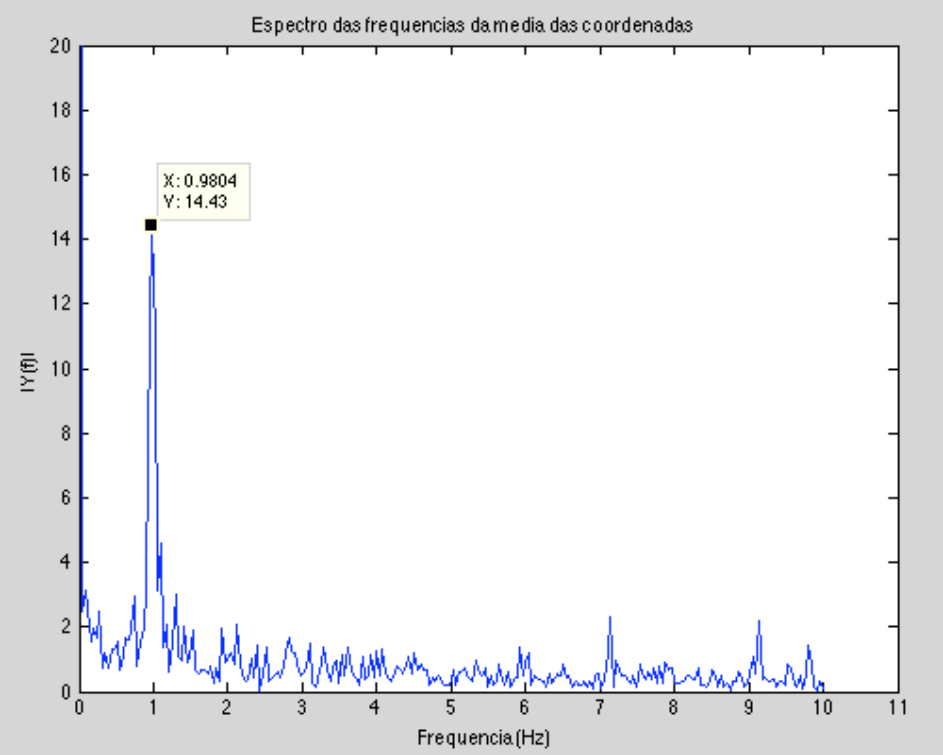


Figura 21 - Gráfico da variação da frequência calculada com recurso à equação da média, frequência fundamental é igual a 0,9804 ≈1Hz.

Relativamente à equação da norma, verificou-se que esta permitia determinar 0,9804Hz de frequência fundamental com amplitude de 14,77 (Figura 22).

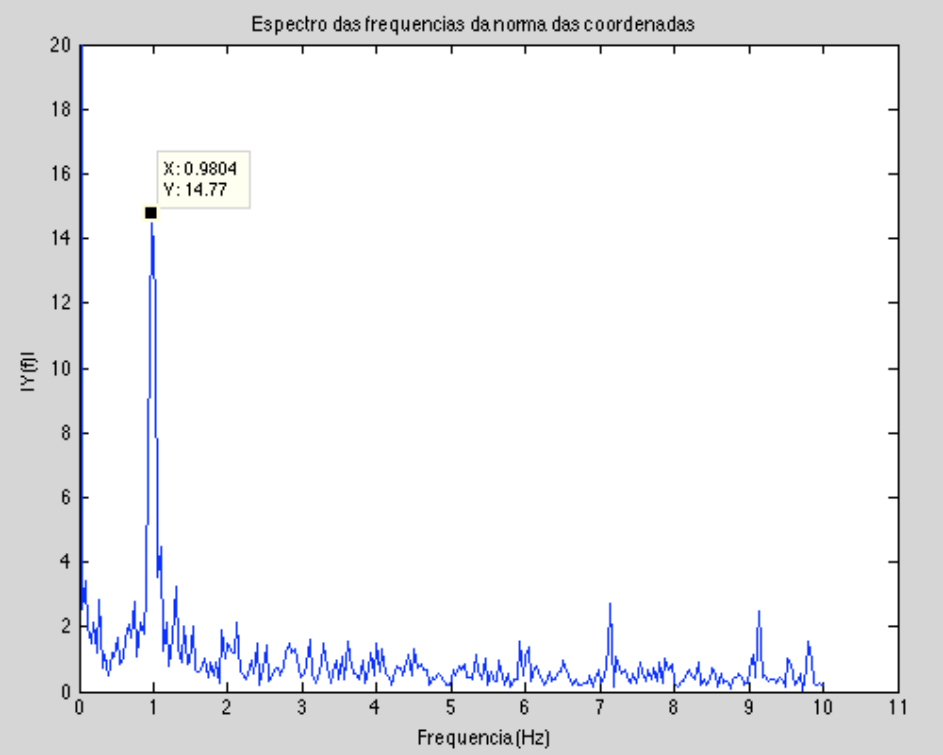


Figura 22 - Gráfico da variação da frequência calculada com recurso à equação das normas, frequência fundamental = 0,9804Hz ≈1Hz.

4.2.2 Uma abordagem alternativa ao cálculo do ritmo

Todas as características de estimação do ritmo apresentadas até ao momento baseiam-se no seguimento espacial do objecto de interesse ao longo do tempo. O conhecimento da trajectória do objecto p(x,y,t), sob qualquer uma das formas apresentadas no Capítulo 3 (pontos, silhueta, etc.), permite calcular sinais temporais (a posição x do centro de massa, a posição y do centro de massa, a distância do centro de massa do objecto à origem, etc) que tentam capturar o ritmo do objecto. Um problema com estas abordagens é que qualquer uma delas não detecta determinado tipo de movimentos. Exemplifiquemos com a distância do centro de massa do objecto à origem: se o objecto se mover sobre uma circunferência de centro na origem, este sinal será uma constante no tempo. Daí que não seja possível estimar o ritmo a partir deste sinal.

Uma abordagem alternativa é não tentar seguir a evolução espacial do objecto mas sim fixarmo-nos numa posição (x,y) e estudarmos a evolução do valor observado nessa posição. Se nenhum objecto passar por essa posição, o sinal observado será o sinal nulo. Contudo,

se um dado objecto passar periodicamente por essa posição, o sinal observado nessa posição fixa será um sinal periódico com o período do movimento.

