Universidade de São Paulo - USP

Escola de Engenharia de São Carlos - EESC

Departamento de Engenharia Mecânica - SEM

Grupo de Mecatrônica

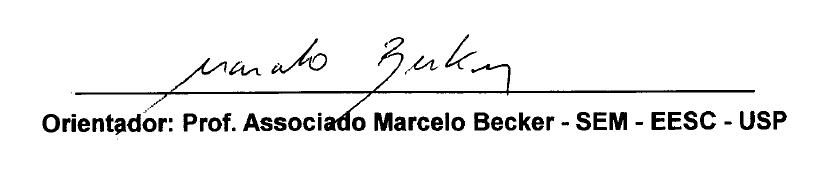
Laboratório de Robótica Móvel - LabRoM

Relatório Final 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**

****

Dezembro / 2013

Conteúdo

[1 – RESUMO DO PLANO INICIAl 3](#_Toc374351004)

[2 – RESUMO DO TRABALHO 4](#_Toc374351005)

[3 – INTRODUÇÃO 5](#_Toc374351006)

[4 – METODOLOGIA 9](#_Toc374351007)

[4.1 – DETECTORES DE BORDA 10](#_Toc374351008)

[4.1.1 – OPERADOR DE CANNY 11](#_Toc374351009)

[4.1.2 – LAPLACIANO DA GAUSSIANA (LoG) 13](#_Toc374351010)

[4.1.3 – OPERADOR DE SOBEL 15](#_Toc374351011)

[4.1.4 – DETECTOR COM MÁSCARA INCLINADA 16](#_Toc374351012)

[4.2 – TRANSFORMADA DE HOUGH 17](#_Toc374351013)

[4.2.1 – SOBREPOSIÇÃO DE IMAGENS 22](#_Toc374351014)

[4.3 – IPM COM HISTOGRAMAS 24](#_Toc374351015)

[4.5 – CRIAÇÃO DO FUNDO VERDADE 28](#_Toc374351016)

[4.6 – ALGORITMO GENÉTICO 30](#_Toc374351017)

[4.6.1 – PROBLEMA A SER OTIMIZADO 31](#_Toc374351018)

[4.6.2 – REPRESENTAÇÃO DAS SOLUÇÕES DO PROBLEMA 31](#_Toc374351019)

[4.6.3 – DECODIFICAÇÃO DOS CROMOSSOMOS 33](#_Toc374351020)

[4.6.4 – AVALIAÇÃO 33](#_Toc374351021)

[4.6.5 – SELEÇÃO 35](#_Toc374351022)

[4.6.6 – OPERADORES GENÉTICOS 36](#_Toc374351023)

[4.6.7 – INICIALIZAÇÃO DA POPULAÇÃO 36](#_Toc374351024)

[4.6.8 – PARÂMETROS E CRITÉRIOS DE PARADA 37](#_Toc374351025)

[5 – RESULTADOS 39](#_Toc374351026)

[6 - CONCLUSÃO 51](#_Toc374351027)

[7 – EVOLUÇÃO DO PROJETO 52](#_Toc374351028)

[8 – REFERÊNCIAS 55](#_Toc374351029)

# 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 a finalização 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 financiou 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. Todos esses métodos tiveram seus parâmetros otimizados com o auxílio de um algoritmo genético para se obter uma melhor detecção das faixas.

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 – 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 março de 2010 foi lançada pela ONU (Organização das Nações Unidas) com parceria da Organização Mundial da Saúde (OMS) a década de ação pelo trânsito seguro (2011 – 2020) [ONUBR, 2011], com o objetivo de estabilizar e reduzir as fatalidades no trânsito salvando a vida de aproximadamente cinco milhões de vidas nesse período [OMS, 2013]. A OMS também classificou 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 1,24 milhões de pessoas por ano, sendo previsto que até o ano de 2030 essa possa se tornar a quinta maior causa de mortes [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 [OMS, 2009]. Pode-se observar na Tabela 1 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 partir do modelo Fusion 2013, é oferecido um sistema opcional que consiste em uma câmera instalada próximo ao espelho retrovisor e que tem 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 assistivos embarcados já existe há 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. Isto gerou um grande desenvolvimento nesse tipo de tecnologia. Mais recentemente, novos carros começaram a serem desenvolvidos, 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 (LabRoM) 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 (assistida) 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, 2013]. Para seu funcionamento o veículo possui diversos sensores: um sensor LIDAR Velodyne HDL – 32E, dois sensores LIDAR ibeo LUX, uma câmera AVT – Stingray, uma IMU IG – 500N, um GPS – RTK da Septentrio e dois odômetros. Na Figura 1 é possível observar a posição dos sensores e na Figura 2 tem-se uma foto do veículo de testes.

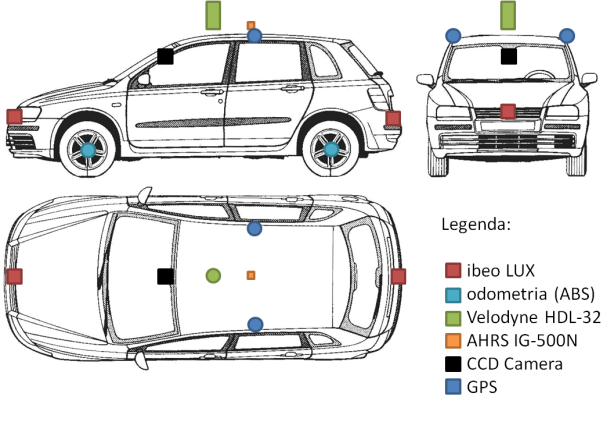


Figura 1 - Ilustração dos sensores embarcados no veículo de testes



Figura 2 – Veículo de testes do projeto SENA

# 4 – 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 que foram utilizados para fazer o 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 *frames* de vídeos obtidos durante “passeios” do veículo de teste para aquisição de dados. As fotos tiverem seu tamanho pré-fixado em 640x480, pois essa é a resolução ideal, já que imagens maiores teriam o tempo de processamento muito alto, e fotos com resolução menor deixam a desejar no detalhamento da imagem [Khalifa, 2010].

