Faculdade de Engenharia da Universidade do Porto

Reconhecimento e seguimento de objectos num cenário multi-câmara

Telmo Afonso Vilar Gonçalves Oliveira

Relatório Final realizado no âmbito da Unidade Curricular
Preparação da Dissertação – EEC0035

Orientador: Prof. Dr. Luís Corte-Real
Proponente do tema: Eng. Pedro Carvalho

15 de Fevereiro de 2011

Índice

1 – Introdução

O seguimento e reconhecimento de objectos (VOT – *Video Object Tracking*) tem vindo a merecer crescente atenção nos últimos anos devido a uma série de aplicações que têm vindo a ser associados a este tema, como por exemplo *assisted living* ou aplicações militares. O VOT permite, através do uso de câmaras, a identificação de novos artefactos no seu campo de visão e o seu posterior reconhecimento. Na domótica, isto poderia facilmente traduzir-se em métodos biométricos de validação de entrada, ou como forma de melhor gerir os consumos energéticos, em função da localização dos utilizadores.

No domínio de VOT, pode ainda ser estudada uma área de cenários em que operam mais do que uma câmara. Se se tiver em mente o caso de sistemas de videovigilância, o recurso a um sistema de VOT multi-câmara consegue uma automatização de todos os processos inerentes a este domínio, com uma série de vantagens do ponto de vista da eficácia do sistema, associados a esta melhoria.

Pode-se assim ter um vislumbre de toda uma série de usos e aplicações para qual o VOT poderia ser adaptado, desde a simples automatização de tarefas ou registo de eventos, até ferramentas de observação de ambientes, cenários ou outros meios.

2 - Contextualização e objectivos

Esta dissertação enquadra-se no estudo de uma aproximação de uma solução que permita a detecção, localização e seguimento de objectos num cenário multi-câmara. Esta solução deverá ter um carácter automático (sem qualquer intervenção externa para controlo) e escalável (1 a N câmaras). Serão assim estudadas e abordadas as seguintes temáticas: técnicas de representação e identificação de objectos em sequências de vídeo; técnicas de detecção e seguimento de objectos em ambiente multi-câmara; desenvolvimento de novos modelos para a detecção e seguimento de objectos em ambiente multi-câmara; desenvolvimento de algoritmos para aprendizagem de relações entre câmaras.

Conduzir-se-ão testes dos modelos elaborados para responder aos problemas propostos e que permitirão inferir sobre a sua validade através da análise dos resultados finais de precisão, desempenho e eficiência. Pretende-se desta forma conseguir uma análise crítica do trabalho realizado. De forma a suportar a realização dos testes referidos, apresenta-se também neste documento uma ferramenta para o efeito - "*Framework* de avaliação de algoritmos de *tracking*".

No final do documento são indicadas as principais conclusões do estudo feito e apontado um plano de trabalhos para o desenvolvimento do tema.

3 - Estudo do estado-da-arte

Várias questões surgem ao estudar uma possível solução para o que é proposto. Nesse sentido, foi conduzido um estudo do estado-da-arte para uma familiarização e apreciação de diversas abordagens ao tema proposto, permitindo uma maior sensibilização às áreas abrangidas (multimédia, aprendizagem incremental, correlação de dados, etc.). Através da análise de artigos científicos, foi feita a familiarização com conceitos essenciais para o sistema.

O estudo pode ser subdividido em três tópicos principais: como detectar e segmentar um novo objecto; como o classificar e posteriormente identificar; como o localizar em cenários multi-câmara.

O primeiro tópico aborda o estudo feito no domínio da sinalização de novas ocorrências, nomeadamente como discernir entre um novo objecto que se move no cenário e outros artefactos presentes (ruído na imagem, mudanças de iluminação ou sombras e reflexos).

Na segunda parte são discutidos os métodos de classificação dos objectos segmentados anteriormente. Vários parâmetros são tidos em conta neste caso, que vão desde o tempo dispendido para a classificação do objecto e para sua identificação, até à forma de representação, podendo optar-se por um método de classificação que seja invariante a uma série de factores como escala ou rotação.

Por último será estudado o modo de localização em cenários multi-câmara, ou seja, que objecto da câmara X corresponde ao objecto detectado na câmara Y. Até este ponto apenas tinham sido estudados procedimentos numa única câmara. Nesta secção discutir-se-ão abordagens para a modelação de um relacionamento inter-câmara, nomeadamente para efeitos de *object handover* (transição de objectos detectados entre câmaras). É também referida a temática da aprendizagem incremental, de forma a que o sistema seja totalmente automático e possa tomar decisões baseadas em eventos passados.