Mesmo que dois ou mais objectos passem periodicamente por essa posição fixa, é possível estimar o período do movimento dos objectos. Para tal, basta criar um sinal para cada objecto presente; o sinal captura a evolução da etiqueta (cor) de um dado objecto.

Por exemplo, se numa dada posição (x,y), entre t=0 e t=2, estivesse presente o objecto A, entre t=2 e t=4 não estivesse presente nenhum objecto, entre t=4 e t=5 estivesse presente o objecto B, entre t=5 e t=7 estivesse o objecto A, entre t=7 e t=10 o objecto C, repetindo-se a partir daqui, obteríamos os sinais representados na Figura 23.

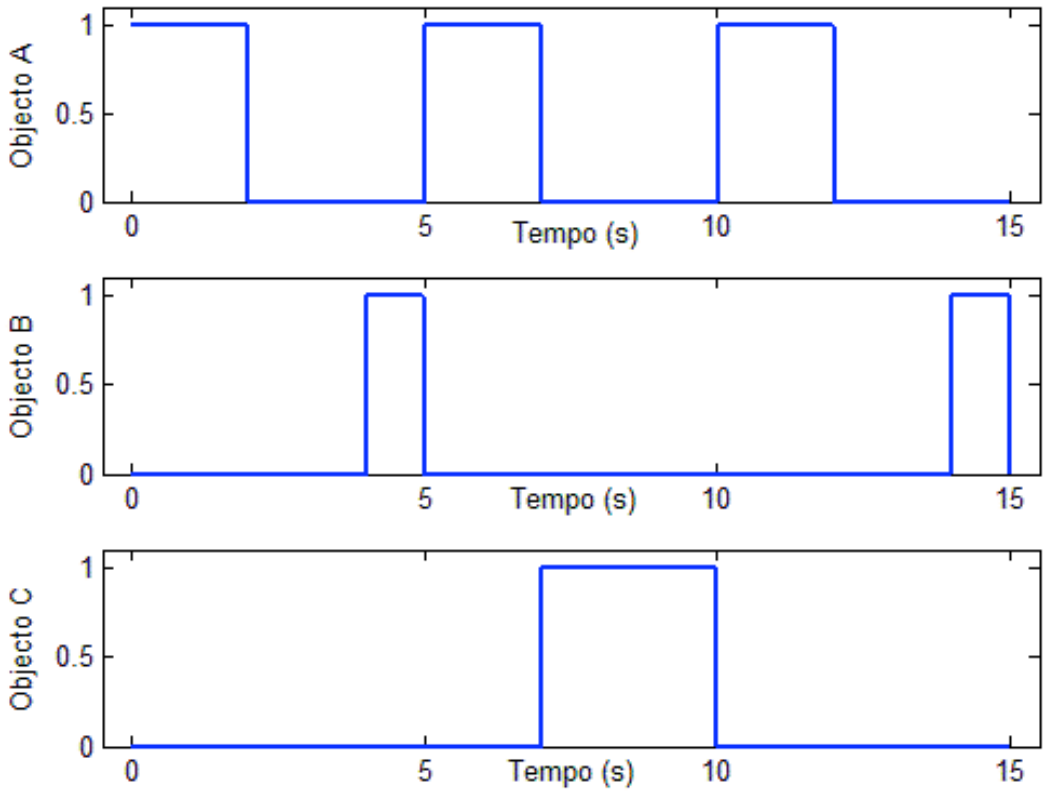


Figura 23 - Exemplo da evolução temporal do valor dum determinado *pixel* numa dada posição (x,y).

A inexistência de informação para preferir uma posição (x,y) em relação a outra para fazer esta estimativa e de forma a obter-se uma estimativa robusta leva a que se execute esta análise para cada posição (x,y) da imagem. Finalmente, seria necessário fundir as diversas estimativas obtidas em cada posição. Esta técnica acaba por se revelar

computacionalmente pesada, dificultando um desempenho em tempo real. Como tal, o seu estudo não foi aprofundado nesta tese.

4.3 Algoritmos de estimação de ritmo

Extrapolados os dados acerca do objecto, e analisando o gráfico da evolução temporal das suas coordenadas (Figura 24), torna-se evidente que estamos perante um sinal periódico. Assim, tal como foi referido no estado da arte, Guedes [11] constata que "*se aplicar um algoritmo capaz de detectar as periodicidades deste sinal, como a Transformada Rápida de Fourier (Fast-Fourier Transform - FFT), por exemplo, pode-se detectar o ritmo presente nesse sinal, calcular a sua frequência fundamental e assim obter o tempo que pode ser usado para fazer algo relevante musicalmente, tal como a geração de ritmos a partir do movimento de dança ou possibilitar ao bailarino o controlo, em tempo real, do tempo musical*" (pag. 16).