## 4.1 – 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 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, 2008].



Imagens

Perfil de uma linha horizontal

Primeira Derivada

Segunda Derivada

Figura 3 – Esquema simplificado de identificação de bordas

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 3.

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

### 4.1.1 – 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, 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 [Silva, 2004] [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.

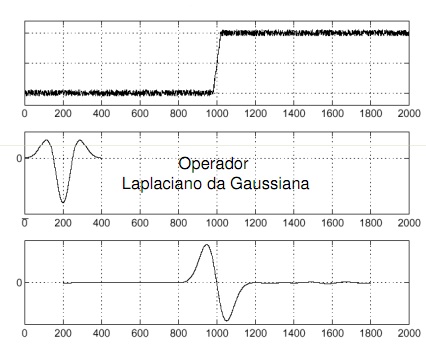
Ao fim desse processo a imagem estará binarizada e deverá apresentar apenas as bordas de maior intensidade.

### 4.1.2 – 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” (em inglês *blur*) *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 Figura 4.



Sinal

Função

Convolução

Figura 4 - Funcionamento do operador Laplaciano da Gaussiana

### 

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.

### 4.1.3 – OPERADOR DE SOBEL

O operador 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.

### 4.1.4 – 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 [Gonzalez e Woods, 2007].

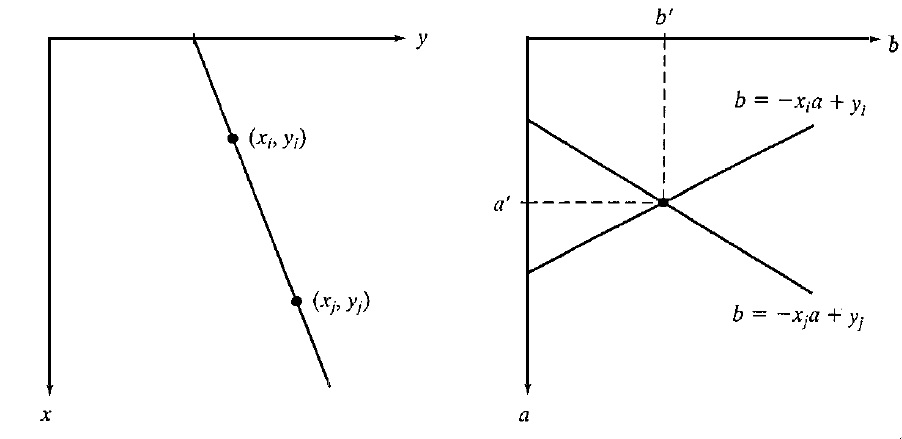
|  |  |
| --- | --- |
| Tabela 5 - Máscara de 45° | Tabela 6 - Máscara de -45° |
| |  |  |  | | --- | --- | --- | | -2 | -1 | 0 | | 1 | 0 | 1 | | 0 | 1 | 2 | | | |  |  |  | | --- | --- | --- | | 0 | -1 | -2 | | 1 | 0 | -1 | | 2 | 1 | 0 | |

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.

## 4.2 – TRANSFORMADA DE HOUGH

Depois de fazer a identificação de todas as bordas através dos detectores apresentados, têm-se imagens binarizadas 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 [Gonzalez e Woods, 2007]. Com isso tem-se uma pequena amostra da transformada, como pode ser observado na Figura 5.

**Figura 5 - 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 tem-se 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 Figura 6 um exemplo dessa transformada:

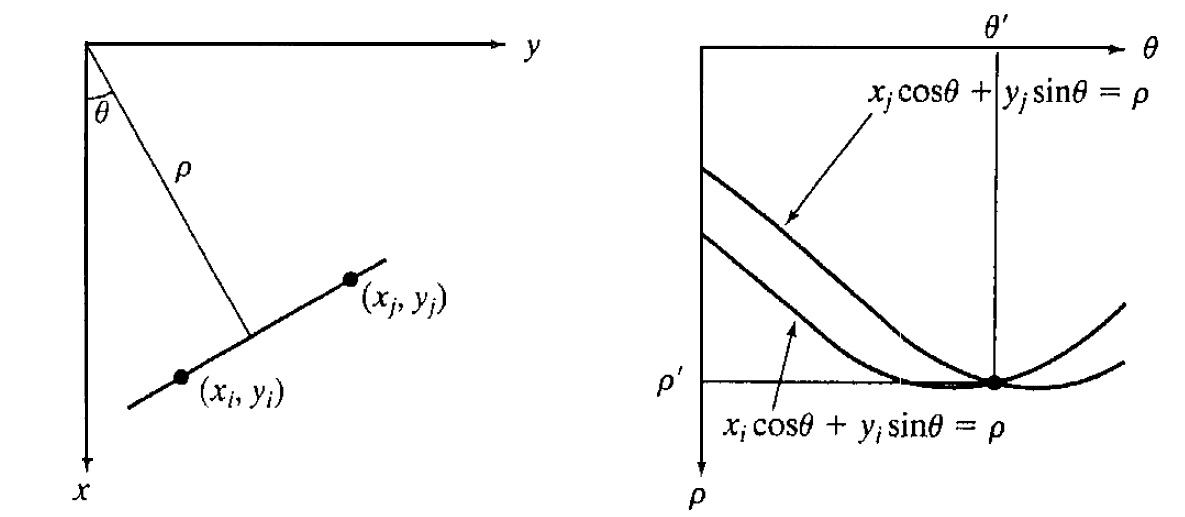


Figura 6 - Exemplo de transformada do espaço cartesiano para o espaço dos parâmetros senoidais (Gonzalez e Woods, 2007).