3.1- Detecção e segmentação de novos objectos

Este é um primeiro e importante passo em *tracking* de objectos pois se não for correctamente executado, podem ser detectadas erroneamente como novas ocorrências pequenas variações de iluminação, ruído na imagem ou objectos estáticos no cenário. Por outras palavras, esta etapa tem como objectivo conseguir distinguir correctamente o *background* (cenário de operação, estático) do *foreground* (objectos móveis, pessoas).

Ao dimensionar uma possível solução, uma primeira abordagem para lidar com este problema é a "subtracção do *background*". Resumidamente, perante uma *frame* corrente captada, é-lhe subtraída uma imagem de referência do cenário obtida previamente (ou uma média de N imagens referência anteriores). Apesar da sua baixa complexidade esta operação revela-se ingénua, sendo demasiado susceptível a variações em ambientes não controlados como a iluminação ou novos objectos que se tornam estáticos (como por exemplo o simples mover de uma cadeira), assinalando muitos "falsos positivos".

Métodos mais elaborados (como [1], [2] ou [3]) modelam o background e vão actualizando o modelo à medida que novas ocorrências vão sendo detectadas. Estes métodos podem ser classificados como sendo preditivos ou não-preditivos. Os métodos preditivos modelam as cenas como um fluxo no tempo, usando um filtro de Kalman (como é o caso de [4]) para actualizar as mudanças detectadas de forma gradual, sendo portanto orientados a ambientes maioritariamente estáticos. Os métodos não-preditivos consistem na elaboração de uma representação probabilística de observações em pixéis. Tenta-se desta forma emular o comportamento de um pixel às variações a que este é submetido.

Várias abordagens destacam-se neste último campo, tendo-se revelado o método apresentado em [1] bastante eficaz nos testes efectuados pelos autores. Depois de modelar o ruído através de uma função gaussiana, e da definição de um *threshold* adequado para identificação de variações de luz, os autores sugerem 5 métodos diferentes para a identificação de comportamentos dinâmicos no *background*: *Running Average, Mixture of Gaussians* (MoG) (apresentado em [5]), *Kernel Density Function* (apresentado em [6]), *Principal Features* (apresentado em [7]) e *Mean Shift*. Depois de serem executados vários testes com diferentes sequências e usando uma métrica de avaliação (*Perceptual Spacial Quality*, métrica proposta em [8]), os autores chegam à conclusão de que o método MoG é o que leva a melhores resultados sem sacrificar demasiado o custo computacional.

O método MoG, ao invés de outros métodos que representam directamente o *background*, tem uma abordagem mais eficaz, ao estimar um modelo para o *background* que consiga prever o comportamento e variação de cada pixel por observação do "historial" desse mesmo pixel. Isto é conseguido com recurso à estimação da função densidade probabilidade (f.d.p.), como é mostrado em [1] e [5].

Referente à segmentação apresentada em [1], o método aplicado é o seguinte: (1) eliminação de pixéis detectados como *foreground* devido ao ruído (variações de iluminação ou

mudanças estruturais); (2) teste é feito para verificar se as mudanças detectadas não ocorreram devido a modificações das condições de iluminação; (3) refinamento final em que são eliminadas mudanças estruturais resultantes de comportamento dinâmico repetitivo no *background*. O resultado final destas operações é a classificação de um conjunto de pixéis como *foreground*. Existe ainda a vantagem de, devido à detecção das variações de iluminação, não serem detectadas as sombras projectadas no cenário, sem recorrer a técnicas de pós-processamento.

Esta é também a metodologia seguida em [3], diferindo em alguns pontos. Neste artigo, de forma a remover ruído, é efectuada a operação morfológica *close*. Para diferenciar entre vários objectos humanos, é usado um algoritmo que tem por base a cabeça das pessoas. É escolhido este método porque é a parte do corpo que mais se destaca, sendo-o naturalmente também no *blob* (conjunto de pixéis segmentados) localizado. Começando no topo da imagem, é feita uma pesquisa por essa parte, resultando numa aproximação grosseira do número de pessoas presentes (igual ao número de cabeças). No final, a abordagem descrita funciona em situações como: pequenos números de pessoas que se movimentam de forma conjunta; situações de oclusão; situações de sombra ou reflexos. Contudo, os autores admitem que em casos de, por exemplo, utilização de um guarda-chuva, o algoritmo de pesquisa baseado na cabeça humana não consegue bons resultados.

