

**Projeto SPDI – UFAM/FUCAPI/INdT**  
**Temas dos Protótipos**

1. Implementar o Algoritmo de Calibração de Zhang
2. Reconstrução Estéreo
3. Realidade Aumentada
4. Detecção de Objetos Abandonados
5. Detecção de Automóveis
6. Detecção de Olhos
7. Detecção de Placas
8. Reconhecimento de Faces
9. Detecção de Pedestres

## 1. Implementar o Algoritmo de Calibração de Zhang

Neste projeto, será necessário que o grupo leia o artigo de Zhang (A flexible new technique for camera calibration". IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11):1330-1334, 2000) e também implemente o algoritmo de Levenberg-Marquardt.

Os passos são os seguintes:

- 1 - Imprimir em uma folha de papel um padrão de calibração e fixá-lo na parede;
- 2 - Fotografar este padrão com uma câmera pessoal de vários ângulo distintos (~50);  
(OBS: Uma opção é fornecer as imagens)
- 3 - Detectar os cantos do padrão:
  - 3.1 - Aplicar um detector de bordas (p. ex., Canny) às imagens;
  - 3.2 - Encontrar melhores linhas que passam pelas bordas (Hough transform);
  - 3.3 - Definir os cantos do padrão como a interseção entre as retas encontradas;
- 4 - Associar números distintos a cada canto da imagem. Mostrar estes números na imagem. Mesmos cantos em imagens diferentes devem ter os mesmos números associados;
- 5 - Extrair os parâmetros intrínsecos e extrínsecos da câmera usando o Algoritmo de Zhang;
- 6 - Refinar os parâmetros encontrados usando o algoritmo de Levenberg-Marquardt;
- 7 - Como demonstrar a precisão dos parâmetros encontrados? Comparar os resultados obtidos antes e depois do passo 6.

## 2. Reconstrução Estéreo

Este projeto requer pesquisa adicional, uma vez que alguns conceitos abordados não foram cobertos em profundidade durante o curso. Segue os seguintes passos:

- 1 - Fotografar um objeto ou grupo de objetos simples (p. ex., caixas) em uma mesa a partir de dois pontos de vista com uma câmera. Sugere-se que o objeto esteja a uma distância de aproximadamente 2m e que a uma vista (posição da câmera) esteja entre 15 a 25cm da outra.
- 2 - Marcar manualmente um conjunto de pontos correspondentes (pelo menos 8) entre as duas imagens.
- 3 - Estimar a matriz fundamental da câmera usando o algoritmo de 8-pontos.
- 4 - Detectar bordas das imagens.
- 5 - Encontrar os pontos correspondentes nas duas imagens, assumindo restrição epipolar. Definir uma estratégia para lidar com casos de múltiplos candidatos.
- 6 - Construir uma configuração canônica para as câmeras e gere uma reconstrução da estrutura da cena (calcule as coordenadas 3D dos pontos correspondentes)
- 7 - Plotar um modelo 3D da cena;

### 3. Realidade Aumentada

A partir de um padrão conhecido desenhado em uma folha de papel, inserir modelos tridimensionais de sólido conhecidos (cubos, tetraedros) na perspectiva correta em cada frame de um vídeo. Passos:

(OBS: Deve ser usada a folha com o padrão e deve ser especificada a origem do sistema de coordenadas no mundo real. Deverão ser especificadas também a coordenadas onde os sólidos sintéticos serão colocados).

1 - Imprima um padrão conhecido (distâncias entre marcadores conhecidas) em uma folha de papel. Defina um sistema de coordenadas no mundo real em função dos marcadores impressos.

2 - Gere um vídeo do padrão impresso, no qual você movimenta a câmera ao redor do padrão, sem retirá-lo da cena.

3 - A partir dos frames do vídeo, calcule a homografia que projeta o padrão do mundo real (folha impressa) no plano da câmera;

4 - Crie manualmente modelos de sólidos geométricos básicos (cubos, tetraedros). Você pode fazer isso especificando as coordenadas dos cantos dos sólidos em função do sistema de coordenadas no mundo real. (OBS: neste passo você está livre para criar objetos tridimensionais sintéticos tão complexos quanto queira)

5 - Para cada frame do vídeo, calcule a projeção das arestas dos sólidos no plano da câmera, e insira artificialmente os objetos na perspectiva correta conforme a câmera é movimentada. Utilize cores diferentes para cada sólido (ou use seu objeto complexo!).

