Universidade de São Paulo - USP

Escola de Engenharia de São Carlos - EESC

Departamento de Engenharia Mecânica - SEM

Grupo de Mecatrônica

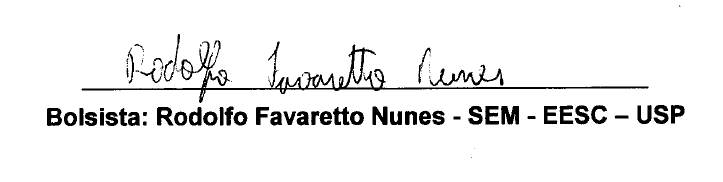
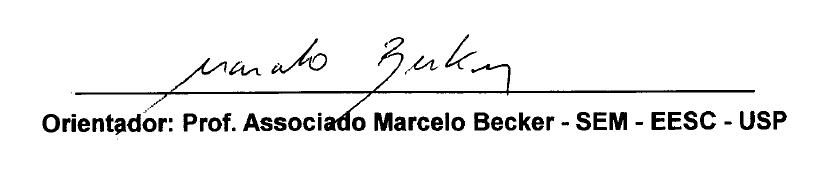
Laboratório de Robótica Móvel - LabRoM

Relatório Parcial de Projeto de Iniciação Científica

Projeto:

**Sistema Embarcado de Navegação Autônoma – Detecção de Faixas com Câmera Monocular e Otimização Através de Algoritmo Genético**

**D:\Meus Documentos\FACULDADE\IC\Becker\Logo Fapesp.jpg**

****

Maio / 2012

Conteúdo

[1 – RESUMO DO PLANO INICIAl 3](#_Toc323300760)

[2 – RESUMO DO TRABALHO 4](#_Toc323300761)

[3 – DETALHAMENTO DO TRABALHO 5](#_Toc323300762)

[3.1 – INTRODUÇÃO 5](#_Toc323300763)

[3.2 – METODOLOGIA 8](#_Toc323300764)

[3.2.1 – DETECTORES DE BORDA 9](#_Toc323300765)

[3.2.1.1 – OPERADOR DE CANNY 10](#_Toc323300766)

[3.2.1.2 – LAPLACIANO DA GAUSSIANA (LOG) 13](#_Toc323300767)

[3.2.1.3 – OPERADOR DE SOBEL 14](#_Toc323300768)

[3.2.1.4 – DETECTOR COM MÁSCARA INCLINADA 16](#_Toc323300769)

[3.2.2 – TRANSFORMADA DE HOUGH 16](#_Toc323300770)

[3.2.2.1 – SOBREPOSIÇÃO DE IMAGENS 20](#_Toc323300771)

[3.2.3 – IPM COM HISTOGRAMAS 21](#_Toc323300772)

[3.2.5 – CRIAÇÃO DO FUNDO VERDADE 25](#_Toc323300773)

[3.2.6 – ALGORITMO GENÉTICO 28](#_Toc323300774)

[3.3 – RESULTADOS 30](#_Toc323300775)

[3.4 – CONCLUSÕES 32](#_Toc323300776)

[4 – EVOLUÇÃO DO PROJETO 33](#_Toc323300777)

[5 - REFERÊNCIAS 35](#_Toc323300778)

# 1 – RESUMO DO PLANO INICIAl

O Projeto de Iniciação Científica 2011/17619-8, aprovado pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), tem por objetivo desenvolver algoritmos que façam a detecção de faixas de trânsito de maneira pós-processada buscando obter uma boa detecção em ruas e rodovias utilizando a técnica de algoritmo genético para melhoramento dos parâmetros utilizados, de modo que contribua para a pesquisa desenvolvida no Laboratório de Robótica Móvel (LabRoM) da Escola de Engenharia de São Carlos, Universidade de São Paulo (EESC-USP), mais especificamente no projeto intitulado “Sistema Embarcado de Navegação Autônoma - SENA” que objetiva o desenvolvimento de um veículo com capacidade de se locomover de forma semi-autônoma (assistida) e autônoma.

Para tanto, foi determinado que, em um primeiro momento, seria realizada a implementação em Matlab de três algoritmos para a detecção de borda: Operador de Canny, Operador de Sobel e Operador Laplaciano da Gaussiana. E então a partir disso, os parâmetros desses detectores seriam otimizados com o auxílio de um algoritmo genético com a finalidade de gerar uma rotina mais robusta que faça boas detecções tanto em ruas urbanas quanto em rodovias.

Além disso, o projeto prevê que será analisada a eficiência da utilização da Transformada de Hough para fazer a descrição geométrica das faixas e que serão procuradas outras técnicas que possam ser utilizadas para essa mesma finalidade.

# 2 – RESUMO DO TRABALHO

O presente relatório tem por finalidade apresentar o andamento do Projeto de Iniciação Científica 2011/17619-8, iniciado em 1º de Dezembro de 2011, à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), que financia o seu desenvolvimento.

A pesquisa, desenvolvida no Laboratório de Robótica Móvel (LabRoM) da EESC - USP, visou desenvolver ferramentas computacionais capazes de fazer a detecção de bordas em imagens de ruas urbanas e rodovias de maneira pós-processada em Matlab com a finalidade de encontrar faixas de trânsito.

Para realizar a detecção de bordas foram implementados quatro algoritmos: Operador de Canny, Operador de Sobel, Operador Laplaciano da Gaussiana e um detector com máscaras inclinadas. Este último foi desenvolvido tendo por base o Operador de Sobel, como uma alternativa que não estava prevista na proposta inicial do projeto.

Para se obter a descrição geométrica das faixas, foi implementado o algoritmo de Transformada de Hough nas imagens depois de já terem sido detectadas as bordas. Além disso, foram propostas duas alternativas para a descrição das faixas, uma para ser utilizado em vídeos, que consiste em fazer a sobreposição de seguidas imagens e depois utilizar a Transformada de Hough, e outra que seria fazer a transformada inversa perspectiva da imagem e em seguida analisar os histogramas verticais dessa imagem para se identificar as faixas.

# 3 – DETALHAMENTO DO TRABALHO

## 3.1 – INTRODUÇÃO

A cada ano é verificado um aumento na quantidade de acidentes e mortes no trânsito brasileiro e mundial. Os números são tão alarmantes nos últimos tempos que, em 2011 foi lançada pela ONU (Organização das Nações Unidas) a década de ação pelo trânsito seguro (2011 – 2020) com a finalidade de prevenir tais acidentes (ONUBR, 2011), e a OMS (Organização Mundial da Saúde) considerou as mortes em acidentes de trânsito como um problema mundial de saúde pública, alcançando a marca de nona maior causa de morte no mundo, um total de aproximadamente 1,3 milhões de pessoas por ano (OMS, 2009). O Brasil está entre os dez países com o maior número de mortes no trânsito no mundo, juntamente com África do Sul, China, Egito, Estados Unidos, Índia, Irã, México e Rússia segundo dados da OMS (2009). Pode-se observar na um comparativo nesse quesito entre o Brasil e outros países da América do Sul.