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.

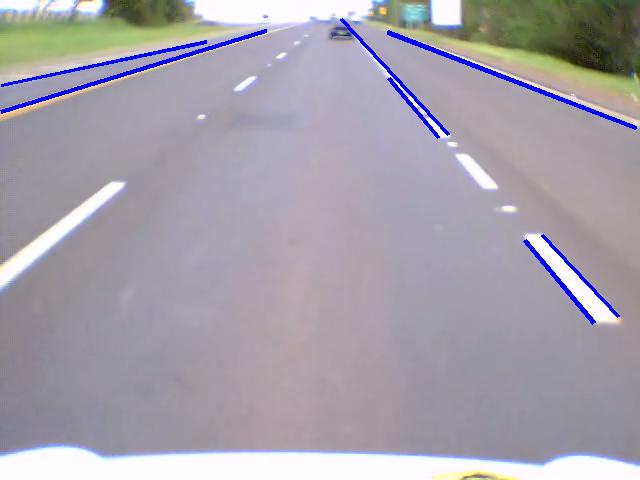
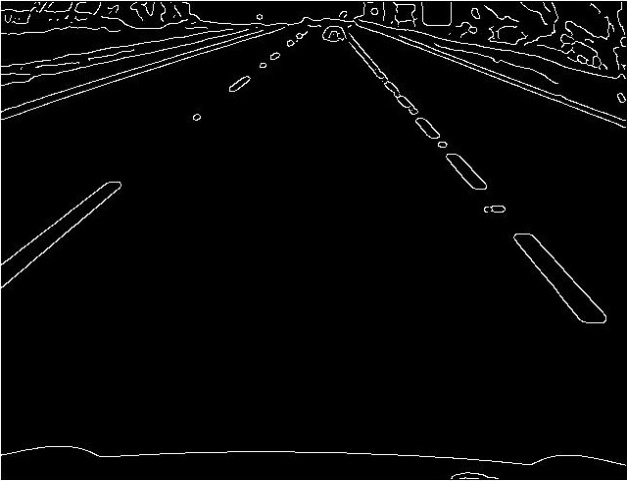


Figura 7 – Imagem após aplicação do operador de Canny e foto original com a transformada de Hough.

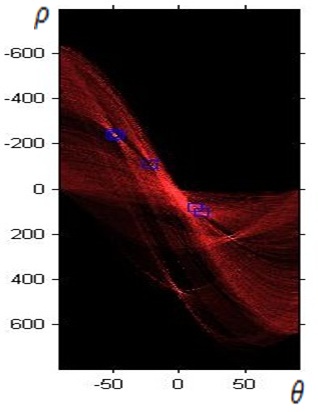


Figura 8 - Espaço dos parâmetros da Transformada de Hough

Na Figura 8 acima, pode-se observar a transformada de Hough da Figura 7 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 Figura 7 com transformada de Hough tem-se a representação dessas retas em azul sobre a foto original.

Existem alguns parâmetros que podem ser modificados na Transformada de Hough para que o algoritmo se enquadre melhor no que se necessita, pode-se destacar três principais parâmetros: *numpeaks, FillGap* e *MinLength*.

* *Numpeaks* está relacionado ao número de picos de intersecção das senóides que estamos interessados, ou seja, qual a quantidade máxima de retas que se deseja encontrar com esse método. Por padrão esse número é de apenas um, porém como se deseja encontrar diversas faixas optou-se por trabalhar com um máximo de 15 picos. Esse número foi obtido após diversos testes em diversas condições e se mostrou suficiente para encontrar as retas de interesse e não muito excessivo a ponto de aumentar a quantidade de retas não desejadas encontradas.
* *FillGap* é um número escalar positivo que especifica a distância entre dois segmentos de retas associados com um mesmo pico. Quando a distância entre os dois segmentos é menor que o número especificado o algoritmo transforma as duas linhas em apenas um único segmento de reta. Decidiu-se definir que essa distância como 30 *pixels,* pois é um valor que torna possível a aglutinação de algumas faixas descontínuas aumentando a probabilidade de identificação de tais faixas.
* *MinLength* é um número escalar positivo que especifica se uma determinada linha deve ser descartada ou não. Segmentos de reta menores que o número especificado são descartados por serem muito pequenos e, logo, serem provavelmente originarias de ruídos da imagem. Esse parâmetro foi definido como 60 *pixels* após a realização de testes onde pode se verificar que com esse valor existia uma valorização das bordas em detrimento dos ruídos.

### 4.2.1 – SOBREPOSIÇÃO DE IMAGENS

Pode-se observar na Figura 7 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 por Nieto [Nieto et. al, 2008]. 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 *pixel* a *pixel* dos quadros que se deseja sobrepor, em cada comparação é obtido o *pixel* 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 tem-se 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 Figura 9 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 9 - 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

Um problema desse método é que ele exige que as faixas se mantenham em posição constante nas imagens para que ele seja eficiente, em situações como curvas ou movimentação lateral do carro observa-se que ocorre uma multiplicação dos pontos de interesse da imagem, aumentando a incerteza da localização da faixa e os erros inerentes disso. Fato parecido ocorre com obstáculos nas vias como, por exemplo, outros carros que causam um borramento na foto dificultando a identificação das faixas e aumentando a taxa de erros.