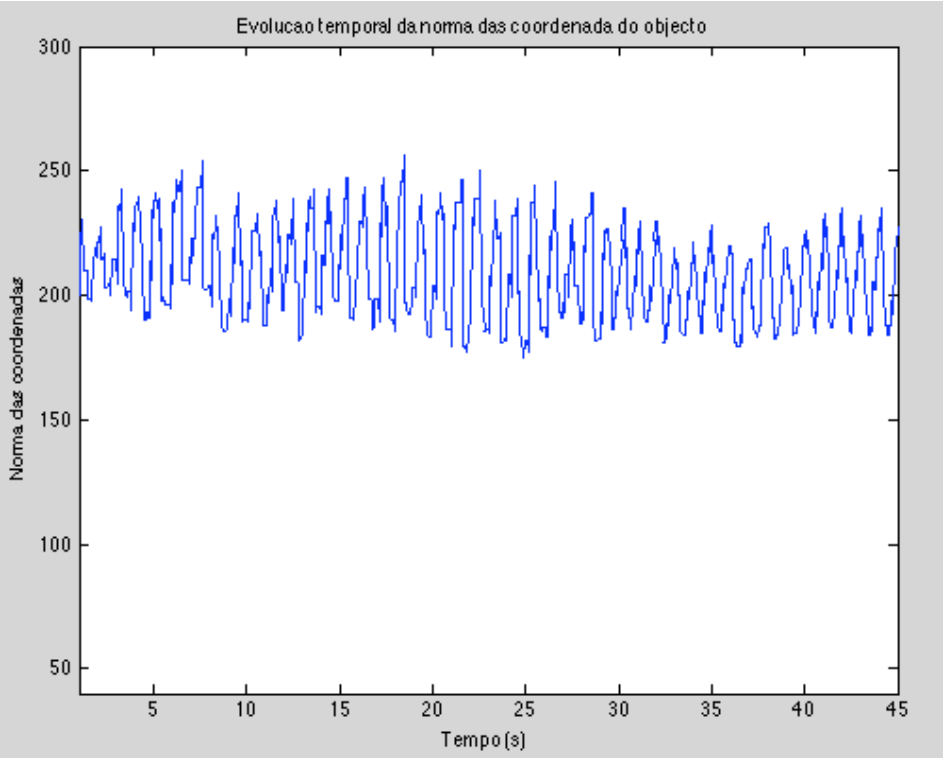


Figura 24 – Gráfico da evolução temporal das coordenadas do objecto em análise.

Mas, uma vez que o cálculo de uma FFT implica um processamento demasiado elevado, pois não permite escolher a banda de frequências em que se pretende executar, foi investigada também uma alternativa mais eficaz, o algoritmo de Goertzel. Este algoritmo é capaz de calcular a DFT (Transformada de Fourier Discreta) de uma banda limitada de frequência mais rapidamente que a FFT. Neste projecto foram testados e comparados ambos os algoritmos.

4.3.1 Algoritmo FFT

A Transformada Rápida de Fourier é um algoritmo computacional optimizado para implementar a Transformada Discreta de Fourier num vector de 2^N amostras. Permite determinar a frequência dum sinal discreto, representar a sua evolução no domínio das frequências e calcular a frequência fundamental do mesmo. Este algoritmo é baseado na derivação do algoritmo de FFT desenvolvido por Danielson e Lanczos [56]. Eles demonstraram que uma transformada discreta de Fourier de tamanho N pode ser reescrita como a soma de duas transformadas discretas de tamanho $N/2$, uma delas formada pelos índices pares do vector original e a outra pelos índices ímpares. A demonstração é simples [57]:

$$F_k = \sum_{j=0}^{N-1} e^{2\pi i jk/N} f_j \qquad (5)$$

$$= \sum_{j=0}^{N/2-1} e^{2\pi ik(2j)/N} f_{2j} + \sum_{j=0}^{N/2-1} e^{2\pi ik(2j+1)/N} f_{2j+1} \qquad (6)$$

$$= \sum_{j=0}^{N/2-1} e^{2\pi ikj/(N/2)} f_{2j} + W^k \sum_{j=0}^{N/2-1} e^{2\pi ikj/(N/2)} f_{2j+1} \qquad (7)$$

$$= F_k^e + W^k F_k^o \qquad (8)$$

Na última linha, F_k^e é o k-gésimo valor da transformada de Fourier de tamanho $N/2$ formada pelos componentes pares (even) da F_k original. Do mesmo modo a F_k^o é a correspondente transformada formada pelos componentes ímpares (odd) e $W \equiv e^{2\pi i/N}$. O "trunfo" do Teorema de Danielson-Lanczos é que pode ser usado recursivamente. Podemos dividir cada um dos componentes F_k^e e F_k^o novamente nos seus índices pares e ímpares,

obtendo assim vectores de tamanho $N/4$. Ou seja, podemos definir F_k^{ee} e F_k^{eo} como a transformada discreta dos pontos que são respectivamente par-par (even-even) e par-ímpar (even-odd) nas sucessivas divisões do vector original. Assim se garantirmos um vector original de tamanho 2^N podemos aplicar consecutivamente o Teorema de Danielson-Lanczos até que a transformada seja apenas de tamanho 1. Assim, a transformada será simplesmente calculada sobre o valor elementar de entrada para o respectivo índice do vector de saída $F^{eoeeoeo...oee} = f_n$. Uma vez obtido o vector de saída, para descobrir a frequência fundamental do sinal, basta calcular o máximo absoluto do vector e a frequência fundamental será dada pelo índice do vector sobre dois.

4.3.2 Algoritmo de Goertzel

O algoritmo de Goertzel permite calcular a DFT de uma banda de frequências eficazmente [58]. Essa banda pode ser especificada recorrendo a uma cadeia de filtros passa-banda deixando assim de fora as frequências que não interessam para o cálculo. Neste caso, a banda vai da frequência de 0,66Hz a 3,33Hz[14] que equivale a respectivamente 40bpm e 200bpm.

O espectro de magnitude e fase do movimento é calculado em duas fases. A primeira consiste em passar as coordenadas de variação de movimento pela cadeia de filtros passa-banda. Segundo [59] a equação às diferenças que define o filtro passa-banda de segunda ordem é:

$$Y_{(n)} = X_{(n)} + 2CR \times Y_{(n-1)} - R^2 \times Y_{(n-2)} \qquad (9)$$

onde $C = \cos\left(\frac{2\pi cf}{sr}\right)$, $R = e^{\left(\frac{-\pi bw}{sr}\right)}$, cf é a frequência central, bw a largura de banda, sr a taxa de amostragem e e = 2,718 (número de Euler).

---

[14] De acordo com Parncutt ([11] Cap. II) a sensação de ritmo surge da complexa interacção de todas as periodicidades envolvidas numa dada sequência de ritmo. Esta sensação está também dependente da nossa memória de curto prazo, sendo que deixa de existir para valores aproximadamente acima dos 1800ms e abaixo dos 200ms (Fraisse, 1982; Parncutt, 1987).

