

UNIVERSIDADE FEDERAL DO ABC

CENTRO DE ENGENHARIA, MODELAGEM E CIÊNCIAS SOCIAIS
APLICADAS

FABIANA DA SILVA GUIZI

**Sistema Rastreador de Pessoas para Ambientes Internos com
Câmeras**

Santo André - SP

2018

Fabiana da Silva Guizi

**Sistema Rastreador de Pessoas para Ambientes Internos com
Câmeras**

Relatório de Trabalho de Graduação

Relatório apresentado ao Curso de Graduação em Engenharia de Informação da Universidade Federal do ABC, como requisito para aprovação na disciplina Trabalho de Graduação III.

Professor Dr. Celso Setsuo Kurashima

Docente Orientador, Universidade Federal do ABC

Santo André – SP

2018

Aos meus pais, Ana Maria da Silva Guizi e José Luiz Guizi, e aos meus irmãos, Fernanda Silva Guizi e Luiz Henrique Silva Guizi, pelo amor, suporte e (muita) paciência.

AGRADECIMENTOS

À Ana Maria da Silva Guizi, José Luiz Guizi, Fernanda Silva Guizi e Luiz Henrique Silva Guizi, pelo amor, incentivo e suporte constante.

À minha falecida avó, Aparecida Rozeira Guizi, pelo suporte, amor e por ter sempre me aconselhado a ter perseverança e paciência para atingir meus objetivos.

A minha família em geral, pelo suporte e incentivo.

Ao orientador Prof. Doutor Celso Setsuo Kurashima, pelo acompanhamento competente e dedicado na elaboração deste trabalho e durante os anos de orientação de projeto de iniciação científica.

A todos amigos da UFABC que me ajudaram a seguir com o curso.

Aos professores da UFABC que compartilharam conhecimentos de forma eficiente e pela inspiração.

A todos os que direta ou indiretamente contribuíram para a realização desta pesquisa.

The Doctor: “The shepherd’s boy says, “There’s this mountain of pure diamond. It takes an hour to climb it, and an hour to go around it. Every hundred years, a little bird comes. It sharpens its beak on the diamond mountain. And when the entire mountain is chiseled away, the first second of eternity will have passed. ” You must think that’s a hell of a long time. Personally, I think that’s a hell of a— *{the wall breaks.}* Personally, I think that’s a hell of a bird. ” *{The Doctor steps through a portal, and the water castle world then becomes his Confession Dial}*

“The Heaven Sent”. *Doctor Who*, BBC Wales, Cardiff, 2015.

RESUMO

A detecção e o rastreamento de pessoas em vídeo encontram uma grande variedade de aplicações em visão computacional, tais como monitoramento, segurança, entretenimento, navegação de robôs, ambientes inteligentes, entre outras. No entanto, o desenvolvimento de metodologias robustas para estes propósitos é desafiante devido variação de aparência, pose e posição das pessoas em vídeo, condições de iluminação, complexidade do ambiente e ocorrência de oclusões. Metodologias clássicas de detecção e rastreamento de objetos ou pessoas envolvem a identificação de padrões e outros atributos, que correspondem a aspectos das imagens relacionados à informação de cor dos elementos de interesse. Trabalhos recentes sobre o assunto têm apontado vantagens de abordagens com uso de múltiplas câmeras e/ ou informação 3D do ambiente, pois apresentam maior robustez a variação de iluminação e melhor desempenho na ocorrência de oclusões. A presente proposta de trabalho de graduação em Engenharia de Informação visou à investigação de sistemas de detecção e rastreamento com uso de múltiplas câmeras para aplicações em ambientes fechados. Neste relatório são apresentados detalhes de arquitetura e da implementação do sistema de detecção e rastreamento desenvolvido, baseado em metodologia com integração de informação de imagens de diferentes perspectivas da cena analisada. Os resultados mostraram que o sistema implementado apresenta desempenho satisfatório na detecção e rastreamentos de até duas pessoas em ambiente fechado, mesmo na ocorrência de perdas de detecção, mas com restrição de proximidade entre os elementos de interesse. O sistema ainda apresentou pequena diminuição na taxa de quadros para as configurações do *hardware* de processamento e taxa de detecção e acerto relativamente altas para as condições de testes realizados, com taxas superiores a 0.8.

Palavras-chave: Processamento de Imagens, Visão de Máquina, Múltiplas Câmeras;

ABSTRACT

People detection and tracking in video have a wide variety of applications in computer vision which include monitoring, surveillance, entertainment, rover navigation, intelligent environments, etc. However, the development of robust methodologies for these purposes is challenging due to variation people appearance, pose and position in videos, lighting conditions, complexity of environments and the occurrence of occlusions. Classical methodologies of object or people detection and tracking involve the identification of patterns and other features, which correspond to aspects of the images related to the color information of the targets. Recent works on this subject have pointed to the advantages of multiple camera and/ or 3D environment information approaches, since they present greater robustness to the lighting variation and better performance in the occurrence of occlusions. The present proposal of graduation work in Information Engineering aimed at the investigation of people detection and tracking systems based on multiple cameras for indoor environment applications. In This report presents details of architecture and implementation of the detection and tracking system developed based on methodology with integration of information of images from different perspectives of the analyzed scene. The results showed the implemented system presents satisfactory performance for detection and tracking of up to two people indoor environment, even in the occurrence of detection losses, but with a restriction of proximity between the elements of interest. The system still showed a small decrease in the frame rate for the processing hardware configurations and relatively high detection and hit rate for the tests conditions performed, with rates higher than 0.8.

Keywords: Image Processing, Machine Vision, Multiple Cameras.

LISTA DE FIGURAS

Figura 1-1 – Fluxo de processamento típico de sistemas de detecção e rastreamento de pessoas. Fonte: Adaptado de (CAMPLANI, 2017).	2
Figura 4-1 – Esquema de configuração de ambiente de uso do sistema.	7
Figura 4-2 – Esquema de módulos de processamento e fluxos principais de informações do sistema proposto.	8
Figura 4-3 – Submódulos de processamento do sistema proposto e suas entradas e saídas.	9
Figura 4-4 – Fluxo de processamento do submódulo de gerenciamento de rastreamentos.	10
Figura 4-5 – Fluxo de processamento do submódulo de subtração de fundo.	11
Figura 4-6 – Fluxo de processamento do submódulo de filtragem de máscaras de <i>foreground</i>	11
Figura 4-7 – Exemplo de erro de paralaxe: (a) Visão da câmera com dois elementos de interesse; (b) Imagem combinada de diferentes visões com vários pontos de máximo de intensidade. ...	12
Figura 4-8 – Fluxo de processamento do submódulo de intensificação de presença.	13
Figura 4-9 – Fluxo de processamento do submódulo de filtragem de imagem de intensificação de presença.	14
Figura 4-10 – Fluxo de processamento do submódulo de transformação de perspectiva.	14
Figura 4-11 – Fluxo de processamento do submódulo de combinação de imagens.....	15
Figura 4-12 – Fluxo de processamento do submódulo de filtragem de imagem combinada. ...	15
Figura 4-13 – Fluxo de processamento do submódulo de detecção de regiões de interesse.	16
Figura 4-14 – Fluxo de dados entre submódulos do módulo de rastreamento.	17
Figura 4-15 – Fluxo de processamento do submódulo de gerenciamento de rastreamentos. ..	18
Figura 4-16 – Fluxo de processamento do submódulo de correspondência.	19
Figura 4-17 – Fluxo de processamento do submódulo de predição de posição de rastreamento.	19
Figura 4-18 – Fluxo de processamento de construção de imagem de saída do sistema.....	20
Figura 6-1 – Quadros de entrada – 210° iteração.	43
Figura 6-2 – Saídas do submódulo de subtração de fundo: (a) Modelos de fundo; (b) Máscara de <i>background</i>	44
Figura 6-3 – Saídas de submódulo de filtragem de máscara de <i>foreground</i>	44
Figura 6-4 – Saídas do submódulo de intensificação de presença: (a) Teste 1; (b) Teste 2; (c) Teste 3; (d) Teste 4.	46
Figura 6-5 – Saídas do submódulo de combinação de imagens: (a) Teste 1; (b) Teste 2; (c) Teste 3; (d) Teste 4.....	48
Figura 6-6 – Saídas do submódulo de filtragem de combinação de imagens: (a) Teste 1; (b) Teste 2; (c) Teste 3; (d) Teste 4.....	49
Figura 6-7 – Imagens de saídas do sistema: (a) Teste 1; (b) Teste 2; (c) Teste 3; (d) Teste 4.	50
Figura 6-8 – Sequência de resultados de processamento dos 525° e 530° quadros de entrada das sequências: (a) Teste 1; (b) Teste 2; (c) Teste 3; (d) Teste 4.....	52
Figura 6-9 – Resultados de processamento dos 635° quadros de entradas dos vídeos: (a) Teste 1; (b) Teste 2; (c) Teste 3; (d) Teste 4.	53

Figura 6-10 – Resultados de processamento dos 710º quadros de entradas dos vídeos: (a) Teste 1; (b) Teste 2; (c) Teste 3; (d) Teste 4.	54
Figura 6-11 – Resultados de processamento dos 810º quadros de entradas dos vídeos: (a) Teste 1; (b) Teste 2; (c) Teste 3; (d) Teste 4.	55

LISTA DE TABELAS

Tabela 5-1 – Características de vídeos “ <i>Laboratory Sequency</i> ”. (EPFL, 2018)	23
Tabela 5-2 – Equações do filtro de Kalman. Adaptado de (WELCH e BISHOP, 2006)	35
Tabela 6-1 – Configuração de entradas e parâmetros para testes.....	42
Tabela 6-2 – Configuração de teste comuns.....	43
Tabela 6-3 – Resultados de eficiência de detecção e rastreamento.	56

SUMÁRIO

1.	INTRODUÇÃO.....	1
2.	OBJETIVO	4
3.	TRABALHOS CORRELATOS	5
4.	VISÃO GERAL DO SISTEMA PROPOSTO	7
4.1.	Configuração do ambiente de aplicação	7
4.2.	Organização funcional do sistema	8
5.	MATERIAIS E MÉTODOS.....	21
5.1.	Hardware.....	21
5.2.	Biblioteca OpenCV	21
5.3.	Linguagem de Programação	22
5.4.	Ambiente de desenvolvimento.....	22
5.5.	Base de vídeos	22
5.6.	Implementação do sistema proposto.....	23
5.6.1.	Entrada.....	23
5.6.2.	Módulo de detecção.....	25
5.6.3.	Módulo de rastreamento	31
5.6.4.	Saída	39
5.7.	Procedimentos de análise de resultados.....	39
5.8.	Metodologia de avaliação de desempenho	40
6.	RESULTADOS E ANÁLISE.....	42
6.1.	Análise geral de aspectos de processamento	43
6.2.	Análise geral de resultados de detecção e rastreamentos.....	50
6.3.	Análise da eficiência de detecção e rastreamento.....	55
7.	CONCLUSÃO.....	58
	REFERÊNCIAS BIBLIOGRÁFICAS	60

1. INTRODUÇÃO

A detecção e o rastreamento de pessoas em ambientes representam tópicos de pesquisa com relevância em visão computacional, pois possuem importantes aplicações nas áreas de vigilância, entretenimento, ambientes inteligentes, esportes, análise de comportamento de pedestres, entre outras (IOCCHI e BOLLES, 2005; JIANG et al., 2010). No entanto, o desenvolvimento de metodologias robustas para estes propósitos é desafiante devido à variação de posições, condições de iluminação, complexidade do ambiente e ocorrência de oclusões. Metodologias clássicas de detecção e rastreamento de objetos ou pessoas baseiam-se no uso de sensores RGB e envolvem a identificação de padrões e outros atributos, que correspondem a aspectos das imagens relacionados à informação de cor. Apesar da eficiência dessas abordagens em ambientes controlados, muitos são os problemas de desempenho quando essas metodologias são aplicadas para análise de cenários reais.

Trabalhos recentes sobre o assunto têm apontado vantagens de abordagens com uso de múltiplas câmeras e /ou informação 3D do ambiente, pois apresentam maior robustez a variação de iluminação e melhor desempenho na ocorrência de oclusões (DARELL et al. 2001; XIA et al., 2011). Com isso, metodologias com uso de câmeras com pelo menos duas perspectivas, tais como apresentadas em (BEYMER e KONOLIGE, 1999), (DARELL et al. 2001) e (KELLY, 2008) utilizam a informação 3D da cena para lidar com variações de cor e iluminação e obtenção de importantes informações e reconhecimento de objetos de interesse. Já os métodos com múltiplas perspectivas como os propostos em (BLACK et al., 2002), (COLLINS et al., 1999), (FLEURET et al., 2008) e (TAJ e CAVALLARO, 2013) são capazes de lidar de forma eficiente com a ocorrência de oclusões sem a necessidade de aplicação de algoritmos correspondência esparsa ou densa entre pares de pontos de imagens em diferentes perspectivas. Através da combinação da informação de calibração das câmeras em suas diversas perspectivas e da remoção de fundo estes métodos, obtêm-se meios para a integração da informação de fontes de vídeo com diferentes perspectivas e modelos de pessoas e outras categorias de elementos em movimento nos vídeos.

O fluxo de processamento de sistemas de detecção e rastreamento de objetos ou pessoas pode ser representado de forma genérica conforme o esquema apresentado na Figura 1-1. Os processos que recorrentemente são implementados nesses sistemas, tais como etapa de detecção, correspondência e atualização de trajetórias, são indicados por linhas sólidas. A etapa opcional de processamento, no caso a identificação de regiões de interesse (do inglês, *Region Of Interest* ou *ROI*), é representada por um bloco com linhas tracejadas.

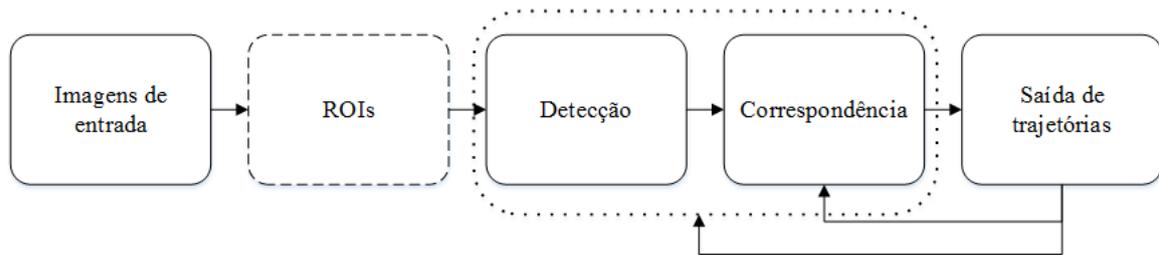


Figura 1-1 – Fluxo de processamento típico de sistemas de detecção e rastreamento de pessoas. Fonte: Adaptado de (CAMPLANI, 2017).

As entradas de sistemas de detecção e rastreamento são geralmente imagens digitais da cena em análise. A etapa de detecção, executada pelo bloco *Detecção* (Figura 1-1), consiste na busca por formas humanas ou outras formas de interesse nas imagens de entrada para um dado instante de processamento, baseando-se na informações de intensidade cor ou de profundidade. Opcionalmente, uma etapa de seleção de área de interesse (bloco tracejado denominado *ROI* na Figura 1-1) pode ser executada a fim de reduzir o espaço de busca da etapa de detecção e diminuir o tempo de processamento.

Na etapa de correspondência (bloco *Correspondência* na Figura 1-1), as detecções de pessoas realizadas na etapa anterior são associadas a um rastreamento independente, com a finalidade de estabelecer a descrição da trajetória das pessoas em vídeo ao longo do tempo. Tais trajetórias comumente estão relacionadas a informações como posição ou descrições de modelo de aparência ou de movimento. Modelos de aparência são representações dos elementos de interesse através de histograma de cor, formas geométricas, atributos de profundidade, etc., que auxiliam na identificação o elemento de interesse nas imagens. Modelos de movimento, por sua vez, auxiliam na descrição da trajetória dos elementos rastreados, sendo recorrente o emprego de métodos matemáticos como os filtros bayesianos para predição de posição desses elementos. Dentre os principais métodos de estimativa de movimento destacam-se o filtro de Kalman em sua versão clássica e a estendida, e os filtros de partículas. O filtro de Kalman estima processos utilizando um controle de *feedback*: o filtro estima o estado do processo em um determinado instante de tempo e obtém dessa estimativa o *feedback* na forma de medidas com ruído (WELCH e BISHOP, 2006). Esse modelo, apesar de ser baseado em hipóteses bastante restritas – isto é, (1) o sistema modelado é linear, (2) o ruído das medições é aproximadamente branco e (3) também de natureza gaussiana –, encontra aplicação em várias situações gerais (BRADSKI et al., 2008). A versão estendida do filtro de Kalman já é utilizada para lidar com estimativas relacionadas a processos ou medidas não lineares. Modelos mais complexos de

movimento que consideram múltiplas hipóteses são tratados por filtros de partículas, com destaque para sua implementação através do algoritmo *condensation* (ISARD e BLAKE, 1996).

Tendo em vista os principais aspectos relacionados a detecção e o rastreamento de pessoas e objetos em vídeos, o presente projeto propôs-se a realizar a investigação de metodologias de detecção e rastreamento de pessoas baseadas em múltiplas perspectivas para aplicações em ambientes fechados. A partir dessa investigação foi elaborada uma proposta de arquitetura de sistema para estes propósitos fundamentada na combinação de imagens em diferentes perspectivas da cena analisada em uma única imagem no plano do chão. Neste relatório são apresentados os principais detalhes da arquitetura, resultados e conclusões obtidos a partir da implementação do protótipo de sistema planejado.

2. OBJETIVO

O projeto apresentado neste trabalho de graduação de Engenharia de Informação teve por objetivo a investigação de sistemas rastreadores de pessoas com uso de múltiplas câmeras para operação em ambientes fechados. A meta específica deste trabalho foi a elaboração do protótipo de um “Sistema Rastreador de Pessoas para Ambientes Internos com Câmeras” que fosse capaz de realizar a detecção e o rastreamento de pessoas em recintos fechados empregando uma metodologia baseada no uso de informações de câmeras em diferentes perspectivas da cena analisada.

3. TRABALHOS CORRELATOS

A presente proposta de projeto visou realizar a detecção de o rastreamento de pessoas com uso de câmera em múltiplas perspectivas. Na literatura recente, trabalhos como (FLEURET et al., 2008), (SANTOS e MORIMOTO, 2011) e (TAJ e CAVALLARO, 2013) ilustram abordagens consideradas interessantes para a solução do problema de interesse.

O sistema proposto por (FLEURET et al., 2008) faz uso de um *framework* probabilístico para executar a detecção e o rastreamento simultaneamente. O modelo do sistema proposto baseia-se na combinação de um modelo simples de movimento com um modelo de aparência. O modelo de aparência é composto da combinação das informações de densidade de cor RGB (histogramas) e um mapa de ocupação (*occupancy map*) no plano do chão. No mapa de ocupação o chão é particionado em células regulares, sendo que a probabilidade de ocupação de cada uma dessas células é estimada por subtração de fundo. Dessa maneira, o modelo de mapa de ocupação é uma distribuição condicional entre o *foreground* e a configuração de células ocupadas. O método ainda utiliza o algoritmo de Viterbi para encontrar a trajetória mais provável para cada indivíduo.

Já no método proposto por (SANTOS e MORIMOTO, 2011) os pixels de *foreground* resultantes da subtração de fundo de câmeras em várias perspectivas são utilizados como evidências da localização de pessoas, tendo também como referência o plano do chão (*ground plane*). Assim, o algoritmo proposto calcula a chamada *foreground mass* acima de cada pixel e é adotado o conceito de imagem de suporte, a qual representa justamente *foreground mass* para cada pixel de imagem em sua perspectiva original. A imagem de suporte é normalizada para compensar os efeitos de perspectiva e acumulados na referência do plano para cada câmera, de modo que a detecção de indivíduos se resume a uma busca por regiões de máximos no acumulador combinada com a informação de histograma de cor para as regiões de *foreground* em que estes estão localizados nas perspectivas originais das imagens analisada. Para estes métodos, a maioria dos candidatos falso positivos são filtrados a partir verificação da consistência das detecções candidatas em relação a todas as perspectivas da cena consideradas. Ao final, apenas os candidatos que passaram pelo processo de filtragem são rastreados com uso do filtro de Kalman e o modelo de aparência.

No trabalho (TAJ e CAVALLARO, 2013) é proposto um sistema de rastreamento de pessoas que utiliza o método de *track-first*, no qual o gerenciamento dos rastreamentos é efetuado a partir da localização e rastreamento dos elementos de interesse em cada vídeo de

entrada e nas perspectivas originais dos quadros com uso de filtros de partículas. Em seguida, as informações individuais de rastreamento também são combinadas e em uma única imagem em *top view* e projetadas no plano do chão. A principal vantagem desse método em relação as metodologias de *detect-and-track* como as utilizadas por (FLEURET et al., 2008) e (SANTOS e MORIMOTO, 2011) é a eliminação do passo de detecção após a fusão das informações das imagens, o que implica na redução da taxa de falsos positivos relacionados ao ruído de processo de detecção. No entanto, o uso de filtro de partículas nos vídeos em várias etapas de processamento torna a implementação da estratégia uma tarefa relativamente trabalhosa.

Para a solução proposta neste projeto, buscou-se uma representação equivalente do chamado mapa de ocupação (FLEURET et al., 2008) e da imagem de suporte (SANTOS e MORIMOTO, 2011) também a partir da remoção de fundo das imagens em diferentes perspectivas. No entanto, diferentemente dos trabalhos citados em que os modelos de aparência também são compostos por histogramas de cores RGB dos elementos de interesse nas perspectivas em que estes estão visíveis, o modelo de aparência utilizado neste trabalho considera apenas a representação dos elementos em *foreground* no plano do chão da cena.