Tabela 1. Mortes em acidentes de Trânsito (OMS, 2009).

|  |  |  |
| --- | --- | --- |
| País | Número absoluto | Número a cada 100.000 habitantes |
| Argentina | 4.063 | 10,28 |
| Bolívia | 1.073 | 11,27 |
| Brasil | 35.155 | 18,33 |
| Chile | 2.280 | 13,71 |
| Colômbia | 5.409 | 11,72 |
| Paraguai | 845 | 13,79 |
| Peru | 3.510 | 12,58 |
| Uruguai | 427 | 12,79 |

Infelizmente esses números não tendem a sofrer uma queda significativa nos próximos anos, já que o número de carros novos vendidos é superado a cada ano e os investimentos na malha rodoviária em diversos estados estão aquém das necessidades. Em apenas seis anos o número de carros novos vendidos dobrou e o Brasil passou da 10ª posição para a 4ª colocação no *ranking* mundial nesse quesito (Fenabrave, 2010).

Diante dessa situação, a indústria automobilística vem sendo pressionada por regulamentações governamentais e pela opinião pública e como consequência vem investindo muito em novos sistemas de segurança para veículos (Folha, 2010). Também nos últimos anos, presenciou-se um grande desenvolvimento nas áreas de eletrônica e sensores embarcados, tornando mais viável a aplicação de técnicas de robótica móvel e inteligência artificial em veículos de passeio, sendo possível a instalação de sistemas avançados de auxílio ao motorista ou *Advanced Driver-Assistance Systems* (ADAS). Sistemas tais como o Controle Ativo de Cruzeiro, Frenagem Automática de Emergência, Proteção Pré-Colisão e o Sistema de Permanência na Pista (Kowalenko, 2010). Esse tipo de tecnologia pode diminuir o número de acidentes de trânsito que ocorrem devido a falhas do motorista, causas como: excesso de velocidade, fadiga, sono, imperícia e imprudência. Um exemplo desse tipo de sistema em veículos de passeio pode ser observado no carro Fusion da empresa Ford. A montadora anunciou recentemente que usará a tecnologia de permanência na pista no carro Fusion modelo 2013, que apresentará um sistema opcional que consiste em uma câmera instalada próximo ao espelho retrovisor e que terá a função de identificar as faixas de trânsito e alertar o motorista caso ele atravesse uma faixa sem utilizar o sinal luminoso de mudança de direção, isso tem por objetivo diminuir principalmente acidentes causados por cansaço e sono do motorista (Ford, 2011).

Esse tipo de sistema de transporte pessoal com sistemas autônomos já existe faz algum tempo, entretanto começou a ganhar mais visibilidade a partir de 2003 quando a agência americana DARPA (*Defense Advanced Research Projects Agency*) propôs uma competição entre carros autônomos, o que gerou um grande desenvolvimento nesse tipo de tecnologia. Mais atualmente, novos carros começaram a serem elaborados, como exemplo pode-se citar o carro desenvolvido pela Google. Em sua aparição no ano de 2010, o mesmo executou aproximadamente 225.000 km de testes com apenas uma colisão em sua traseira (Ackerman, 2010).

É nesse contexto que o Laboratório de Robótica Móvel do Grupo de Mecatrônica da EESC-USP desenvolve o Projeto SENA – Sistema Embarcado de Navegação Autônoma – que tem por objetivo principal o desenvolvimento de um veículo com capacidade de se locomover de forma semi-autônoma (assitiva) e autônoma em ambientes urbanos e rodovias. O projeto inclui também o controle total do veículo em momentos que o motorista não possa atuar de modo eficiente, fazendo que a ação de dirigir seja monitorada pelos computadores embarcados, o que corresponderia a uma atuação autônoma do veículo (SENA, 2012). Para seu funcionamento o veículo possui diversos sensores como pode ser observado na .

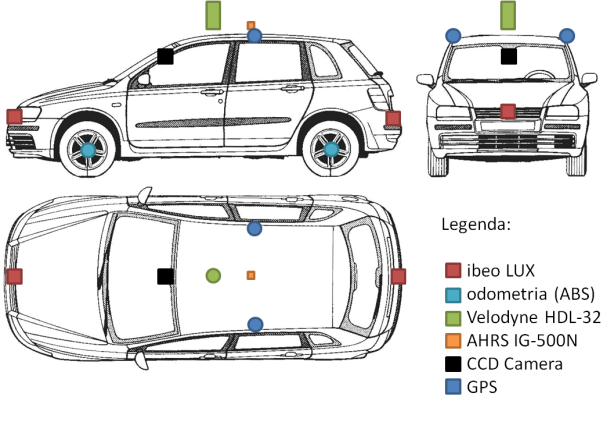


Figura 1 Ilustração dos sensores embarcados no veículo de testes.

## 3.2 – METODOLOGIA

O objetivo principal dessa pesquisa baseia-se em localizar e identificar as faixas de trânsito, para isso inicialmente precisa-se detectar as bordas presentes na imagem e então utilizar algum método para identificar quais são as bordas relevantes, ou seja, quais representam as faixas e eventualmente as sarjetas das ruas. Como citado anteriormente, para a primeira parte, a detecção de bordas, foram estudados e implementados quatro algoritmos: Operador de Canny, Operador Laplaciano da Gaussiana, Operador de Sobel e um detector com máscara inclinada. Para a segunda parte, a identificação da faixa, foram utilizadas duas técnicas: a Transformada de Hough e o Mapeamento Inversa da Perspectiva com análise de histogramas. Paralelamente a isso, foi construído o fundo verdade de um banco de imagens coletadas com o veículo de teste, com a finalidade de futuramente (na segunda etapa do projeto) ser utilizado no treinamento dos parâmetros dos detectores de borda através de um algoritmo genético.

Para fazer o desenvolvimento dos algoritmos foi utilizado o *software* Matlab. Esse ambiente de desenvolvimento foi escolhido devido a sua eficiência e facilidade no trabalho com matrizes, além de possuir um alto desempenho e ser voltado para o cálculo numérico (Matsumoto, 2008).

Todas as imagens utilizadas nos testes foram coletadas na forma de fotos e vídeos obtidos durante passeios do veículo de teste. As fotos tiverem seu tamanho pré-fixado em 640x480, pois essa é a resolução ideal segundo Khalifa et al. (2010) já que imagens maiores teriam o tempo de processamento muito alto, e fotos com resolução menor deixam a desejar no detalhamento da imagem.

## 3.2.1 – DETECTORES DE BORDA

O princípio básico dos detectores de borda é encontrar a fronteira entre regiões com diferentes intensidades de nível de cinza. Para isso é preciso inicialmente utilizar uma transformação na imagem, passando de um espaço de cor RGB para uma escala de cinza que se utiliza apenas da iluminação, eliminando-se assim a matiz e a saturação da imagem (Mathworks, 2012). Isso é feito se utilizando a Equação 1:

(1)

Sendo Y a intensidade em escala de cinza, R a componente vermelha do valor do pixel, G a componente verde e B a de azul. Com isso é possível observar que a imagem que antes possuía três parâmetros passa a ter apenas um componente, agilizando o processamento da imagem (Assidiq et al., 2008).

Tendo a imagem em escala de cinza os detectores utilizam normalmente da derivada da intensidade dos pixels ao longo da imagem para localizar as bordas, isso acontece, pois essa transição de nível de cinza apresenta uma função característica, como se pode observar na .