Em [3] são feitas também algumas críticas à referida modelação de background. É afirmado que existem várias desvantagens nesta abordagem, porque não incorpora restrições da forma do objecto e que cada *blob* pode não corresponder a um único objecto em movimento, sobretudo em aplicações no mundo real, em que devido à proximidade muitos *blobs* vão ser segmentados erradamente (a adicionar a isto pode ser também enunciado o facto de serem detectadas sombras e/ou reflexões, e que devido a baixos contrastes de cor, uma única pessoa pode ser fragmentada em diversos *blobs*).

Seguindo os modelos da forma humana (com recurso a elipsóides), os autores afirmam que são eliminados os problemas acima descritos, além de ser mais robusto a ruído e processamento de baixo-nível. No procedimento de identificação dos objectos, os *blobs* são extraídos do background, e são computados analisando a forma das *edges* detectadas (de forma a definir se são objectos humanos ou não).

3.2 - Classificação e Identificação de Objectos

Uma vez segmentado o objecto na imagem, este terá de ser assinalado de forma a ser processado posteriormente. Surge aqui a questão da sua classificação. Poderiam ser guardados todos os pixéis pertencentes aos objectos. No entanto este não é o melhor método, pois qualquer objecto que se mova no cenário irá variar na sua forma ao longo das suas várias instâncias, além das modificações que pode sofrer tais como um mudança de escala ou de

posição, mudanças no ponto-de-vista (rotação no eixo horizontal ou vertical) e oclusões parciais.

É neste contexto que surgem os descritores de *features*, que servem como uma "impressão digital" da imagem. Em suma, ao invés de serem representados os pixéis segmentados numericamente, é-lhes antes associado um descritor que represente as suas propriedades, sendo estas exploradas de formas diferentes consoante o descritor escolhido. Os descritores de *features* são também bastante utilizados em sistemas CBIR (*content based image retrieval*) que tiram partido da eficácia deste meio de representação de informação de imagem para gerir as operações de pesquisa e selecção de imagens (combinados muitas vezes com meta-dados). Alguns aspectos a ter em conta na selecção de descritores é o seu tempo de extracção, tempo de *query* e robustez a variações.

O trabalho realizado e apresentado em [9] é especialmente útil neste domínio. Este artigo propõe-se à realização de um estudo comparativo dos vários descritores visuais, analisando ainda quais deles poderiam ser combinados para conduzir a melhores resultados, discriminando ainda que descritores estão mais adaptados a determinadas tarefas (extracção de informação, relevância da informação, redundância, tempo de pesquisa, etc.). Segundo os autores, os descritores podem ser subdivididos em 4 campos: representação de cor, textura, forma e *features* locais. É com base nesta subdivisão que é elaborada uma tabela com os resultados de testes de extracção e pesquisa dos descritores numa base de dados, sendo estes agrupados por tipo. Neste campo, os descritores baseados em cor e *features* locais têm um melhor desempenho que os restantes.

Depois de realizados os testes de precisão com todos os descritores, são apontadas várias conclusões relevantes: os histogramas de cor e descritores MPEG7 não apresentam bons resultados em imagens em escala de cinzento; por outro lado, os histogramas de cor, invariantes à escala e distribuição espacial, têm um óptimo desempenho em todas as imagens com uma gama alargada do espectro de cor; para uso generalizado, os descritores SIFT (*scale invariante feature transform*) apresentam os melhores resultados.

Ao analisar as correlações entre descritores, os autores chegam à conclusão de que: os histogramas invariantes de cor e descritores MPEG7 apresentam uma forte relação, pelo que não há ganhos significativos na sua combinação; apesar de 30 anos de pesquisa em descritores de textura, nenhum apresenta uma performance de destaque, podendo no entanto ser combinados com outros descritores; adicionando a um descritor SIFT qualquer outro descritor, conduzirá a uma melhor performance em qualquer sistema CBIR. Ao terminar, os autores concluem que os histogramas de cor representam uma boa base para tarefas que envolvam fotografias de cor, tendo no entanto os descritores SIFT um melhor desempenho na maioria das tarefas exigidas (acarretando contudo um maior custo computacional).