O sistema proposto neste trabalho é caracterizado por executar a detecção e o rastreamento de elementos em *foreground* da cena em análise a partir da representação desses elementos no plano do chão da cena. As etapas de processamento do sistema proposto foram desenvolvidas com a finalidade de aperfeiçoar a representação da integração das informações das múltiplas perspectivas da cena para que as etapas de detecção e gerenciamento dos rastreamentos fossem executadas utilizando apenas as informações disponíveis e deduzíveis dessa representação.

4. VISÃO GERAL DO SISTEMA PROPOSTO

Nesta seção são apresentados aspectos de caracterização do sistema proposto, como a configuração do ambiente idealizado para aplicação e organização funcional.

4.1. Configuração do ambiente de aplicação

A Figura 4-1 apresenta o esquema de configuração do ambiente de aplicação do sistema proposto.

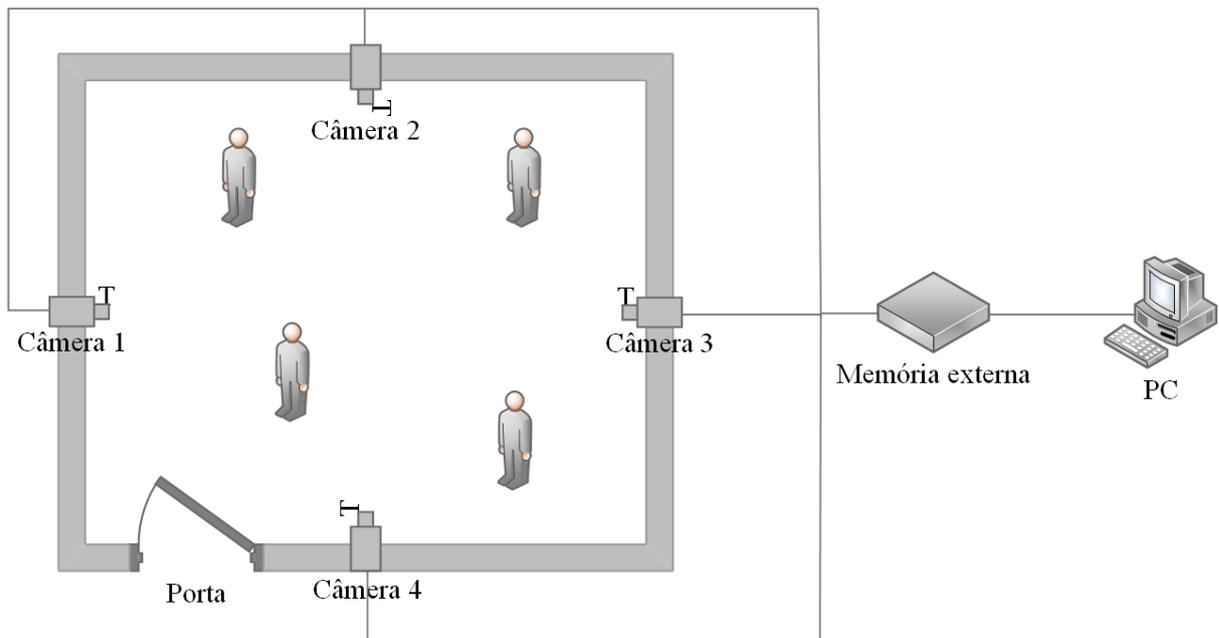


Figura 4-1 – Esquema de configuração de ambiente de uso do sistema.

O sistema proposto foi idealizado para análise de imagens de recintos fechados nos quais as informações sobre movimento de pessoas fossem de relevância e caracterizados por uma densidade de pessoas moderada, tais como lojas, escritórios, salas de espera, saguões, corredores, salas de aula, cárceres, etc. Nesses ambientes, câmeras posicionadas em regiões estratégicas devem realizar o registro e transferência simultânea de imagens com diferentes perspectivas de forma a maximizar o campo de visão do ambiente. As imagens registradas pelas câmeras devem ainda apresentar variações de iluminação e cor moderadas, conforme requisitos para desempenho adequado de algoritmos sensíveis a este tipo de oscilação de intensidade, como o algoritmo de subtração de fundo. As imagens obtidas podem ser armazenadas em memória para poderem ser posteriormente processadas por um computador ou sistema embarcado com a implementação do sistema de processamento proposto implementado.

4.2. Organização funcional do sistema

A organização funcional do sistema para o protótipo do sistema consiste em dois módulos principais: o módulo de detecção e o módulo de rastreamento. De uma forma simplificada, a função do módulo de detecção é efetuar a detecção de pessoas a partir das imagens de várias câmeras, considerando uma técnica de combinação das várias imagens processadas em uma única perspectiva de referência. O módulo de rastreamento, por sua vez, tem como entrada as informações de relevância da saída do módulo de detecção, sendo responsável pela identificação dos indivíduos em cena ao longo do tempo através da combinação de técnicas para correspondência entre os indivíduos detectados e rastreados e de predição de movimento. A organização funcional simplificada do sistema através de representação por módulos principais de processamento e os principais fluxos de dados é ilustrada na Figura 4-2.

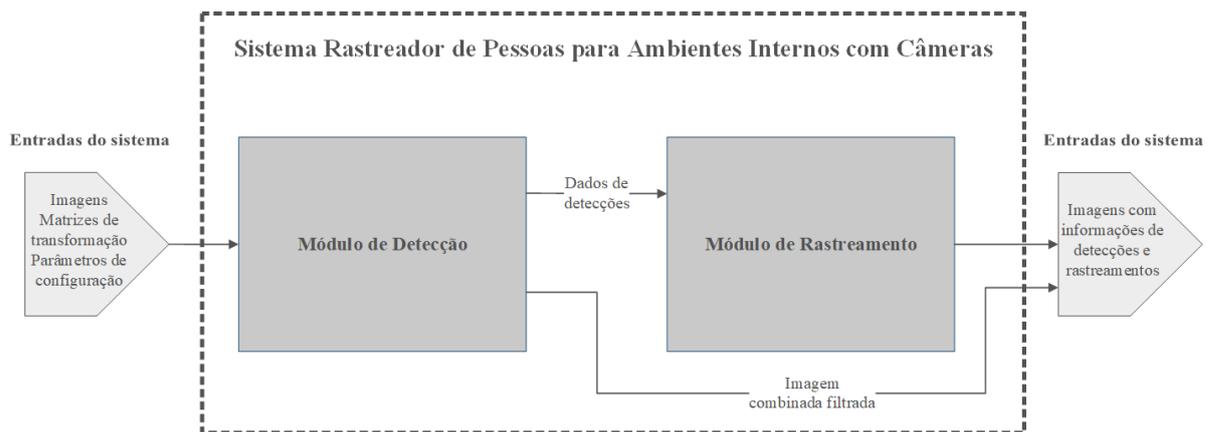


Figura 4-2 – Esquema de módulos de processamento e fluxos principais de informações do sistema proposto.

Um modelo mais detalhado da organização funcional do sistema apresentado ainda pode ser definido, considerando a subdivisão de cada módulo em submódulos de processamento, conforme mostrado na Figura 4-3.

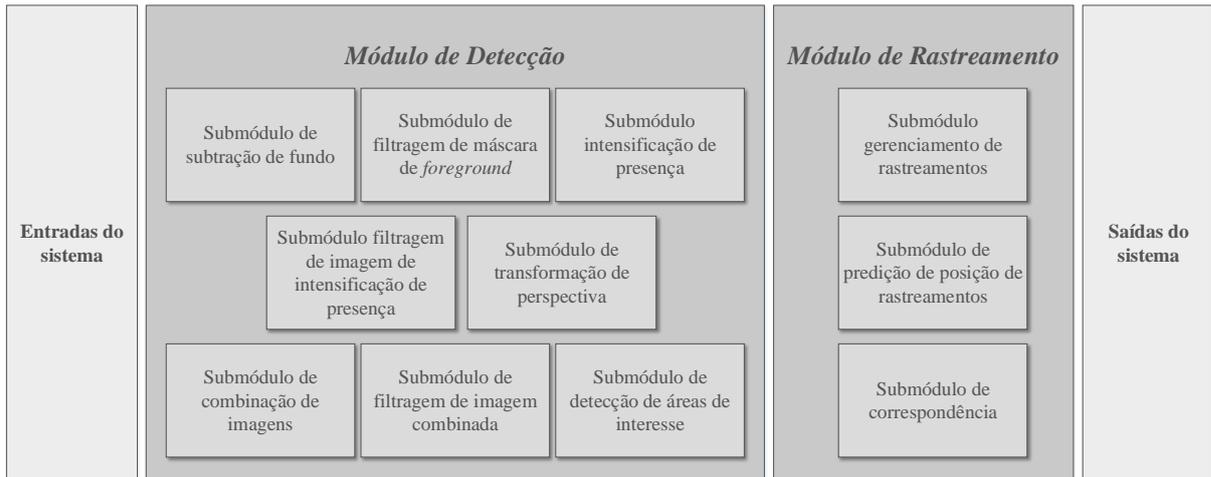


Figura 4-3 – Submódulos de processamento do sistema proposto e suas entradas e saídas.

No esquema representado na Figura 4-3 as *entradas do sistema* referem-se não apenas as imagens digitais de múltiplas perspectivas e em sincronia, mas também as informações para auxiliar no processamento dessas imagens, como matrizes de transformação de perspectiva (também chamadas de homografias) e os parâmetros de configuração para as funções dos constituintes dos submódulos do sistema.

A identificação dos elementos de interesse é executada pelo módulo de detecção, cujo fluxo de informações externo e interno entre seus submódulos constituintes está representado na Figura 4-4.

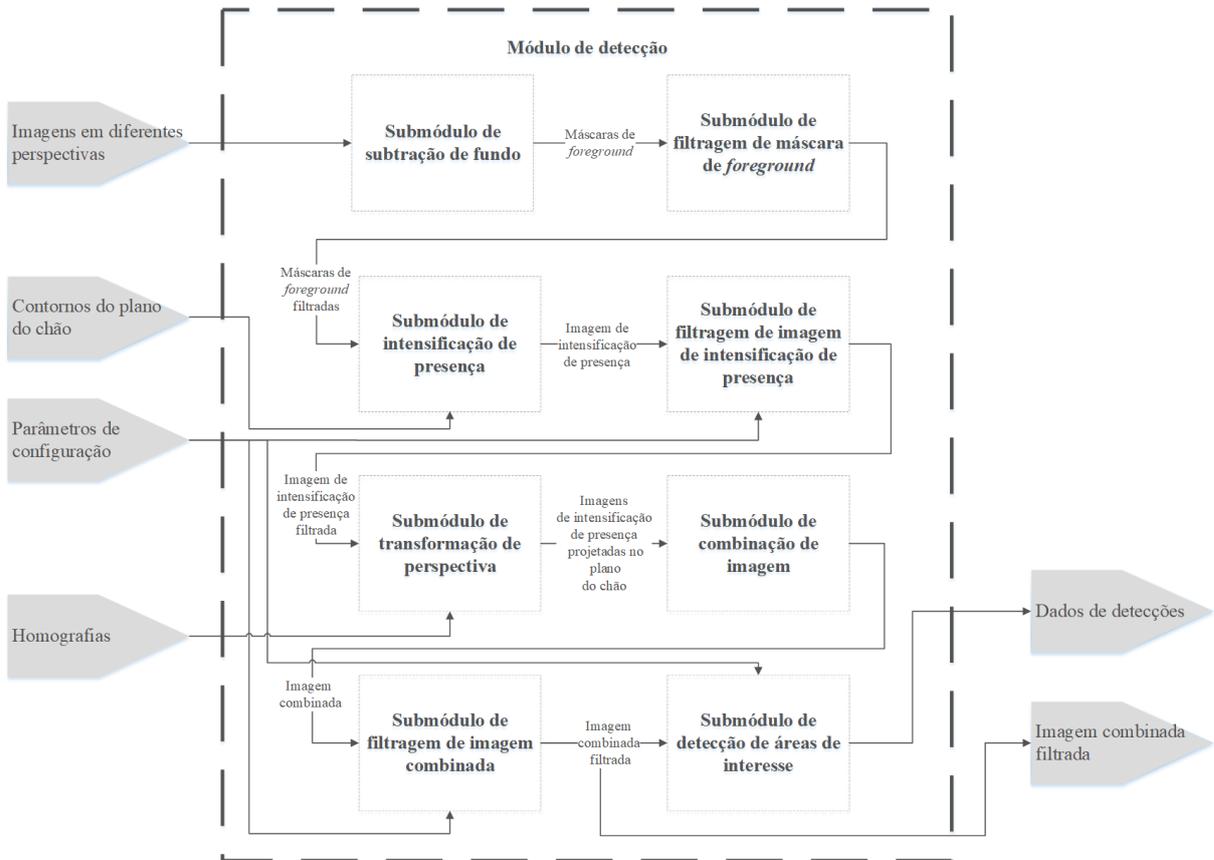


Figura 4-4 – Fluxo de processamento do submódulo de gerenciamento de rastreamentos.

A primeira etapa de processamento é efetuada pelo *submódulo de remoção do fundo*, (Figura 4-5). Nela, um modelo de fundo é criado a partir das informações estatísticas da cena sem a presença das pessoas. A partir desse modelo e da comparação de cada quadro de entrada em um dado instante de análise são criadas as chamadas *máscaras de foreground*. Na máscara de *foreground* os pixels brancos representam as regiões unitárias das imagens para as quais o critério de decisão do algoritmo de remoção de fundo julgou como não pertencente ao fundo. Por outro lado, os pixels em preto constituem regiões considerada como representativas do fundo da imagem.

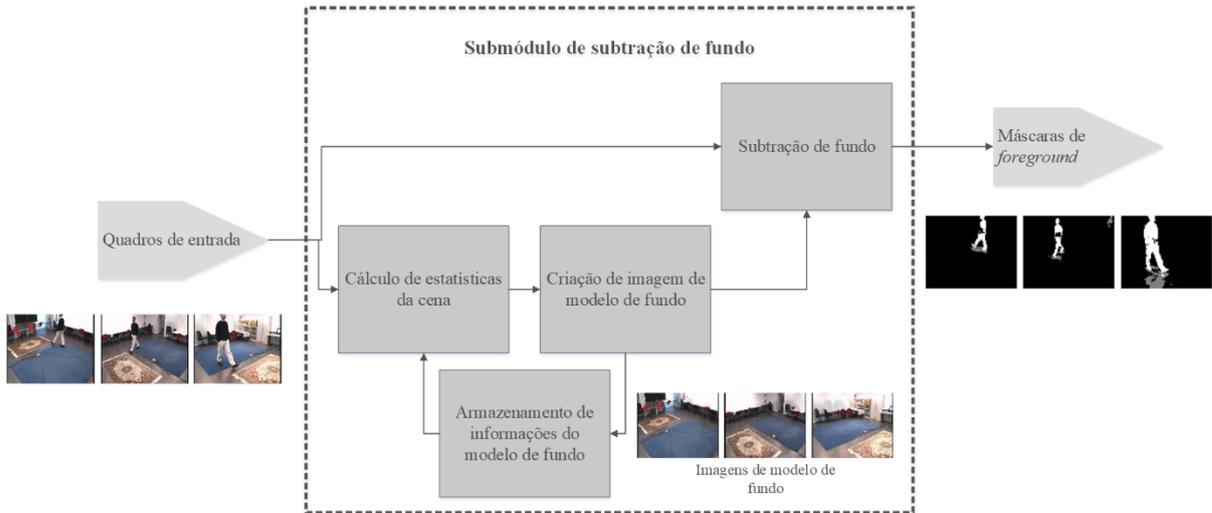


Figura 4-5 – Fluxo de processamento do submódulo de subtração de fundo.

Em geral, as *máscaras de foreground* são computadas com certo grau de ruído que pode ser minimizado através de operações morfológica e de filtragem. Tais funções são realizadas pelo *submódulo de filtragem de máscara de foreground*. Com isso, as inconsistências mais prováveis neste tipo de imagem são eliminadas, diminuindo-se as chances de obtenção de resultados ruins nas etapas subsequentes de processamento. A Figura 4-6 mostra o fluxo de processamento executado neste submódulo. A filtragem se inicia com aplicação operações morfológicas sobre as imagens de entrada, tais como erosão e dilatação (BRADSKI et al, 2008), para eliminação dos pequenos aglomerados de pixels brancos ou pretos em regiões predominantemente pretas e brancas, respectivamente. Em seguida, um filtro de suavização é aplicado para uniformizar as regiões de *foreground* e fundo das imagens. Por fim, a geração de imagens de binárias é realizada através de definição de limiar de intensidade e polarização dos valores dos pixels em preto e branco em relação a este limiar.

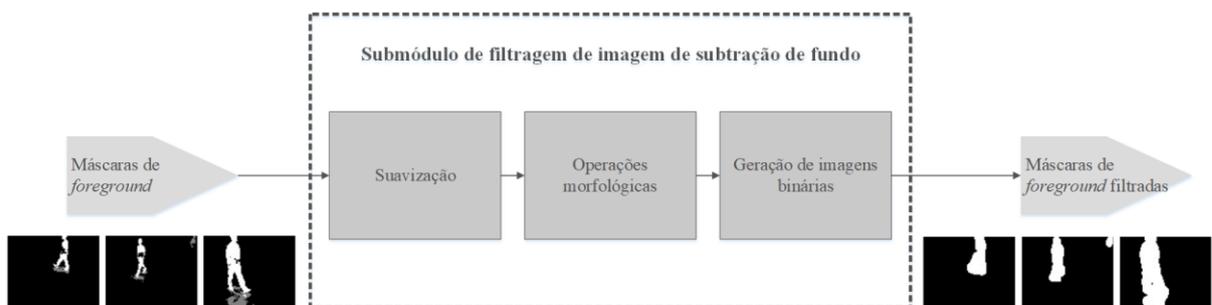


Figura 4-6 – Fluxo de processamento do submódulo de filtragem de máscaras de *foreground*.

Visto que a etapa de detecção se processa através da identificação de máximos de intensidade da combinação das informações de *foreground* no plano do chão, uma das principais fontes de erro de detecção é a interseção de regiões das imagens projetadas neste plano e que não representam necessariamente as posições que os elementos de interesse ocupam na cena real. No caso da projeção das máscaras de *foreground*, tais regiões correspondem aos pixels de *foreground* que representam a extensão do corpo dos indivíduos em vídeo. A Figura 4-7 ilustra a ocorrência de erros de paralaxe da combinação direta das informações de *foreground* no plano do chão. Este instante em que há duas pessoas na cena (Figura 4-7 (a)), na imagem em Figura 4-7 (b) esperava-se dois pontos de máximo nas posições ocupadas pelas pessoas no plano de chão. No entanto, outros três pontos com intensidade relativamente alta podem ser identificados.

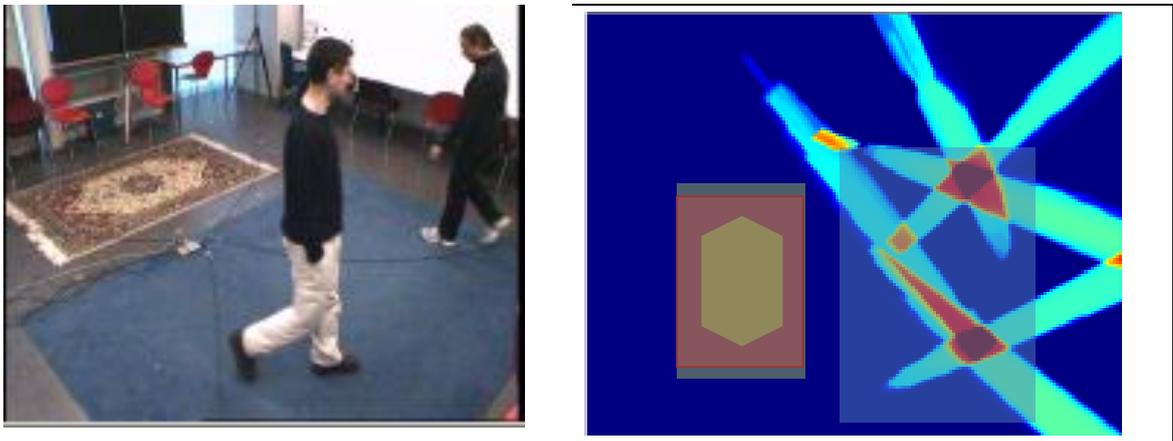


Figura 4-7 – Exemplo de erro de paralaxe: (a) Visão da câmera com dois elementos de interesse; (b) Imagem combinada de diferentes visões com vários pontos de máximo de intensidade.

Dessa maneira, antes serem projetadas para o plano do chão, uma nova representação das imagens filtradas deve ser construída a fim de evitar os possíveis erros de paralaxe no processo de combinação das imagens de diferentes perspectivas. Assim, no *submódulo de intensificação de presença* são geradas imagens nas quais busca-se projetar a altura dos aglomerados de pixels de *foreground* – também chamados de *blobs* – no plano do chão através do somatório do número de pixels de *foreground* acima das regiões de chão e da normalização dos aglomerados de pixels com valores não nulos – este último processo aplicado para lidar com variações de escala das pessoas em vídeo em diferentes distâncias em relação aos dispositivos de captura. A informação de contorno do chão é também utilizada na etapa de normalização para garantir que apenas as regiões que tangem o plano do chão sejam representadas com máximos de intensidade.

Nas ilustrações que representam as saídas desse submódulo mostradas na Figura 4-8, é possível notar que nas imagens de intensificação de presença os pixels *blobs* que representam as regiões próximas ao chão apresentam maior intensidade do que as extremidades opostas. Este tipo de resultado diminui as chances de erros de paralaxe na medida em que intensifica as regiões das imagens que ao serem projetadas no plano do chão representam a posição aproximada dos elementos de interesse no plano do chão da cena e diminui a intensidade dos pixels relativos as porções do corpo dos indivíduos que não são projetados nesta real posição.