Figura 2 – Esquema simplificado de identificação de bordas

Agora serão apresentados os algoritmos detectores de borda que foram estudados e implementados.

### 3.2.1.1 – OPERADOR DE CANNY

Canny estabeleceu que um bom algoritmo detector de bordas deve atender a três necessidades básicas (Canny, 1986):

1. Baixa taxa de erro: O primeiro critério define que deve haver uma baixa probabilidade de o detector identificar um *pixel* que esteja fora da região de borda.
2. Boa Localização: Os pontos detectados como borda devem estar o mais próximo possível da real borda.
3. Resposta única: Deve haver apenas um ponto de máxima que será dito borda, ou seja, evitar bordas duplas.

Baseado nessas premissas, Canny desenvolveu um algoritmo que passa por diversas etapas com o objetivo de se fazer uma boa detecção das bordas presentes na imagem.

Essas etapas serão descritas abaixo:

1. Filtragem: A filtragem é uma parte essencial em qualquer detector de bordas, pois tem a finalidade de reduzir a quantidade de ruído presente na imagem, que quando em excesso pode provocar uma mudança no gradiente e assim produzir uma borda falsa. O filtro construído por Canny, Equação 3, pode ser aproximado pela primeira derivada da função Gaussiana (Silva et al., 2004), apresentado na Equação 2. Onde o desvio-padrão (σ) pode ser modificado manualmente, e será um dos parâmetros otimizados pelo algoritmo genérico posteriormente.

(2)

(3)

1. Duas máscaras unidimensionais são criadas, uma horizontal e uma vertical que são aplicadas sobre a imagem filtrada para fazer sua diferenciação e assim gerar Ix e Iy respectivamente. Com isso pode-se calcular a magnitude em cada ponto da imagem utilizando a Equação 4.

(4)

1. Supressão não máxima: esse processo tem por objetivo provocar um afinamento das bordas, tendo por princípio que o ponto de borda deverá ser aquele que possui um valor máximo. Assim, a supressão não máxima provoca a anulação dos *pixels* que não sejam máximos locais levando-se em consideração a direção perpendicular à borda. Esse processo também proporciona uma melhor localização das fronteiras e diminui a incidência de bordas duplas. Pode-se observar que este processo é muito importante para o que o algoritmo atenda o terceiro critério de Canny como visto em Silva et al.(2004) e em Vale e Dal Poz (2002).
2. Histerese: esse é um processo de limiarização que completará a segmentação da imagem. São aplicados dois limiares, também chamados de *Threshold values,* sendo um de valor baixo T1 e outro de valor alto T2. O processo consiste em percorrer todos os valores de magnitude da imagem classificando-os, aqueles que possuírem valor maior que T2 serão ditos “bordas fortes”, os que possuírem valor entre T1 e T2 serão chamados de “bordas fracas” e os que possuírem magnitude menor que T1 serão eliminados (Gonzalez e Woods, 2007). Em seguida os pontos de borda fraca são analisados, se um ponto de borda fraca tiver algum vizinho 8-conectado que seja borda forte, ele é incorporado a este, tornando-se assim um ponto de borda, por outro lado os que não possuírem alguma borda forte como vizinha serão eliminados.

Ao fim desse processo a imagem estará binarizada e deverá apresentar apenas as bordas de maior intensidade.

### 3.2.1.2 – LAPLACIANO DA GAUSSIANA (LOG)

A função Gaussiana é muito utilizada na visão computacional como um filtro, normalmente utilizado para remover ruídos na imagem devido ao “borramento” provocado com a convolução dessa função com a imagem. O algoritmo apresentado se utilizada da segunda derivada da função Gaussiana, ou seja, de seu Laplaciano como mostrado na Equação (5), calculado na direção *x* e *y* ou em r, sendo e σ o desvio padrão.

(5)

Como a derivada secundária se trata de uma operação linear, então aplicar o LoG na imagem é o mesmo que aplicar a Gaussiana na imagem e depois calcular o Laplaciano do resultado. Assim pode-se observar que ao aplicar o LoG na imagem, dois efeitos são provocados: ele borra a imagem provocando a redução dos ruídos e calcula o Laplaciano, ou seja calcula a derivada secundária, assim encontrando as bordas presentes na imagem. Entretanto, ao se utilizar esse processo são encontradas bordas duplas como resposta, para resolver esse problema o algoritmo procura pelos pontos que cruzam pelo zero entre as duas bordas, e então assume que essa seja verdadeiramente a borda e não as anteriores. Pode-se ver isso mais facilmente através da .

Depois de serem encontradas todas as bordas da imagem elas são analisadas e comparadas com o valor de limiar imposto, eliminando as de valor menor que do *threshold*. Pretende-se que esse valor seja otimizado pelo algoritmo genético, assim como o valor de σ, que é o desvio-padrão, responsável pela intensidade do borramento da imagem.

Figura 3 - Funcionamento do operador Laplaciano da Gaussiana

### 

### 3.2.1.3 – OPERADOR DE SOBEL

O operado de Sobel consiste na aplicação de duas máscaras de filtragem de tamanho 3x3 que passam por convolução com a imagem, sendo uma máscara para detecção vertical e uma para detecção horizontal, gerando duas imagens que podem ser analisados separadamente ou comporem um único resultado. Abaixo são mostradas as matrizes utilizadas na convolução (Yong-Dong et al., 2005).

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Tabela 2 - Tabela de valor dos pixels   |  |  |  | | --- | --- | --- | | Z1 | Z2 | Z3 | | Z4 | Z5 | Z6 | | Z7 | Z8 | Z9 | |  |
| Tabela 3 - Máscara de Gx |  | Tabela 4 - Máscara de Gy |
| |  |  |  | | --- | --- | --- | | -1 | -2 | -1 | | 0 | 0 | 0 | | 1 | 2 | 1 | |  | |  |  |  | | --- | --- | --- | | -1 | 0 | 1 | | -2 | 0 | 2 | | -1 | 0 | 1 | |

De posse das máscaras e dos valores dos pixels vizinhos ao que se quer analisar pode-se encontrar o valor do gradiente para cada ponto da imagem seguindo as seguintes equações (Gonzalez e Woods, 2007):

(6)   
 (7)  
 Para se determinar um ponto de borda basta estabelecer um valor de limiar e analisar o valor do gradiente em cada ponto. Se o valor do ponto for maior que o do *theshold* imposto diz-se que é um ponto de borda. Além dessa analise na vertical e na horizontal separadamente, pode-se encontrar os pontos de borda fazendo o módulo dos gradientes das duas imagens como pode ser visto nas Equações 8 e 9:

(8)

(9)

É importante ressaltar que mesmo depois de encontrar o módulo dos gradientes, as bordas serão mais ressaltadas nas direções verticais e horizontais, falhando ao encontrar algumas bordas inclinadas.