Em [10], além dos já mencionados, são apresentados novos descritores e conclusões relevantes sobre estes. Neste trabalho, o processo de detecção e *matching* é acelerado com o uso de localização de pontos de interesse, que levam a uma área de pesquisa menor. No final, os autores concluem que: os histogramas de cor têm uma boa performance em cenários de câmara única, sendo no entanto vulneráveis a má calibração e variações de iluminação, o que faz com que sejam uma má escolha para o caso de multi-câmara; os histogramas de gradientes orientados (HOG – *Histogram of Oriented Gradientes*, apresentado em [11]) apresentam óptimos resultados e são computacionalmente eficientes; o método SIFT é invariante a todas as condições acima descritas, sendo-o também o descritor SURF, estudado em [12] (com a vantagem de este ter melhor prestação em termos de velocidade e *accuraccy*). Este último representa uma grande vantagem em relação ao SIFT, pois apesar de este conseguir bons resultados, o seu peso computacional leva a que muitas vezes seja descartado.

Na área da identificação de objectos, o artigo [13] também apresenta técnicas interessantes: depois de detectado um novo indivíduo no cenário, é mostrado que são conseguidos melhores resultados na sua posterior pesquisa e identificação se se optar por uma subdivisão em duas partes do corpo (uma superior, outra inferior) como demonstrado em [14]. Como descritores dos segmentos detectados são utilizados histogramas de cor RGB normalizados para uma escala de cinzento, segundo [14].

O autor analisa depois as possíveis soluções para o passo da identificação. Partindo do princípio de que todas as instâncias estão registadas, estas são comparadas entre si, de forma a discriminar se um par de instâncias pertence ao mesmo indivíduo ou não. Na comparação, são testados vários métodos diferentes, sendo ainda estudado se a aprendizagem incremental de um método de comparação seria mais vantajoso do que um método previamente definido. A comparação é feita como sendo a distância no espaço entre dois descritores. Os métodos apresentados para o cálculo desta distância são: distância euclidiana, *Symmetrical Mahalanobis Distance* e *Diffusion Distance* (estudada em [15]). Para a aprendizagem incremental, foi realizada uma implementação do algoritmo sugerido em [16]. É ainda testado um método de identificação utilizado maioritariamente em aplicações biométricas de reconhecimento facial (*Sparse Recognition Classifier*, apresentado em [17]). No final do artigo, depois de testados todos os métodos, chega-se à conclusão de que a divisão do corpo dos indivíduos detectados em duas partes conduz a melhores resultados. Todos os métodos apresentados têm prestações semelhantes, excepto o *Symmetrical Mahalanobis Distance* que resulta sempre em piores performances. É de realçar também o facto de que o método mais elementar e directo de entre todos, a distância euclidiana, figura sempre nos melhores valores obtidos.

Para fins de descrição de cenário existe um descritor que, segundo [18] e [19], tem óptimos resultados (sobretudo em aplicações de CBIR). Estes descritores (GIST, estudado em

[18]) não usam informação de cor, apenas de estrutura, sendo invariantes a mudanças de luminância, focagem ou redimensionamento. É dito em [19] que são superiores aos histogramas de cor pois estes, pelo facto de na sua generalidade serem invariantes à translação ou rotação, representam também uma desvantagem, pois a orientação da imagem pode ter relevância em algumas aplicações. A grande vantagem na utilização dos descritores GIST reside na sua rapidez de execução. No quadro mostrado em [20], vemos que os resultados conseguidos para uma pesquisa de uma imagem são de 1,3 segundos para uma base de dados de 10M imagens. Se se recorrer a uma variante optimizada deste descritor (GISTIS, mencionada em [20]), é conseguido para a mesma pesquisa um tempo de 38 ms.

No artigo [2] é seguida outra abordagem. Ao ser detectado um objecto, é extraído e armazenado o seu descritor, sendo-lhe atribuída uma identificação. O descritor inicialmente eleito foi o MCSHR (*major color spectrum histogram representation*, usado pela primeira vez em [21]). No entanto na implementação foi usada uma variante, o IMCSHR (incremental MCSHR) para compensar pequenas variações do objecto. É seguidamente aplicada uma transformada para atenuar variações de iluminação. São também utilizados descritores locais em conjunto com pontos-chave (*key interest points*). Para este último caso são utilizados descritores SIFT, que são depois quantizados de forma a serem identificados por uma "palavra" num dicionário (elaborado através da relação hierárquica entre os descritores). Em suma, é usado um *bag-of-visterms*, sendo realizado um *cluster* de descritores SIFT, emulando um método de dicionário (*codebook*), ou seja, as palavras são usadas para descrever os objectos detectados. Esta abordagem consegue assim juntar a alta capacidade de um descritor SIFT com o poder de generalização de um *cluster* palavras.