A segunda fase consiste em extrair a parte real e imaginária do sinal filtrado por aplicação do algoritmo de Goertzel [11]. As partes real e imaginária do filtro são calculadas da seguinte forma:

$$Y_{real} = Y_{(n)} - R\cos\left(\frac{2\pi bw}{sr}\right) \times Y_{(n-1)} \qquad (10)$$

$$Y_{imag} = -R\sin\left(\frac{2\pi bw}{sr}\right) \times Y_{(n-1)} \qquad (11)$$

A magnitude e fase são dadas por:

$$Y_{mag} = \sqrt{Y_{real}^2 + Y_{imag}^2} \qquad (12)$$

$$Y_{fase} = \arctan\left(\frac{Y_{real}}{Y_{imag}}\right) \qquad (13)$$

Uma vez obtido o espectro das frequências do movimento, este é comparado com o espectro de um impulso de 1Hz previamente estabelecido e por correlação cruzada a frequência fundamental é calculada.

4.3.3 Implementação e teste dos algoritmos de estimação de ritmo

Para implementação dos algoritmos foram criados dois objectos externos para o Max. A FFT foi programada de raiz baseada na proposta de [57] e o algoritmo de Goertzel foi implementado a partir do objecto criado previamente por Guedes em 2005 (o m.bandit).

Para testar a execução e precisão de ambos os algoritmos recorreu-se às sequências de teste previamente mencionadas (arco vertical e arco diagonal, às frequências de 0,5Hz, 1Hz e 2Hz).

4.3.4 Resultados

Os resultados encontram-se nas tabelas seguintes (Tabela 5 **e 6**). Estas contêm, para cada sequência, o valor da frequência fundamental e respectiva amplitude obtida pela FFT e a frequência fundamental obtida, por correlação cruzada, pelo algoritmo de Goertzel.

Tabela 5 – Resultados dos algoritmos FFT e Goertzel na análise da frequência fundamental da sequência de vídeo vertical.

Sequência Vertical	0,5Hz			1Hz			2Hz		
Algoritmos	FFT		Goertzel	FFT		Goertzel	FFT		Goertzel
	F(Hz)	Amp.	F(Hz)	F(Hz)	Amp.	F(Hz)	F(Hz)	Amp.	F(Hz)
Y	0,471	40,81	Idle	0,980	32,26	1,006	2,078	28,10	2,1
X	0,039	5,33	idle	0,039	6,48	idle	0.039	6,53	idle
$d = \sqrt{x^2 + y^2}$	0.471	17,81	idle	0,980	14,77	1,053	2,078	14,87	2,120
$m = \frac{x + y}{2}$	0,471	16,61	idle	0,980	14,43	1,126	2,078	13,50	2,165

Tabela 6 -Resultados dos algoritmos FFT e Goertzel na análise da frequência fundamental da sequência de vídeo diagonal.

Sequência Diagonal	0,5Hz			1Hz			2Hz		
Algoritmos	FFT		Goertzel	FFT		Goertzel	FFT		Goertzel
	F(Hz)	Amp.	F(Hz)	F(Hz)	Amp.	F(Hz)	F(Hz)	Amp.	F(Hz)
Y	0,471	26,76	idle	0,980	21,29	1,047	2,039	18,05	2,061
X	0,471	29,90	idle	0,980	22,64	1,006	1,922	14,42	2,064
$d = \sqrt{x^2 + y^2}$	0,471	7,58	idle	0,980	7,92	1,110	2,039	19,11	2,045
$m = \frac{x + y}{2}$	0,471	3,34	idle	0,980	4,84	1,152	2,039	13,92	2,105

Na análise das Tabelas 5 e 6 terá de se ter em atenção que o algoritmo de Goertzel está especificado para procurar valores entre 0,66Hz e 3,33Hz, por isso é que na sequência a 0,5Hz surge a palavra *idle* (parado).

Também se pode constatar que os valores de frequência fundamental da FFT praticamente não variam nas diferentes coordenadas/equações utilizadas. No entanto, o pico de amplitude de cada valor varia ligeiramente, sendo que o da equação da norma é sempre superior ao da média. Daqui podemos concluir que a equação da norma prevalece sobre a da média.

Por outro lado, na sequência diagonal as coordenadas isoladas fornecem valores de frequência fundamental com maior amplitude que a equação da norma. No entanto, na sequência vertical a coordenada em x fornece resultados insatisfatórios. Assim de forma a garantir que movimentos em qualquer direcção fornecem resultados precisos de frequência, optou-se pela equação da norma.

Relativamente aos algoritmos FFT e Goertzel, é de salientar que ambos promovem bons resultados, sendo que a FFT é ligeiramente mais precisa. No entanto, para providenciar os resultados enunciados nas tabelas 5 e 6 a FFT utiliza uma janela de 512 amostras. O algoritmo de Goertzel, sendo baseado em filtros passa-banda recursivos, constrói o espectro utilizando as 2 amostras mais recentes do sinal, só precisando dum período completo de movimento para fornecer resultados precisos (no caso de movimento com período de 1Hz e taxa de amostragem de 20fps necessita de apenas 20 amostras). Esta característica torna-o bastante mais rápido que a FFT. Tendo em conta a limitação de tempo real que se pretende para este projecto, o algoritmo de Goertzel apresenta-se como a metodologia a adoptar.

4.4 Resumo e Análise Crítica

Nesta secção foram enunciadas e comparadas as principais equações que permitem correlacionar um sistema de coordenadas. Foi também exposta uma metodologia alternativa baseada apenas no cruzamento de objectos sobre determinados *pixels* da imagem. Do mesmo modo foram descritos detalhadamente os algoritmos desenvolvidos e testados para a estimação da frequência fundamental de um sinal em tempo real.

Foi escolhida a equação das normas para relacionar as coordenadas e o algoritmo de Goertzel para o cálculo da frequência fundamental.

Capítulo 5

Integração

Após a análise cuidada, acerca de qual o algoritmo ou método ideal para desempenhar as funções de cada bloco proposto no primeiro capítulo (Figura 1) neste capítulo é descrito como foi desenvolvido e integrado o sistema.

5.1 Plataforma de Desenvolvimento

A plataforma escolhida para o desenvolvimento e integração da aplicação foi o MAX/MSP/Jitter[15] da Cycling'74. A escolha foi baseada na capacidade do MAX executar, em tempo real, as várias funções requeridas, assim como o facto de ser uma das plataformas mais utilizadas por compositores e coreógrafos entusiastas da dança interactiva. Além disso, uma vez que é modular, permite facilmente que os blocos criados sejam integrados entre si ou, futuramente, com outras plataformas como o Eyesweb ou o Isadora.