## 4.3 – 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 mostrado por Kurdziel [Kurdziel, 2008]. Para fazer essa transformação na imagem o algoritmo utiliza parâmetros internos e externos da câmera que devem ser previamente conhecidos. A Figura 10 abaixo representa como a mudança da perspectiva modifica a imagem:

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

Figura 10 - Representação da modificação gerada na imagem pelo método IPM. À esquerda tem-se 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 *pixel* com a sua nova posição na imagem sem perspectiva. Assim, basta percorrer toda a imagem trocando os *pixel*s de lugar para se obter uma nova figura com visão aérea. Para se construir essa tabela de conversão de uma imagem para uma imagem utilizam-se as 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 obtida uma tabela que relacione a posição de cada *pixel* da imagem original com sua nova posição na imagem sem o efeito da perspectiva para utilizar nas fotos de rua adquiridas com o veículo de testes. Na Figura 11 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 11 - Exemplo da utilização de transformada de IPM.

Como pode-se visualizar na Figura 11 que o processo de troca do lugar dos *pixels* não é perfeito já que a imagem aparece distorcida e com falta de informações. Observa-se também que esse comportamento piora à medida que os pixels se encontram mais afastados da câmera embarcada no veículo. 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 apresentar defeitos, a imagem fica muito boa a distâncias próximas do veículo e com a técnica de interpolação de linhas é possível suprir a falta de informações para *pixels* distantes. A Figura 12 apresenta um exemplo desse processo de interpolação onde se pode observar que a imagem fica completa, porém apresenta maior efeito de desfoque.

|  |  |
| --- | --- |
| 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 12 - Exemplo de interpolação da imagem para preenchimento de *pixels*.

Mas, deve-se frisar que 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. abaixo 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 13 - 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 [Nieto et al. 2007].

## 4.5 – 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 tanto durante o dia como no período da noite. 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, foram selecionadas no total 58 fotos para situações de boa luminosidade e 30 fotos para baixa luminosidade. A seguir são mostrados alguns exemplos de fotos utilizadas e seus respectivos fundo verdade.

|  |  |
| --- | --- |
| 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\Avell B153\USP\Íniciação Cientifica\Relatório Final Fapesp\Imagens\noite_orig.jpg | C:\Users\Avell B153\USP\Íniciação Cientifica\Visão Computacional\Testes\AG - novo\Canny\noturno\fv_26.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 14 - Exemplo de fotos utilizadas no banco de dados e seus respectivos fundo verdade.

## 4.6 – ALGORITMO GENÉTICO

Inspirado no princípio das evoluções das espécies proposto por Darwin, os algoritmos genéticos foram desenvolvidos por John Holland em 1975 [Goldberg, 1989]. Esse modelo computacional funciona de maneira iterativa com o objetivo de encontrar soluções para problemas, por meio da evolução das populações seguindo o princípio de seleção natural e reprodução genética. Essa técnica está sendo muito utilizada nos últimos anos por causa da flexibilidade na resolução de problemas de otimização, podendo ser utilizado, por exemplo, em diversas áreas da engenhariam, em soluções para Biologia, Medicina, Teoria de Controle e Mercado Financeiro [Pearson et al., 2003].

Existem várias maneiras de se desenvolver um AG, o que pode influenciar na qualidade e eficiência da resposta obtida. Segundo Pacheco [Pacheco, 1999] podemos caracterizar algoritmos genéticos através das seguintes componentes:

1. Problema a ser otimizado

2. Representação das Soluções de Problema

3. Decodificação dos Cromossomos

4. Avaliação

5. Seleção

6. Operadores Genéticos

7. Inicialização da População

8. Parâmetros e Critérios de Parada

A seguir será discutido como a otimização foi desenvolvida baseando-se nesses tópicos.

### 4.6.1 – PROBLEMA A SER OTIMIZADO

Os detectores de borda apresentados anteriormente são os problemas que queremos otimizar. Cada um deles possui diferentes parâmetros que podem ser otimizados e o que buscamos é aumentar a taxa de verdadeiros positivos e reduzir a quantidade de falso positivos.

Normalmente a técnica de AG é utilizada para resolver problemas complexos, com diversos parâmetros ou características combinadas. Nesse trabalho o foco é em problemas menos complexos, já que possuem menos incógnitas, tem-se para o operador de Canny três parâmetros de otimização, dois para o LoG e apenas um para o operador de Sobel e o de máscara inclinada. Por se tratar de problemas mais simples seria esperado que o programa encontrasse de maneira rápida e eficiente uma resposta adequada ao sistema.

### 4.6.2 – REPRESENTAÇÃO DAS SOLUÇÕES DO PROBLEMA

Nesse tópico buscou-se definir a estrutura de cada cromossomo a ser manipulado pelo algoritmo e o espaço de busca das possíveis soluções.

Começou-se a análise pelo algoritmo de Canny, o primeiro parâmetro é o σ que possui influência no processo de filtro da imagem, durante testes preliminares foi notado que para um valor superior a cinco a imagem começa a ficar muito desfocada e quase não era possível fazer a localização das bordas. Com a intenção de abranger uma maior quantidade de resultados possíveis foi escolhido que seriam aceitos valores de σ de zero até um valor máximo a dez, pois acima disso se tornava quase impossível a identificação de qualquer borda. Os valores de *Threshold* estão relacionados à intensidade dos pontos de borda e são definidos entre zero e um, sendo necessário, por definição do programa, que T2 seja maior que T1.