3.3 - Localização em cenários multi-câmara

Uma vez realizados os passos anteriores, resta a análise de modelos para o tratamento da informação em ambientes multi-câmara. Dado o facto de este ser um domínio, relativamente recente na área de seguimento de objectos, não existe ainda nenhum método estabelecido e consensual para a modelização das relações inter-câmara. No entanto esta é uma área fulcral, pois através de um bom modelo é conseguido um aumento da eficiência algorítmica da pesquisa por objectos ou *object handover*. Neste âmbito, devem também ser estudadas as questões referentes à gestão/atenuação de diferenças ente câmaras (nomeadamente a nível das diferenças de iluminação, calibragem, posicionamento do nó de observação, etc.)

É apresentado em [10], um sistema de *tracking* que, segundo os autores, funcionará em qualquer rede usando câmaras fixas e móveis não calibradas (designadas como *master* e *slave*, respectivamente). Os objectos são detectados com as câmaras móveis a partir de observações feitas pelas câmaras fixas (seja por processamento do *foreground* ou por

selecção manual dos objectos). O funcionamento do descritor de objectos (OD – *object descriptor*) proposto é o seguinte: o objecto é segmentado com recurso a uma *Bounding Box* (BB) e subdividido em rectângulos de tamanho igual, sendo usados os rectângulos menores para informação local e a BB maior usada para análise do comportamento global. Várias observações do mesmo objecto são usadas no processo de *matching*.

Para a localização dos objectos são usadas 2 estratégias:

- *dense scan*: uma janela proporcional à BB do objecto é usada para pesquisar a imagem, comparando o número de *edges*.

- *sparse scan*: é comparado o ponto de interesse do objecto com os pontos de interesse da imagem, de forma a reduzir o número de regiões a pesquisar. As regiões a pesquisar vão sendo reduzidas até restar uma única.

Depois de conduzidas as experiências, é mostrado na conclusão que a abordagem tem uma melhor performance do que se se tivesse recorrido apenas a um único descritor, e que o sistema não é afectado por variações de iluminação, pontos-de-observação, distribuição de cores, qualidade de imagem e oclusão parcial. É enfatizado também o facto de que a redução do espaço de pesquisa leva a um desempenho que se aproxima do "tempo-real" (ou seja, apesar de haver um inevitável tempo de atraso, este é mínimo).

Os autores de [4] seguem um procedimento diferente. O sistema proposto baseia-se em 3 parâmetros para estabelecer a relação entre as câmaras: cor, tamanho e movimento. Esses parâmetros são também usados para classificar uma nova instância como sendo um novo objecto, ou um objecto já observado. Os autores afirmam que utilizando aprendizagem incremental, *links* probabilísticos e com recurso à BB de um objecto para a sua representação, é conseguido um elevado desempenho. Os *links* probabilísticos entre regiões de entrada/saída (do ambiente em que está inserido) providenciam a informação sobre o movimento. A grande vantagem desta abordagem, segundo os autores, é o facto de a precisão do sistema ser melhorada à medida que o tempo passa e mais objectos vão sendo observados e processados, tudo isto sem informação *a priori*.

Segundo a metodologia seguida, os movimentos são observados ao longo do tempo para estabelecer períodos de reaparecimento dos objectos. Novas instâncias são detectadas e, intra-câmara, o *foreground* é correlacionado temporalmente com as *frames* anteriores usando um filtro Kalman. Se a instância já tiver sido observada, é classificada como um "objecto observado", e o seu descritor é actualizado; caso não tenha sido observada, é classificada como "novo objecto". Caso um objecto da *frame* anterior não tenha sido encontrado na actual, é então classificado de "*exiting object*" (passando para o processamento inter-câmara, onde o descritor do objecto é comparado com os descritores processados, conjuntamente com as novas instâncias observadas em outras câmaras). A relação entre câmaras é estabelecida dividindo cada câmara numa grelha composta por 16 regiões. Assim que um objecto for detectado é assinalado com uma BB, sendo o tamanho

desta utilizado para fazer a correspondência (*link*) entre as regiões de cada câmara (ou seja, um objecto que saia do campo de visão de uma câmara, tendo uma BB de determinado tamanho, irá reaparecer numa outra câmara com uma BB de outro tamanho, mapeando assim as probabilidades de regiões de entrada/saída).