6 - Gere um vídeo de saída, com os sólidos/objetos inseridos por você no vídeo original;

#### 4. Detecção de Objetos Abandonados

Este projeto se baseia em vídeos de monitoramento, visando a detecção de objetos abandonados em um ambiente desordenado. Os vídeos são gerados por uma câmera-robô que se desloca em velocidade constante indo e vindo sobre um eixo fixo em uma calha. Uma sequência de referência apresenta o vídeo gerado pelo robô no caso onde não há objetos abandonados. O objetivo do projeto é indicar a presença de objetos nos vídeos de teste ausentes no vídeo de referência. As sequências podem ser obtidas em:

<http://www.smt.ufrj.br/~tvdigital/database/objects>

Passos:

Parte 1: Alinhamento dos vídeos

- a) Uma vez que a câmera apresenta apenas movimento translacional, é possível utilizar as homografias entre quadros sucessivos para encontrar o deslocamento relativo entre quadros.
- b) Extraia as características para cada quadro do vídeo de referência (utilizando, por exemplo, SIFT);
- c) Calcule as homografias entre os quadros da sequência e um quadro inicial, e plote a curva de deslocamentos em função do número do frame. (Utilize o método RANSAC para remover outlier e descobrir a melhor homografia). Encontre o ponto de retorno da câmera no seu eixo de deslocamento.
- d) Repita os passos 1 e 2 para o vídeo de teste e utilize as informações obtidas para alinhar temporalmente os frames de referência e de teste (sugestão: você pode modelar as curvas obtidas para cada sequência como segmentos de reta e descobrir a defasagem temporal entre elas).

Parte 2: Detecção

- a) Agora, com os vídeos das duas sequências sincronizados temporalmente, é possível obter precisamente os pares de imagens (uma proveniente de cada vídeo) a serem comparados para fins de detecção de objetos abandonados. Uma técnica simples de detecção seria então a simples diferença entre imagens. Calcule a diferença entre os frames correspondentes e defina uma estratégia que identifique objetos abandonados. Quais os problemas com essa abordagem?
- b) Utilize seus conhecimentos sobre descritores e extração de características para propor uma nova metodologia para detecção de objetos abandonados baseada na comparação de descritores nos frames sincronizados. Que vantagens esta abordagem apresenta em relação à utilizada no passo 4?

## 5. Detecção de Automóveis

Neste projeto, um banco de dados contendo sequências de vídeo de automóveis girando em uma plataforma será usado para definir padrões representativos de automóveis em diferentes poses.

Técnicas de detecção de características deverão ser utilizadas para reduzir a dimensionalidade dos descritores associados a diferentes poses de automóveis, que por sua vez serão usadas na detecção e rastreamento de veículos em vídeos provenientes de câmeras de monitoramento de tráfego.

Passos:

- 1 - A partir dos vídeos contidos na base disponível em (<http://cvlab.epfl.ch/data/pose>), defina intervalos de rotação dos automóveis que representem diferentes poses de veículos (diferentes intervalos de ângulos de rotação podem estar associados a diferentes poses)
- 2 - Utilize técnicas de extração de características (p. ex. SIFT, SURF) para gerar descritores que representem diferentes poses de automóveis;
- 3 - Utilize o algoritmo *k-means* para definir classes representativas de descritores para diferentes orientações de veículos;
- 4 - Pré-processe os vídeos das câmeras de monitoramento utilizando técnicas de extração de foreground para definir as regiões de objetos em movimento;
- 5 - Com base no dicionário de poses de veículo definido no passo 3, detecte a presença (ou não) de veículos em cada frame dos vídeos de monitoramento (destaque os automóveis detectados em cada frame dos vídeos);
- 6 - Categorize os veículos detectados nos vídeos de monitoramento, indicando a presença (ou não) de veículos e, quando presentes, sua orientação. Dica: Usem algum método de classificação como redes neurais, SVM, etc.