5.2 Arquitectura do Sistema

De seguida é apresentada uma breve descrição dos blocos que compõem a arquitectura do sistema, como representado na Figura 25. A pormenorização de cada um dos blocos/algoritmos é realizada nas restantes secções deste capítulo.

[15] Daqui em diante o MAX/MSP/Jitter será apenas referenciado como Max.

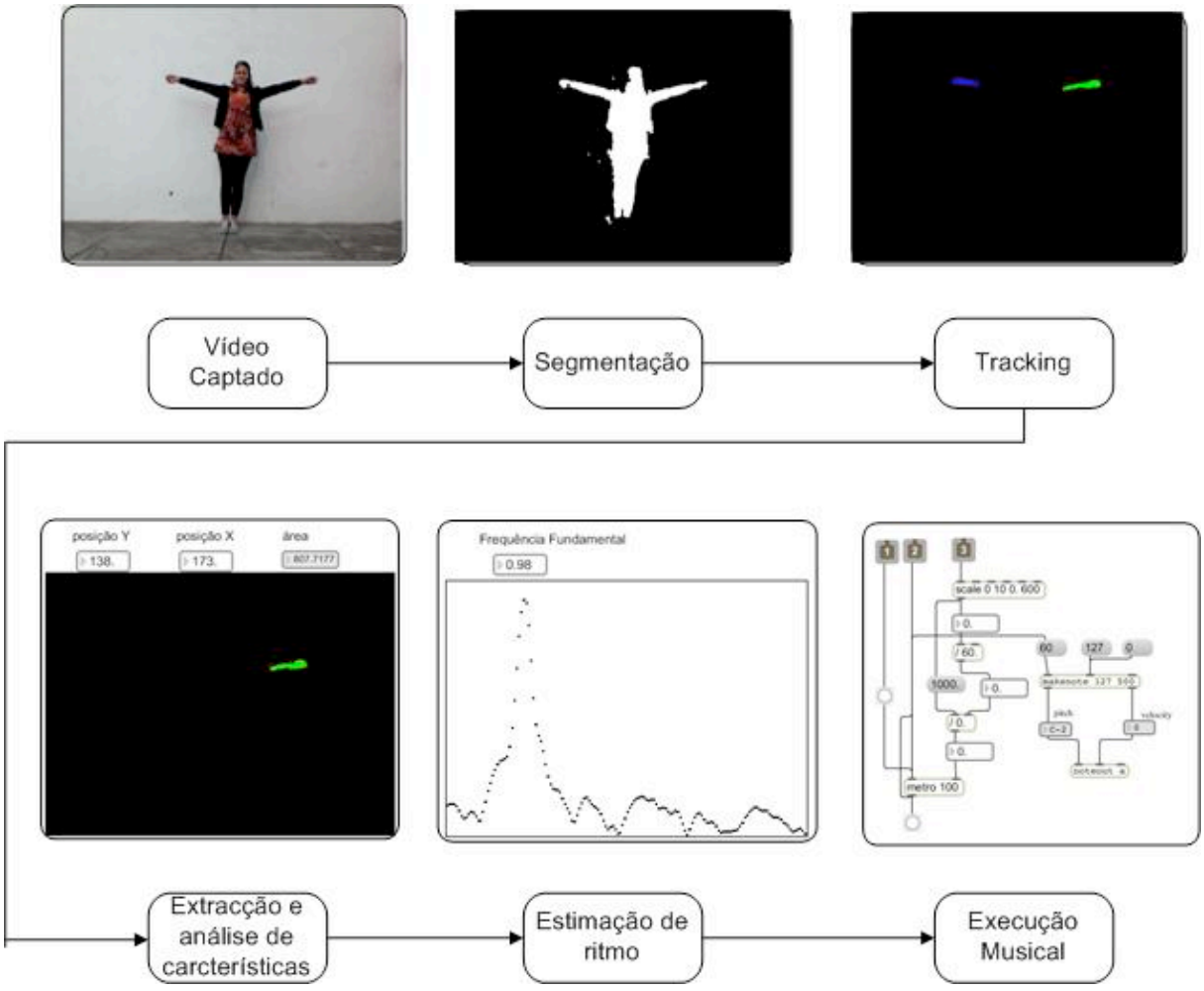


Figura 25 – Arquitectura do sistema RitMoVídeo.

5.2.1 Segmentação *background/foreground*

De acordo com o estudo descrito no capítulo três, a segmentação de imagens foi efectuada com base no algoritmo de Mistura de Gaussianos. Com recurso à biblioteca do OpenCv o software foi desenvolvido[16] em C++ e separa o *background* do *foreground* usando a metodologia explicada no ponto 3.1.2 desta tese (pag. 25).

[16] Agradeço a colaboração do Engenheiro Hélder Oliveira nesta tarefa.

5.2.2 *Tracking*

A nível de tracking a implementação foi efectuada também em C++ e recorrendo à biblioteca do OpenCv. O algoritmo foi implementado conforme descrito na secção 3.4 (pag. 34).

5.2.3 Extracção e análise de características

De entre as várias características possíveis de extrair, reconhecemos que as de maior interesse neste caso seriam as coordenadas do centro de massa de cada componente, assim como a sua área. Nesse sentido foram utilizados ou desenvolvidos objectos no Max responsáveis por:

1) Cálculo da área de cada componente

A área de cada componente pode ser encontrada simplesmente contando o número de *pixels* que cada componente contém. Felizmente a biblioteca de imagem CV.Jit para o MAX, disponibilizada por Jean Marc Pelletier [60], já contém um objecto capaz de fazer esta contagem.

2) Cálculo centro de massa de cada componente

Para encontrar o centro de massa de cada componente recorreu-se à equação dos momentos centrados definida da seguinte forma:

$$m_{pq} = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} x^p \times y^q \times I(x,y) \qquad \textbf{(14)}$$

Onde o I(x,y) indica se o pixel pertence ao objecto ou não. Se pertencer I=1 senão I=0. O centro de massa em x é obtido por resolução da equação para o momento m_{10} e o centro de massa em y para o momento m_{01}. Se quisermos validar o cálculo da área para cada componente basta fazer o momento m_{00}.