Referente ao estabelecimento de *links*, em [2] é também apresentado um procedimento interessante, igualmente com recurso à aprendizagem incremental por observação dos objectos, o que caracteriza este tipo de metodologias como sendo soluções escaláveis. No trabalho os autores afirmam que devido ao facto de os objectos sofrerem alterações ao longo do tempo, é necessária a sua actualização constante. Para abordar este problema, é utilizada *adaptative learning* através do algoritmo Learn++, solução proposta em [22]. Desta forma, os autores do texto afirmam que o método pode ser implementado em cenários com pontos-de-observação independentes, sendo este o propósito do artigo.

4 – *Framework*

Tal como mencionado para alguns dos casos apresentados, de forma a melhor poder avaliar a performance dos métodos escolhidos são aplicadas métricas de avaliação de imagem. Neste sentido, colaborou-se na elaboração de uma "*Framework* de avaliação de algoritmos de *tracking*" para, como o nome indica, realizar um *benchmark* dos métodos seleccionados (tendo como *input* as sequências de imagem que serão alvo dos algoritmos de *tracking*, será registada numericamente a avaliação da precisão dos resultados obtidos). Uma completa e detalhada *overview* da *Framework* pode ser consultada em [24]. Esta plataforma recorre a 4 tipos de métricas (informação adicional sobre estas pode ser encontrada em [23] ou [25]):

-NGT (*non ground truth*): ao contrário dos mais frequentemente utilizados métodos GT (*ground truth*) que comparam informação de referência com o resultado dos algoritmos, esta métrica utiliza as imagens da sequência e as máscaras etiquetadas resultantes do algoritmo de *tracking*.

-PD (*partition distance*): utiliza as máscaras de referência e as máscaras etiquetadas que resultam do algoritmo, comparando-as.

-baseada em BB (*bounding box*): o objecto na sequência é assinalado com uma BB e é depois comparado com informação de referência. Tanto o resultado do algoritmo como a informação de referência são comparados com recurso a registos CVML que são gerados (variante do XML adaptado para fins de *Computer Vision*).

-métricas híbridas: combinação das métricas mencionadas anteriormente, nomeadamente PD-BB, PD-NGT e PD-BB-NGT.

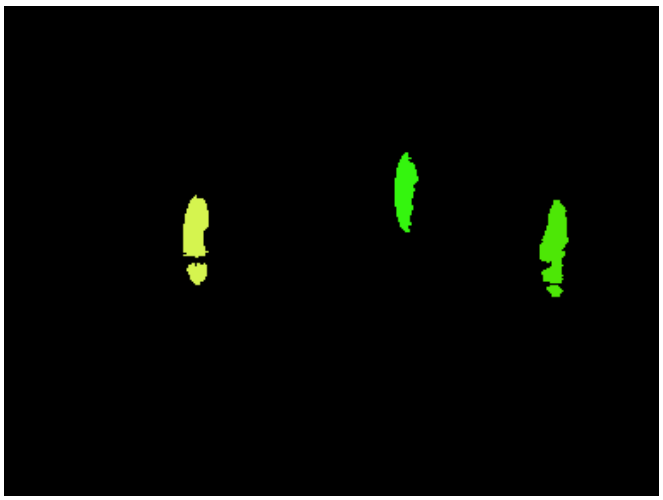

Figura 1 – Exemplo 1 de utilização de máscaras. *Dataset* OneShopOneWait1

Figura 2 – Exemplo 2 de utilização de máscaras *Dataset* One ShopOneWait1

Na imagem da Figura 3 encontra-se assinalada a componente do projecto na qual se focou o trabalho inicialmente desenvolvido: *binding* das métricas implementadas em Python para C/C++. Contudo, após uma análise de todas as opções disponíveis, optou-se por seguir o método mais imediato da chamada ao sistema para invocação das métricas.