### 3.2.1.4 – DETECTOR COM MÁSCARA INCLINADA

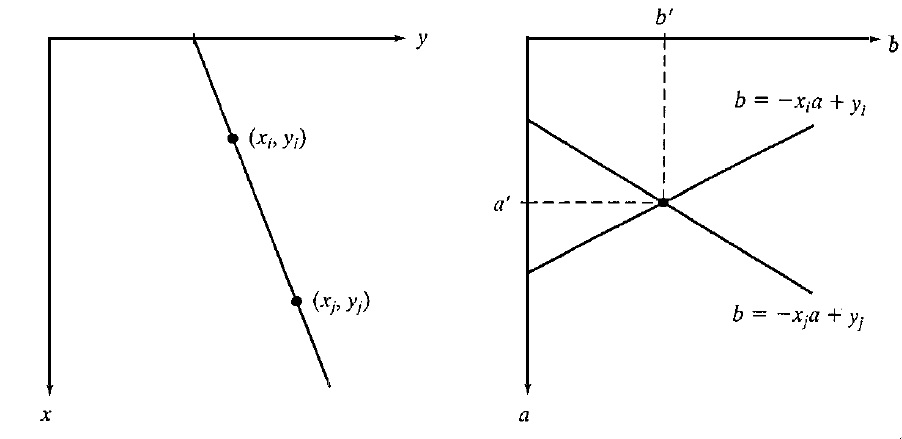
A técnica de Sobel é muito interessante, pois ressalta a borda em direções determinadas, entretanto as direções ressaltadas não são de tanto interesse na determinação de faixas de trânsito. Devido a isso, optou-se por fazer outro algoritmo com a mesma ideia do Sobel, porém que buscasse bordas que estavam na direção que as faixas normalmente se encontram nas fotos, de aproximadamente 45º. As máscaras utilizadas para isso são mostradas abaixo.

|  |  |
| --- | --- |
| Tabela 5 - Máscara de 45° | Tabela 6 - Máscara de -45° |
| |  |  |  | | --- | --- | --- | | -2 | -1 | 0 | | 1 | 0 | 1 | | 0 | 1 | 2 | | | |  |  |  | | --- | --- | --- | | 0 | -1 | -2 | | 1 | 0 | -1 | | 2 | 1 | 0 | |

Semelhante ao desenvolvimento de Sobel, o módulo do gradiente é encontrado e desse modo, são obtidas duas imagens que podem ser analisadas separadamente ou podem ser sobrepostas para fazer uma análise geral. Ao ser comparado com o valor de limiar imposto, as imagens são binarizadas e tem-se assim, as bordas encontradas.

## 3.2.2 – TRANSFORMADA DE HOUGH

Depois de fazer a identificação de todas as bordas através dos detectores apresentados, teremos imagens binarizadas e com muitas informações desnecessárias e ruídos. Com a utilização da Transformada de Hough pretende-se extrair dessa imagem apenas a localização e a angulação das faixas de trânsito e quando possível também das sarjetas da rua. Basicamente o objetivo desse método é localizar um conjunto de pontos que sejam colineares.

Para aplicar essa transformada, tomamos um ponto da imagem que seja considerado borda e é possível observar que por ele passam infinitas retas, sendo que todas obedecem à equação para diferentes valores de e . Reescrevendo a equação da forma teremos o chamado plano dos parâmetros e nesse caso faz-se com que e sejam as variáveis. Pode-se tomar outro ponto da reta que possua os mesmos e e fazer o mesmo processo descrito acima. Com isso tem-se uma pequena amostra da transformada, como pode ser observado na .**Figura 4 - Exemplo de transformada do espaço cartesiano para o espaço dos parâmetros (Gonzalez e Woods, 2007).**

Pode-se observar que um ponto no espaço cartesiano corresponde a uma reta no espaço dos parâmetros, e uma reta no espaço cartesiano corresponde a um ponto no espaço dos parâmetros, assim temos que pontos colineares no plano *xy* serão retas concorrentes com um ponto em comum, esse ponto é exatamente o que procuramos, pois com ele encontramos e que são os coeficientes angular e linear da reta que passa pelos dois pontos no espaço cartesiano. Esse é um exemplo muito simples, apresentando apenas dois pontos, mas esse processo pode ser feito para imagens binárias muito mais complexas, o que pode gerar um grande problema, pois como o plano ab é linear, é possível que as retas tendam ao infinito, e nesse caso fazer uma análise para localizar os pontos de encontro entre as retas pode ter um custo computacional muito grande e desnecessário. Por esse motivo, a Transformada de Hough não faz uma transformação para reta e sim para funções senóides (Gonzalez e Woods, 2007). Para isso utiliza-se a Equação 10 mostrada abaixo.

(10)

Dessa forma a função fica limitada e o algoritmo encontra mais facilmente o ponto de intersecção. Pode-se ver na um exemplo dessa transformada:

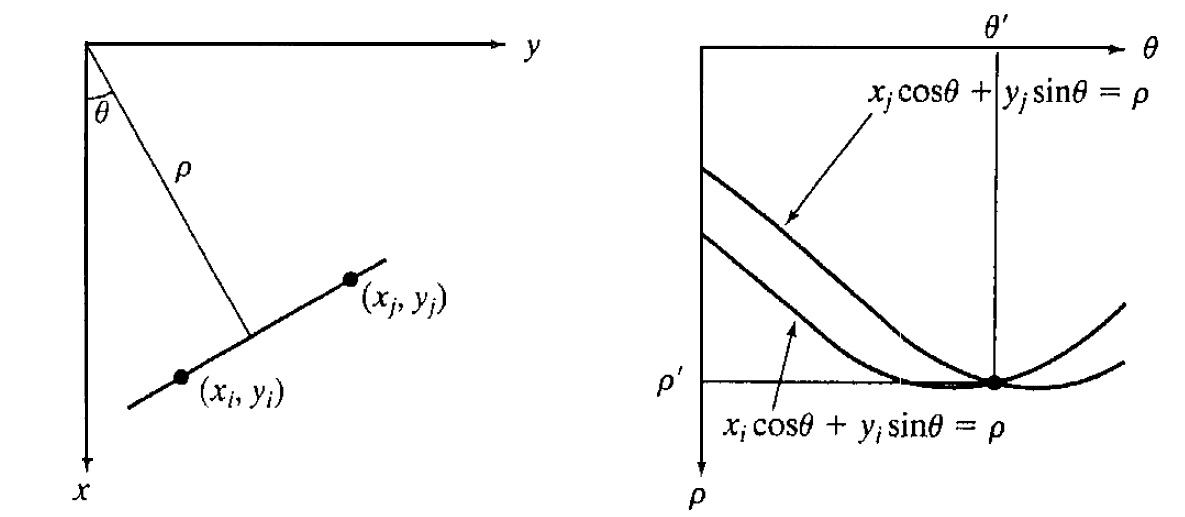


Figura 5 - Exemplo de transformada do espaço cartesiano para o espaço dos parâmetros senoidais (Gonzalez e Woods, 2007).

Pode-se observar que nesse caso, cada ponto do espaço cartesiano corresponde a uma senóide no espaço dos parâmetros e cada ponto do espaço dos parâmetros corresponde a uma reta no espaço cartesiano. Assim o ponto de intersecção entre as senóides será utilizado para encontrar e , assim temos a equação da reta que passa por ambos os pontos. Abaixo é mostrado um exemplo de transformada de Hough aplicada em uma foto retirada com o veículo de teste em uma rodovia.