A partir dos momentos podemos também obter qual é a orientação de cada objecto. Para isso basta calcular o ângulo que o objecto faz com o eixo dos xx.

O ângulo pode ser calculado pela seguinte equação:

$$\theta = \frac{1}{2} \times \tan^{-1} \times \left[\frac{2 \times m_{11}}{m_{02} - m_{20}} \right] \quad (15)$$

A partir do ângulo podemos fazer uma mudança de base de coordenadas, que será necessário para destacar a inclinação dos objectos, segundo um rectângulo. Definido esse rectângulo é possível saber, para cada, qual o seu comprimento e altura.

A mudança de coordenas é feita através da fórmula:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \cos\theta & -sen\theta \\ sen\theta & \cos\theta \end{bmatrix} \times \begin{bmatrix} x_c \\ y_c \end{bmatrix} \quad (16)$$

O sistema de coordenadas está evidenciado na Figura 26:

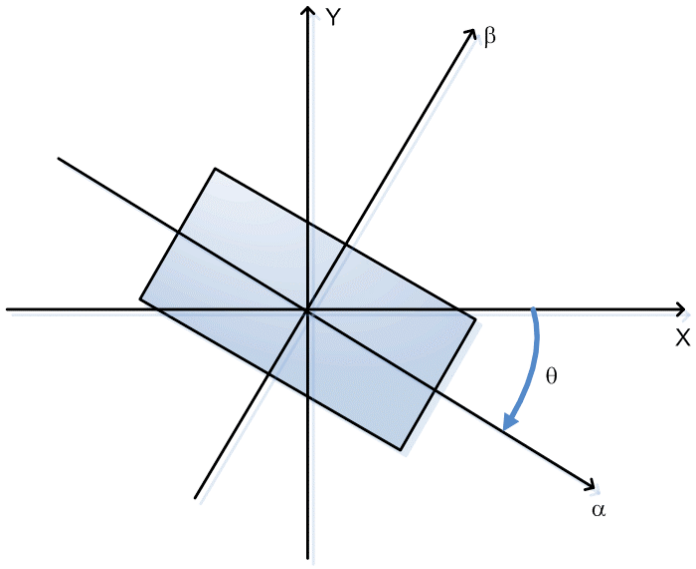


Figura 26 – Sistema de coordenadas que permite calcular a mudança de base.

Através da definição dos pontos na nova base podemos desenhar o rectângulo que cobre toda a área do componente como exemplificado na Figura 27.

Rectângulo: $\left[\alpha_{\min} \quad \beta_{\min} \quad \alpha_{\max} \quad \beta_{\max}\right]$

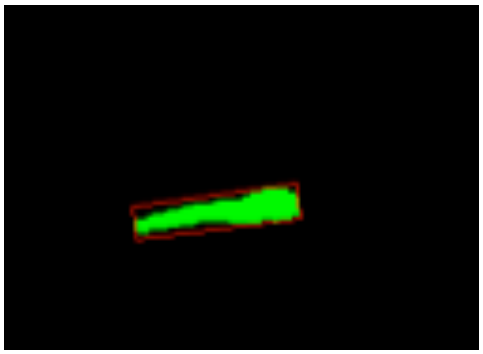


Figura 27 – Rectângulo que limita o componente.

3) Análise das características

A análise das características foi efectuada tendo em conta a equação da norma de um vector. Para isso foi desenvolvido um objecto no Max. Este tem como entrada as coordenadas x e y do centro de massa do objecto e como saída o resultado da equação da norma das coordenadas.

5.2.4 Estimação de ritmo

As características processadas no ponto anterior são enviadas para o objecto responsável pela extracção da frequência fundamental do movimento. Este objecto implementa o algoritmo de Goertzel tal como definido na secção 4.3.2 (pag. 52). Este tem como saída a frequência fundamental e os primeiros quatro harmónicos.

5.2.5 Execução Musical

Para uma percepção auditiva e visual do ritmo foi criado um objecto que calcula as batidas por minuto correspondentes à frequência e executa um trecho musical a esse tempo.

5.3 Dificuldades e Limitações

Uma limitação deste projecto está no cenário de utilização, é conveniente que este seja controlado. O ideal será um cenário estático de uma só cor e o uso de uma cor distinta do mesmo para o figurino do bailarino.

O *tracking* revela um desempenho médio para sequências complexas. Conforme se exemplifica na Figura 28, se ocorre um movimento demasiado brusco o algoritmo não consegue efectuar o *tracking* correcto (deveria manter a cor azul e o rectângulo em todas as imagens).

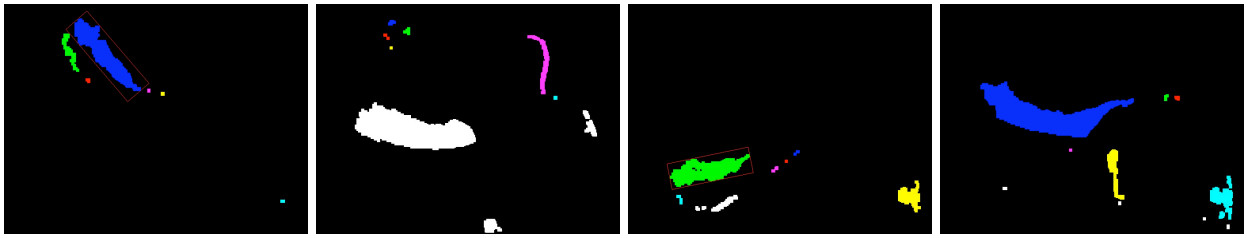


Figura 28 - Sequência de 4 imagens em que o *tracking* falha na coloração e delimitação do objecto.

A nível da extracção das características a equação da norma apresenta uma lacuna. Caso o movimento seja puramente circular as coordenadas em x e y compensam-se no cálculo e a equação providencia sempre o mesmo resultado ao longo do tempo. Obviamente que a execução de um movimento circular perfeito repetidamente é algo improvável num humano, mas não deixa de ser uma limitação. O aperfeiçoamento da técnica de extracção de características baseada em *pixels* poderá permitir ultrapassar estas limitações.