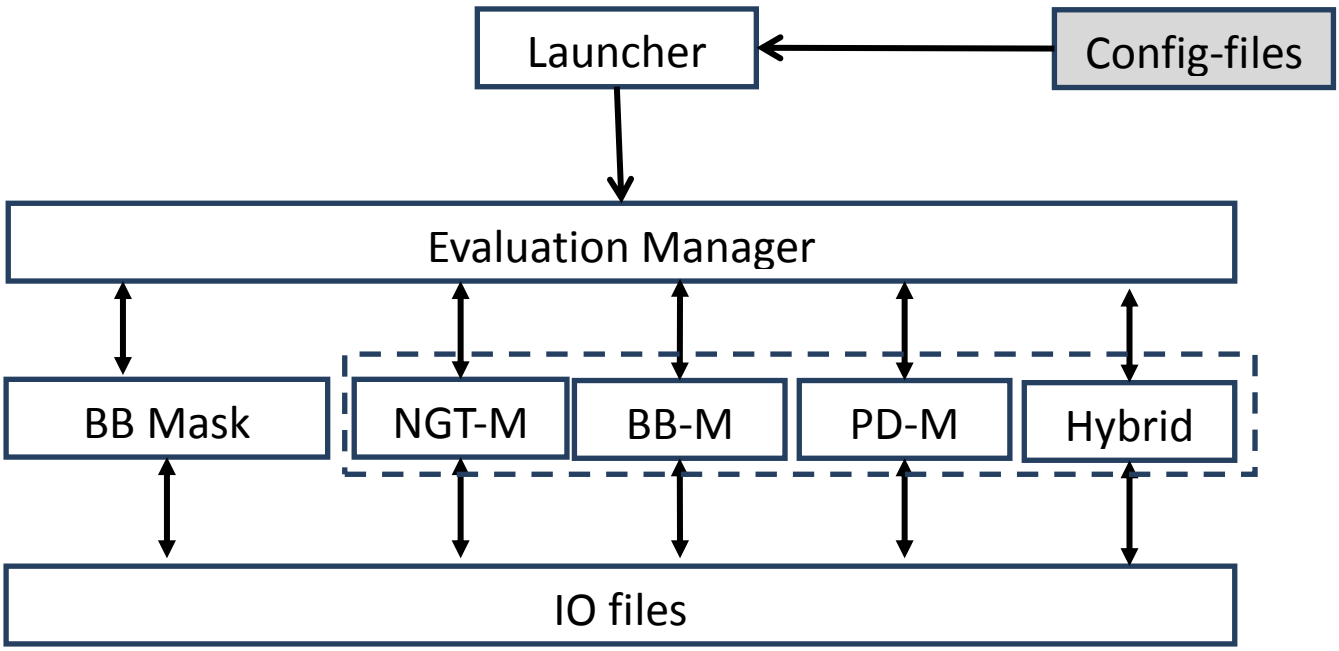

Figura 3 - Arquitectura funcional da *Framework*. Componente em que se colaborou assinalada a tracejado

5 – Plano de Trabalhos e Conclusão

Depois de propostos os vários métodos para as diferentes fases do projecto, prevê-se como trabalho imediato o registo das sequências de vídeo que serão alvo dos testes a serem executados.

Como trabalho futuro, planeia-se a elaboração de um modelo de detecção de objectos, seguido da escolha de um OD adequado. A este último passo, dada a sua relevância e complexidade, será dada especial atenção. Realizar-se-ão diferentes testes para cada implementação de cada OD. Enquanto alguns dos autores dos artigos estudados disponibilizam a implementação dos algoritmos apresentados, há casos em que tal não acontece, pelo que se prevê também uma faze, no plano de trabalhos, que seja dedicada à modelação das soluções analisadas.

Com recurso à *Framework* mencionada anteriormente, serão realizados diferentes testes, de forma a seleccionar o melhor OD (ou conjunto de OD), consoante a sua complexidade (tempo de execução, extracção, pesquisa) e os seus resultados finais.

Por último, será elaborado um modelo para o cruzamento de dados inter-câmara, estando prevista a utilização de uma aprendizagem incremental, sendo esta uma fase em que muito trabalho será realizado, dado que os resultados finais serão fortemente influenciados por este passo.

Relativamente às tecnologias e ferramentas a utilizar, estas ficarão restringidas pelas implementações existentes das soluções estudadas. Como a grande maioria destas se encontra disponível em C/C++ e MATLAB®, o ambiente de trabalho será o Visual Studio 2010, sendo a linguagem de programação escolhida o C/C++ pois esta permite a utilização de funções MATLAB®, tornando assim possível a agregação de todas as soluções numa única plataforma.

6 – Referências

[1] L. F. Teixeira e L. Corte-Real. (2007). "Cascaded change detection for foreground segmentation". IEEE Workshop on Motion and Video Computing (WMVC'07).