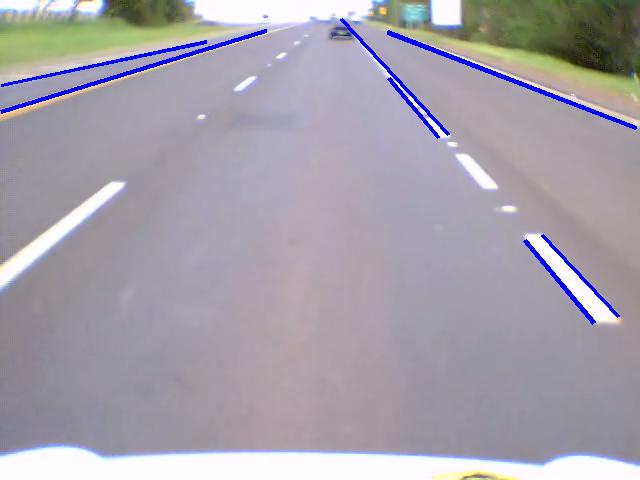
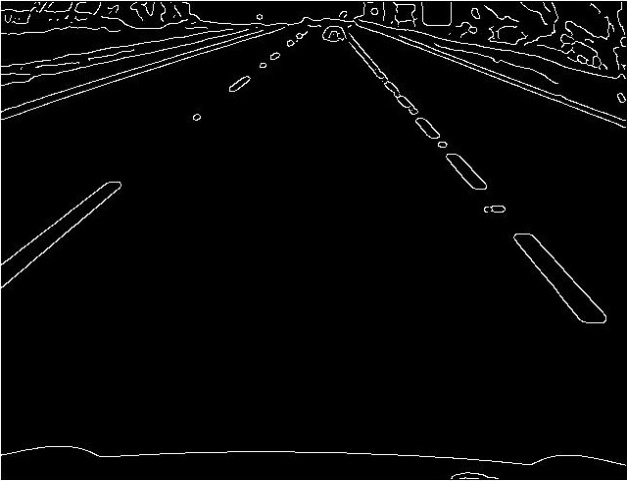


Figura 6 - Imagem com Canny e foto original com Hough

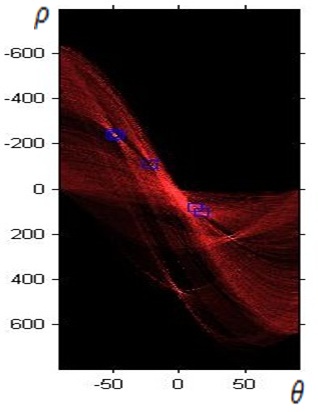


Figura - Espaço dos parâmetros da Transformada de Hough

Na acima, pode-se observar a transformada de Hough da binarizada. Cada senóide em vermelho representa um ponto da imagem binarizada e os retângulos azuis representam os pontos de maior intersecção de senóides. A posição desses retângulos nos proporciona e das retas que possuem mais pontos colineares e, portanto serão consideradas bordas. Na com transformada de Hough tem-se a representação dessas retas em azul sobre a foto original.

### 3.2.2.1 – SOBREPOSIÇÃO DE IMAGENS

Pôde-se observar na que a Transformada de Hough possui um problema em localizar faixas descontínuas, pois como possui normalmente uma menor quantidade de pontos colineares, não é classificado nesse algoritmo. Uma solução para isso é fazer a sobreposição de imagens como proposto em Nieto et al. (2009), essa técnica é válida quando se deseja localizar as bordas em um vídeo e não para fotos isoladas. O processo consiste em utilizar uma sobreposição de imagens de uma sequência de *frames* com o objetivo de reforçar as faixas descontínuas, transformando-as em faixas contínuas. Para isso é feito uma comparação *píxel* a *píxel* dos quadros que se deseja sobrepor, em cada comparação é obtido o píxel de maior intensidade e esse ponto é utilizado na imagem. Depois de feito isso para todos os pontos observa-se um grande realce principalmente nas partes brancas da imagem, já que a intensidade desses pontos é muito grande, com isso temos um destaque das bordas, porém também se observa um aumento dos ruídos e uma certa distorção em algumas partes da imagem, o que pode ser prejudicial.

A mostra um exemplo da aplicação dessa técnica em um vídeo feito com o veículo de testes em uma rodovia, a sobreposição utiliza-se de uma sequência de sete quadros. Pode-se observar uma melhor detecção das faixas quando é feito a sobreposição de imagens, principalmente da faixa descontínua apresentada na imagem.

|  |  |
| --- | --- |
| sem_sobreposição.jpg | sem_sobreposição_hough.jpg |
| a) | b) |
| sobreposição.jpg | sobreposição_hough.jpg |
| c) | d) |

Figura 8 - Exemplo de sobreposição de imagens e suas consequências: (a) Imagem original. (b) Transformada de Hough aplicada à imagem original. (c) Imagem sobreposta de sete frames seguidos. (d) Transformada de Hough na imagem sobreposta

## 3.2.3 – IPM COM HISTOGRAMAS

O mapeamento inverso da perspectiva (*Inverse Perspective Mapping* – IPM) é uma transformação que pode ser utilizada em imagens com o objetivo de retirar o efeito da perspectiva de uma imagem. A imagem resultante que se busca obter por esse método será como uma foto aérea do local (também chamada *birds-eye view*), que não possui o efeito da perspectiva como visto em Kurdziel (2008). Para fazer essa transformação na imagem o algoritmo utiliza parâmetros internos e externos da câmera que devem ser previamente conhecidos. A abaixo representa como a mudança da perspectiva modifica a imagem:

|  |  |
| --- | --- |
|  |  |

Figura 9 - Representação da modificação gerada na imagem pelo método IPM. À esquerda temos a imagem original e à direita a imagem sem efeito da perspectiva (Kurdziel, 2008).

O método IPM baseia-se na construção de uma tabela que relaciona a posição de cada píxel com a sua nova posição na imagem sem perspectiva. Assim, basta percorrer toda a imagem trocando os *pixels* de lugar para se obter uma nova figura com visão aérea. Para se construir essa tabela de conversão de uma imagem para uma imagem utiliza-se duas equações mostradas a seguir:

(11)

(12)

Sendo que:

* é a altura da câmera em relação ao chão.
* é a meia abertura da câmera em coordenadas horizontais.
* é a meia abertura da câmera em coordenadas verticais.
* é o ângulo de inclinação da câmera em relação a horizontal.
* é o tamanho da imagem original na horizontal.
* é o tamanho da imagem original na vertical.

Baseado nessas equações, foi desenvolvido uma tabela que relacione a posição de cada píxel da imagem original com sua nova posição na imagem sem o efeito da perspectiva para utilizar nas fotos de rua retiradas com o carro de teste. Abaixo pode-se observar um exemplo dessa transformação:

|  |  |
| --- | --- |
| C:\Users\RODOLFO\USP\Íniciação Cientifica\Relatório Parcial Fapesp\IPM\USP_normal.jpg | C:\Users\RODOLFO\USP\Íniciação Cientifica\Relatório Parcial Fapesp\IPM\USP_IPM.jpg |

Figura 10 - Exemplo da utilização de transformada de IPM.