Por fim, a principal dificuldade deste projecto é que tudo tem de ser executado em tempo real. A segmentação, extracção de características, determinação da frequência fundamental e controlo do ritmo, tudo isto tem de ser realizado com o mínimo atraso temporal relativamente à realidade captada. Esta característica foi o maior desafio no desenvolvimento deste projecto.

Além destas, existem outras limitações que ainda não foram totalmente superadas, como uma segmentação e uma separação das diversas componentes mais robusta.

Capítulo 6

Conclusões e Perspectivas de Trabalho Futuro

Para o tempo disponibilizado para este projecto (20 semanas), creio que os objectivos foram alcançados e foi desenvolvido trabalho importante em diversas áreas.

O intuito do projecto RitMoVídeo era construir uma cadeia de blocos que permitisse a análise de movimentos presentes numa sequência de vídeo e o cálculo do ritmo desses movimentos, em tempo real.

Para cada bloco foram estudados e comparados os principais algoritmos. Uma vez determinadas as metodologias a utilizar, estas foram integradas na cadeia de blocos e o RitMoVídeo ficou operacional.

O projecto funciona bem, mas tem as suas limitações. No entanto, estas podem ser ultrapassadas com um pouco mais de tempo de investigação e desenvolvimento.

6.1 Trabalho Futuro

Como trabalho futuro, uma das prioridades deverá ser tornar a separação das componentes mais robusta. O *tracking* apresenta um desempenho médio para sequências complexas.

A nível da equação que relaciona as coordenadas, seria interessante eliminar a lacuna previamente referida, talvez através do desenvolvimento de uma equação de regressão linear ou atribuindo pesos a cada coordenada tendo em conta a quantidade de movimento

efectuada. O método de extracção de ritmo através do posicionamento de *pixels* enunciado anteriormente também se pode revelar como uma boa alternativa.

Seria interessante também aumentar o nível de complexidade do ritmo gerado automaticamente, talvez através da composição de melodias que, de certa forma, enriquecessem o sentimento da exibição.

Referências

[1] - Mulder Axel, 'Human movement tracking technology', Report 94-1 of the Hand Centered Studies of Human Movement Project, Burnaby, Columbia Britânica, Canadá: Universidade Simon Fraser, Julho 1994.

[2] - http://music.arts.uci.edu/dobrian/motioncapture/, acesso em 25/01/2009.

[3] – Mark Feldmeier, Mateusz Malinowski, Joseph A. Paradiso, "Large Group Musical Interaction using Disposable Wireless motion sensors", (2002)

[4] - U. Enke. DanSense: Rhythmic analysis of dance movements using acceleration-onset times. Master's thesis, RWTH Aachen University, Aachen, Alemanha, Setembro 2006.

[5] - http://music.arts.uci.edu/dobrian/motioncapture/mcm.htm, acesso em 25/01/2009.

[6] – www.vicon.com, acesso em 25/01/2009.

[7] – Atsushi Nakazawa, Katsushi Ikeuchi, Takaaki Shiratori, "Detecting Dance Motion Structure Using Motion Capture and Musical Information", Instituto de Ciência Industrial, Universidade de Tokyo 4-6-1, Komaba, Meguro-ku, Tokyo 153-8505, Cybermedia Center, Universidade de Osaka 1-32, Machikaneyama, Toyonaka, Osaka 560-0043, Japão.

[8] – Mikhail Gorman, Margrit Betke, Elliot Saltzman, e Amir Lahav, "Music Maker – A Camera-based Music Making Tool for Physical Rehabilitation", Computer Science Technical Report No. 2005-032, Universidade de Boston.

[9] - http://www.infomus.dist.unige.it/EywMain.html, acesso em 25/01/2009.

[10] – Antonio Camurri, Carol L. Krumhansl, Barbara Mazzarino, Gualtiero Volpe, "An Exploratory Study of Anticipating Human Movement in Dance", 2nd International

Symposium on Measurement, Analysis and Modeling of Human Functions, 1st Mediterranean Conference on Measurement, Genova, Itália, Junho de 2004.

[11] – Guedes, C. Mapping Movement to Musical Rhythm: A Study in Interactive Dance. Ph.D. Thesis, Universidade de Nova-Iorque, Nova-Iorque, 2005.

[12] - Oppenheim, A. V. & Schafer, R. V. "Digital Signal Processing", Prentice-Hall International, Londres, 1975.

[13] - Naveda, L. and M. Leman (2008). Representation of Samba dance gestures, using a multi-modal analysis approach. 5th International Conference on Enactive Interfaces. Pisa, European Enactive Network of Excellence ENACTIVE.

[14] - http://www.cycling74.com/products/max5, acesso em 25/01/2009.

[15] – http://puredata.info/ , acesso em 25/01/2009.

[16] – http://www.troikatronix.com/isadora.html, acesso em 25/01/2009.

[17] – http://www.processing.org/, acesso em 25/01/2009

[18] - R. F. Gonzalez, R. E. Woods, "Digital Image Processing", Addison-Wesley Publishing Company, 1993.

[19] – Luís F. Teixeira, Jaime S. Cardoso, Luís Corte-Real, "Object Segmentation Using Background Modelling and Cascaded Change Detection", Journal of Multimedia, vol. 2, no. 5, Setembro 2007

[20] - D.-S. Lee, "Effective Gaussian mixture learning for video background subtraction", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 5, pp. 827–832, Maio 2005.

[21] - A. Elgammal, D. Hardwood, e L. Davis, "Non-parametric model for background subtraction," Proceedings of European Conference on Computer Vision, vol. 2, 2000, pp. 751–767.

[22] - L. Li, W. Huang, I. Y.-H. Gu, e Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," IEEE Transactions on Image Processing, vol. 13, no. 11, pp. 1459–1472, Novembro 2004.

[23] - J. S. Cardoso and L. Corte-Real, “Toward a generic evaluation of image segmentation,” IEEE Transactions on Image Processing, vol. 14, pp. 1773–1782, Novembro 2005.

[24] - ——, “A measure for mutual refinements of image segmentations,” IEEE Transactions on Image Processing, vol. 15, pp. 2358–2363, Agosto 2006.