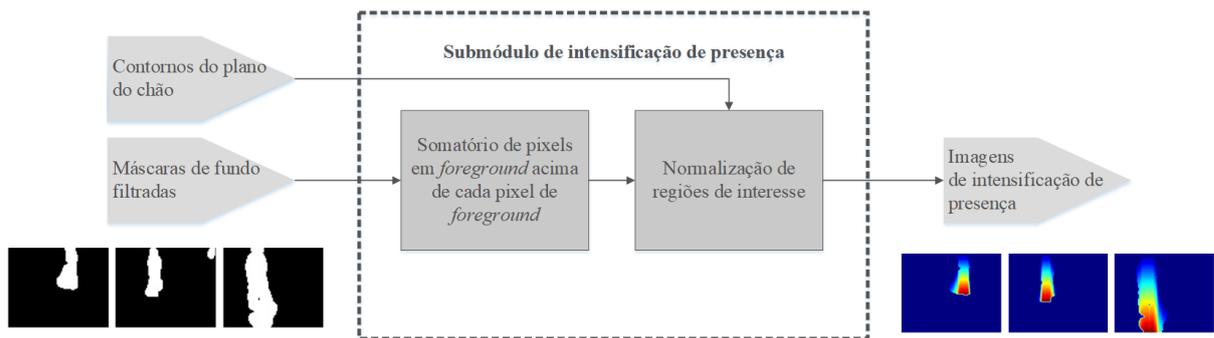


Figura 4-8 – Fluxo de processamento do submódulo de intensificação de presença.

É possível limitar ainda mais a quantidade de pixels *foreground* da imagem de intensificação de presença estabelecendo um limiar de valor mínimo para representação de pixels com valores não nulos. Com isso, também é definido o ***submódulo de filtragem de imagem de intensificação de presença***, cujo processamento, conforme especificado pelo fluxo de processamento apresentado na Figura 4-9, consiste de um filtro com valor de limiar definido como entrada do sistema dentro da faixa de 0 a 255. As informações de contorno do chão ainda são utilizadas pelo mesmo submódulo a fim de que apenas regiões de *blobs* no plano original sejam computadas com pontos de máximo, prevenindo a computação de *blobs* resultantes da segmentação do corpo dos indivíduos com valores de máximos.

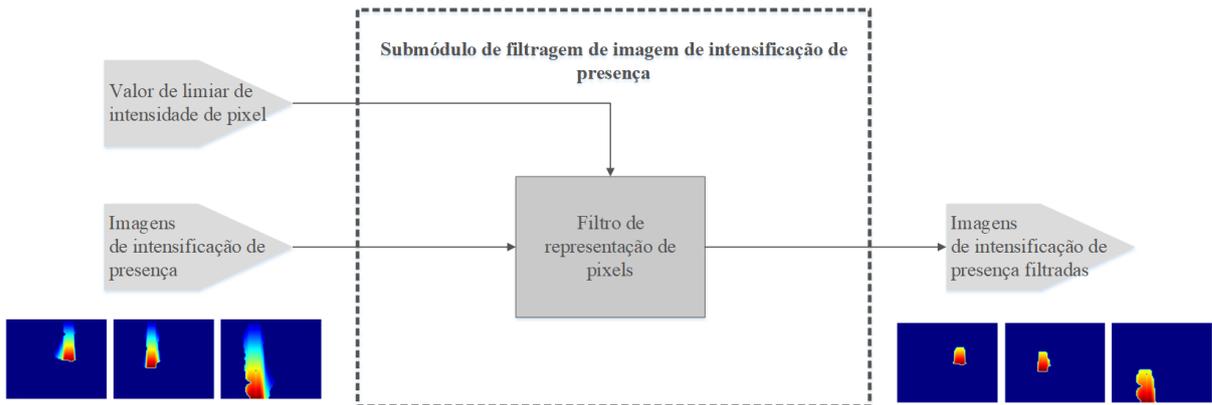


Figura 4-9 – Fluxo de processamento do submódulo de filtragem de imagem de intensificação de presença.

O *submódulo de transformação de perspectiva* é responsável por executar a mudança de perspectiva das imagens de saída do *submódulo de intensificação de presença*. Dessa maneira, essas imagens são projetadas do plano original de acordo com a perspectiva das imagens que as originaram para o plano do chão. Conforme mostrado na Figura 4-10, para que esta transformação seja realizada é necessária uma matriz de transformação de perspectiva ou homografia, a qual é específica para cada uma das imagens de entradas de acordo com sua perspectiva original.

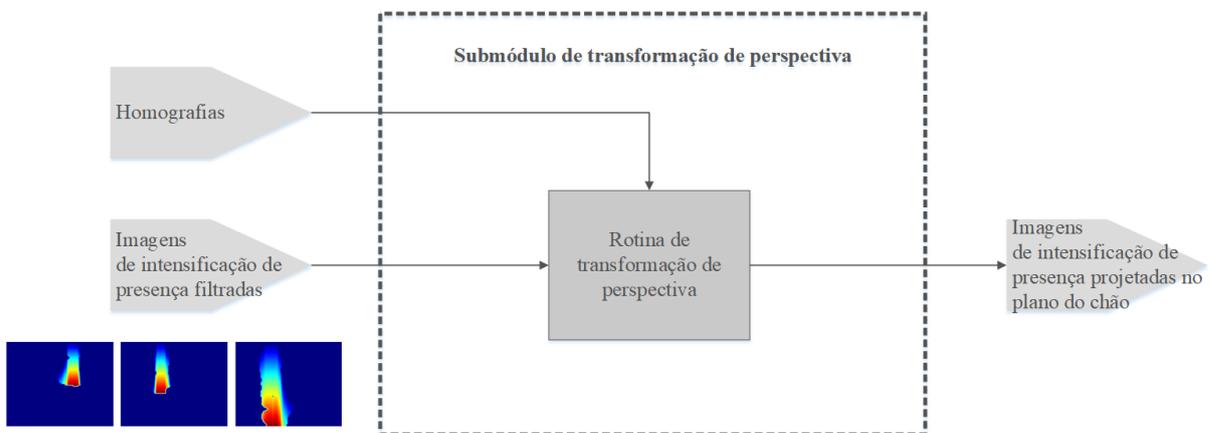


Figura 4-10 – Fluxo de processamento do submódulo de transformação de perspectiva.

No *submódulo de combinação de imagens*, as imagens projetadas no plano do chão são combinadas em uma única imagem visando intensificar regiões de interseção de informações e a agregar informações complementares entre as imagens de diferentes perspectivas. Assim como pode observado na Figura 4-11, a combinação de imagens conta com duas etapas principais: a primeira de *soma pixel a pixel das intensidades de pixel de cada uma das entradas* para compor o valor de pixel na posição correspondente em uma nova imagem e a de

normalização dos valores dessa imagem para representação do resultado da combinação na escala compatível para processamento nas etapas posteriores – isto é, intensidades de 0 a 255.

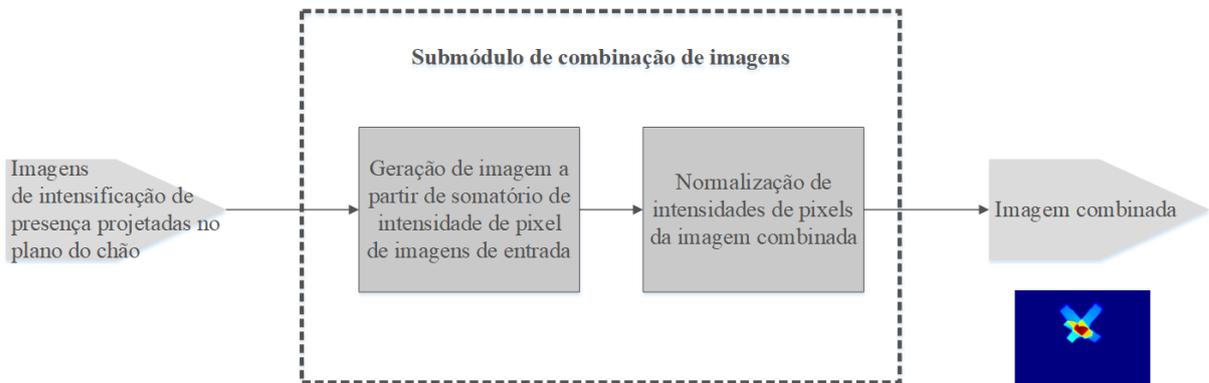


Figura 4-11 – Fluxo de processamento do submódulo de combinação de imagens.

No **submódulo de filtragem de imagem combinada** (Figura 4-12) cria-se uma imagem binária a partir da definição de um valor de limiar de intensidade de pixel para a imagem combinada. Assim como na etapa de filtragem das imagens de intensificação de presença o valor do limiar de representação dos pixels é também definido na entrada do sistema rastreador (parâmetro de configuração). É neste submódulo que são definidas as regiões de máximo que, de acordo com as características de tamanho e posição, poderão ser identificadas como elementos de interesse no **submódulo de detecção de áreas de interesse**.

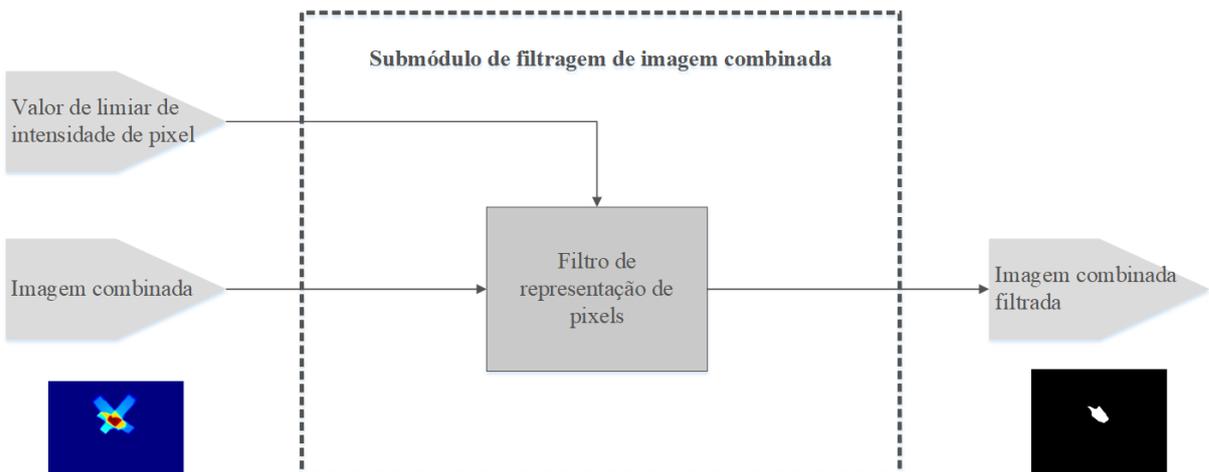


Figura 4-12 – Fluxo de processamento do submódulo de filtragem de imagem combinada.

Da imagem binária no **submódulo de detecção de áreas de interesse** são detectadas as regiões de *blobs* de acordo com os parâmetros de tamanho mínimo e máximo de *blobs* definidos experimentalmente e em conformidade com as características das imagens de entrada. As posições dos centroides das detecções juntamente com seus retângulos delimitadores são

armazenadas e lançadas como entradas do módulo de rastreamento e da etapa de geração de imagem de saída. Vale ressaltar que em razão das distorções das imagens nas próximas as regiões de borda – oriundas das projeções das imagens de seus planos originais para o plano do chão –, a área de análise para esta etapa de processamento é reduzida, sendo que a porcentagem da redução de dimensões de largura e altura são também definidas como parâmetros de entrada do sistema. Na Figura 4-13 são identificadas as principais etapas de processamento desse módulo, suas entradas e saídas.

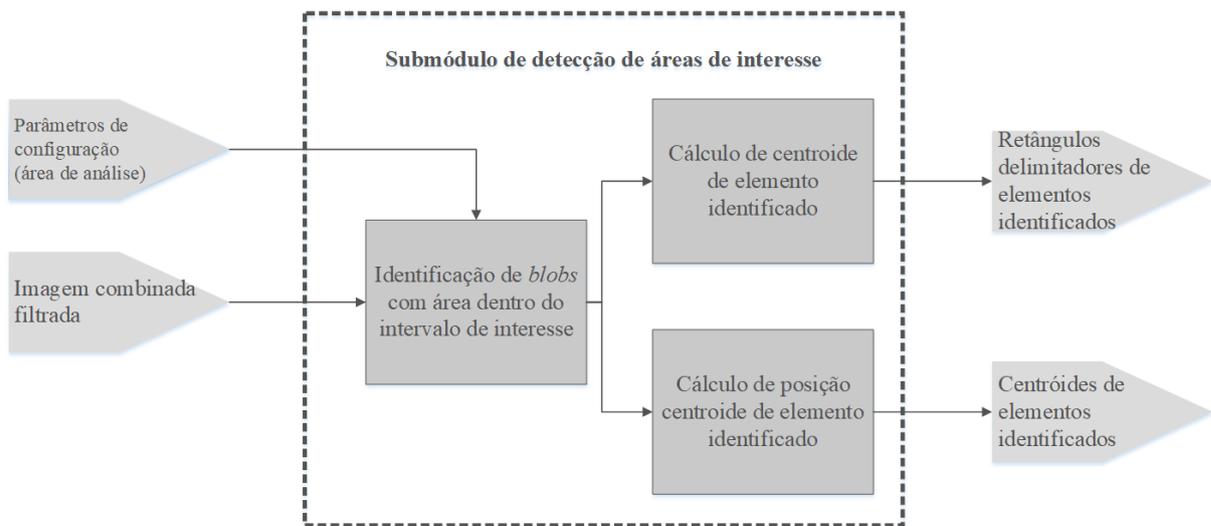


Figura 4-13 – Fluxo de processamento do submódulo de detecção de regiões de interesse.

O módulo de rastreamento funciona através da combinação das atividades dos submódulos relacionados ao gerenciamento, correspondência e predição de trajetórias dos elementos de interesse. Ao contrário do módulo de detecção em que o processamento realizado pelos submódulos constituintes é sequencial, no módulo de rastreamento os submódulos de correspondência e de predição de movimento trabalham em função das demandas de processamento geradas pelo submódulo de gerenciamento de rastreamentos. As principais relações de fluxos de dados entre submódulos internos e outras estruturas externas ao *módulo de rastreamento* são mostradas na Figura 4-14.

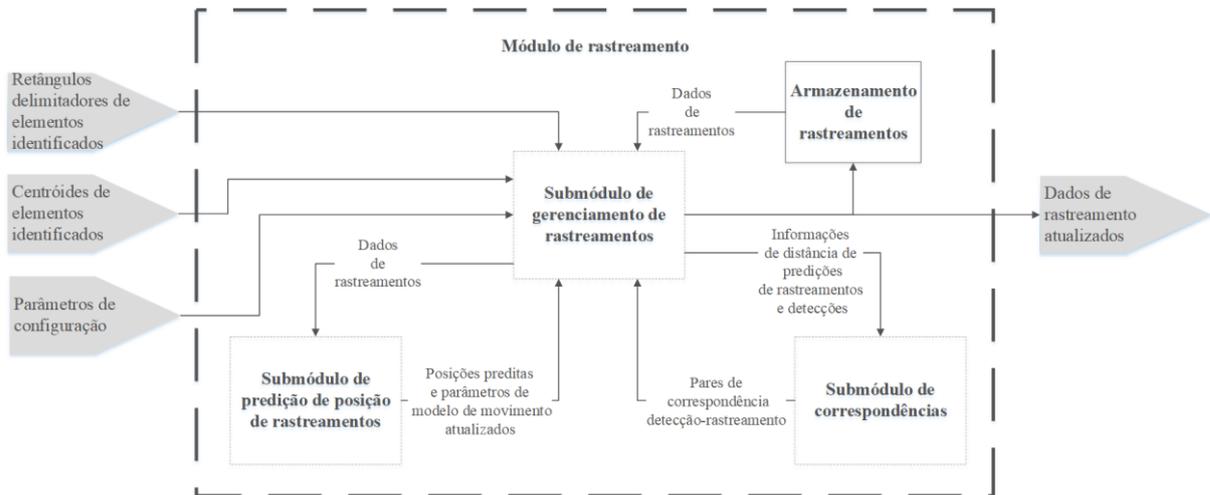


Figura 4-14 – Fluxo de dados entre submódulos do módulo de rastreamento.

As entradas do *módulo de rastreamento* são as informações de detecções – retângulos e centroides de detecções – e os parâmetros de configuração desse módulo que determinam, por exemplo, a distância máxima tolerada para correspondência entre uma detecção e um rastreamento registrado, o número máximo de quadros para o qual os rastreamentos podem ficar sem serem pareados com uma detecção e alguns parâmetros para construção de matrizes utilizadas pelo filtro de Kalman para modelagem do movimento dos elementos rastreados e que serão mais bem detalhados na subseção 5.6.3.2. Internamente ainda trabalha com os dados de rastreamentos registrados na estrutura denominada *Armazenamento de rastreamentos* (Figura 4-15). A saída desse módulo é composta das informações dos rastreamentos, isto é, os dados de rastreamentos atualizados, em especial, a posição prevista para os rastreamentos no instante de processamento corrente.

Conforme mencionado anteriormente, os fluxos de informações dentro do *módulo de rastreamento* não são unidirecionais, sendo continuamente processados e trocados entre o *submódulo de gerenciamento de rastreamentos* e os *submódulos de correspondência* e de *predição de posição*. Na Figura 4-14 esses dados são identificados como os dados de rastreamento, dados de detecções (centroides e retângulos com regiões de interesse), predição de posição de rastreamento, parâmetros de configurações (entradas do sistema), pares de correspondência, informações de distância de predições de rastreamentos e dados de rastreamentos (corrente e atualizados).

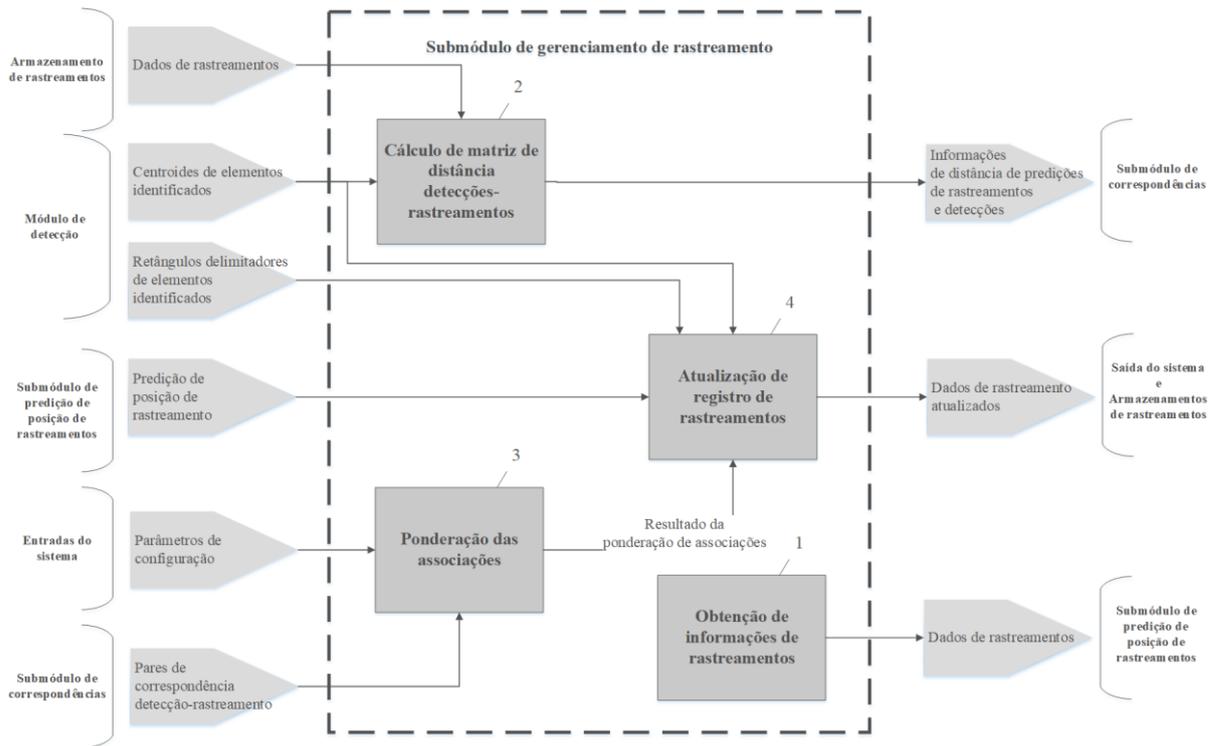


Figura 4-15 – Fluxo de processamento do submódulo de gerenciamento de rastreamentos.

O *submódulo de gerenciamento de rastreamentos* é responsável por controlar as atividades que permitem que o *módulo de rastreamento* estabelecer a correlação temporal dos elementos identificados em vídeo para diferentes instantes de tempo. Essas atividades envolvem a recuperação das informações de rastreamento registrados, pareamento de elementos rastreados e detectados pelo *submódulo de correspondência*, verificação de consistência dos pareamentos, registro e atualização de estados dos elementos rastreados e cálculo de posição prevista para o próximo quadro de análise pelo *submódulo de previsão de posição*.

O *submódulo de correspondência* tem como entradas os dados de distância entre detecções e previsões de posição dos rastreamentos para o instante de análise e para todas as combinações possíveis e saída a correspondência de detecções e rastreamentos que representam o menor custo global de correspondência. O único bloco de processamento desse submódulo é responsável por encontrar os pares detecção-rastreamento que representam a associação correta entre os indivíduos detectados em vídeo no instante de processamento considerado e os identificados até o momento de processamento considerado. A implementação de sistema apresentado neste trabalho conta com uso de um método de combinação ótima conhecido como “Algoritmo Húngaro”. A Figura 4-16 esquematiza o processamento do *submódulo de correspondência*.