Como se pode observar na , o processo de troca do lugar dos *pixels* não é perfeito já que a imagem aparece distorcida e com falta de informações, sendo pior quanto mais longe se encontra do carro. Isso já era esperado, pois a imagem original não possui todas as informações para produzir uma imagem sem perspectiva perfeita. Por outro lado, apesar de defeitos a imagem fica muito boa a distâncias próximas do carro e com a técnica de interpolação de linhas é possível suprir a falta de informações para pixels distantes. A apresenta um exemplo desse processo de interpolação, pode-se observar que a imagem fica completa, porém apresenta um desfocamento para pontos a longa distância.

|  |  |
| --- | --- |
| C:\Users\RODOLFO\USP\Íniciação Cientifica\Relatório Parcial Fapesp\IPM\IPM_sem_interpolação.jpg | C:\Users\RODOLFO\USP\Íniciação Cientifica\Relatório Parcial Fapesp\IPM\IPM_interpolação.jpg |

Figura 11 - Exemplo de interpolação da imagem para preenchimento de pixels.

Mas, a utilização de IPM nesse trabalho não é feita sobre a imagem original coletada e sim na sua aplicação nas imagens que passaram pelos detectores de borda. Depois de aplicada a transformação, pode-se fazer um histograma vertical da imagem, que basicamente consiste em fazer uma soma do valor dos pixels em todas as colunas da imagem, para assim identificar os locais que possuem faixas na foto. é mostrado um exemplo dessa técnica.

|  |  |
| --- | --- |
| C:\Users\RODOLFO\USP\Íniciação Cientifica\Relatório Parcial Fapesp\IPM\original_canny.jpg | C:\Users\RODOLFO\USP\Íniciação Cientifica\Relatório Parcial Fapesp\IPM\IPM_original.jpg |
| a) | b) |
| C:\Users\RODOLFO\USP\Íniciação Cientifica\Relatório Parcial Fapesp\IPM\canny_original.jpg | C:\Users\RODOLFO\USP\Íniciação Cientifica\Relatório Parcial Fapesp\IPM\cannny.jpg |
| c) | d) |
| C:\Users\RODOLFO\USP\Íniciação Cientifica\Relatório Parcial Fapesp\IPM\canny_completo.jpg | C:\Users\RODOLFO\USP\Íniciação Cientifica\Relatório Parcial Fapesp\IPM\histograma.jpg |
| e) | f) |

Figura 12 - Aplicação de IPM e histograma: (a) Imagem original. (b) IPM aplicado à imagem original com interpolação. (c) Bordas detectadas com Operador de Canny. (d) IPM aplicada à imagem "c". (e) Interpolação aplicada à imagem "d". (f) Histograma da imagem "e".

Com esse histograma pode-se fazer um estudo da imagem, a fim de localizar os picos e fazer uma avaliação de quais desses picos são realmente faixas. Essa análise trata-se de uma área relacionada ao sistema de cognição do veículo e não faz parte do projeto inicial, porém é uma técnica que merece destaque, e que pode ser uma forma de substituir a Transformada de Hough para fazer a identificação das faixas, como é possível verificar em Nieto et al. (2007).

## 3.2.5 – CRIAÇÃO DO FUNDO VERDADE

A geração do fundo verdade é uma parte essencial do projeto já que com ele é possível fazer uma avaliação da eficiência dos algoritmos, e além disso, pode ser usado no algoritmo genético para determinar os melhores parâmetros a serem usados nos detectores de borda.

Inicialmente pensou-se em fazer a criação do fundo verdade baseado na descrição matemática das sinalizações horizontais de trânsito. Porém, notou-se que essa técnica é muito trabalhosa, pois é necessário fazer grande parte da análise manualmente, e seria necessário fazer uma estrutura de controle condicional para cada faixa de cada foto, o que seria inviável para uma quantidade grande de imagens. Para resolver esse problema optou-se por gerar uma imagem que representasse o fundo verdade.

Primeiramente foram escolhidas as fotos que formariam um banco de dados, essas imagens foram coletadas com o veículo de testes dentro do Campus, nas ruas próximas a faculdade e em rodovias nos arredores da cidade. Então foi desenvolvido um algoritmo para fazer a leitura das imagens e através de algumas rotinas disponíveis no Matlab foi possível fazer o fundo verdade com o auxílio do *mouse*. Assim, basta apenas fazer uma comparação para verificar se as bordas detectadas estão dentro da área classificada como faixa ou sarjeta, e assim fazer a avaliação dos algoritmos para cada foto.

Essa decisão de utilizar uma imagem como fundo verdade foi muito importante já que reduziu drasticamente o tempo para criação do “gabarito” de cada foto, possibilitando a geração de um banco de dados maior do que o planejado inicialmente. Isso será de grande utilização na implementação do algoritmo genético, pois quanto maior o banco de imagens melhor é o treinamento dos parâmetros e mais robusto fica a detecção.

A seguir são mostrados alguns exemplos de fotos utilizadas e seus respectivos fundo verdade. É interessante ressaltar que com essa rotina, a criação de fundo verdade não será um problema, assim o banco de dados pode ser aumentado caso necessário.

|  |  |
| --- | --- |
| C:\Users\RODOLFO\USP\Íniciação Cientifica\Relatório Parcial Fapesp\Fundo Verdade\17.jpg | C:\Users\RODOLFO\USP\Íniciação Cientifica\Relatório Parcial Fapesp\Fundo Verdade\fv_17.jpg |
| C:\Users\RODOLFO\USP\Íniciação Cientifica\Relatório Parcial Fapesp\Fundo Verdade\30.jpg | C:\Users\RODOLFO\USP\Íniciação Cientifica\Relatório Parcial Fapesp\Fundo Verdade\fv_30.jpg |
| C:\Users\RODOLFO\USP\Íniciação Cientifica\Relatório Parcial Fapesp\Fundo Verdade\44.jpg | C:\Users\RODOLFO\USP\Íniciação Cientifica\Relatório Parcial Fapesp\Fundo Verdade\fv_44.jpg |

Figura 13 - Exemplo de fotos utilizadas no banco de dados e seus respectivos fundo verdade.

## 3.2.6 – ALGORITMO GENÉTICO

A técnica de algoritmo genético é um modelo computacional iterativo inspirado no princípio de seleção natural e reprodução genética para encontrar soluções para um determinado problema, através da evolução das populações. Essa técnica esta sendo muito utilizada nos últimos anos por causa da sua flexibilidade e versatilidade na resolução de problemas de otimização, podendo ser utilizado, por exemplo, em diversas áreas da Engenharia, em soluções para Biologia, Medicina e Teoria de Controle (Pearson et al., 2003). Abaixo, é apresentado um fluxograma () com o funcionamento básico de um algoritmo genético que pode ser visto mais detalhadamente em Pacheco (1999).