[25] – Yilmaz, A., Javed, O., and Shah, M. 2006. Object tracking: A survey. ACM Comput. Surv. 38, 4, Article 13, Dezembro 2006

[26] - VEENMAN, C., REINDERS, M., AND BACKER, E. 2001. Resolving motion correspondence for densely moving points. IEEE Trans. Patt. Analy. Mach. Intell. 23, 1, 54–72.

[27] - SERBY, D., KOLLER-MEIER, S., AND GOOL, L. V. 2004. Probabilistic object tracking using multiple features. In IEEE International Conference of Pattern Recognition (ICPR). 184–187.

[28] - COMANICIU, D., RAMESH, V., AND MEER, P. 2003. Kernel-based object tracking. IEEE Trans. Patt. Analy. Mach. Intell. 25, 564–575.

[29] – YILMAZ, A., LI, X., AND SHAH, M. 2004. Contour based object tracking with occlusion handling in video acquired using mobile cameras. IEEE Trans. Patt. Analy. Mach. Intell. 26, 11, 1531–1536.

[30] - ALI, A. AND AGGARWAL, J. 2001. Segmentation and recognition of continuous human activity. In IEEE Workshop on Detection and Recognition of Events in Video. 28–35.

[31] - COOTES, T., EDWARDS, G., AND TAYLOR, C. 2001. Robust real-time periodic motion detection, analysis, and applications. IEEE Trans. Patt. Analy. Mach. Intell. 23, 6, 681–685.

[32] - ZHU, S. AND YUILLE, A. 1996. Region competition: unifying snakes, region growing, and bayes/mdl for multiband image segmentation. IEEE Trans. Patt. Analy. Mach. Intell. 18, 9, 884–900.

[33] - PARAGIOS, N. AND DERICHE, R. 2002. Geodesic active regions and level set methods for supervised texture segmentation. Int. J. Comput. Vision 46, 3, 223–247.

[34] - ELGAMMAL, A., DURAISWAMI, R., HARWOOD, D., AND DAVIS, L. 2002. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. Proceedings of IEEE 90, 7, 1151–1163.

[35] - FIEGUTH, P. AND TERZOPOULOS, D. 1997. Color-based tracking of heads and other mobile objects at video frame rates. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 21–27.

[36] - EDWARDS, G., TAYLOR, C., AND COOTES, T. 1998. Interpreting face images using active appearance models. In International Conference on Face and Gesture Recognition. 300–305.

[37] - MUGHADAM, B. AND PENTLAND, A. 1997. Probabilistic visual learning for object representation. IEEE Trans. Patt. Analy. Mach. Intell. 19, 7, 696–710.

[38] - BLACK, M. AND JEPSON, A. 1998. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. Int. J. Comput. Vision 26, 1, 63–84.

[39] - AVIDAN, S. 2001. Support vector tracking. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 184–191.

[40] - PARK, S. AND AGGARWAL, J. K. 2004. A hierarchical bayesian network for event recognition of human actions and interactions. Multimed. Syst. 10, 2, 164–179.

[41] - SHAFIQUE, K. AND SHAH, M. 2003. A non-iterative greedy algorithm for multi-frame point correspondence. In IEEE International Conference on Computer Vision (ICCV). 110–115.

[42] - HARITAOGLU, I., HARWOOD, D., AND DAVIS, L. 2000. W4: real-time surveillance of people and their activities. IEEE Trans. Patt. Analy. Mach. Intell. 22, 8, 809–830.

[43] - PASCHOS, G. 2001. Perceptually uniform color spaces for color texture analysis: an empirical evaluation. IEEE Trans. Image Process. 10, 932–937.

[44] - SONG, K. Y., KITTLER, J., AND PETROU, M. 1996. Defect detection in random color textures. Israel Verj. Cap. J. 14, 9, 667–683.

[45] - CANNY, J. 1986. A computational approach to edge detection. IEEE Trans. Patt. Analy. Mach. Intell. 8, 6, 679–698.

[46] - BOWYER, K., KRANENBURG, C., AND DOUGHERTY, S. 2001. Edge detector evaluation using empirical roc curve. Comput. Vision Image Understand. 10, 77–103.

[47] - HORN, B. AND SCHUNK, B. 1981. Determining optical flow. Artific. Intell. 17, 185–203.

[48] - LUCAS, B. D. AND KANADE., T. 1981. An iterative image registration technique with an application to stereo vision. In International Joint Conference on Artificial Intelligence.

[49] - BLACK, M. AND ANANDAN, P. 1996. The robust estimation of multiple motions: Parametric and piecewise- smooth flow fields. Comput. Vision Image Understand. 63, 1, 75–104.

[50] – SZELISKI, R. AND COUGHLAN, J. 1997. Spline-based image registration. Int. J. Comput. Vision 16, 1-3, 185–203.

[51] - BARRON, J., FLEET, D., AND BEAUCHEMIN, S. 1994. Performance of optical flow techniques. Int. J. Comput. Vision 12, 43–77.

[52] - HARALICK, R., SHANMUGAM, B., AND DINSTEIN, I. 1973. Textural features for image classification. IEEE Trans. Syst. Man Cybern. 33, 3, 610–622.

[53] - LAWS, K. 1980. Textured image segmentation. PhD thesis, Electrical Engineering, Universidade da Califórnia do Sul.

[54] - MALLAT, S. 1989. A theory for multiresolution signal decomposition: The wavelet representation. IEEE Trans. Patt. Analy. Mach. Intell. 11, 7, 674–693.

[55] – P. KadewTraKuPong and R. Bowden,An improved adaptive background mixture model for real-time tracking with shadow detection, in Proc. 2nd European Workshp on Advanced Video-Based Surveillance Systems, 2001

[56] - J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19:297–301, 1965

[57] - William H. Press, Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery, “Numerical Recipes: The Art of Scientific Computing, Third Edition”, Cambridge University Press

[58] - Blahut, R. (1985). *Fast algorithms for digital signal processing*. Reading, MA: Addison-Wesley Publishing Company

[59] - Tempelaars, S., “Signal processing, speech and music”, Swets & Zeitlinger, Lisse, The Netherlands, 1996

[60] - http://www.iamas.ac.jp/~jovan02/cv/, acesso em 27/01/2009