Para o Laplaciano da Gaussiana foi realizado uma análise semelhante ao exposto acima. Esse detector possui um parâmetro relacionado ao desvio-padrão do filtro gaussiano (σ) que pelas mesmas razões pode variar de um valor mínimo de zero até um máximo igual a dez. A outra incógnita é o valor de *threshold* que é definido entre zero e um.

Tanto o operador de Sobel quanto o detector com máscara inclinada possuem apenas um parâmetro a ser otimizado, esse valor de limiar é também definido entre zero e um.

Assim, para iniciar a primeira geração do AG foram gerados de maneira randômica dez cromossomos diferentes para cada parâmetro citado acima. Os valores máximos e mínimos para cada detector podem ser observados na tabela abaixo:

Tabela 7 - Valores máximos e mínimos para os parâmetros

|  |  |  |  |
| --- | --- | --- | --- |
| Operador | Parâmetro | Mínimo | Máximo |
| Canny | σ | 0 | 10 |
| T1 | 0 | 1 |
| T2 | T1 | 1 |
| LoG | σ | 0 | 10 |
| *Threshold* | 0 | 1 |
| Sobel | *Threshold* | 0 | 1 |
| Máscara Inclinada | *Threshold* | 0 | 1 |

### 4.6.3 – DECODIFICAÇÃO DOS CROMOSSOMOS

Essa etapa consiste basicamente na construção da solução real do problema a partir dos cromossomos, para que depois possa ser feita uma avaliação.

Foi-se escolhido escrever diversos programas ao invés de um único com todos os algoritmos, assim foram desenvolvidos oito programas de otimização, pois para cada detector, já que as imagens foram separadas entre “bem iluminadas” (imagens capturadas em dias ensolarados) e com “pouca iluminação” (durante o período noturno), já que nesses dois casos os parâmetros possuem valores ótimos muito diferentes.

### 4.6.4 – AVALIAÇÃO

Para fazer a avaliação, uma imagem era gerada para cada indivíduo e então comparada com o fundo verdade para obter sua pontuação, baseado na quantidade de verdadeiros positivos e falsos positivos. A quantidade de verdadeiros negativos e falsos negativos não foi considerada para a otimização já que a ideia não é que o detector encontre toda a faixa, mas apenas as bordas dessa. Para a avaliação foram desconsideradas as regiões inferior e superior da imagem, por conterem muitos ruídos e não serem regiões de interesse. A imagem a seguir mostra como era cortada a imagem inicial para se fazer a análise.



Figura - Linhas em vermelho representando o corte da imagem

Nas fotos diurnas, para cada *pixel* de borda encontrado que se enquadrava dentro da região estipulada pelo gabarito, o cromossomo ganhava um ponto e para cada *pixel* fora da região estipulada era decrementado um ponto. Para as fotos noturnas esse valor foi alterado, pois a detecção se torna mais difícil e para que o programa tenha uma boa resposta foi necessário estipular acréscimo de três pontos para verdadeiros positivos e uma punição de um ponto para falsos positivos. Na imagem abaixo pode-se observar em azul os pontos que corretamente foram classificados como faixa e, em vermelho, os que foram encontrados fora da região desejada, as bordas são referentes à foto apresentada na Figura 15.

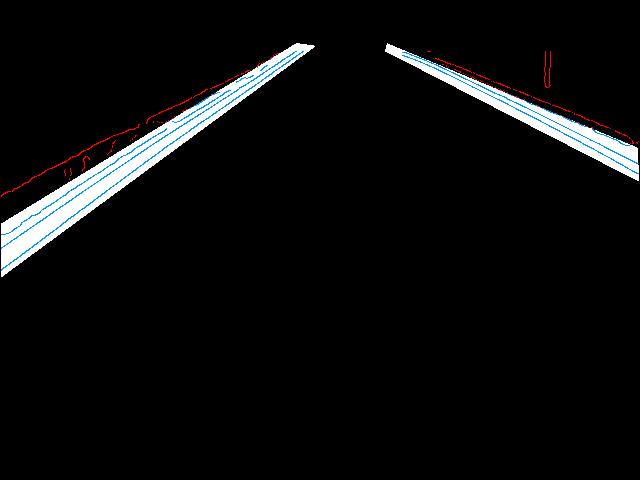


Figura 16 – Pontos de borda encontrados plotados sobre o fundo verdade

### 4.6.5 – SELEÇÃO

O processo de seleção em algoritmos genéticos define quais indivíduos irão se reproduzir e é esperado que os mais aptos passem parte de seus genes para as próximas gerações de modo a evoluir a resposta do algoritmo. Existem diversas formas de selecionar os indivíduos e determinar de que modo será a reprodução deles, no caso desse problema foi escolhido fazer uma classificação para definir qual gene apresenta a melhor resposta de sua geração, esse é definido como o mais apto e se reproduz com todos os outros indivíduos.

### 4.6.6 – OPERADORES GENÉTICOS

Os indivíduos da geração anterior são recombinados através de um operador de *crossover*, que são submetidos a uma operação de mutação com o objetivo de aumentar a diversidade da população. Essa é uma característica essencial do algoritmo genético, e é um dos principais responsáveis pela convergência do programa em uma boa resposta. Para se aplicar a mutação era acrescentado ou decrementado de maneira randômica um valor para o *treshhold* e para σ de no máximo 5%, para que a variação não seja tão grande.