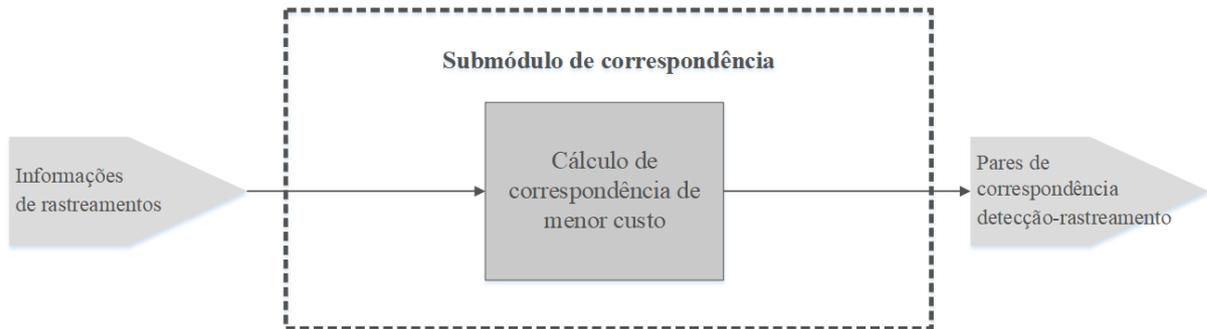


Figura 4-16 – Fluxo de processamento do submódulo de correspondência.

O *submódulo de predição de posição de rastreamentos* apresenta as rotinas de predição de movimento que ajudam a aumentar a confiabilidade do gerenciamento dos rastreamentos na medida em que a comparação de posição de um instante anterior em relação a um instante específico de análise não se baseia apenas no último registro de posição do elemento rastreado e sim em uma posição prevista para este dado elemento considerando seu histórico de movimentação – isto é, o modelo de movimento do rastreamento. Como pode ser observado na Figura 4-17, as entradas desse submódulo são informações para atualização de rastreamento: que, compreendem os atributos dos elementos rastreados para o instante de análise anterior, como parâmetros de modelo de movimento como as matrizes do filtro de Kalman e as posições desse rastreamento, a correspondência de detecção e rastreamento e outros dados de conclusão da etapa de ponderação de rastreamentos.

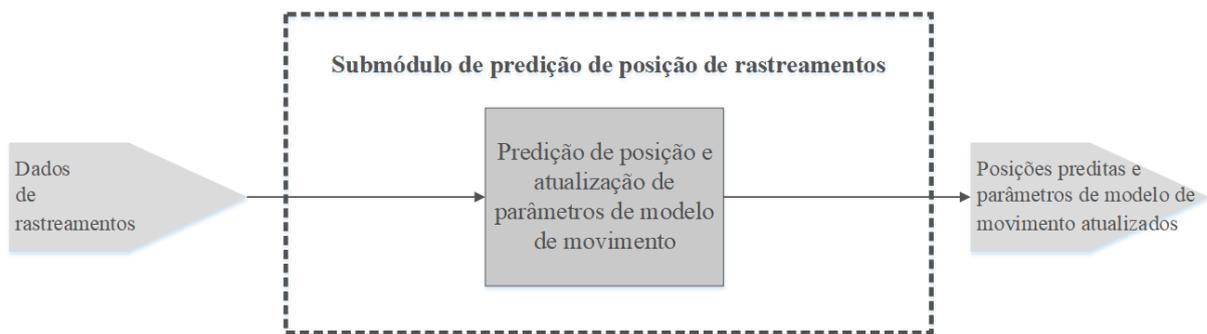


Figura 4-17 – Fluxo de processamento do submódulo de predição de posição de rastreamento.

Por fim, como *saídas do sistema* tem-se a representação binária da imagem combinada com os elementos de interesse englobados por retângulos delimitadores e os elementos de rastreamento sendo representados pelo ponto centroeide acompanhado por um índice identificador de cada elemento. O parâmetro de entrada relacionado à distância máxima entre uma detecção e rastreamento é ainda apresentado através do círculo centralizado na posição do

rastreamento. Outra característica das imagens de saída do sistema é a representação da região de análise de detecção e rastreamentos, representada pelo retângulo vermelho. A Figura 4-18 mostra as etapas principais de processamento que permitem a apresentação dos dados de saída em uma imagem.

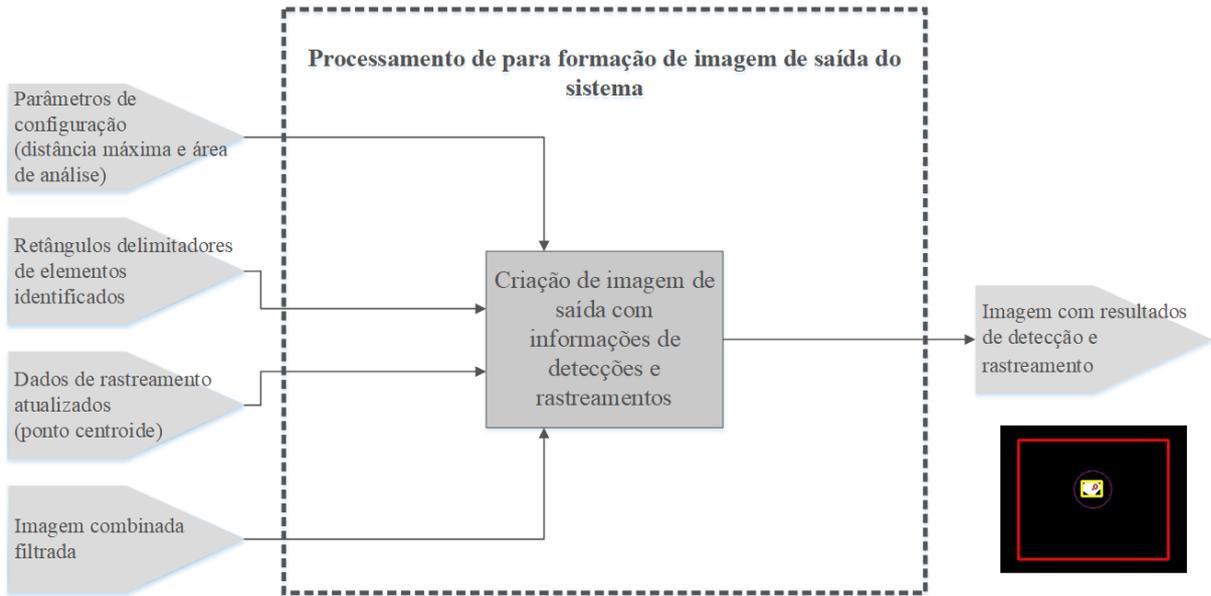


Figura 4-18 – Fluxo de processamento de construção de imagem de saída do sistema.

5. MATERIAIS E MÉTODOS

Nesta seção são apresentados e descritos os materiais e métodos utilizados no desenvolvimento do sistema proposto.

5.1. Hardware

O protótipo do sistema desenvolvido utilizou os hardwares com as seguintes características:

- Notebook Lenovo Ideapad 320
 - Intel® Core™ i7-7500U CPU @ 2.70GHz;
 - 16 GB de memória RAM;
 - Placa de vídeo NVIDIA GeForce 940MX 4GB.

5.2. Biblioteca OpenCV

OpenCV (*OpenSource Computer Vision Library*) é uma biblioteca de código aberto que conta com mais de 2500 algoritmos para desenvolvimento de aplicações em visão computacional. O projeto da biblioteca surgiu do intuito de promover uma infraestrutura comum para aplicações de visão computacional e alavancar o uso da percepção de máquina em produtos comerciais. Atualmente, a comunidade da biblioteca tem mais de quarenta e sete mil usuários e o número de *downloads* já excederam os sete milhões.

A biblioteca conta com interfaces de C++, C, Python, Java e MATLAB e suporta os sistemas operacionais Windows, Linux, Android e iOS. Além disso, OpenCV possui licença BSD (*Berkeley Software Distribution*) o que significa que ela pode ser livremente copiada, adaptada e distribuída.

A versão da biblioteca utilizada para o desenvolvimento deste trabalho foi a 3.2.0, lançada em 23 de dezembro de 2018.

5.3. Linguagem de Programação

A linguagem de programação escolhida para o desenvolvimento do projeto foi C++, dentre outras características, por ser uma linguagem multi-paradigma e muito difundida entre a comunidade usuária da biblioteca OpenCV.

5.4. Ambiente de desenvolvimento

Por questões de desempenho, tais como facilidade e rapidez na compilação e execução de códigos e na instalação das bibliotecas e módulos necessários, optou-se pelo uso do ambiente Ubuntu 16.04.4 LTS.

5.5. Base de vídeos

Para análise do protótipo de sistema desenvolvido foram utilizados vídeos do banco de dados de vídeos de pedestres do laboratório de visão computacional (*Computer Vision Laboratory, CVLAB*) da Escola Politécnica Federal da Lausana (*École Polytechnique Fédérale de Lausanne*). Este banco de dados possui sequências de vídeos de vários ambientes obtidas por três ou quatro câmeras em diferentes posições. Os conjuntos de vídeos selecionados para análise pertenciam a chamada “*Laboratory Sequency*”, cujas características do ambiente de gravação eram mais semelhantes a caracterização do ambiente de aplicação do sistema idealizado especificado na subseção 4.1. Outro fator determinante para a escolha dos vídeos dessas sequências foi a disponibilidade das informações de calibração de cada uma das câmeras com as homografias para transformações de perspectiva dos planos originais de cada imagem para o plano do chão em *top view*. A Tabela 5-1 apresenta as principais características desses vídeos (EPFL, 2018).

Tabela 5-1 – Características de vídeos “*Laboratory Sequency*”. (EPFL, 2018)

Parâmetro	Descrição
<i>Nomes dos arquivos de vídeo</i>	“4p-c0.avi”, “4p-c1.avi”, “4p-c2.avi” e “4p-c3.avi”
<i>Espaço</i>	O espaço utilizado foi o laboratório.
<i>Iluminação</i>	Ambiente com iluminação interna artificial e constante.
<i>Pedestres</i>	Até quatro pessoas se movimentam aleatoriamente no campo de visão da câmera.
<i>Vista da câmera</i>	Ortogonal ao plano do chão.
<i>Movimentação da câmera</i>	Câmera estática.
<i>Dispositivo de captura de vídeo</i>	Sem informações.
<i>Duração</i>	Dois minutos e meio.
<i>Taxa de quadros</i>	25 quadros por segundo
<i>Resolução</i>	360 × 288 pixels

5.6. Implementação do sistema proposto

Nesta subseção são apresentadas as principais técnicas e algoritmos utilizados na implementação de cada submódulo do protótipo de sistema desenvolvido.

5.6.1. Entrada

As entradas do sistema do sistema consistem das sequências de vídeos digitais armazenados em disco no formato AVI (*Audio Video Interleave*), arquivos em formato XML (*eXtensible Markup Language*) com as homografias para a transformação de perspectiva das imagens de seu plano original para o plano do chão, arquivos em formato TXT com os dados dos contornos das regiões de chão nas imagens de entrada em seus planos originais e os parâmetros de configuração do sistema a serem utilizados pelos submódulos.

As homografias de cada um dos vídeos de entrada foram fornecidas juntamente os vídeos disponibilizados em (EPFL, 2018). Os dados do contorno do chão foram obtidos a partir da projeção das imagens de entrada de seu plano original para o plano do chão e, em seguida, de volta para o plano original das imagens. No processo de transformação da imagem do plano original para o do chão a informação das regiões não pertencentes ao chão são perdidas de forma que ao reprojeter as imagens transformadas de volta para seu plano original apenas as regiões pertencentes ao chão apresentam valores de pixel não nulos para todos os canais. Considerando que a posição de cada pixel em uma imagem pode ser dada com base em coordenadas bidimensionais x e y , o algoritmo responsável por obter as informações de contorno do chão das perspectivas das imagens de entradas armazenava a posição y (linha da imagem) a partir da qual os valores de intensidade de pixel para cada canal da imagem em cor RGB eram todos diferentes de 0. Um vetor com o número de elementos igual ao número de colunas da imagem – ou pixels horizontais – foi utilizado para armazenar essa posição y . Em seguida, os dados do vetor foram armazenados em arquivo TXT para, conforme mencionado, serem utilizados como dados de na entrada do sistema.

O campo Algoritmo 5-1 apresenta o pseudocódigo para a obtenção dos contornos da região do chão das imagens e armazenamento em vetor.

Algoritmo 5-1 – Pseudocódigo de algoritmo para obtenção das informações de contorno do chão.

Variáveis: Matrizes **Imagem**, **ImagemChao**, **ImagemReprojetada**, **Homografia**; vetor de inteiros **contorno[]**; inteiro **N, M, i, j**

Procedimento Contorno_do_chao(**Imagem**, **Homografia**)

ImagemChao \leftarrow transformaPerspectivaChao (**Imagem**, **Homografia**) % Projeção da imagem do
% plano original para o plano do chão

ImagemReprojetada \leftarrow transformaPerspectivaOriginal(**ImagemChao**, **Homografia**) % Projeção da
% imagem do plano do chão de volta para o plano original

M \leftarrow número de linhas da matriz **Imagem**

N \leftarrow número de colunas da matriz **Imagem**

Cria vetor contorno[N] com número de elementos iguais ao número de colunas da Imagem original

para **i** = 0 até **N**, **faça**

para **j** = 0 até **M**, **faça**

se **ImagemReprojetada(j, i)** \neq 0 para todos os canais

contorno[i] \leftarrow **j**

Interrompa

fim se

j \leftarrow **j+1**

fim para

i \leftarrow **i+1**

fim para

retorna **contorno[]**

fim do procedimento

5.6.2. Módulo de detecção

5.6.2.1. Submódulo de subtração de fundo

Para o protótipo do sistema desenvolvido optou-se pela utilização da implementação do algoritmo de segmentação *background/foreground* baseado em mistura de gaussianas proposto por (ZIVKOVIC, 2004; ZIVKOVIC e HEIJDEN, 2006). Este algoritmo faz uso de um modelo de fundo para o qual considera-se que cada pixel pertencente ao fundo da cena pode ser modelado a partir de uma mistura de funções com distribuição gaussiana e com variação adaptativa do número de parâmetros e componentes da mistura ao longo do tempo.

A classe OpenCV responsável pela implementação desse algoritmo é a *BackgroundSubtractorMOG2*. Assim, o instanciamento de objetos dessa classe foi efetuado com uso da função *createBackgroundMOG2()* para cada uma das imagens de entrada. Para este instanciamento optou-se por utilizar os parâmetros padrão, de modo que os atributos da classe relacionados ao número de amostras para a formação do modelo de fundo (*history*), a distância limiar para que um determinado valor de pixel seja considerado como pertencente ao modelo de fundo (*varThreshold*) e a opção de detecção de sombras (*detectShadows*) receberam valores iguais a 500, 16 e 'true', respectivamente. Atualização do modelo de fundo e a criação das *máscaras de foreground* a partir das imagens de entrada são executados utilizando a função *apply()* a cada iteração do código principal. As entradas da função *apply()* são a matriz ou imagem de entrada (*inputImage*), a matriz para armazenamento da máscara de *foreground* (*fgmask*) e a taxa de aprendizado do plano de fundo. Visto que as condições de iluminação e de componentes do fundo da sequência de vídeo analisada não se alterava abruptamente ou de forma significativa ao longo do tempo, optou-se por não realizar a atualização do modelo de fundo, escolhendo uma taxa de aprendizado nula.

Apesar de não haver necessidade de criação de matriz com a imagem do modelo de fundo construído, pois a função *apply()* já utiliza de suas informações para realizar a subtração de fundo, esta imagem pode ser obtida para visualização de resultados de processamento a partir da função *getBackgroundImage()* que tem como parâmetro de entrada apenas uma matriz para armazenamento do modelo (*backgroundImage*).

5.6.2.2. *Submódulo de filtragem de máscara de foreground*

Conforme mencionado na subseção 4.2, a *máscara de foreground* geralmente é computada com determinado nível de ruído que pode ser minimizado através de operações de filtragem de imagem. As regiões de ruído correspondem a pixels ou pequenos aglomerados de pixels que foram computados com pertencentes ao *foreground* ou ao *background* de forma errada. Essas regiões podem ser eliminadas ou reduzidas através das seguintes operações morfológicas (BRADSKI et al, 2008):

- **Erosão:** é o processo resultante da convolução de uma imagem por uma matriz unitária onde o valor de cada pixel da imagem é substituído pelo valor mínimo da região retangular no seu entorno. O principal efeito dessa operação – implementada pela função OpenCV *erode()* – é a invasão de regiões escuras nas brancas. As entradas principais utilizadas para essa função foram a imagem de entrada que se desejava modificar a estrutura morfológica, uma matriz para armazenamento do resultado e um *kernel* 3×3 que determina área de cobertura de filtragem de cada pixel, englobando o pixel central e oito pixels mais próximos ao seu redor;
- **Dilatação:** é o processo resultante da convolução de uma imagem por uma matriz unitária onde o valor de cada pixel da imagem é substituído pelo valor máximo da região retangular no seu entorno. A invasão de regiões brancas nas escuras é o efeito dessa operação, implementada pela função OpenCV *dilate()* que, por sua vez, utiliza os mesmos parâmetros de entrada que a função *erode()*;

A operação de suavização é o processo resultante da aplicação de um filtro passa-baixa para reduzir a componentes de alta frequência da imagem. A função OpenCV utilizada responsável por realizar esta operação de filtragem foi a denominada *blur()*. De forma semelhante as funções *erode()* e *dilate()*, as entradas utilizadas para esta função foram uma imagem de entrada, uma matriz para armazenamento da saída e o *kernel* 3×3 para delimitar a vizinhança para a filtragem de cada pixel.

No código desenvolvido para o sistema proposto a filtragem da máscara de *foreground* foi executada encadeando das operações de suavização, dilatação, erosão e novamente dilatação, de forma que a saída de uma função era utilizada com entrada da seguinte.

O passo de geração da imagem binária contou com uso da função OpenCV *threshold()* na imagem final do processo de filtragem. O limiar de intensidade escolhido nesta etapa foi de

50, fazendo com que os pixels da imagem filtrada com valores inferiores a este nível fossem computados com intensidade 0 e os superiores, com intensidade 255 na imagem de saída.

5.6.2.3. Submódulo de intensificação de presença

Na solução proposta implementada pelo módulo de intensificação de presença, a partir de cada uma das *máscaras de foreground* filtradas uma nova imagem é construída executando os seguintes passos:

- **Geração de imagem de soma de pixels de foreground:** a imagem de soma de pixels de *foreground* trata-se de uma matriz com as dimensões da *máscara de foreground* filtrada cujos valores dos pixels correspondem a soma da quantidade de pixels de *foreground* acima do pixel de *foreground* correspondente na máscara. Uma vez que dependendo da altura da imagem do vídeo analisado os valores de um dado pixel podem exceder a representação máxima com 8 bits – ou seja, a intensidade 255 –, as imagens de soma de pixels são codificadas como matrizes de profundidade de 32 bits do tipo *float*.

A implementação da função para a criação da imagem de somatório de pixels não contou com uso de funções específicas OpenCV, apenas a manipulação de estruturas básicas como matrizes. O pseudocódigo do algoritmo para a construção da imagem de somatório de pixels é apresentado no campo Algoritmo 5-2.

Algoritmo 5-2 – Pseudocódigo de algoritmo para obtenção de imagem de somatório de pixels do *foreground*.

Variáveis: Matrizes **Imagem**, **ImageSomatorio**, **ImagemForeground**; inteiros **N**, **M**, **i**, **j**, **k**;

```

Procedimento geracaoImagemSomatorio(Imagem)
  M ← número de linhas da matriz Imagem
  N ← número de colunas da matriz Imagem
  para i = 0 até N, faça
    para j = 0 até M, faça
      se Imagem(j, i) ≠ 0
        para k = j até 0, faça
          ImageSomatorio(j, i) ← ImageSomatorio(j, i) + 1
          k ← k - 1
        fim para
      fim se
      j ← j + 1
    fim para
    i ← i + 1
  retorna ImageSomatorio
fim do procedimento

```

- ***Geração de imagem de soma de pixels normalizada:*** a partir da combinação de imagens de soma de pixels e de vetor com regiões de interesse são representações das imagens de soma de pixel em que os valores dos pixels dentro de cada região de interesse são normalizados para ser representada em uma imagem de 8 bits. Esse procedimento não só soluciona o problema de diferentes representações da imagem de somatório de pixel devido a diferenças de escala (ou distância) das pessoas em vídeo, como também elimina regiões de *foreground* que não tangem o plano do chão nas imagens. Em geral, tais regiões podem aparecer já que muito frequentemente as regiões da imagem de *foreground* que representam uma pessoa ou outro elemento de interesse podem aparecer segmentadas por diversos fatores. Com isso, as regiões de interesse identificadas são armazenadas em um vetor de elementos do tipo de retângulo (`cv::Rect` no OpenCV) para serem utilizados na etapa de normalização de regiões de interesse. Uma das principais funções da biblioteca OpenCV utilizadas no processo de normalização foi a função `minMaxIdx()` que, na presente implementação, tinha como entrada as regiões da imagem que se desejava normalizar, e variáveis para armazenamento dos valores máximo e mínimo de intensidade de pixel para região de análise. Com isso, construindo uma escala de intensidades de pixels normalizada de acordo com os valores de máximo e mínimos obtidos na função anterior, a normalização da imagem pode ser efetuada, de fato, usando a função OpenCV `convertTo()`.