## 6. Detecção de Olhos

Neste trabalho iremos tratar da detecção de olhos em uma base de dados bastante conhecida. A BioID está disponível em (<https://www.bioid.com/About/BioID-Face-Database>). A detecção de olhos pode ser utilizada em etapas iniciais de diversos sistemas como detecção de fadiga de motoristas, estimação de pose da cabeça, sistemas de auxílio a pessoas com deficiência, etc. Passos:

Parte 1 - Modelo de distribuição da posição dos olhos.

a) Muitas bases de dados possuem anotações manuais da posição dos olhos. A BioID é uma delas. Separe a BioID em dois Conjuntos, um para treino e outro para teste. Use o conjunto de treino para estimar um modelo de distribuição da posição dos olhos. Assuma que a o posicionamento dos olhos possuem distribuição Gaussiana. Como cada face da base de dados possui um tamanho diferente, não se esqueça de levar isso em consideração quando for estimar o modelo.

b) Aplique este modelo no conjunto de teste e, com isso, obtenha a posição mais provável dos olhos. Calcule o erro médio com relação às anotações manuais. Quais os problemas desta abordagem?

Parte 2 - Detecção de olhos baseada em modelos.

a) Alguns trabalhos efetuam a detecção dos olhos usando detectores baseados em modelo. Em (Valenti e Gevers, 2012) e (Timm e Barth, 2011) temos dois exemplos. Implemente um destes trabalhos e aplique o detector no conjunto de teste. Compare os resultados com o item b) da primeira parte. Aponte as principais vantagens e desvantagens.

Parte 3 - Detecção de olhos baseada em aprendizado.

a) Por outro lado, muitos métodos envolvem algum tipo de aprendizagem. Em (Araujo et al., 2014) temos um exemplo. Implemente este trabalho e obtenha o erro médio de detecção no conjunto de teste. Compare este resultado com os obtidos nas etapas anteriores. Quais as principais vantagens e desvantagens deste tipo de método?

-R.Valenti and T.Gevers, "Accurate eye center location through invariant isocentric patterns," IEEE Trans. on Pattern Analysis and Machine Intel., vol. 34, no. 9, pp. 1785–1798, Sept 2012.

- F. Timm and E. Barth, "Accurate eye centre localisation by means of gradients," in Intl. Conf. on Computer Theory and Applications, 2011.

- G. M. Araujo et al., "Fast eye localization without a face model using inner product detectors," IEEE Intl. Conf. on Image Processing, 2014.

## 7. Detecção de Placas

Neste projeto, pretendemos detectar placas de carros em uma base de dados contendo imagens frontais de automóveis. As imagens contêm condições adversas como sombra, ruído, condições extremas de iluminação. Iremos tratar diversos aspectos de um sistema de detecção de placas.

Passos:

### Parte 1 - pré-processamento

a) A partir do banco de dados disponível em ([www.ononononon.com](http://www.ononononon.com)) monte um conjunto de treinamento e de teste. Notem que temos somente uma placa em cada imagem, mas temos diversas amostras negativas. Precisamos tratar o problema de desbalanceamento.

b) as imagens foram obtidas em condições adversas de luminosidade e usando câmeras de baixo custo. Use estratégias de pré-processamento para tornar o seu sistema robusto a esses problemas. Dica: testem várias técnicas de suavização, correção de iluminação, etc.

### Parte 2 - extração de características

a) Mesmo após o pré-processamento as amostras positivas e negativas ainda possuem uma dimensão muito grande. Isso tende a tornar a etapa de detecção muito lenta ou mesmo inviável. Monte uma estratégia de extração de características. Dica: tentem várias técnicas como extração de pontos salientes, SIFT, SURF, PCA, etc.

### Parte 3 - detecção

a) Utilize algum detector para encontrar o retângulo da placa nas imagens de teste. Dica: existem algumas possibilidades, como Redes neurais, SVM, etc. Não se esqueça de avaliar os resultados usando os arquivos de referência.

b) Elaborem alguma heurística para separar as caixas contendo letras.

c) Apliquem alguma técnica de Reconhecimento de caracteres ópticos para descobrir a sequência de letras e números nas placas. Avaliem os resultados, confrontando a saída do sistema com a referência.