### 4.6.7 – INICIALIZAÇÃO DA POPULAÇÃO

Esse processo determina como será a criação dos indivíduos para o primeiro ciclo do algoritmo. Essa inicialização normalmente ocorre de modo randômico, mas podem ser semeados genes com bons cromossomos quando se conhece aproximadamente o valor esperado para o resultado.

Nesse trabalho a inicialização foi feita de maneira randômica e para as rodadas sucessivas a população era gerada através do modo já descrito, entretanto o individuo de maior pontuação era passado pra a próxima geração sem sofrer nenhuma mutação, isso foi feito para garantir que o melhor resultado de cada geração não fosse perdido com as mutações e a pontuação máxima nunca fosse menor do que o valor já encontrado.

Também foi usada uma técnica de extermínio da população durante o processo, a ideia desse procedimento é matar todos os indivíduos, menos o melhor, e gerar uma nova população de maneira randômica com o objetivo de diversificar mais os indivíduos e buscar uma resposta que pode ser melhor do que a encontrada anteriormente. Essa técnica é usada quando o algoritmo já chegou a uma convergência, então é praticamente como se fazer um segundo algoritmo genético, este tendo um dos indivíduos iniciais com o gene já determinado, sendo assim uma segunda forma de inicialização do programa.

### 4.6.8 – PARÂMETROS E CRITÉRIOS DE PARADA

Os dois parâmetros geralmente usados como critério de parada são o número de gerações ou o total de indivíduos, mas também pode ser usado um critério de convergência da população, nesse caso se avalia a diferença entre o melhor resultado de gerações sucessivas e a parada ocorre quando o valor da diferença for menor que o estipulado previamente. A princípio, foi escolhido para esse problema um critério de parada de número de gerações iguais a cem, posteriormente foi observado que a convergência acontecia muito mais rapidamente, após a trigésima rodada não havia mais alteração perceptível e, portanto, para haver uma melhor eficiência no quesito resultado versus gasto computacional foi definido como critério de parada um total de cinquenta ciclos.

Com todos esses tópicos supracitados realizados tem-se o algoritmo genético pronto para ser utilizado na otimização do problema e os resultados serão discutidos mais adiante. A seguir pode-se observar um esquema reduzido com o funcionamento básico do AG.



Figura 17 - Esquema simplificado de um algoritmo genético

# 5 – RESULTADOS

As algoritmos expostos acima foram implementados em Matlab e testados de maneira pós-processada em imagens retiradas de vídeos captados com o veículo de testes em ruas da cidade, rodovias e dentro do Campus da USP, tanto no período diurno quanto no período noturno.

O algoritmo genético foi desenvolvido conforme os oito passos descritos por Pacheco [Pacheco, 1999] e aplicado nos quatro detectores de borda, de modo a obter parâmetros ótimos para a detecção de faixas. Na Tabela 8 estão expostos os valores encontrados para tais parâmetros.

Tabela 8 - Valores dos parâmetros otimizados

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Detector** | **Período** | **Threshold** | **σ** | **T1** | **T2** |
| **Canny** | Diurno | - | 1,4601 | 0,2125 | 0,2612 |
| Noturno | - | 3,0848 | 0,0236 | 0,0735 |
| **LoG** | Diurno | 0,7092 | 0,3384 | - | - |
| Noturno | 0,4033 | 0,2124 | - | - |
| **Sobel** | Diurno | 0,0171 | - | - | - |
| Noturno | 0,0050 | - | - | - |
| **Inclinado** | Diurno | 0,0443 | - | - | - |
| Noturno | 0,0759 | - | - | - |

De posse de todos esses valores é possível fazer uma análise para identificar qual deles apresenta o melhor resultado. Um algoritmo foi desenvolvido para analisar os seguintes parâmetros e encontrar um vencedor:

* Quantidade média de pontos de borda encontradas por cada detector por foto
* Quantidade média de pontos de borda que estavam dentro da região dita como faixa (verdadeiro positivo)
* Quantidade média de pontos de borda que não estavam dentro da região dita como borda (falso positivo)
* Razão entre a quantidade média de verdadeiros positivos e o número médio de pontos de borda
* Razão entre a quantidade média de falsos positivos e o número médio de pontos de borda

A fim de obter uma análise mais detalhada, as fotos com boa iluminação foram classificadas em duas categorias: rodovia e rua urbana. Essa divisão foi feita porque é possível observar que a quantidade de obstáculos no ambiente urbano é muito maior, o que diminui a pontuação dos algoritmos.

Tabela 9 - Resultado para detecção no período diurno

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Diurno** | | | |
|  | Canny | Log | Sobel | Inclinado |
| Média de pontos de borda | 3397,1 | 2361,4 | 3484,3 | 3889,9 |
| Média de verdadeiros positivos | 1932,2 | 884,8 | 1652,9 | 1507,3 |
| Média de falsos positivos | 1465,0 | 1476,6 | 1831,5 | 2382,7 |
| Porcentagem verdadeiros positivos | 0,59% | 0,49% | 0,56% | 0,46% |
| Porcentagem falsos positivos | 0,41% | 0,51% | 0,44% | 0,54% |

Tabela 10 - Resultado para detecção em rodovias no período diurno

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Diurno – Rodovia** | | | |
|  | Canny | Log | Sobel | Inclinado |
| Média de pontos de borda | 2940,4 | 1083,7 | 2216,6 | 2300,0 |
| Média de verdadeiros positivos | 2408,8 | 856,8 | 1696,2 | 1510,4 |
| Média de falsos positivos | 531,6 | 226,9 | 520,5 | 789,6 |
| Porcentagem verdadeiros positivos | 0,82% | 0,76% | 0,78% | 0,65% |
| Porcentagem falsos positivos | 0,18% | 0,24% | 0,22% | 0,35% |