No campo Algoritmo 5-3 é mostrado o pseudocódigo para computação da imagem de somatório do procedimento de normalização de valores de intensidade da imagem de somatória de pixels de *foreground*.

Algoritmo 5-3 – Pseudocódigo de algoritmo de normalização de imagem de somatório de pixels de *foreground*.

Variáveis: Matrizes **ImageSomatorio**, **ImagemForeground**, **ImagemNormalizada**; vetor de inteiro **contorno[]**; vetor de retângulos **ROI[]**; double **max**, **min**, **escale**; inteiro **i**;

Procedimento normalizaçãoROI(**ImageSomatorio**, **ImagemForeground**, **contorno**)

ROI = detectaROI(**ImagemForeground**, **contorno**) % Retorna os retângulos delimitadores dos blobs
% para as regiões de interesse

para **i** = 0 até **ROI.tamanho**, **faça**

obterMinMax(**ImageSomatorio**(**ROI[i]**), **&min**, **&max**) % Retorna os valores máximo e
% mínimo de intensidade de pixel da imagem de somatório de pixels

escale ← $255 / (\mathbf{max} - \mathbf{min})$ % Escala normalizada com valores de intensidade mínimos e
% máximos

ImagemNormalizada (**ROI[i]**) ← **normaliza**(**ImageSomatorio**(**ROI[i]**), **escale**)

% Normaliza regiões da imagem de somatório de pixels em *foreground* de acordo
% com o valor de escala encontrada

fim de para

retorna **ImagemNormalizada**

fim do procedimento

5.6.2.4. *Submódulo de filtragem de imagem de intensificação de presença*

Para a filtragem dos pixels com valores de intensidade inferiores ao limiar mínimo definido no antes do início do processamento do sistema a função OpenCV *threshold()* é novamente empregada, tendo como entradas principais a imagem de intensificação de presença e o valor desse limiar (parâmetro de entrada do sistema). Diferentemente do processo de binarização do *submódulo de filtragem de máscara de foreground* para o qual os valores dos pixels eram polarizados para as intensidades mínimas e máxima da escala de 8 bits, a filtragem realizada pelo *submódulo de filtragem de intensificação de presença* apenas os pixels da imagem de entrada com valores de intensidade inferiores ao valor do limiar passam a ser representados pela intensidade mínima de pixel.

5.6.2.5. *Submódulo de transformação de perspectiva*

A mudança de perspectiva de perspectiva das imagens de intensificação de presença de seu plano original para o plano do chão é realizada através do uso da função OpenCV *warpPerspective()*. Os parâmetros de entrada usados por esta função são a imagem de intensificação de presença filtrada, a homografia que mapeia a informação do plano original das imagens para o plano do chão para uma perspectiva do tipo *top view* de referência e o tipo de interpolação para transformação a ser realizada que, no caso, é interpolação linear.

5.6.2.6. Submódulo de combinação de imagens

Conforme mencionado na seção 4.2, a síntese da imagem com informações de todas as perspectivas é obtida por meio da soma das intensidade de pixels de cada uma das projeções da imagem de intensificação de presença no plano do chão. Considerando que essas imagens são de 8 bits e que o somatório dos pixels de uma mesma posição pode superar o limite máximo de intensidade de representação para esta profundidade de cor, a matriz com o somatório de cada uma das entradas é 32 bits do tipo *float*. Ao fim do processo de somatório pixel a pixel, a imagem combinada é também normalizada para que possa ser representada na escala de cinza, visto que funções subsequentes exigem representação de dados em níveis de cinza para processamento utilizando funções da biblioteca OpenCV. Além disso, a representação da imagem combinada em 8 bits permite que na etapa de filtragem sempre seja usado o mesmo intervalo de valores para escolha de limiar. A descrição do algoritmo para a obtenção da imagem combinada das várias entradas projetadas na região do chão é apresentada no campo Algoritmo 5-4.

Algoritmo 5-4 – Pseudocódigo da função de combinação de imagens.

Variáveis: Matrizes **ImageCombinada32F**, **ImagemCombinada8bits**; vetor de matrizes **ImagensTransformadas**; double **max**, **min**, **escala**; inteiro **M**, **N**, **P**, **i**, **j**, **k**;

Procedimento combinaImagens(**ImagensTransformadas**[])
M ← número de linhas da matriz **Imagem**
N ← número de colunas da matriz **Imagem**
P ← número de elementos do vetor de **ImagensTransformadas**
para i = 0 até N, **faça**
 para j = 0 até M, **faça**
 ImageCombinada32F ← $\sum_{k=1}^P \text{ImagemCombinada8bits}[k]$
 j ← j+1
 fim para
 i ← i+1
 obterMinMax(**ImageCombinada32F** &**min**, &**max**) % Retorna os valores máximo e
 % mínimo de intensidade de pixel da imagem de combinação
 escala ← $255/(\text{max} - \text{min})$
 ImagemCombinada8bits ← **normaliza**(**ImageCombinada32F**, **escala**)% Normaliza
 % imagem de combinada de 32 bits do tipo float de acordo com escala calculada
 retorna ImagemCombinada8bits
fim do procedimento

5.6.2.7. Submódulo de filtragem de imagem combinada

No submódulo de detecção de áreas de interesse são detectadas as regiões que devem corresponder as áreas do chão ocupadas pelas pessoas em movimento nos vídeos. Uma vez que as regiões de maior nível de intensidade possuem maior probabilidade de representarem justamente as regiões de ocupadas pelas pessoas em vídeo, então para reduzir ainda mais as

chances de erros e paralaxe aplica-se um limiar de intensidade, o qual na implementação atual do protótipo do sistema pode ser ajustado durante a execução do programa através do recurso de barra de rolagem ou, preferencialmente, definido antes do início da execução. A filtragem neste submódulo é um processo de binarização com polarização dos valores de pixel de acordo com o limiar: pixels com intensidades superiores ao limiar são representados pelo valor máximo e os pixels de intensidade de cor inferior, com os valores mínimo na escala de cor de 8 bits. Aqui, a função responsável pela binarização também é a *threshold()*.

Após a binarização, a imagem resultante também é tratada com operações de suavização, erosão e dilatação encadeadas – respectivamente, funções OpenCV *blur()*, *erode()* e *dilate()* com kernel 3×3 . A semelhança do **submódulo de filtragem da máscara de foreground**, o emprego de operações morfológicas e de suavização tem por objetivo uniformizar as regiões de *foreground* e de fundo.

5.6.2.8. Submódulo de detecção de áreas de interesse

Durante o desenvolvimento de sistema notou-se que nas regiões mais próximas as bordas da imagem combinada havia maior suscetibilidade a erros de detecção devido a distorção de projeção das imagens ser maior nessas regiões. Com isso, considerou-se a redução da área de detecção de elementos para uma determinada porcentagem das dimensões de altura e largura das imagens de entrada originais.

Na presente implementação do sistema, a detecção dos *blobs* é executada dentro da área de busca da imagem binária, considerando apenas como regiões válidas aquelas cuja área está dentro dos limites de área máxima e mínima de *blob* definidos para o cenário de análise. Os retângulos delimitares e os centroides das detecções consideradas válidas são armazenados em vetores apropriados e suas informações utilizadas no módulo de rastreamento.

5.6.3. Módulo de rastreamento

5.6.3.1. Submódulo de gerenciamento de rastreamentos

No submódulo de gerenciamento são realizadas as operações de controle que determinam como os elementos rastreados são associados às detecções e atualizados. Os elementos de rastreamentos são definidos como estruturas de caracterizadas de modo a permitir

sua identificação e o correto acompanhamento de sua trajetória ao longo do tempo. Para tanto, à esta estrutura são associados atributos como posição do centroide, retângulo delimitador de área, número de quadros sem associação com detecção, estrutura de predição de posição de centroide com filtro de Kalman.

Com isso, a lógica de controle processamento implementada pelo submódulo de gerenciamento de rastreamento se processa nas seguintes etapas principais:

- **Associação de elementos detectados com elementos rastreados:** as pessoas identificadas em vídeo estão registradas, podendo ser identificadas ao longo do tempo através de *tags* de identificação de registro. Como estratégia de associação utiliza-se o chamado algoritmo húngaro que se trata de um método de correspondência ótima implementado pelo *submódulo de correspondência*;
- **Ponderação sobre associações:** fase na qual são verificados os resultados dos pareamentos realizados a fim de identificar casos em que não ocorreu pareamento de uma detecção com um rastreamento registrado ou de um rastreamento com uma detecção. Em linhas gerais, os procedimentos de tratamentos se resumem aos seguintes casos:
 - *Correspondência sem sobras:* nenhum tipo de tratamento é realizado;
 - *Correspondência com sobras elementos de rastreamentos:* situação relacionada a erros no módulo de detecção. Neste caso, a predição associada ao elemento rastreado não pareado é atualizada sem a medida de correção, apenas a partir da matriz de estados que armazena as informações de movimento do rastreamento;
 - *Correspondências com sobras de detecções:* para o caso em que alguma detecção não foi pareada com uns elementos rastreado já existente então cria-se um elemento de rastreamento com informações associadas a este rastreamento;
- **Ponderação sobre associações e rastreamentos:** nesta etapa é realizada a verificação da consistência dos pareamentos para as seguintes situações:
 - *Elementos de rastreamento não atualizados há mais tempo do que o limite de quadros permitidos:* quando os elementos rastreados não são associados a detecções após a quantidade de quadros definida como limite, então o rastreamento é considerado como inválido e é terminado, sendo associado a

eles uma *flag* com estado indicando que a este elemento de rastreamento não será mais utilizado para verificação de correspondência;

- **Associação de elementos com distância Euclidiana maior do que a do limiar permitido:** quando uma associação é realizada e a distância entre o centroide da detecção e do elemento de rastreamento associado é superior ao mínimo valor necessário, então a associação é descartada. Para esta situação considera-se o caso de correspondência com sobra de elementos de rastreamento e detecções sem associação;

- **Atualização dos rastreamentos:** fase em que são realizadas a atualização das informações de rastreamentos, seja para os elementos rastreados que foram associados a uma detecção válida, seja para aquelas que não foram associados a quaisquer detecções no instante de análise. Essas previsões são fornecidas pelo **submódulo de previsão de posição de rastreamentos** que utiliza o filtro de Kalman, implementado através do uso de recursos da classe OpenCV *KalmanFilter()*;

Na solução desenvolvida, a classe que controla o processo geral de rastreamento é denominada *CTracker*. Tal classe utiliza dados de entrada definidos antes do loop de execução para definir parâmetros do rastreamento bem como métodos de controle durante o processamento dos frames. A sua função principal, o *CTracker::Update()* que é executado a cada quadro executa toda a verificação em que é executada a correspondência das detecções e dos rastreamentos, atualização dos dados dos rastreamentos, bem como inserção e remoção de elementos de rastreamentos do vetor de rastreamento que armazena todas os rastreamentos.

Os parâmetros de entrada para a construção da classe *CTracker* consistem de um intervalo de tempo relacionado a transição entre dois estados consecutivos do filtro de Kalman (*dt*), a magnitude de ruído de aceleração também utilizado pelo filtro de Kalman (*accelNoiseMag*), a distância limiar entre os centroides de detecção e rastreamento para que estes sejam considerados como pertencentes ao mesmo indivíduo em cena (*dist_thresh*) e o número máximo de quadros para o qual um rastreamento registrado pode ficar ativo para comparação com outros elementos de detecção (*max_skipped_frames*). Já os parâmetros de entrada para a função de atualização dos rastreamentos são apenas os vetores com os pontos centroides e de regiões.

O pseudocódigo da classe que implementa o gerenciamento dos rastreamentos é detalhado no campo Algoritmo 5-5.

Algoritmo 5-5 – Pseudocódigo de algoritmo de gerenciamento de rastreamentos.

Variáveis: Matriz **correspondencias**, **custo**; vetor de posições **deteccões**, **regiões**, **rastreamentos**; inteiro **M**, **N**, **i**, **j**;

Procedimento atualizacaoRastreamentos(**deteccoes**[], **regioes**[])

M ← tamanho de vetor **deteccoes**

N ← tamanho de vetor **rastreamentos**

 se **N** == 0

para i = 0 até **M**, **faça**

Cria elementos de vetor rastreamentos[] com informações de **deteccoes**[**i**]

fim para

caso contrário

continua

fim se

Cálculo de distância entre elementos do vetor deteccoes[] e **rastreamentos**[], considerando apenas elementos de rastreamento válidos

Construção de matriz de custo

Estimativa de combinação ótima a partir de método de combinação ótima

 % Ponderação das associações

para cada associação, faça

 se há sobras de **deteccões**

Cria-se rastreamento associado a deteção em sobra

fim se

fim para

 % Ponderação de associações e rastreamentos

para cada associação

 se **distancia entre deteção-rastreamento** > **limiar máximo de distancia**

Descarta associação

fim se

 se **elemento rastreado passa intervalo de tempo sem se associar a uma deteção** > **limiar elemento rastreado fica inválido para associação**

fim se

fim para

N ← tamanho de vetor **rastreamentos** % Atualiza variável associada ao número de rastreamentos

 % Atualização dos rastreamentos

para i = 0 até **N**, **faça**

 se **rastreamento** foi associado a **deteção**

Atualiza rastreamento[**i**] com **informação de deteção associada**

caso contrário

Atualiza rastreamento[**i**] sem **informações de deteção**

fim se

fim para

fim do procedimento

Vale ressaltar que a caracterização completa dos elementos rastreados é feita utilizando dados do ponto centroide e da região em que possivelmente se situa o elemento de interesse, informações relacionadas ao filtro de Kalman para a criação das respectivas matrizes relacionadas, conforme será detalhado na subseção 5.6.3.2 e número de quadros em que este elemento não foi relacionado com uma deteção através da classe denominada *STrack*. A principal função relacionada com esta classe é denominada *STrack::updateState()*, a qual atualização individual do modelo de movimento de cada elemento rastreado. Seus principais detalhes também são apresentados na subseção (5.6.3.2).

5.6.3.2. Submódulo de predição de posição de rastreamentos

Conforme mencionado anteriormente, o submódulo de predição de posição de rastreamento é responsável por implementar a estrutura do filtro de Kalman para a predição do movimento dos elementos rastreados. De acordo com (WELCH e BISHOP, 2006), o filtro de Kalman estima processos utilizando um controle de *feedback*: o filtro estima o estado do processo em um determinado instante de tempo e obtém dessa estimativa o *feedback* na forma de medidas com ruído. As equações que definem o filtro são divididas em duas categorias: as equações de atualização de tempo e as equações de atuação de medidas. As equações do primeiro tipo projetam o estado atual para o seguinte no tempo e estimava do erro de covariância para obter as estimativas *a priori* para o próximo passo. Já as equações de atualização de medidas são responsáveis pelo *feedback* que corresponde à incorporação de uma nova medida em uma estimativa *a priori* para obter uma estimativa *a posteriori* melhorada. Com isso, as equações de atualização de tempo podem ser pensadas como equações de predição e as equações de atualização de medições, como equações de correção. Essas equações são mostradas na Tabela 5-2.

Tabela 5-2 – Equações do filtro de Kalman. Adaptado de (WELCH e BISHOP, 2006)

Equações de atualização de tempo (Predição)	Equações de atualização de medições (Correção)
(1) Projeção para o próximo estado: $\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$ (2) Projeção do erro de covariância para o próximo instante: $P_k^- = AP_{k-1}A^T + Q$	(3) Cálculo do ganho de Kalman: $K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$ (4) Atualização do estado com a medida de z_k : $\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$ (5) Atualização o erro de covariância: $P_k = (I - K_k H)P_k^-$

Nas equações da Tabela 5-2, \hat{x}_k^- é uma estimativa *a priori* da variável de estado x para o passo k dado que temos a estimativa de seu valor para instante $k - 1$, sendo u_{k-1} a entrada. A matriz A é uma matriz de transição de estados do sistema, enquanto que B é a matriz de controle. A variável de estado z representa a medida, sendo H uma matriz de medidas relacionada ao estado do sistema no instante z_k . As matrizes Q e R representam, respectivamente, os processos de covariância do ruído e a covariância do ruído das medidas, as quais se assumem constância

apesar de, na prática, mudarem ao longo do tempo. As estimativas dos erros de covariância *a priori* e *a posteriori* em um instante k são representadas por P_k^- e P_k . Já a matriz K é escolhida de modo a ser um ganho que minimiza o erro de covariância *a posteriori*.

A classe OpenCV responsável pela implementação do filtro de Kalman é chamada de *KalmanFilter*. No protótipo de sistema desenvolvido optou-se pela criação de uma classe para gerenciamento das funções relacionadas a classe *KalmanFilter* – no caso, as funções de criação da estrutura, de atualização e correção dos estados do objeto. A solução adotada foi baseada na solução apresentada por (GITHUB – *Multitarget-tracker*, 2018) que, dada as entradas de região de interesse, a velocidade (ou variação de tempo) e a dimensão de ruído de aceleração, cria-se uma matriz de estado para este elemento. Uma vez que se considerou o rastreamento de objetos extensos, a matriz de estados do sistema foi modelada com quatro variáveis de estado – coordenadas horizontal e vertical do centroide da região de interesse onde se localiza o rastreamento, velocidade vertical e horizontal do objeto rastreado – e também quatro variáveis de entrada – coordenadas horizontal e vertical do centroide na imagem e altura e largura da região de interesse do rastreamento.

Os principais parâmetros de entrada associados a função *TKalmanFilter* são o ponto associado ao centroide do indivíduo que se deseja criar um registro (*pt*) e os já citados valores de variação de tempo (*dt*) e de magnitude de ruído de aceleração (*accelNoiseMag*).

Visto que para cada dos elementos rastreados deve existir um modelo de movimento, a função *STrack::updateState()* era responsável por intermediar as informações relacionadas ao modelo de rastreamento de cada rastreamento e as funções da classe *TKalmanFilter* para predição e atualização dos estados do modelo. Os principais parâmetros de entradas considerados para esta função são o próprio objeto *Strack*, o ponto de detecção associado ao rastreio (se existente) e a região da imagem em que essa detecção está situada (também se existir). Já para a classe *TKalmanFilter* são definidas rotinas que determinam os procedimentos para obtenção das predições (*TKalmanFilter::GetPointPrediction()*), atualização dos elementos da matriz de estados a partir de predições ou medidas (*TKalmanFilter::Update()*) e que são utilizadas pela função *STrack::updateState()*.

A função *TKalmanFilter::GetPointPrediction()* utiliza objetos do tipo Kalman gerados com a classe de implementação do filtro de Kalman na biblioteca OpenCV para gerar a predição do próximo estado do objeto (Equações 1 e 2). Já a função de atualização do modelo de

movimento através do filtro de Kalman utiliza o objeto relacionado ao filtro de Kalman para realizar a correção ou atualização do estado do elemento (Equações 3, 4 e 5).

5.6.3.3. *Submódulo de correspondência*

O submódulo de correspondência efetua a associação entre as detecções e os elementos rastreados baseado na distância do ponto centroide de uma detecção e da posição prevista do centroide de um elemento rastreado para o instante de tempo do quadro analisado. Uma vez que se deseja associar univocamente detecções a rastreamentos e rastreamentos a detecções, o problema da correspondência se resume a encontrar a configuração de associações que gere o menor custo total. A fim de se evitar o uso de lógica com várias hipóteses para se obter a associação de detecções e elementos rastreados, optou-se por empregar um método ótimo para a resolução desse tipo de problema, conhecido como algoritmo húngaro. Segundo (GUIRADO e ROCHA, 2014), o Algoritmo Húngaro é um algoritmo de otimização combinacional desenvolvido por Harold W. Kuhn que atribuiu o nome de “*Método Húngaro*” ao algoritmo, pois o algoritmo foi baseado em grande parte nos trabalhos dos matemáticos húngaros Jenő Egerváry e Dénes König. Supondo uma matriz de custo $C_{N \times N}$ com todas as entradas não negativas e que possua N zeros de tal forma que dois deles não estejam na mesma linha ou coluna. Para que o método húngaro possa ser aplicado é necessário que a matriz de custo seja quadrada, que as entradas sejam inteiras e que o problema seja de minimização. A implementação do método húngaro é efetuada com auxílio do código disponibilizado em (GITHUB – *C++ Implementation of the hungarian algorithm*, 2018). A matriz de custo para o problema de correspondência de detecções e rastreamentos é a distância do ponto centroide de uma detecção até o ponto predito de um rastreamento para aquele instante. Para que esta correspondência funcione, considera-se que se o número de detecções é maior do que o de rastreamentos, então calcula-se o custo do rastreamento (linhas) a uma detecção (colunas). Caso contrário, a matriz de custo consiste do custo das detecções (linhas) a um dado rastreamento (colunas). Este procedimento garante que sempre haja uma solução ótima para a matriz de custo. Após definida a matriz de custo, cria-se um objeto *Hungarian* para encontrar a solução ótima da matriz de custo gerada. Na matriz com a solução ótima as posições não nulas indicam a combinação de detecção e rastreamento para a solução encontrada. Com isso, as informações das associações ótimas são usadas para os procedimentos de controle citados para o submódulo de gerenciamento de rastreamento.

O pseudocódigo relacionado ao processo de geração da matriz ótima para a obtenção das associações através do Algoritmo Húngaro é mostrada no campo Algoritmo 5-6.

Algoritmo 5-6 – Pseudocódigo do Algoritmo Húngaro.