## 8. Reconhecimento de Faces

Deteção de faces é útil como etapa inicial de diversos sistemas, como sistemas de segurança, estimação de pose da cabeça, etc. Neste projeto iremos fazer o reconhecimento de indivíduos usando faces uma base de dados disponível em:

[http://www02.smt.ufrj.br/~tvdigital/muque/page\\_02.html#SEC2.1](http://www02.smt.ufrj.br/~tvdigital/muque/page_02.html#SEC2.1).

Este projeto deve seguir aproximadamente os seguintes passos:

### Parte 1 - Deteção da face

a) O algoritmo de deteção Viola-Jones é um dos mais utilizados atualmente. Aplique este algoritmo para fazer a deteção da face quadro a quadro. Não se esqueçam de ajustar os parâmetros do algoritmo para obter um desempenho melhor.

b) É comum uma pequena variação no tamanho da janela da face. Mesmo quando o indivíduo se move pouco. Elabore uma heurística para encontrar uma correlação temporal entre o tamanho da janela ao longo dos frames. Utilize isso para suavizar a variação da janela. Métodos que detectam a um objeto quadro a quadro e rastreiam utilizando técnicas para encontrar correlação temporal entre as deteções são também conhecidos como métodos de "tracking by detection".

Dica: utilize uma abordagem simples como média móvel.

### Parte 2 - Reconhecimento de faces.

a) Também é possível efetuar o reconhecimento de indivíduos em vídeo usando um banco com faces previamente cadastradas. Crie um banco como este usando a base de dados disponível.

b) Existem alguns algoritmos estado-da-arte que são amplamente utilizados para fazer o reconhecimento de faces. Em (Prates et al., 2012) podemos encontrar dois deles, o CFA e o SIFT/SURF. Utilize o CFA, como descrito neste trabalho, para efetuar o reconhecimento.

c) Utilize o SIFT/SURF, como descrito em (Prates et al., 2012) para efetuar o reconhecimento. Compare os resultados obtidos com o item b).

d) O SIFT/ SURF possui certa robustez à variação de iluminação. Refaça o reconhecimento usando CFA, mas desta vez aplique o método de normalização da iluminação descrito em (Tan e Triggs, 2007) como etapa de pré-processamento. Compare os resultados obtidos. Dica: como a robustez deste método de normalização da iluminação depende muito dos parâmetros escolhidos, talvez seja uma boa ideia testar algumas combinações.

- R. L. Prates et al., "Filtros de correlação e características Invariantes à escala para o reconhecimento de faces," in Simpósio Brasileiro de Telecomunicações, 2012.

- X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," in AMFG, 2007.

## 9. Detecção de Pedestres

Detecção e rastreamento de pedestre é muito útil em sistemas de segurança. Podemos encontrar um survey de métodos e bases de dados usados na detecção de pedestres em (Zheng e Chen, 2012). Iremos utilizar as bases de dados:

MIT: <http://cbcl.mit.edu/cbcl/software-datasets/PedestrianData.html>,

PETS2009: <http://cs.binghamton.edu/~mrldata/pets2009.html>,

INRIA: <http://pascal.inrialpes.fr/data/human/>.

Este projeto deve seguir aproximadamente os seguintes passos:

### Parte 1 - Detecção de movimento

a) Pré-processe os vídeos das câmeras de monitoramento utilizando técnicas de extração de foreground para definir as regiões de objetos em movimento;

### Parte 2 - Extração de características

a) Utilize o HoG ou SIFT/SURF para fazer extração de features das áreas em movimento.

### Parte 3 - Detecção de pedestres

a) Utilize o SVM Para fazer o reconhecimento dos pedestres. Compare os resultados da combinação HoG-SVM com SIFT/SURF-SVM. Dica: exemplos de HoG com SVM para detecção de pedestres pode ser encontrado em (Xu et al., 2011) e em (Dalal e Triggs, 2005).

- G. Zheng and Y. Chen, "A review on vision-based pedestrian detection," in IEEE Global High Tech Congress on Electronics, 2012.

- G. Xu, et al., "Real-time pedestrian detection based on edge factor and histogram of oriented gradient," in IEEE Intl. Conf. on Information and Automation, 2011.

- N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in CVPR, 2005.