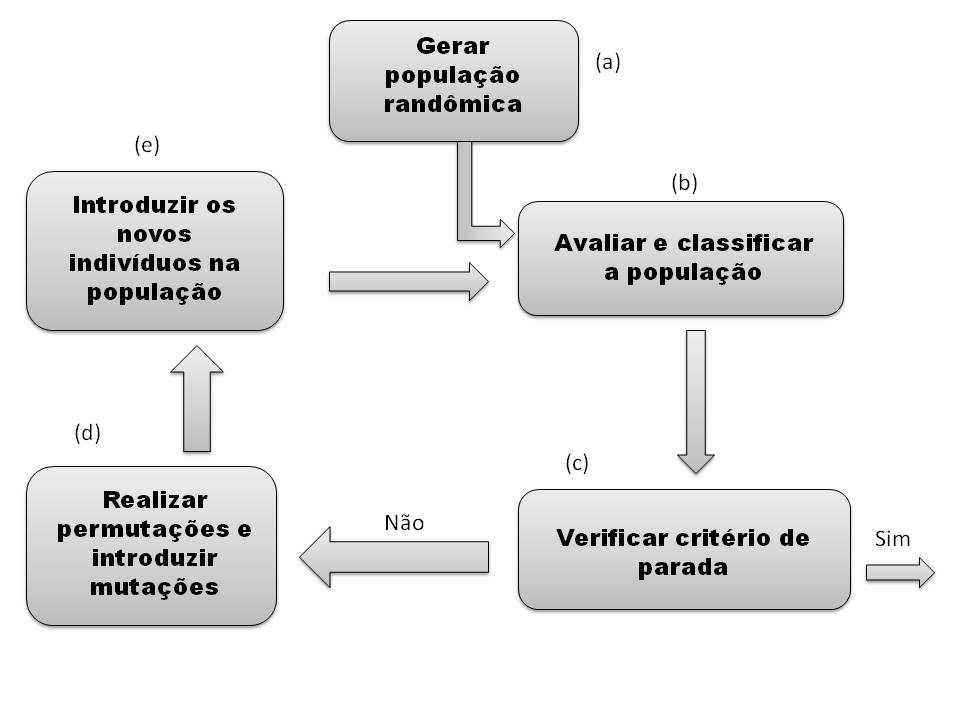


Figura 14 - Esquema simplificado de um Algoritmo Genético.

Pela , tem-se que:

1. Gera-se uma população inicial randômica que possuam os parâmetros a serem otimizados.
2. Em seguida, deve-se avaliar cada indivíduo tendo por base o fundo verdade, e realizar a classificação do desempenho destes, selecionando os que obtiveram melhor desempenho.
3. Verifica-se então se a condição de otimização foi alcançada.
4. Se não foi alcançada, deve-se realizar permutações entre os indivíduos e introduzir pequenas mutações em seus “genes”, mantendo inalterado apenas o melhor indivíduo da geração.
5. Finalmente, os novos indivíduos são introduzidos na população, ocupando o lugar de seus geradores.

As rotinas de “b” a “e” serão repetidas até que uma condição de parada seja satisfeita.

Com essa técnica espera-se que haja um refinamento nos parâmetros utilizados nos detectores de borda, a fim de fazer uma detecção que priorize as faixas de trânsito. É importante frisar que os parâmetros a serem utilizados irão depender das condições ambientais, assim as detecções noturnas, por exemplo, irão ter parâmetros diferentes das detecções para dias claros. Essa identificação poderá ser feita por um sensor de iluminação e através desse dado o sistema de controle do carro irá escolher o grupo de parâmetros adequado para fazer uma melhor detecção.

Nessa fase do projeto o aluno teve contato com as técnicas de algoritmo genético apenas com fins de familiarização. Esse assunto será estudado e implementado posteriormente, sendo aprofundado no próximo relatório.

## 3.3 – RESULTADOS

Todos os algoritmos explicitados acima foram desenvolvidos e testados em Matlab em fotos e vídeos captadas com o veículo de testes em ruas da cidade, rodovias e dentro do Campus da USP. Neste tópico serão comparados os algoritmos em testes iniciais realizados.

Primeiramente foram realizados os testes com detectores de borda, sendo os parâmetros modificados manualmente em busca de um melhor resultado. A e a mostram um comparativo entre os quatro detectores de borda utilizados.

Como pode-se observar nas imagens, todos os detectores apresentam bons resultados, porém cada um possui sua particularidade. Nos testes iniciais realizados pôde-se observar que o Operador de Canny obteve melhores detecções que os demais, entretanto o detector com máscara inclinada realça de maneira mais eficiente as bordas, o que pode ser muito importante no momento de fazer a identificação das faixas. Ainda será usado um algoritmo genético para otimizar todos essas rotinas na segunda parte da pesquisa, para então ser feita a comparação dos detectores e se verificar qual será utilizado de modo embarcado no veículo de testes.

|  |  |
| --- | --- |
| C:\Users\RODOLFO\USP\Íniciação Cientifica\Relatório Parcial Fapesp\resultados\2.jpg | |
| (a) | |
| (b) | (c) |
| (d) | (e) |

Figura 15 - Comparativo entre detectores de borda: (a) Imagem original. (b) Operador de Canny. (c) Operador de Sobel. (d) Operador Laplaciano da Gaussiana. (e) Detector com máscara inclinada.

|  |  |
| --- | --- |
| C:\Users\RODOLFO\USP\Íniciação Cientifica\Relatório Parcial Fapesp\resultados\4.jpg | |
| (a) | |
| (b) | (c) |
| (d) | (e) |

Figura 16 - Comparativo entre detectores de borda. (a) Imagem original. (b) Operador de Canny. (c) Operador de Sobel. (d) Operador Laplaciano da Gaussiana. (e) Detector com máscara inclinada.

## 3.4 – CONCLUSÕES

Nesta primeira parte do projeto o aluno pôde ter contato com o conceito de visão computacional, em especial com métodos de detecção de borda e Transformada de Hough, além de se familiarizar com a linguagem de programação Matlab, amplamente usado para cálculo numérico e trabalho com matrizes.

Os resultados iniciais se mostraram bons, obtendo uma grande quantidade de detecção de faixas apenas modificando-se os parâmetros manualmente. Para a segunda parte do projeto é esperado que os algoritmos já desenvolvidos sejam aprimorados com uma maior robustez através da aplicação de um algoritmo genético para encontrar os melhores parâmetros a serem usados.

# 4 – EVOLUÇÃO DO PROJETO

Este item visa comparar o cronograma e o plano de trabalho propostos com o atual estágio de desenvolvimento do projeto, citando os passos futuros. As tarefas inicialmente propostas, o cronograma proposto de execução (), bem como a evolução do projeto até o atual momento são apresentados a seguir:

# 1 – Revisão Bibliográfica:

A revisão bibliográfica, que tem por objetivo levantar informações sobre o problema estudado e dos métodos utilizados, está sendo realizada constantemente desde o início do projeto e continuará sendo efetuado. Em execução.

# 2 – Familiarização com a técnica de Algoritmo Genético e detecção de bordas:

Essas técnicas foram brevemente estudadas e testes foram realizados utilizando-as, tendo por finalidade conhecer os métodos que seriam posteriormente aplicados no projeto, assim, realizando uma atividade paralela aos estudos teóricos de modo a ter um maior aprendizado. Atividade concluída.

# 3 – Aquisição de dados e imagens para posterior processamento:

Com a câmera instalada na plataforma de testes, foram realizados passeios e assim, obtiveram-se diversas filmagens e fotos de vias em diversas condições climáticas e qualidades de pista. Os passeios foram realizados no interior do Campus, em ruas urbanas e em rodovias. Atividade concluída.

# 4 – Geração dos fundos verdade:

De posse das imagens, algumas foram selecionadas para montar um banco de dados juntamente com fundo verdade, a fim de avaliar a qualidade dos algoritmos e de ser utilizado como “gabarito” para o algoritmo genético. Atividade concluída.