Tabela 11 - Resultado para detecção em ambiente urbano no período diurno

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Diurno – Rua** | | | |
|  | Canny | Log | Sobel | Inclinado |
| média pontos de borda | 3743,2 | 3329,4 | 4444,7 | 5094,5 |
| média verdadeiros positivos | 1571,1 | 906,1 | 1620,1 | 1504,9 |
| média falsos positivos | 2172,1 | 2423,2 | 2824,6 | 3589,5 |
| porcentagem verdadeiros positivos | 0,42% | 0,29% | 0,40% | 0,32% |
| porcentagem falsos positivos | 0,58% | 0,71% | 0,60% | 0,68% |

Tabela 12 - Resultado para detecção no período noturno

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Noturno** | | | |
|  | Canny | Log | Sobel | Inclinado |
| média pontos de borda | 1272,0 | 1741,6 | 1221,1 | 395,1 |
| média verdadeiros positivos | 1146,5 | 1392,7 | 1020,6 | 309,1 |
| média falsos positivos | 125,5 | 348,9 | 200,5 | 75,4 |
| porcentagem verdadeiros positivos | 0,91% | 0,79% | 0,85% | 0,92% |
| porcentagem falsos positivos | 0,09% | 0,21% | 0,15% | 0,08% |

A avaliação foi feita baseando-se nas fotos utilizadas para o desenvolvimento do AG com seus respectivos fundos verdade. Foram utilizadas 58 fotos com boa iluminação, sendo 33 em ambiente urbano e 25 em rodovias e 30 imagens com baixa iluminação.

O parâmetro mais importante na detecção é a porcentagem de verdadeiros positivos, é com esse valor que se pode saber qual a confiabilidade do algoritmo para a detecção de faixas. Pode-se observar que para esse quesito o operador de Canny obteve vantagem sobre os outros algoritmos em quase todas as situações. A única exceção é para imagens noturnas, em que o operador com máscara inclinada obteve um valor de maior eficiência, entretanto se for feita uma avaliação conjunta com a quantidade média de pontos encontrada para borda verifica-se que o algoritmo proposto por Canny encontrou mais que o triplo que o outro avaliado, isso significa que ele encontrou muito mais pontos com uma confiabilidade muito semelhante. Por esses motivos, é facilmente observável que o operador de Canny obteve o melhor resultado em todas as situações propostas e esse será o algoritmo utilizado para a continuidade do projeto.

Apesar de Canny ter sido o melhor detector, pode-se observar pela Tabela 11 que o resultado não é tão satisfatório para algumas situações. Para ambientes urbanos durante o dia apenas 42% dos pontos de borda encontrados pelo algoritmo são realmente pertencentes a faixas. Isso ocorre porque existem muito objetos que atrapalham a detecção, como: árvores, postes, placas de trânsito, carros, muros, entre outros. É importante ressaltar que isso não significa que essa porcentagem será a de retas classificadas como faixas, já que o classificador se utiliza de outros parâmetros como alinhamento e ângulo dos pontos encontrados para a definição das linhas classificadas como faixa. Mas com um resultado de alta confiabilidade como o encontrado para rodovias durante o dia e durante a noite, 82% e 91% respectivamente, é praticamente certo que as retas indicadas pelo classificador serão as que se deseja.

A seguir serão apresentadas algumas imagens para se ter uma comparação visual dos quatro métodos testados, as imagens estão cortadas como mostrado na Figura 15 para que seja observada apenas a região de interesse.

|  |  |
| --- | --- |
| C:\Users\Avell B153\USP\Íniciação Cientifica\Relatório Final Fapesp\Imagens\Comparação\estrada original.jpg | |
| (a) | |
| C:\Users\Avell B153\USP\Íniciação Cientifica\Relatório Final Fapesp\Imagens\Comparação\estrada canny.jpg(b) | C:\Users\Avell B153\USP\Íniciação Cientifica\Relatório Final Fapesp\Imagens\Comparação\estrada inclinado.jpg(c) |
|  |  |
| C:\Users\Avell B153\USP\Íniciação Cientifica\Relatório Final Fapesp\Imagens\Comparação\estrada log.jpg(d) | C:\Users\Avell B153\USP\Íniciação Cientifica\Relatório Final Fapesp\Imagens\Comparação\estrada sobel.jpg(e) |

Figura 18 - Comparativo entre os métodos de detecção para uma rodovia em situação de boa iluminação. (a) Imagem original; (b) Operador de Canny; (c) Operador com máscara inclinada; (d) Laplaciano da Gaussiana; (e) Operador de Sobel

|  |  |
| --- | --- |
| C:\Users\Avell B153\USP\Íniciação Cientifica\Relatório Final Fapesp\Imagens\Comparação\rua original.jpg | |
| (a) | |
| C:\Users\Avell B153\USP\Íniciação Cientifica\Relatório Final Fapesp\Imagens\Comparação\rua canny.jpg(b) | C:\Users\Avell B153\USP\Íniciação Cientifica\Relatório Final Fapesp\Imagens\Comparação\rua inclinado.jpg(c) |
|  |  |
| C:\Users\Avell B153\USP\Íniciação Cientifica\Relatório Final Fapesp\Imagens\Comparação\rua log.jpg(d) | C:\Users\Avell B153\USP\Íniciação Cientifica\Relatório Final Fapesp\Imagens\Comparação\rua sobel.jpg(e) |