Variáveis: Matriz **correspondencias**, **custo**; vetor de posições; inteiro **M, N, i, j, min, numTracos**;

Procedimento algoritmoHungaro(**custo**)

correspondencias \leftarrow **custo**

M \leftarrow número de linhas da matriz **custo**

N \leftarrow número de colunas da matriz **custo**

% Subtração de menor entrada de cada linha

para j = 0 até M, faça

min \leftarrow ∞

 % Verificação de menor entrada de cada coluna da matriz **correspondencia**

para i = 0 até N, faça

se **correspondência[j, i]** < **min**

min \leftarrow **correspondência[j, i]**

fim se

fim para

 % Subtração de menor entrada de uma determinada linha

para i = 0 até N, faça

correspondência[j, i] \leftarrow **correspondência[j, i]** - **min**

fim para

fim para

% Subtração de menor entrada de cada coluna

para i = 0 até N, faça

min \leftarrow ∞

 % Verificação de menor entrada de cada coluna da matriz **correspondencia**

para j = 0 até M, faça

se **correspondência[j, i]** < **min**

min \leftarrow **correspondência[j, i]**

fim se

fim para

 % Subtração de menor entrada de uma determinada coluna

para j = 0 até M, faça

correspondência[j, i] \leftarrow **correspondência[j, i]** - **min**

fim para

fim para

% Cálculo de número de traços / colunas e linhas com zeros

numTracos \leftarrow número total de linhas e colunas com zeros

se **numTracos** == ordem da matriz **custo**

retorna **correspondências**; % Alocação ótima de zeros é atingida, encerra-se procedimento

caso contrário

continua

fim se

% Subtração de menor entrada sem nulos em linhas/coluna

min \leftarrow ∞

para i = 0 até N

para j = 0 até M, faça

se **min** > **correspondência[j, i]** && **correspondência[j, i]** não pertence a uma linha ou coluna com zeros

min \leftarrow **correspondência[j, i]**

fim se

fim para

fim para

para i = 0 até N

para j = 0 até M, faça

se **correspondência[j, i]** \neq 0

correspondência[j, i] \leftarrow **correspondência[j, i]** - **min**

fim se

fim para

fim para

Retorna para passo de cálculo de número de traços / colunas e linhas com zeros

fim do procedimento

5.6.4. Saída

A saída de processamento do sistema é representada pela imagem combinada após a filtragem com os elementos detectados englobados por um retângulo amarelo e as informações de rastreamentos como centroide e *tag* identificadora do elemento rastreado correspondente também representados na mesma imagem. Um círculo no entorno do centroide de rastreamento também é apresentado na imagem de saída, representando a distância máxima com a qual pontos centroides de detecção e rastreamento podem ser correspondidos. Além disso, o retângulo vermelho na saída demarca a região de análise dos processos de detecção e de rastreamentos.

As funções OpenCV utilizadas para gerar as formas indicativas nas imagens foram a *circle()* e a *rectangle()*: ambas as funções dependem da definição da imagem de destino, no caso, a imagem de combinação filtrada, e pontos na imagem definido a posição dessas formas. Para a função *circle()*, além do ponto central, o valor de raio da circunferência também deve ser determinado. Já o posicionamento do retângulo depende dos pontos extremos das áreas de interesse.

5.7. Procedimentos de análise de resultados

Para estimar o desempenho do sistema proposto buscou-se analisar as características das saídas do sistema em termos de qualidade de processamento e de eficiência de processamento, as quais foram avaliadas qualitativamente e quantitativamente. Posto que os resultados de processamento do sistema proposto são dependentes da estratégia para a redução de erros de paralaxe e da quantidade de informação de diferentes regiões da cena na imagem combinada, nos testes de avaliação de desempenho do sistema propõe-se analisar os resultados obtidos pelo sistema variando-se o limiar mínimo para representação de pixels da imagem de intensificação de presença e o número de entradas de vídeo utilizadas. As análises propostas foram as seguintes:

- **Análise geral de aspectos de processamento:** na qual foram avaliadas as principais características dos resultados das etapas principais de processamento. O objetivo dessa análise foi ilustrar as características dos resultados das principais etapas de processamento propostas e que justificam os métodos de processamento utilizados a fim de se obter os resultados desejados;

- **Análise geral dos resultados de detecção:** em que foram avaliadas principais as características dos resultados principais de cada um dos testes e sua relação com os parâmetros de configuração de cada teste.
- **Análise da eficiência de detecção de rastreamento:** nesta análise foram utilizados principalmente parâmetros quantitativos para avaliação dos resultados principais dos testes, a fim de se obter medidas quantitativas do desempenho do sistema nas diferentes condições testadas.

5.8. Metodologia de avaliação de desempenho

Na análise dos resultados do protótipo de sistema desenvolvido se buscou-se avaliar sua capacidade de processamento e de resolução de trajetórias para elementos com movimento aleatório.

A avaliação do desempenho do sistema se processou a partir de avaliação subjetiva e objetiva dos resultados obtidos.

Na avaliação subjetiva foram destacadas observações e conclusões acerca das características dos resultados das detecções e rastreamentos que puderam ser observadas visualmente para as sequências de vídeos analisadas e que não envolvem estimativas de parâmetros quantitativos. A partir dessa análise puderam ser levantadas, por exemplo, características das etapas de processamento e das próprias sequências de vídeo analisadas que implicaram em determinado tipo de resultado, bem como inferir possíveis soluções para melhoria do desempenho do sistema projetado. A avaliação subjetiva foi empregada em todas as etapas de análise propostas na subseção 5.7

A avaliação objetiva levou em consideração parâmetros quantitativos que caracterizam desempenho do sistema, no que se refere velocidade de processamento e de eficiência na detecção e rastreamento dos elementos de interesse nos vídeos analisados. Os parâmetros quantitativos considerados na avaliação objetiva foram:

- **Taxa média de quadros, t_Q :** determina a velocidade média de processamento dos sistemas em termos de quadros por segundo;
- **Taxa de detecção, t_D :** quantifica a eficiência do método de detecção em termos de capacidade de identificação do elemento de interesse em vídeo. Dado um número de pessoas em vídeo dentro da área de análise de detecção (P) e o número de detecções

corretas (DC), a taxa de detecção percentual pode ser definida por:

$$t_{D\%} = \left(\frac{DC}{P} \right) \times 100\% \quad (1)$$

- **Taxa de acerto de detecção, t_{AD} :** quantifica a eficiência do método de detecção. Dado um de detecções corretas ou verdadeiros positivos (VP) e de detecções incorretas ou falsos positivos (FP), a taxa de acerto percentual, $t_{A\%}$, pode ser estimada a partir da seguinte expressão:

$$t_{AD\%} = \left(\frac{VP}{VP + FP} \right) \times 100\% \quad (2)$$

Os parâmetros quantitativos foram definidos na etapa de “Análise de eficiência de detecção e rastreamento”, proposta na seção 5.7.

6. RESULTADOS E ANÁLISE

Nesta seção são apresentados os resultados obtidos a partir da execução da implementação do sistema proposto. Para tanto, a análise dos resultados é realizada de acordo com os procedimentos de análise especificados na subseção 5.7 e com a metodologia de avaliação de desempenho proposta subseção 5.8.

Na Tabela 6-1 são apresentadas as configurações dos testes realizados para os quais os vídeos de entradas e os valores de configuração de parâmetros dos módulos de processamento não são distintos.

Tabela 6-1 – Configuração de entradas e parâmetros para testes.

Teste	Entradas e parâmetros	
	Vídeos de entrada	Limiar de intensidade para imagens de intensificação de presença
1	“4p-c0.avi”, “4p-c1.avi” e “4p-c2.avi”	10
2	“4p-c0.avi”, “4p-c1.avi” e “4p-c2.avi”	150
3	“4p-c0.avi”, “4p-c1.avi”, “4p-c2.avi” e “4p-c3.avi”	10
4	“4p-c0.avi”, “4p-c1.avi”, “4p-c2.avi” e “4p-c3.avi”	150

Na Tabela 6-2 são apresentados os valores dos parâmetros de configuração de entrada para comuns para todos os testes. O valor de limiar para a geração da imagem binária a partir da imagem combinada escolhido foi de 150, visto que se considerou que para cada ponto da imagem haveria pelo menos a intersecção da cobertura dos campos de visão de duas perspectivas. O delta de tempo é um parâmetro utilizado na matriz de transição do filtro de Kalman (matriz A) e que está relacionada com o intervalo de transição entre dois estados consecutivos do filtro de Kalman. A magnitude do ruído de aceleração trata-se de um parâmetro utilizado na definição da matriz de covariância de ruído (matriz Q), visto que o valor real da aceleração de movimento dos elementos rastreados é desconhecido. Tanto o valor do delta de tempo quanto o valor de magnitude de ruído de aceleração foram determinados experimentalmente, sendo escolhidos de modo que a predição da posição centroide dos

rastreamentos resultassem uma estimativa coerente. A distância limiar entre os centroides de detecção e de rastreamento também foram determinados experimentalmente considerando as características dos vídeos analisados e as características das saídas do módulo de detecção.

Tabela 6-2 – Configuração de teste comuns.

Parâmetro	Valor
<i>Limiar de intensidade para imagem combinada</i>	150
<i>Delta de tempo (filtro de Kalman)</i>	0.2
<i>Magnitude do ruído de aceleração (filtro de Kalman)</i>	0.1
<i>Distância limiar entre centroides de detecção e de rastreamento</i>	36 pixels
<i>Número limite de quadros sem atualização</i>	30 quadros

6.1. Análise geral de aspectos de processamento

As seqüências de Figuras 6-1 a 6-7 representam os resultados de processamento principais do Teste 1, Teste 2, Teste 3 e Teste 4. Tais resultados referem-se ao processamento do mesmo conjunto de quadros de entrada – no caso, os 210º quadros de processamento de cada vídeo de entrada. Nesse instante de processamento apenas um indivíduo está presente no ambiente analisado.

A Figura 6-1 mostra as imagens de entrada para os quatro testes. As imagens referentes a Câmera 0, 1 e 2 foram utilizadas como entradas do Teste 1 e Teste 2, já o Teste 3 e o Teste 4 utilizaram todas as quatro entradas de vídeo.

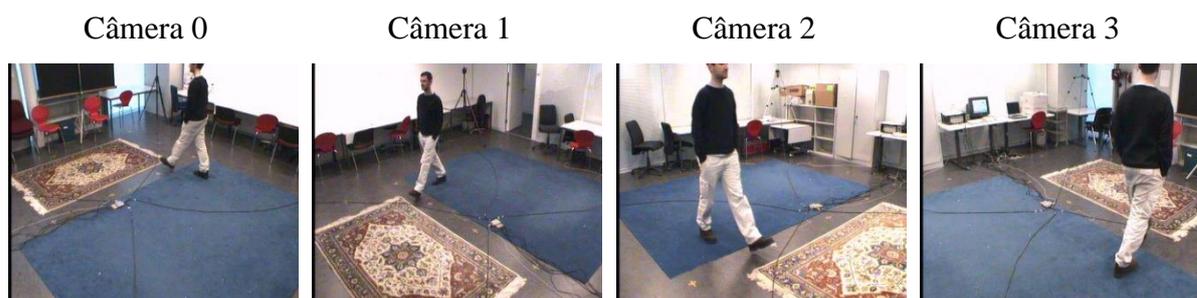


Figura 6-1 – Quadros de entrada – 210º iteração.

Os modelos de fundo e as máscaras de *foreground* de cada uma das perspectivas dos vídeos de entradas resultantes do processamento realizado pelo submódulo de subtração de fundo são mostrados na Figura 6-2. Da mesma maneira que na imagem anterior, as imagens das Câmeras 0 a 3 são representadas saídas para o Teste 1 e o Teste 2 e as imagens das Câmeras 0 a 4, para o Teste 3 e o Teste 4.

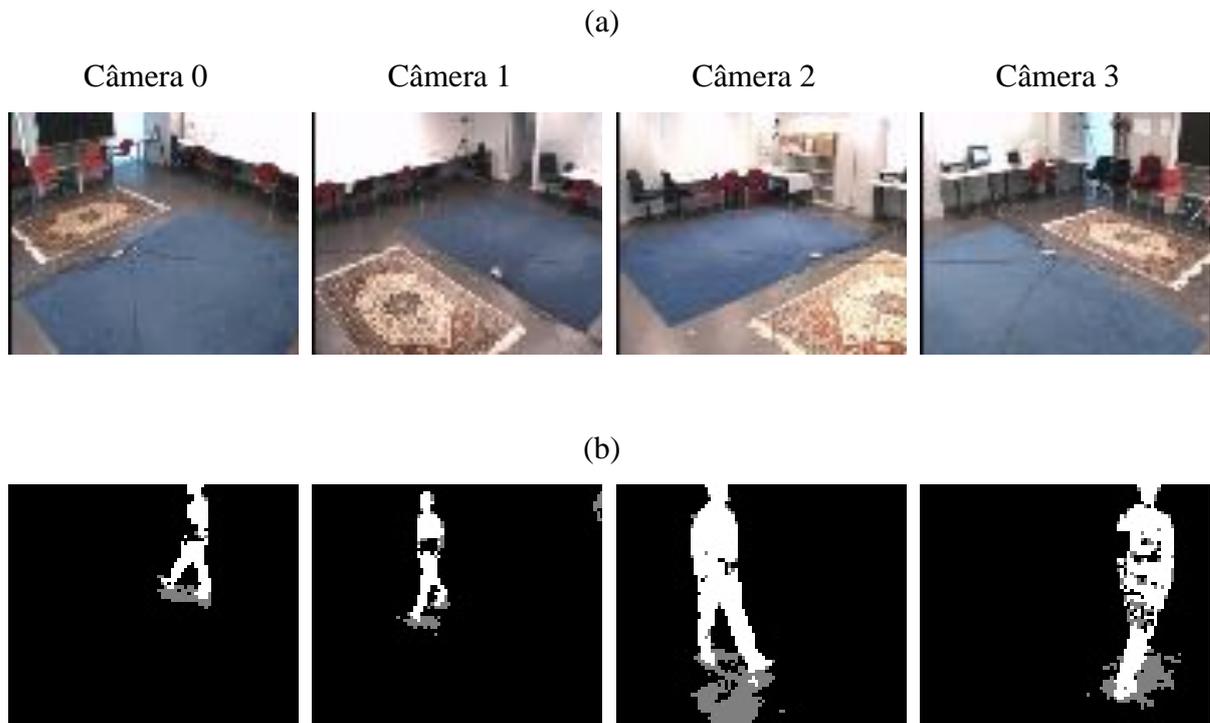


Figura 6-2 – Saídas do submódulo de subtração de fundo: (a) Modelos de fundo; (b) Máscara de *background*.

Verifica-se que o modelo de fundo consiste de uma imagem ligeiramente borrada da cena de fundo na ausência de pessoas para cada uma das perspectivas. As máscaras de *foreground*, apresentam aspecto ruidoso, pois pequenas regiões que deveriam ser computadas como de *foreground* são representadas como pertencentes ao fundo e vice-versa. Tais características justificam a necessidade do submódulo de filtragem de máscara de *foreground* a fim de tornar a regiões de *foreground* mais consistentes, conforme mostrado na Figura 6-3.

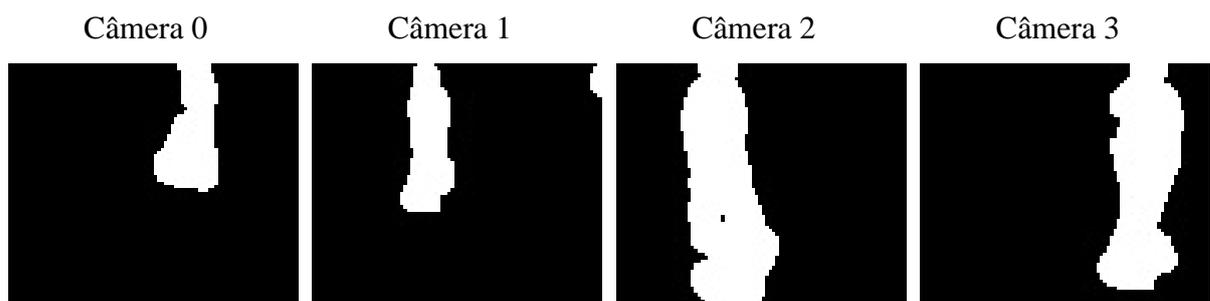


Figura 6-3 – Saídas de submódulo de filtragem de máscara de *foreground*.

Como pode ser constatado comparando as máscaras de *foreground* originais (Figura 6-2(b)) e suas versões filtradas (Figura 6-3), o submódulo de filtragem de máscara de *foreground* faz uso de técnicas de filtragem de imagens em escala de cinza que são eficazes na melhoria do processo de segmentação das regiões de fundo e *foreground*. Um aspecto interessante a ser ressaltado nas imagens é que na metodologia de segmentação de fundo empregada as regiões consideradas como de sombra são computadas em tom de cinza. Uma vez que tais regiões representam uma forma de projeção dos elementos não pertencentes ao fundo da cena diretamente no plano do chão, é de interesse que esses pixels sejam também computados como áreas de *foreground*, o que na prática é realizado.

As saídas de processamento para as quatro condições de testes começam a se diferenciar a partir do submódulo de filtragem de imagem de intensificação de presença. Sendo assim, as saídas relativas a este submódulo para cada teste são representadas separadamente na Figura 6-4.

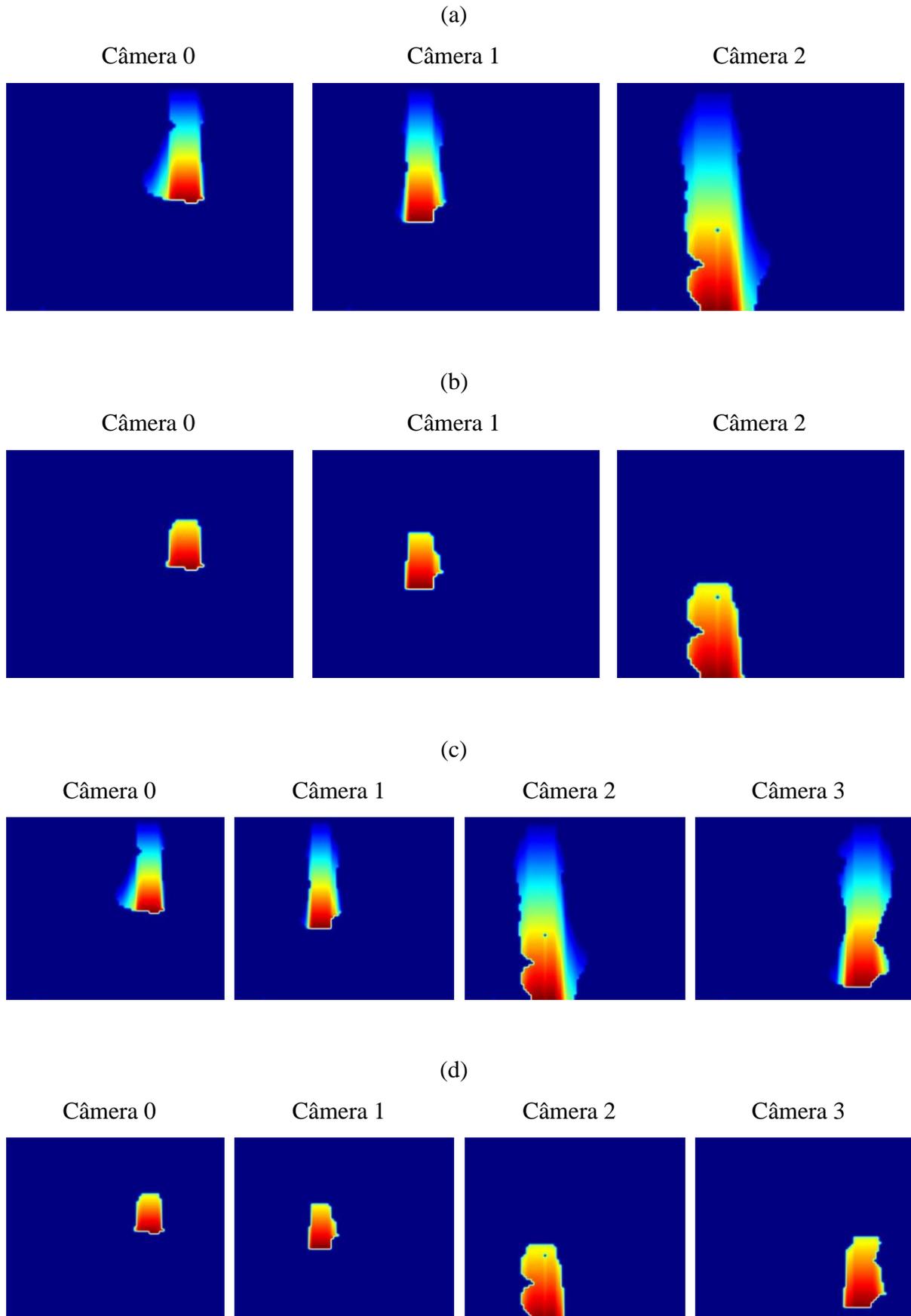


Figura 6-4 – Saídas do submódulo de intensificação de presença: (a) Teste 1; (b) Teste 2; (c) Teste 3; (d) Teste 4.

As imagens de intensificação de presença filtradas estão representadas em escala de cor vermelho-verde-azul, conforme pode ser observado na Figura 6-4. Na prática, as imagens processadas pelos submódulos são codificadas em tons de cinza com profundidade de cor de 8 bits. A representação na escala de cor vermelho-verde-azul foi usada a fim de tornar mais perceptível as diferenças de intensidade de pixel nas imagens na análise dos resultados. De acordo com esta escala, os pixels com maior intensidade de cor são representados em tonalidades avermelhadas – isto é, seus valores estão mais próximos da intensidade de cor 255 na escala de cinza. Os pixels com menor intensidade de cor – ou seja, com valores mais próximos a 0 em escala de cinza – são representados por tonalidades azuladas. Esclarecido este ponto, pode-se constatar que o processamento realizado pelo submódulo de intensificação de presença é capaz de gerar uma representação em que os pixels de *foreground* próximos a região do chão são apresentados com maior peso do que aqueles fora do plano do chão. Tipo de representação é importante para minimizar o erro de paralaxe, pois dessa maneira os pixels de *foreground* que representam a extensão do corpo de uma pessoa apesar de serem projetados com distorção quando aplicada a mudança de plano de projeção da imagem, contudo, por apresentarem menor intensidade há menor chance da combinação de projeções diferentes serem computadas como um máximo de interesse.