[2] L. F. Teixeira e L. Corte-Real. (2009). "Video object matching across multiple independente views using local descriptors and adaptive learning". Pattern Recognition Letters 30, pp. 157-167.

[3] T. Zhao, e R. Nevatia. "Tracking Multiple Humans in Complex Situations". IEEE Transactions on pattern analysis and machine intelligence, Vol 26 No. 9, September 2004.

[4] A. Gilbert, e R. Bowden. "Incremental scalable tracking of objects inter camera". Computer Vision and Image Understanding 111 (2008), pp. 43-58.

[5] D. S. Lee. "Effective Gaussian mixture learning for video background subtraction". IEEE Transactions on Pattern Analisys and Machine Intelligence, 27(5):857-832, May 2005.

[6] A. Elgammal, D. Hardwood, and L. Davis. "Non-parametric model for background subtraction". Proceedings of European Conference on Computer Vision, volume 2, pages 751-767, 2000.

[7] L. Li, W. Huang, I. Y. Gu e Q. Tian. "Statistical modeling of complex backgrounds for foreground object detection". IEEE Transactions on Image Processing, 13(11): 14591472, November 2004.

[8] A. Cavallaro, E.D. Gelasca, e T. Ebrahimi. "Objective evaluation of segmentation quality using spatio-temporal contexto". Proceedings of IEEE International Conference on Image Processing, September 2002.

[9] T. Deselaersm D. Keysers, e H. Ney. "Features for Image Retrieval: An Experimental Comparison", Novembro 29, 2007.

[10] A. Alahi, P. Vandergheynst, M. Bierlaire, e M. Kunt. "Cascade of Descriptors to Detect and Track Objects Across Any Network of Cameras". Computer Vision and Image Understanding, August 11, 2009.

[11] D. Lowe, "Distinctive image features from scale-invariant keypoints". International Journal of Computer Vision 60(2) (2004) 91-110.

[12]H. Bay, T. Tuytelaars, L. Van Gool, "SURF: Speeded Up Robust Features", Lecture Notes in Computer Science 3951 (2006) 404.

[13] D. Figueiredo, "Re-Identification of Visual Targets in Camera Networks – A comparison of techniques". VISAAP 2010.

[14] T. Cong, D.N., Achard, C., Khoudour, L., e Douadi, L. (2009). "Video sequences association for people re-identification across multiple non-overlapping cameras". ICIAP '09: Proceedings of the 15th International Conference on Image Analysis and Processing, pp. 179-189.

[15] H. Ling, e K. Okada. (2006). “Diffusion distance for Histogram comparison”. CVPR ’06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 246-253.

[16] E. P. Xing, A. Y. Ng, M. I. Jordan e S. Russel. (2002). “Distance metric learning with application to clustering with side-information”. Advances in Neural Information Processing Systems 15, pp. 505-512.

[17] D. L. Donoho .(2004). “For most large undetermined systems of linear equations the minimal 11-norm solution is also the sparsest solution”. Comm. Pure Appl. Math, 59:797-829.

[18] M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg, e C. Schmid. “Evaluation of GIST descriptors for web-scale image search”. CIVR 09, July 8-10, 2009.

[19] Disponível em http://cybertron.cg.tu-berlin.de/pdci08/mosaic/ImageDescriptors.pdf - Acesso em 10/02/2011.

[20] Disponível em http://lear.inrialpes.fr/pubs/2009/DJSAS09/gist_evaluation_talk.pdf - Acesso em 10/02/2011.

[21] C. Madden, E.D. Cheng, e M. Piccardi. 2007. “Tracking People across disjoint camera views by an illumination-tolerant appearance representation. Machine Vision Appl. 18 (3), 233-247.

[22] R. Polikar, L. Udpa, S. S. Udpa, e L. Honavar. 2001. “Learn++: Na incremental learning algorithm for supervised neural networks. IEEE Trans. Systems Man Cybernet.- Part C; Appl. Rev. 31 (4), 497-508.

[23] Ç. E. Erdem, B. Sankur, e A. M. Tekalp. “Performance Measures for Video Object Segmentation and Tracking”. IEEE Transactions on Image Processing, Vol. 13, No. 17, July 2004.

[24] P. Carvalho, J. S. Cardoso, e L. Corte-Real. “Hybrid Framework for evaluating video object tracking algorithms”. Electronic Letters, 18th March 2010, Vol. 46 No. 6.

[25] F. Bashir, F. Porikili. “Performance Evaluation of Object Detection and Tracking Systems”.