# 5 – Desenvolvimento e implementação dos algoritmos:

As rotinas utilizadas para fazer a detecção de bordas e a Transformada de Hough foram desenvolvidos, as rotinas utilizando IPM estão em progresso e ainda necessitam de algumas modificações e o AG está começando a ser desenvolvido. Em execução.

# 6 – Testes pós-processsados dos algoritmos:

Alguns testes iniciais foram feitos utilizando-se diferentes detectores de borda, Transformada de Hough e IPM. A partir de agora começarão a ser feitos mais testes para ter uma melhor avaliação dos algoritmos, principalmente após o desenvolvimento e aplicação do AG. Em execução.

# 7 – Elaboração de relatório parcial e final:

Este relatório é o primeiro a ser entregue à FAPESP. Ao final do projeto, mais um relatório será enviado. Primeira parte concluída.

# 8 – Redação de trabalhos e publicações:

Um artigo foi escrito e submetido para o XIX Congresso Brasileiro de Automática, espera-se que em breve saia a avaliação do trabalho. Primeira parte concluída.

Tabela 7 - Cronograma de execução baseado nas tarefas listadas acima.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | dez/11 | jan/12 | fev/12 | mar/12 | abr/12 | mai/12 | jun/12 | jul/12 | ago/12 | set/12 | out/12 | nov/12 | Andamento |
| #1 |  | | | | | | | | |  |  |  | Em execução |
|  | | | | | | | | | | |  |
| #2 |  | | |  |  |  |  |  |  |  |  |  | Concluído |
|  | | | |  |  |  |  |  |  |  |  |
| #3 |  |  | |  |  |  |  |  |  |  |  |  | Concluído |
|  |  | | |  |  |  |  |  |  |  |  |
| #4 |  |  | |  |  |  |  |  |  |  |  |  | Concluído |
|  |  | | |  |  |  |  |  |  |  |  |
| #5 |  |  |  |  | | | | | | | |  | Em execução |
|  |  |  |  | | | | | | | | |
| #6 |  |  |  |  |  |  | | | | | |  | Em execução |
|  |  |  |  |  |  |  |  | | | | |
| #7 |  |  |  |  |  | |  |  |  |  |  | | Primeira parte concluída |
|  |  |  |  |  |  |  |  |  |  |  |  |
| #8 |  |  |  |  |  | |  |  |  |  |  | | Primeira parte concluída |
|  |  |  |  |  |  |  |  |  |  |  |  |

# 5 – REFERÊNCIAS

ACKERMAN, E. (2010). “Google's Autonomous Car Takes To The Streets”. Disponível em: <http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/googles-autonomous-car-takes-to-the-streets>. Acesso em: 27 abr. 2012.

ASSIDIQ, A. A. M.; KHALIFA, O. O.; ISLAM, R.; KHAN, S. (2008). "Real Time Lane Detection for Autonomous Vehicles". Computer and Communication Engineering International Conference, pp.82-88.

CANNY, J. (1986). "A Computational Approach to Edge Detection". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.PAMI-8, No.6, pp.679-698.

FENABRAVE. (2010). “Distribuição de Veículos Automotores no Brasil”. Disponível em: <http://www.fenabrave.org.br/principal/pub/Image/20110614100628anuario2010.pdf >. Acesso em: 27 fev. 2011.

FOLHA. (2010). “Indústria automotiva investirá US$ 11,2 bilhões até 2012, diz Anfavea”. Disponível em: <<http://www1.folha.uol.com.br/folha/dinheiro/ult91u728367.shtml>>. Acesso em: 27 abr. 2012.

FORD. (2011). “Lane Keeping System Helps Fusion Drivers Stay Alert and Between the Lines”. Disponível em: <http://media.ford.com/article\_display.cfm?article\_id=35776>. Acesso em: 27 abr. 2012.

GONZALEZ, R. C.; WOODS, R. E. (2007). “Digital Image Processing”. Pearson Prentice Hall, 3th. Edition.

KHALIFA, O. O. et al. (2010). “A Hyperbola-Pair Based Lane Detection System for Vehicle Guidance”. Proceedings of the World Congress on Engineering and Computer Science, São Francisco, Vol. 1, pp.585-588.

KOWALENKO, K. (2010). “Keeping Cars from Crashing”. The Institute IEEE, v.34, n.3, p.5.

KURDZIEL, M. S. (2008). “A Monocular Color Vision System for Road Intersection Detection”. Rochester Institute of Technology.

MATHWORKS. (2012). “Rgb2gray”. Disponível em: <<http://www.mathworks.com/help/toolbox/images/ref/rgb2gray.html>>. Acesso em: 27 abr. 2012.

MATSUMOTO, E. Y. (2008). “MATLAB 7: Fundamentos”. 2. ed. São Paulo: Érica, 2008. 376 p.

NIETO, M.; SALGADO, L.; JAUREGUIZAR, F.; ARROSPIDE, J. (2008) "Robust Multiple Lane Road Modeling Based on Perspective Analysis" Image Processing, 15th IEEE International Conference, pp.2396-2399, ICIP 12-15 Oct. 2008

NIETO, M.; SALGADO, L.; JAUREGUIZAR, F.; CABRERA, J. (2007). “Stabilization of Inverse Perspective Mapping Images based on Robust Vanishing Point Estimation”. IEEE Intelligent Vehicles Symposium, Turkey, pp. 315-320.

OMS. (2009). “Global Status Report on Road Safety: Time for Action”. Technical report, Organização Mundial da Saúde.

ONUBR. (2011). “Década de ação pelo trânsito seguro 2011-2020 é lançada oficialmente hoje (11) em todo mundo”. Disponível em: <<http://www.onu.org.br/decada-de-acao-pelo-transito-seguro-2011-2020-e-lancada-oficialmente-hoje-11-em-todo-o-mundo/>> Acesso em: 27 abr. 2012.

PACHECO, M. A. C. (1999). “Algoritmos Genéticos: Princípios e Aplicações”. ICA, Laboratório de Inteligência Computacional Aplicada.

PEARSON, D. W.; STEELE, N. C.; ALBRECHT, R. F. (2003). “Artificial Neural Nets and Genetic Algorithms”. New York: Springer Verlag Wien, 271 p.

SENA. “Sistema Embarcado De Navegação Autônoma”. Disponível em: <<http://www.eesc.usp.br/sena/url/pt/index.php>>. Acesso em: 27 abr. 2012.

SILVA, J. F. C.; BARBOSA, R. L.; GALLIS, R. B. A. (2004). “Avaliação Da Qualidade Da Detecção De Bordas Em Uma Sequência De Imagens De Ruas E Rodovias”. Revista Brasileira de Cartografia, No. 56, pp.96-103.

VALE, G. M.; DAL POZ, A. P. (2002). “O Processo de Detecção de Bordas de Canny: Fundamentos, Algoritmos e Avaliação Experimental”. Simpósio Brasileiro de Geomática, Anais, pp.292-303.

YING-DONG, Q.; CHENG-SONG, C.; SAN-BEN, C. (2005). “A Fast Subpixel Edge Detection Method Using Sobel-Zernike Moments Operator”. Image and Vision Computing, pp. 11–17