Outro recurso empregado no processamento para reduzir os possíveis erros de paralaxe na combinação das imagens projetadas no plano do chão é a utilização de um limiar mínimo para representação de intensidade de pixel nas imagens de intensificação de presença. Os resultados da combinação de imagens de diferentes perspectivas após a filtragem realizada pelo respectivo submódulo de filtragem são apresentados na Figura 6-5.

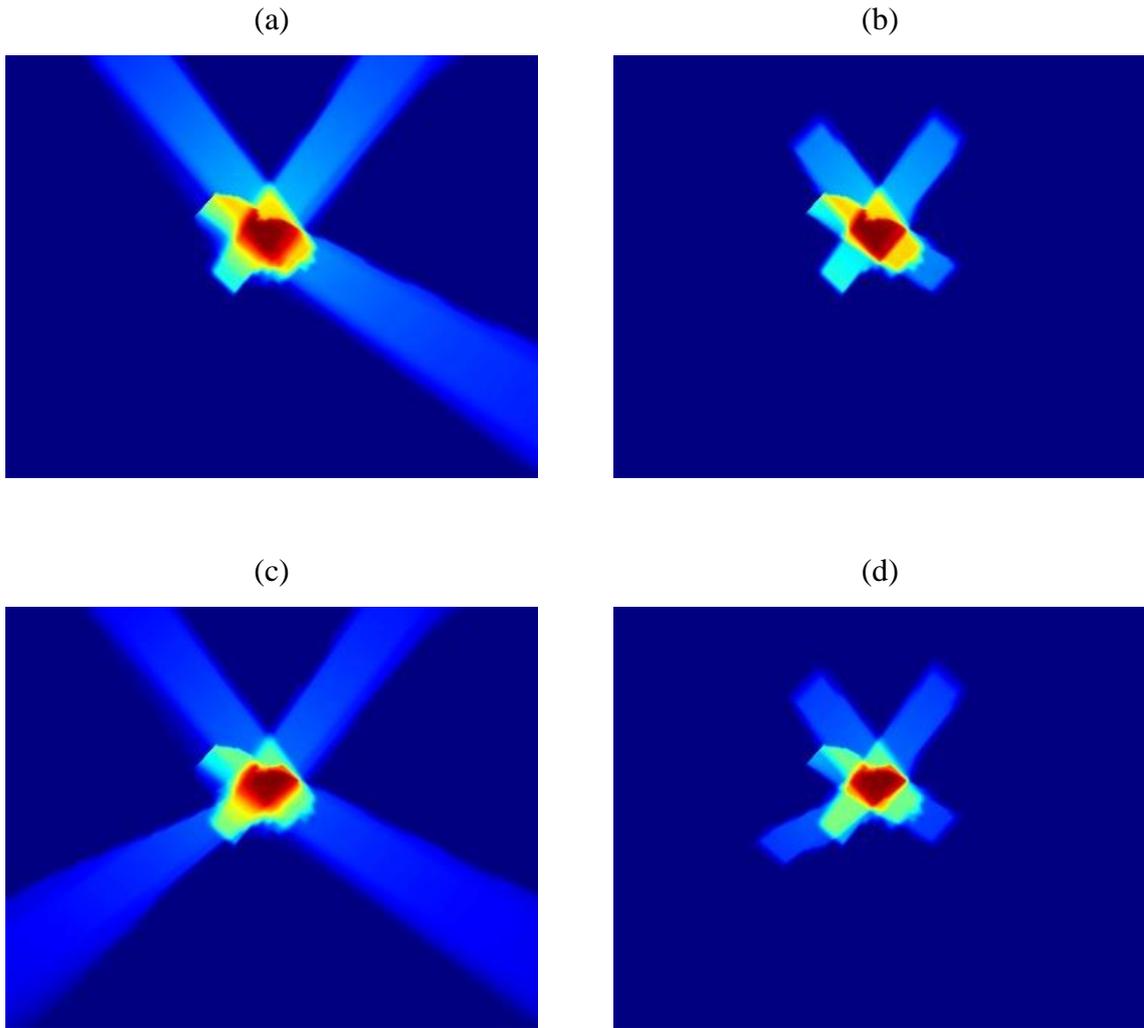


Figura 6-5 – Saídas do submódulo de combinação de imagens: (a) Teste 1; (b) Teste 2; (c) Teste 3; (d) Teste 4.

Das imagens na Figura 6-5 nota-se que nas imagens em que o valor de limiar foi mais baixo (Figura 6-5(a) e Figura 6-5(c)), os pixels de *foreground* representantes da extensão do corpo da pessoa em vídeo são mais extensas do que aquelas em que o limiar considerado foi alto (Figura 6-5(b) e Figura 6-5(d)). Apesar dessas características, por se tratar de um instante de análise em que existe apenas um indivíduo na cena analisada, durante a filtragem da imagem combinada, nas imagens filtradas resultantes essas regiões são removidas de modo que apenas as regiões de maior intensidade são representadas na Figura 6-5.

As imagens binárias obtidas na saída do submódulo de filtragem de imagem combinada para as condições de testes consideradas são apresentadas na Figura 6-6.

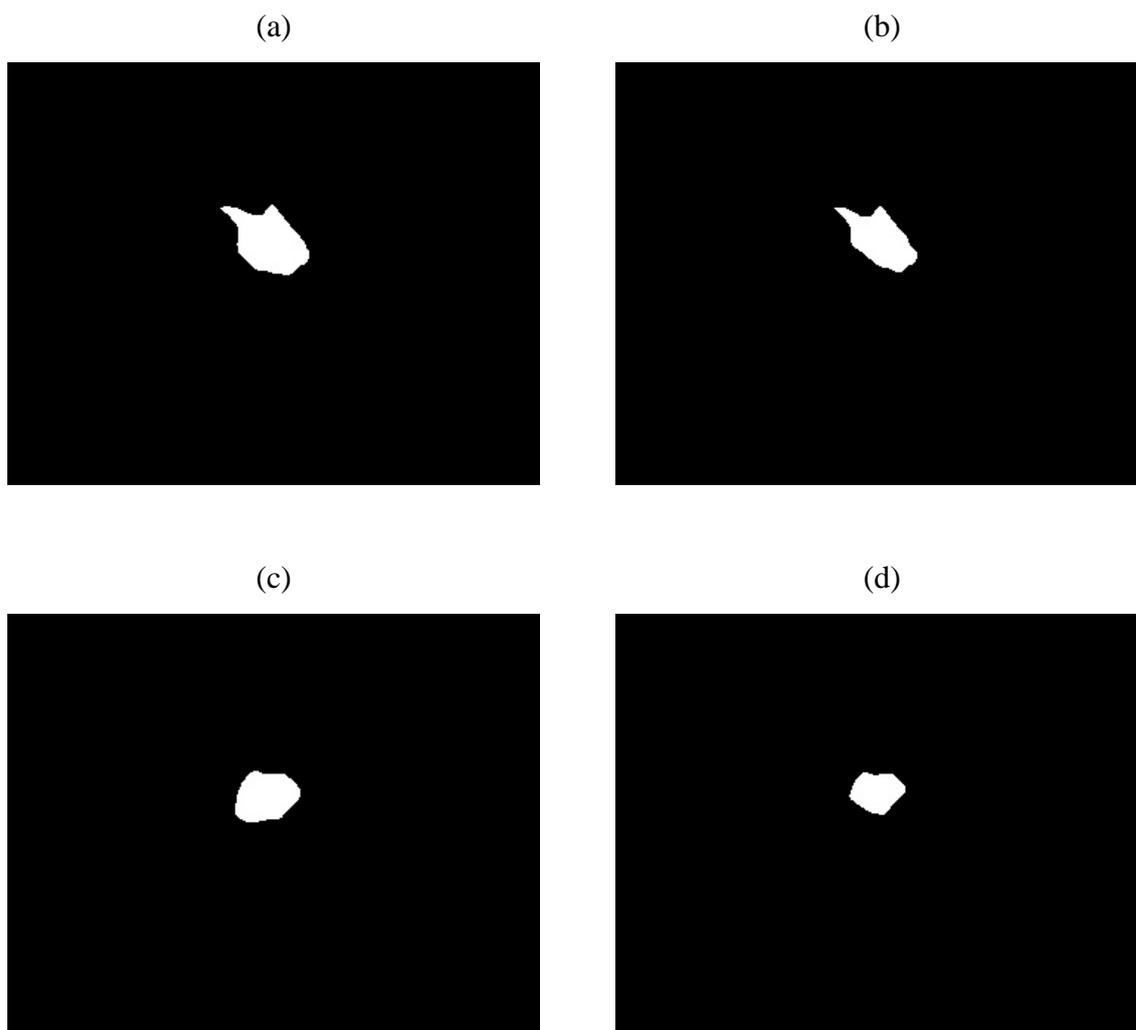


Figura 6-6 – Saídas do submódulo de filtragem de combinação de imagens: (a) Teste 1; (b) Teste 2; (c) Teste 3; (d) Teste 4.

Considerando as diferenças no número de entradas de vídeo para os testes é possível ressaltar apenas da análise isolada dos resultados para o instante considerado que nos resultados dos testes em que foram usadas entradas de quatro vídeos as formas dos *blobs* resultantes da filtragem da imagem combinada apresentam formas mais bem definidas, aproximadamente circulares, enquanto que nos testes em que foram empregadas apenas três das sequências de vídeos, as formas do *blob* representante do indivíduo em vídeo são mais irregulares. Além disso também é possível notar que os *blobs* dos testes filtrados com limiar de intensidade mais alto são ligeiramente menores do que aqueles filtrados com limiar mais baixo.

Na Figura 6-7 tem-se a representação das imagens de saídas dos testes.

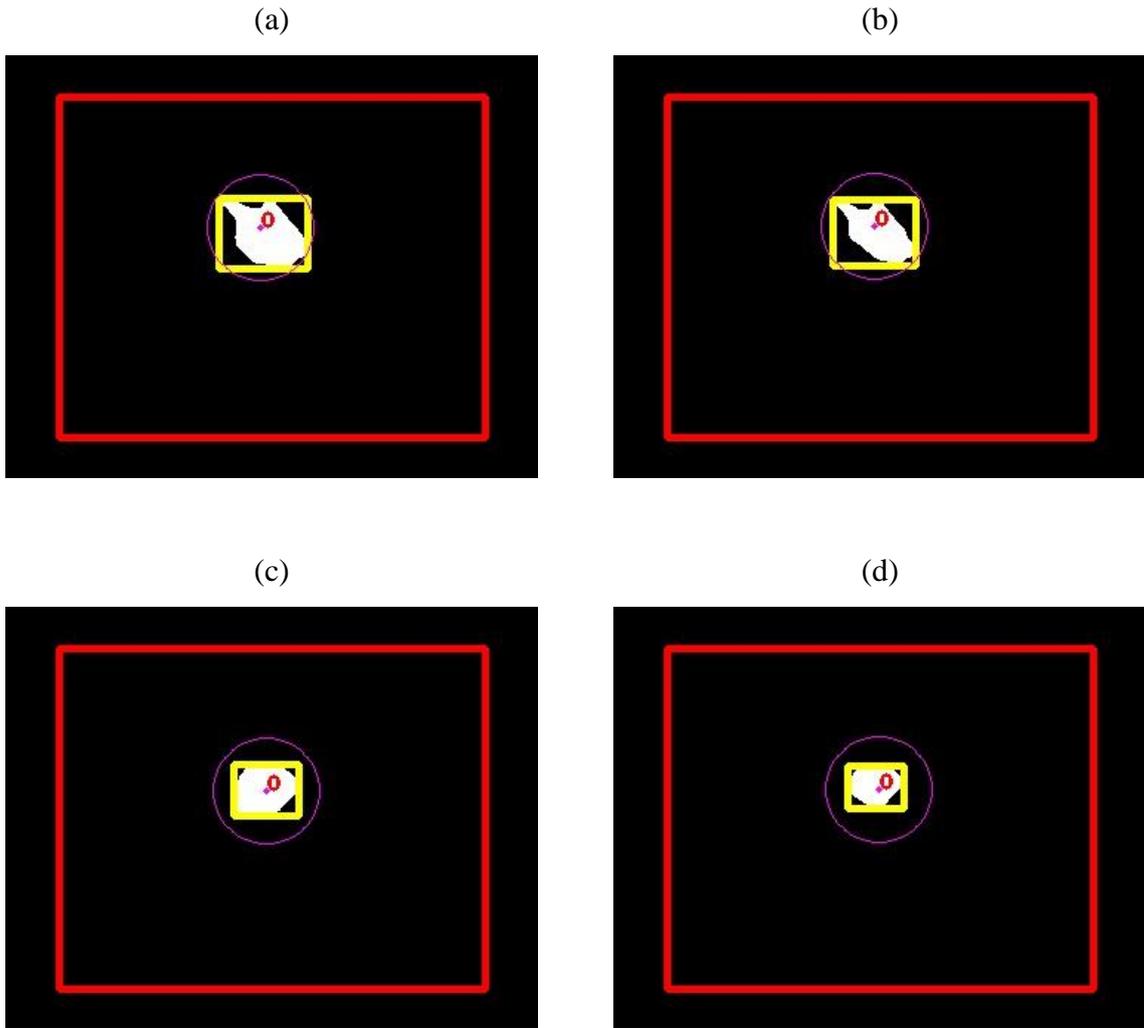


Figura 6-7 – Imagens de saídas do sistema: (a) Teste 1; (b) Teste 2; (c) Teste 3; (d) Teste 4.

Comparando os resultados exibidos na Figura 6-7 pode-se constatar que, apesar das ligeiras diferenças nas etapas de processamento dos testes realizados, os resultados de processamento para os quadros de entrada no instante de tempo analisado são equivalentes: em todos os testes foram detectados *blobs* cuja posição das imagens de fato equivale a da pessoa na cena. Ademais, uma vez que o índice 0 representa o primeiro elemento do vetor de rastreamento, pode-se dizer que o rastreamento desse indivíduo também se processou de maneira correta, pois o identificador do primeiro elemento rastreamento está relacionado com o primeiro indivíduo apresentado na sequência.

6.2. Análise geral de resultados de detecção e rastreamentos

Analisando-se os resultados de cada uma das condições de testes foi possível verificar o efeito da quantidade de vídeos de entrada utilizados para a composição da imagem combinada

e da diferença de limiar mínimo para filtragem das imagens de intensificação de presença, bem como limitações das metodologias de processamento adotadas.

Em geral, todos os testes apresentaram resultados corretos de detecção e rastreamento para os primeiros instantes de análise de vídeo, no qual apenas um indivíduo era detectado e rastreado – ou seja, a cada quadro de análise era atribuído a ele um identificador. Com o surgimento do segundo indivíduo no ambiente os resultados dos vídeos continuaram se processando de modo correto até determinada iteração. Em alguns instantes dos testes com três vídeos de entrada, por exemplo, a detecção de um dos indivíduos não ocorria. Contudo, o módulo de rastreamento era capaz de resolver os casos de correspondência quando a detecção voltava a ficar evidente. Nos Testes 3 e 4 os instantes iniciais a chegada do segundo indivíduo na cena não apresentaram falha na detecção do segundo indivíduo, visto que a adição da informação de mais uma perspectiva na imagem resultou em uma representação com maior peso da região.

Os pares de imagens de saída do sistema exibidos na Figura 6-8 representam os resultados de processamento dos quadros 525 e 530 dos vídeos de entrada. Essa sequência de processamento, no caso das imagens nos itens (b) e (d), exemplifica os instantes de perda de detecção para o qual a modelagem de movimento realizada do rastreamento é importante na predição da posição dos elementos de interesse, mesmo sem a informação de detecção. Outro aspecto relacionado os resultados da Figura 6-8 diz respeito a importância do limiar de filtragem de imagem de intensificação de presença, pois apenas nos casos em que este é maior houve perda de rastreamento.

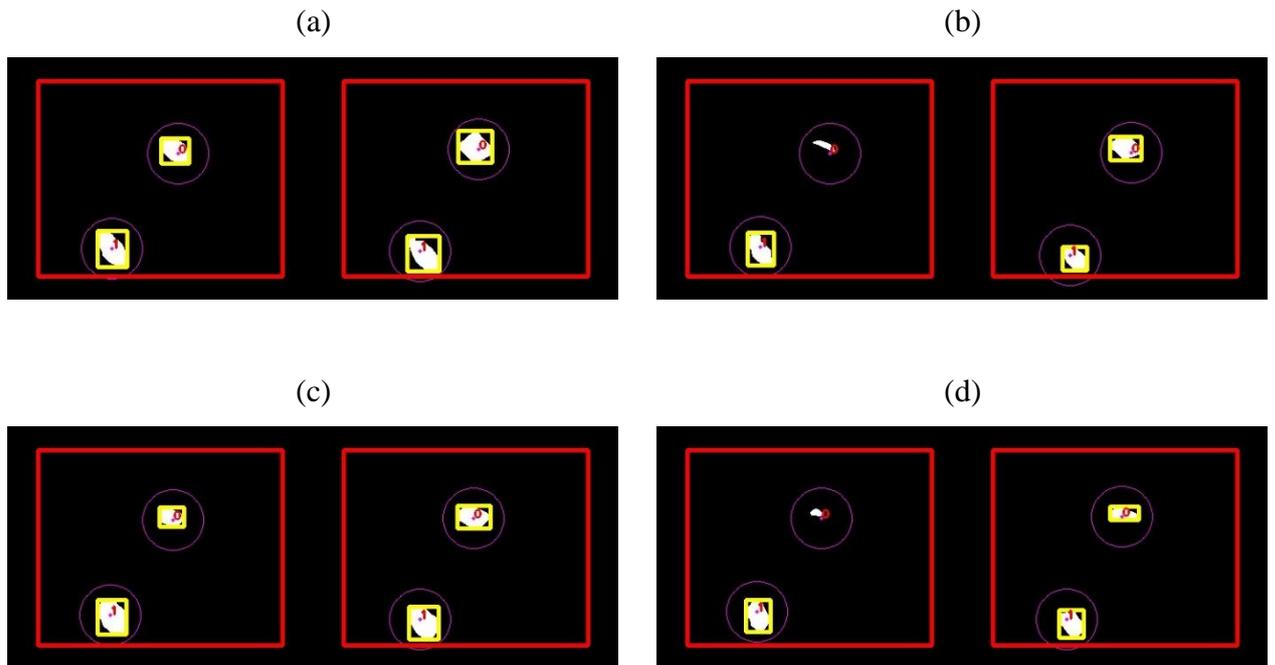


Figura 6-8 – Sequência de resultados de processamento dos 525º e 530º quadros de entrada das sequências: (a) Teste 1; (b) Teste 2; (c) Teste 3; (d) Teste 4.

A partir de aproximadamente da 630ª iteração os indivíduos em vídeo se aproximavam e, para este caso, módulo de detecção e, conseqüentemente o de rastreamento não são capazes de gerar resultados corretos para estes instantes de tempo, como pode ser observado nas imagens da iteração. Isto ocorre os *blobs* dos diferentes indivíduos se fundem em um único grande *blob* e no módulo de rastreamento apenas o centroide dessa detecção é pareado com o elemento de rastreamento para o qual a matriz de custo tem menor custo.

As imagens mostradas na Figura 6-9 mostram os resultados de processamento dos submódulos de filtragem de imagem de intensificação de presença e da saída do sistema para os 635º quadros dos vídeos. Constata-se que para este instante cada os testes alcançam resultados ligeiramente diferente. Primeiramente, o Teste 1 (Figura 4-9(a)) os *blobs* representantes dos indivíduos em cena não são devidamente segmentados pelo valor de limiar de intensidade utilizado, de modo que ficam fundidos em uma única grande região que é identificada com uma única detecção. No Teste 2, a limitação do valor de intensidade consegue segmentar as regiões dos dois indivíduos para este instante, contudo, como no instante anterior estas já haviam se fundido e o limite de distância de relação de rastreamentos é superior ao valor limite definido na entrada do sistema, a região da imagem de saída do sistema que de fato representa o segundo indivíduo em cena passa a ser identificada pela *tag* '2' e não pelo índice 1. Nos Testes 1 e 2 os rastreamentos do segundo elemento rastreado, identificado pelo índice '1', não se ajustam a perda de detecção e o modelo de movimento indica uma trajetória errônea

de movimento para a direção superior esquerda da imagem. Uma vez que esta não é mais atualizada com as informações da detecção do segundo indivíduo, os dados de posição desses rastreamentos continuam errados até suas finalizações. Já nos Teste 3 e 4 para o instante considerado os *blobs* dos indivíduos ainda não se fundiram devido à proximidade, sendo que o rastreamento permite identificar corretamente os indivíduos em vídeo – ou seja, o primeiro indivíduo com o índice ‘0’ e o segundo, com o índice ‘1’.

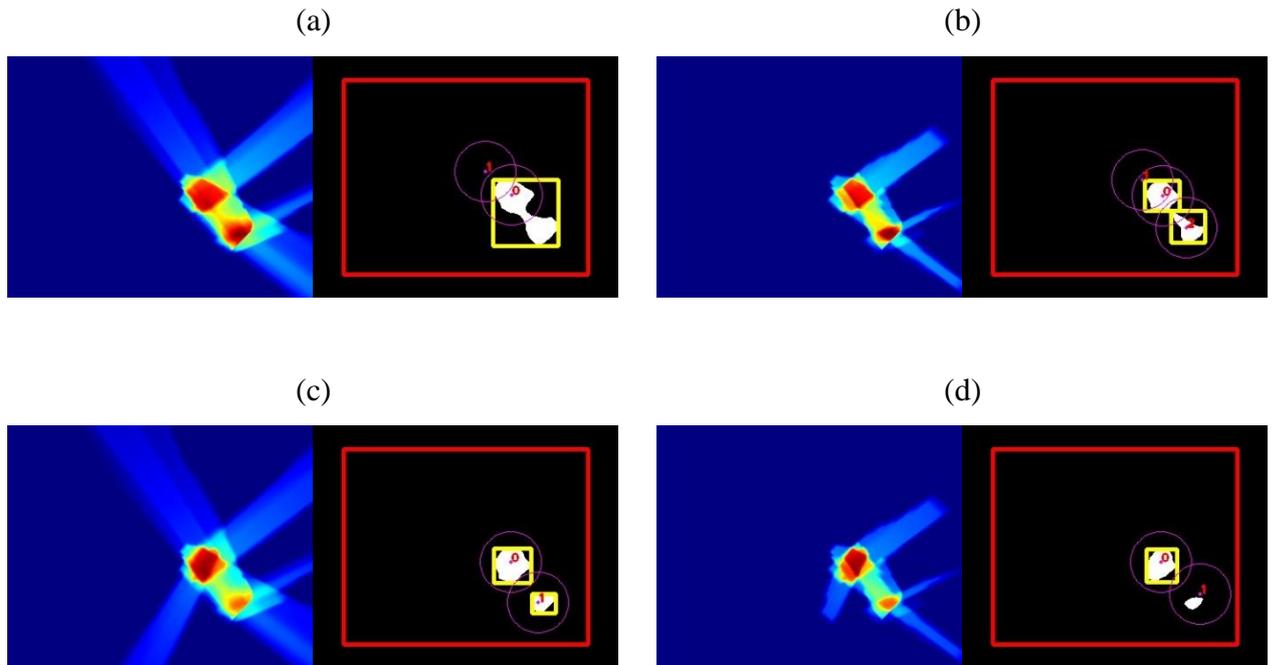


Figura 6-9 – Resultados de processamento dos 635º quadros de entradas dos vídeos: (a) Teste 1; (b) Teste 2; (c) Teste 3; (d) Teste 4.

Nos instantes subsequentes, quando os indivíduos voltam a se distanciar em vídeo um novo identificador é atribuído para o segundo elemento em movimento o índice ‘2’ para todos os testes exceto o Teste 2, para o qual identificador é trocado com o representante do segundo indivíduo, como pode ser observado nas imagens relativas a 710º iteração (Figura 6-10).

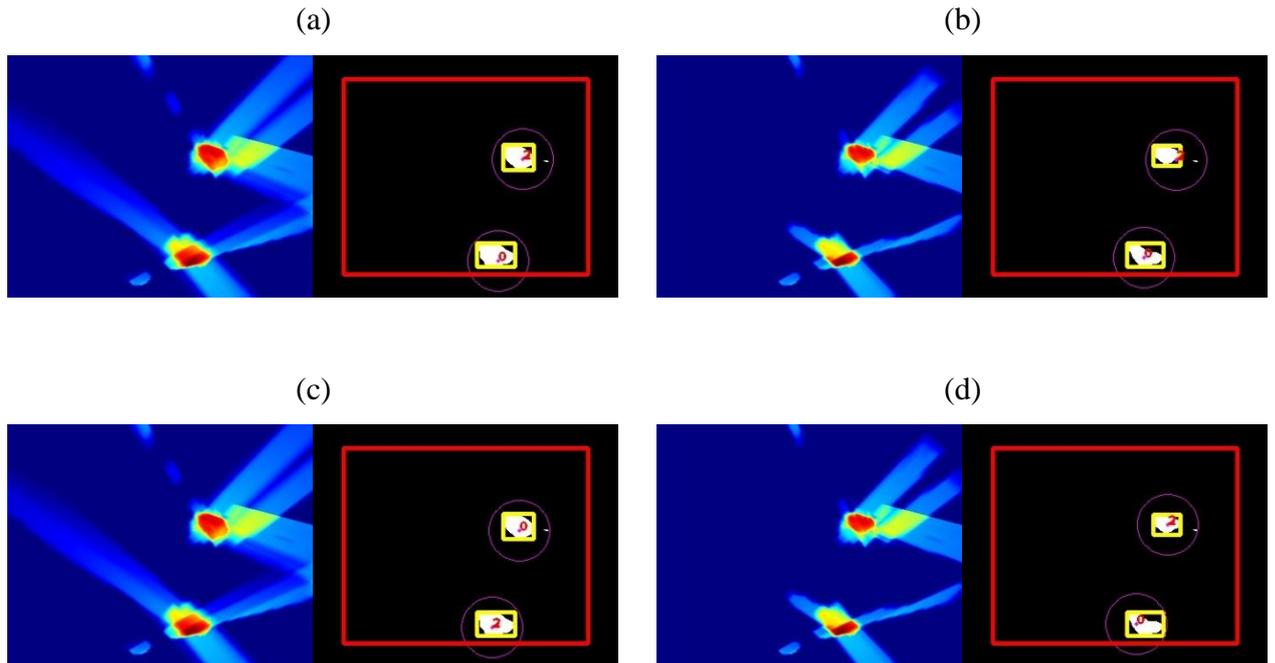


Figura 6-10 – Resultados de processamento dos 710^o quadros de entradas dos vídeos: (a) Teste 1; (b) Teste 2; (c) Teste 3; (d) Teste 4.

A entrada do terceiro indivíduo na cena analisada representa outra perturbação nos resultados. Como pode ser observado nas imagens correspondentes a 810^o iteração de processamento do sistema (Figura 6-11), a presença de vários indivíduos em um espaço reduzido aumenta a probabilidade de ocorrência de erros de paralaxe além da fundição de regiões no chão que representam espaço ocupado por cada indivíduo na cena devido à proximidade. Consequentemente, além das detecções se processarem de forma tumultuosa, novos registros de rastreamentos são criados para serem, em seguida, finalizados por ausência de correspondência com detecções reais consistentes. As imagens com os resultados da filtragem da imagem combinada permitem, por exemplo, constatar que a distorção devido a projeção no plano do chão dos testes com limiar alto de filtragem (Figura 6-11(b) e Figura 6-11(d)) ainda ocorre, embora a intensidade da combinação seja grande o suficiente para gerar erros de paralaxe. Além disso, a ausência de um mecanismo de ponderação das regiões da combinação das imagens para que as regiões com diferentes relações de intersecção de imagem tenham um mesmo peso, além das distorções nas extremidades, comprometem a detecção de elementos de interesse especialmente nas regiões próximas as bordas das imagens.

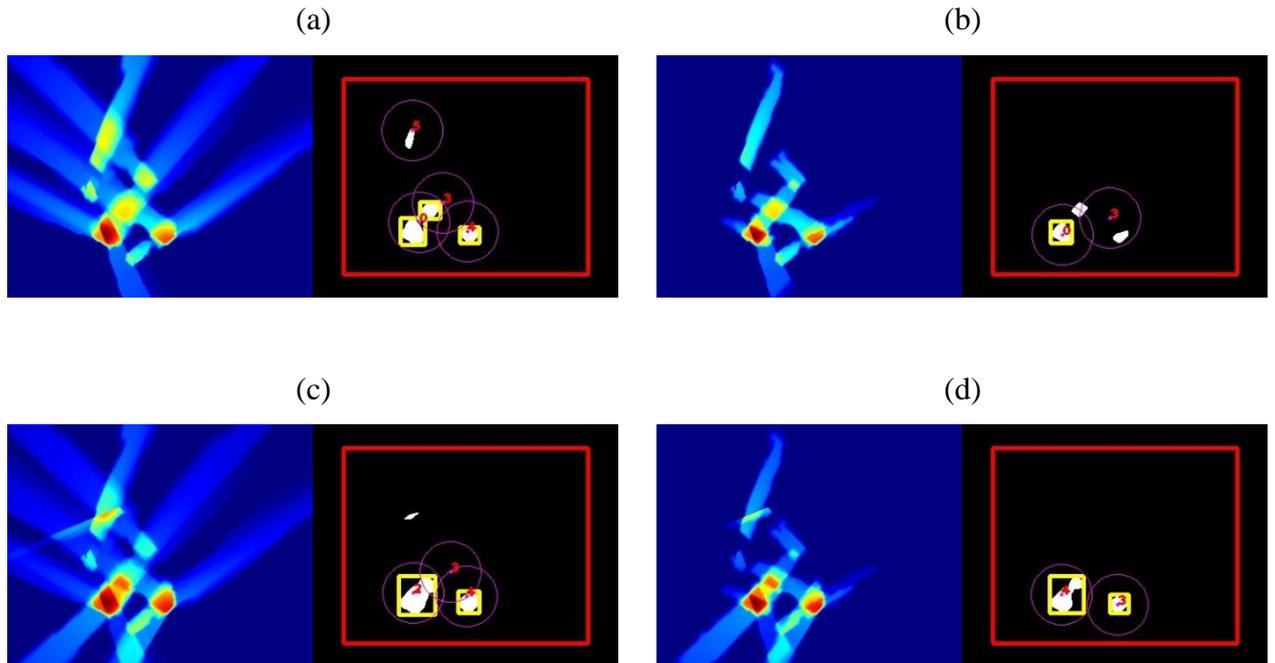


Figura 6-11 – Resultados de processamento dos 810^o quadros de entradas dos vídeos: (a) Teste 1; (b) Teste 2; (c) Teste 3; (d) Teste 4.

Vale ressaltar que, apesar do método de processamento não ter se mostrado eficiente para a situação em que três pessoas aparecem em cena ou quando duas pessoas andam relativamente próxima uma das outras, os resultados ao mínimo evidenciam que é possível utilizar a abordagem adotada para solucionar o problema de oclusão, desde que o indivíduo oculto em uma perspectiva esteja visível em outra.

6.3. Análise da eficiência de detecção e rastreamento

Conforme especificado na subseção 5.7, os parâmetros de eficiência de detecção considerados na avaliação do desempenho do sistema em diferentes cenários foram a taxa média de quadros por segundo, a taxa de detecção e a taxa de acerto de detecção. Na Tabela 6-3 são apresentados os valores médios obtidos para cada uma das condições de teste executadas.

Tabela 6-3 – Resultados de eficiência de detecção e rastreamento.

Teste	Parâmetros		
	Taxa média de quadros	Taxa de detecção	Taxa de acerto de detecção
1	21,39 ±1,09	84,29 %	95,54 %
2	20,39±0,81	74,10 %	94,42 %
3	19,88±0,36	81,82 %	95,41 %
4	19,69±0,34	77,29 %	98,84 %

Ao elaborar os cenários de testes, alterando os valores de limiar e número de vídeos de entrada esperava-se que observar tendências como melhoria na taxa de acertos de detecções com o acréscimo da informação de uma nova perspectiva com a diminuição dos possíveis erros de paralaxe. Os resultados não demonstraram essa premissa não é necessariamente válida: o aumento do limiar não é capaz de deduzir os erros de paralaxe, por exemplo, para os instantes em que três pessoas transladam pelo ambiente ou o uso de uma nova perspectiva.

Em se tratando da taxa de quadros processados por unidade de tempo, verifica-se que nos testes em que se utilizou todas as quatro entradas de vídeo apresenta uma menor taxa de processamento, sendo até 2 quadros a menos processados a cada segundo. No entanto, tais diferenças de valores são tão significativas, pois trata-se de uma diferença que pode ser considerada imperceptível a percepção humana.

Os valores de taxa de detecção mostram que, em geral, as taxas de detecção foram mais altas para os testes em que se considerou o limiar de nível de intensidade para a imagens de intensificação de presença maiores, como é o caso da configuração para o Teste 1 e o Teste 3. Apesar de provável, não se pode falar em tendência de aumento da taxa de acerto de detecção com o aumento desse limiar, pois a taxa de acerto de detecção do Teste 2 foi menor do que do 1.

Apesar de não ter sido definida uma metodologia quantitativa para avaliação da eficiência de rastreamento, o que pode ser dito é que os resultados de rastreamento são dependentes dos resultados de detecção. Com isso, sempre que uma detecção era realizada de

forma correta, o módulo detector foi capaz de realizar o correto gerenciamento e predição de posição do elemento rastreado. Ao final do intervalo de análise das sequências para todos os testes o número de rastreamentos registrados foi maior do que o número de pessoas em vídeo, por fatores como: falso positivo de detecção ou detecção de indivíduo não detectado por intervalo de tempo superior ao tempo limiar definido. Detecções do tipo falso positivo são difícil controle e, de acordo com a metodologia de processamento empregada, geralmente estão relacionadas as regiões em que houve erro de paralaxe. O registro de rastreamentos relacionados a elementos já associados a outros rastreamentos na presente implementação poderia ser contornado tornando maior o intervalo de tempo limite para que um rastreamento que não é relacionado com uma detecção.

7. CONCLUSÃO

A presente proposta de trabalho de graduação em Engenharia de Informação visou à investigação de sistemas de detecção e rastreamento com uso de múltiplas câmeras para aplicações em ambientes fechados. A partir do levantamento bibliográfico realizado na primeira etapa de desenvolvimento do projeto foi possível delinear as principais abordagens e características desejáveis para um sistema robusto de detecção e rastreamento de pessoas e compreender a fundamentação teórica das técnicas adotadas nas principais etapas de processamento do sistema. Assim, em uma segunda etapa, após a definição da arquitetura do sistema, buscou-se realizar a implementação do sistema de acordo com metodologia de detecção baseada na combinação de imagens em diferentes perspectivas da cena.

A arquitetura do sistema implementado pode ser separada em dois módulos principais: o módulo de detecção e o módulo de rastreamento. No módulo de detecção foram utilizadas várias técnicas de processamento de imagem a fim de formar uma representação da imagem que fosse equivalente a visão superior do ambiente analisado e na qual os indivíduos seriam representados pelas regiões ocupadas por eles neste plano. Já o módulo de rastreamento contou com uso de técnicas de predição de posição (filtro de Kalman) e de associação ótima (Algoritmo Húngaro) para prover o gerenciamento dos indivíduos detectados em vídeo ao longo do tempo. Para a verificação de desempenho sistema implementado considerou-se quatro configurações de entrada distintas para a solução elaborada que diferiam entre si pelo número de entradas de vídeo usada no processamento e pelo valor de limiar de filtragem das imagens de intensificação de presença. Para estas condições distintas notou-se diferenças em termos de qualidade e tempo de processamento: no geral, os testes com maior quantidade de entradas apresentavam taxa de processamento de quadros ligeiramente menor, e nos testes em que se considerou o limiar para filtragem de imagem de intensificação menor limitou em certa medida os erros de paralaxe. Quanto a alteração dos parâmetros de número de vídeos em perspectivas distintas ou valor de limiar para a imagem nos testes realizados não possível concluir sobre possíveis influencias

Apesar de todos os procedimentos implementados com o objetivo de aumentar a confiabilidade dos resultados de detecção e rastreamento, o sistema apresentou desempenho ruim situações em que as pessoas se apresentavam mais próximas umas das outras ou quando havia mais de duas pessoas em cena – situação que levava ao surgimento de falsos positivos oriundos de erros de paralaxe. Vale ressaltar, no entanto, que fora estes casos, a solução é ainda uma opção para lidar com problemas de oclusão momentânea de indivíduos em cena, visto que

a informação das perspectivas em que este não se encontra oculto determina a representação desse elemento na imagem combinada.

Dadas as principais características e limitações da solução de implementação apresentada neste relatório, como principais sugestões de melhoria do sistema a serem executadas futuramente pode-se citar:

- Busca de melhoria de representação de imagem de intensidade para resultar em resultados ainda menos propícios a erros de paralaxe;
- Adaptação do código a situações mais gerais sem a necessidade de ajuste de vários parâmetros e entrada, como ocorre na presente implementação;
- Utilização de modelo de movimento que permita lidar melhor com a aleatoriedade do movimento dos indivíduos em vídeo, tal como o filtro partículas;
- Introdução de etapa de classificação de elementos identificados em vídeo, como pessoas ou objetos, para aumentar a confiabilidade das detecções e as funcionalidades do sistema;

Por fim, convém ressaltar que este trabalho proporcionou enorme amadurecimento científico e acadêmico para a discente autora, consolidando conhecimentos adquiridos sobre tópicos específicos de visão computacional e processamento de imagem. Além disso, também proporcionou novas habilidades em programação utilizando a linguagem C++ e recursos da biblioteca OpenCV. Como trabalho para síntese de conhecimentos, pode-se dizer que este projeto atingiu este objetivo no sentido que permitiu aplicar a teoria de tópicos de processamento de imagem estudados na disciplina de Processamento de Vídeo e de filtragem estudados nas disciplinas de Transformadas em Sinais e Sistemas Lineares, Processamento Digital de Sinais e Filtragem Adaptativa do curso de Engenharia de Informação da UFABC, além dos já citados conhecimentos de programação visto em disciplinas como Algoritmos de Estrutura de Dados II.

REFERÊNCIAS BIBLIOGRÁFICAS

- BEYMER, David; KONOLIGE, Kurt. Real-time tracking of multiple people using continuous detection. Artificial Intelligence Center, SRI International, Menlo Park, 1999.
- BLACK, James; ELLIS, Tim J.; ROSIN, Paul. Multi-view image surveillance and tracking. In IEEE Workshop on Motion and Video Computing, 2002.
- BRADSKI, Gary R; KAEHLER, Adrian. Learning OpenCV: Computer Vision with the OpenCV Library. Sebastopol, USA: O'Reilly Media, Inc., 2008. 555 p. ISBN 9780596516130.
- CAMPLANI, Massimo; PAIEMENT, Adeline; MIRMEHDI, Majid; DAMEN, Dima; HANNUNA, Sion; BRUGHARDT, Tilo; TAO, Lili. Multiple human tracking in RGB-D data: A survey. IET Computer Vision, 2017, Vol. 11 Iss. 4, pp. 265-285.
- COLLINS, Robert T.; LIPTON, Alan J.; KANADE Takeo. A System for Video Surveillance and Monitoring, Proceedings of the American Nuclear Society (ANS) Eighth International Topical Meeting on Robotics and Remote Systems, April 1999.
- DARELL, T.; DEMIRDJIAN, D.; CHECKA, N.; FELZENSWALB. Plan-View Trajectory Estimation with Dense Stereo Background Models, Proceedings of the Eighth IEEE International Conference On Computer Vision, July 7–14, Vancouver, British Columbia, Canada, Volume 2, pp. 628–635, 2001.
- FLEURET, François; BERCLAZ, Jérôme; LENGAGNE, Richard; FUA, Pascal. Multi-Camera People Tracking With a Probabilistic Occupancy Map. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 2, pp. 267–282, February 2008.
- EPFL. Multi-camera pedestrians vídeo – "EPFL" data set: Multi-camera Pedestrian Videos. Disponível em: < <https://cvlab.epfl.ch/data/pom> >. Acesso em 12 de abril de 2018.
- GITHUB. C++ Implementation of the hungarian algorithm – RoboJackets. Disponível em: <<https://github.com/RoboJackets/hungarian>>. Acesso em 14 de abril de 2018.
- GITHUB. Multitarget-tracker – Smorodov. Disponível em: <<https://github.com/Smorodov/Multitarget-tracker> >. Acesso em 12 de maio de 2018.
- GUIRADO, J. C.; ROCHA, M. R. O método húngaro na resolução de problemas de otimização. XII EPREM – Encontro Paranaense de Educação Matemática. Campo Mourão, 2014. Disponível em:< http://sbemparana.com.br/arquivos/anais/epremxii/ARQUIVOS/MINICURSOS/autores/MCA_016.pdf>. Acesso em: 14/04/2018.
- IOCCHI, Luca; BOLLES, Robert C. Integrating plan-view tracking and color-based person models for multiple people tracking. In Proceedings of International Conference on Image Processing, ICIP 2005, Genova, Italy, September 2005.
- ISARD, Michael; Andrew BLAKE. Contour tracking by stochastic propagation of conditional density. In Proceedings of European Conference on Computer Vision 96, pages 343–356, 1996.
- JIANG, Zhengqiang; HUYRAN, Du Q.; MORAN, William; CHALLA, Subbash; SPADACCINI, Nick. Multiple Pedestrian Tracking Using Colour and Motion Models.

International Conference on Digital Image Computing: Techniques and Applications, Sidney, 2010, p. 328-334.

KELLY, Philip. Pedestrian Detection and Tracking using Stereo Vision Techniques. 2007. 244 f. Tese de Doutorado – School of Dublin City, University Electronic Engineering, Dublin, 2008.

OPENCV. OpenCV Library. Disponível em: < <https://opencv.org/>>. Acesso em 6 de maio de 2018.

OPENCV. OpenCV Modules. Disponível em: < <https://docs.opencv.org/3.2.0/index.html>>. Acesso em 6 de maio de 2018.

SANTOS, T.T., MORIMOTO, C.H.: Multiple camera people detection and tracking using support integration. PRL 32(1), 47–55, 2011.

XIA, Lu; CHEN, Chia-Chih; AGGARWAL, J. K. Human Detection Using Depth Information by Kinect. International Workshop on Human Activity Understanding from 3D Data in conjunction with CVPR, Colorado Springs, p.15-22, 2011.

WELCH, Greg; BISHOP, Gary. An Introduction to the Kalman filter. Technical Report TR95-041, University of North Carolina, Chapel Hill, NC, 2006.

TAJ, Murtaza; CAVALLARO, Andrea. Simultaneous Detection and Tracking with Multiple Cameras. Studies in Computational Intelligence. 411. 10.1007/978-3-642-28661-2_8. 2013.

ZIVKOVIC, Zoran. Improved adaptive Gaussian mixture model for background subtraction. In: International Conference on Pattern Recognition, Cambridge, UK, 2004.

ZIVKOVIC, Zoran; HEIJDEN, Ferdinand van der. Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recognition Letters, vol. 27, pp. 773–780, 2006.