Figura 19 - Comparativo entre os métodos de detecção para um ambiente urbano em situação de boa iluminação. (a) Imagem original; (b) Operador de Canny; (c) Operador com máscara inclinada; (d) Laplaciano da Gaussiana; (e) Operador de Sobel

|  |  |
| --- | --- |
| C:\Users\Avell B153\USP\Íniciação Cientifica\Relatório Final Fapesp\Imagens\Comparação\noite original.jpg | |
| (a) | |
| C:\Users\Avell B153\USP\Íniciação Cientifica\Relatório Final Fapesp\Imagens\Comparação\noite canny.jpg(b) | C:\Users\Avell B153\USP\Íniciação Cientifica\Relatório Final Fapesp\Imagens\Comparação\noite inclinado.jpg(c) |
|  |  |
| C:\Users\Avell B153\USP\Íniciação Cientifica\Relatório Final Fapesp\Imagens\Comparação\noite log.jpg(d) | C:\Users\Avell B153\USP\Íniciação Cientifica\Relatório Final Fapesp\Imagens\Comparação\noite sobel.jpg(e) |

Figura 20 - Comparativo entre os métodos de detecção para uma situação de baixa luminosidade. (a) Imagem original; (b) Operador de Canny; (c) Operador com máscara inclinada; (d) Laplaciano da Gaussiana; (e) Operador de Sobel

Com esses exemplos podemos ter uma melhor interpretação dos dados recolhidos em forma de pontuação. Para a imagem da rodovia, é muito perceptível que o operador de Canny fez a detecção das bordas de maneira quase perfeita, conseguindo destacar os pontos que temos mais interesse, e é possível observar como são importantes as etapas de supressão não-máxima e histerese provocando uma redução do ruído e uma continuidade da borda, coisas que não são muito observadas nos demais detectores.

Para a imagem no ambiente urbano é observado a quantidade de ruído presenta na imagem, árvores, gramado e principalmente sombras influenciam muito a detecção, esse é o principal motivo pela alta quantidade de falsos positivos detectado pelo algoritmo. Isso pode ser um grande problema quando o algoritmo funcionar no carro em tempo real, e é de grande importância a parte de classificação e cognição desses resultados para que não haja dano ao veículo.

Para a situação de baixa luminosidade todos os algoritmos funcionaram bem, com o direcionamento do farol do carro na frente do veículo as faixas mais próximas possuem um bom destaque e isso facilita a identificação. Mas é também fácil observar que o detector de Canny possui menos ruído e bordas mais evidentes.

É interessante notar que o corte superior e inferior foi de ampla importância, pois essa região possui muitos objetos que poderiam influenciar e aumentar a quantidade de falsos positivos, diminuindo assim a confiabilidade do algoritmo. Ainda assim, pode-se perceber que um corte lateral, principalmente se tratando de ambientes urbanos, poderia ajudar na diminuição dos erros, entretanto é extremamente difícil fazer essa determinação já que a característica da via pode se modificar e um corte poderia significaria perder parte importante da imagem. Uma opção para isso seria se o código fosse associado a um código de identificação da via, assim com a determinação do espaço geométrico da via a detecção de faixas poderia ter uma confiabilidade maior. Isso é algo plausível para esse projeto, pois existe outro aluno atualmente trabalhando com essa identificação da rota. A união de ambos os trabalhos será de grande benefício para o projeto como um todo.

Agora que se possuem as bordas pode-se tentar encontrar a posição geométrica das faixas. Foram testados dois métodos para fazer isso, a Transformada de Hough e a Transformada Inversa da Perspectiva. Após alguns testes o método de Hough foi definido como a operação padrão que seria usada. Isso aconteceu, pois o método de IPM depende de muitos parâmetros do carro e da câmera para que funcione, e esses são valores que mudam constantemente enquanto o carro se movimenta, qualquer tipo de trepidação ou mudança da angulação da via causa uma modificação dos parâmetros de modo complexo. Por essa razão, foi observado que para essa operação é necessário um trabalho muito focado e dispendioso de tempo e que não seria possível dentro do prazo desse projeto de iniciação científica fazer toda essa análise.

Sendo assim, foram realizados testes envolvendo a transformada de Hough aplicada na imagem binarizada extraída com o detector de Canny otimizado, os testes foram realizados tanto nas imagens usadas para o AG como em vídeos capturados com o veículo de teste, a fim de avaliar como seria a detecção ao longo de um caminho. Os resultados foram muito satisfatórios, encontrando com sucesso as faixas de trânsito durante a maioria do percurso, entretanto o algoritmo não conseguiu dar a resposta na mesma taxa de velocidade do vídeo. Isso não é de tanta importância, pois quando o algoritmo for implementado no projeto do carro autônomo o código terá de ser traduzido para a linguagem C++ e os computadores terão maior velocidade de processamento, assim é esperado que a resposta seja em tempo real com a aquisição de imagens.

Para melhorar o resultado da transformada de Hough foi proposto um parâmetro adicional aos já descritos na metodologia. Trata-se do ângulo dos segmentos de reta. A fim de diminuir a detecção de outras coisas que não sejam de interesse, foi desenvolvido uma técnica para evitar segmentos de reta que estivessem quase de modo horizontal na imagem. Para isso foi definido que o segmento deve possuir ao menos um ângulo de 15º a partir da horizontal. Esse valor foi obtido através de diversas experiências, em que foi possível observar que a grande maioria das faixas possuía uma inclinação maior que 20º e para evitar a perda de informações importantes decidiu-se por colocar um valor mínimo mais abrangente, no caso, 15º de angulação.

Feito isso, a seguir são apresentados alguns exemplos de resultados obtidos com essa modificação da técnica.

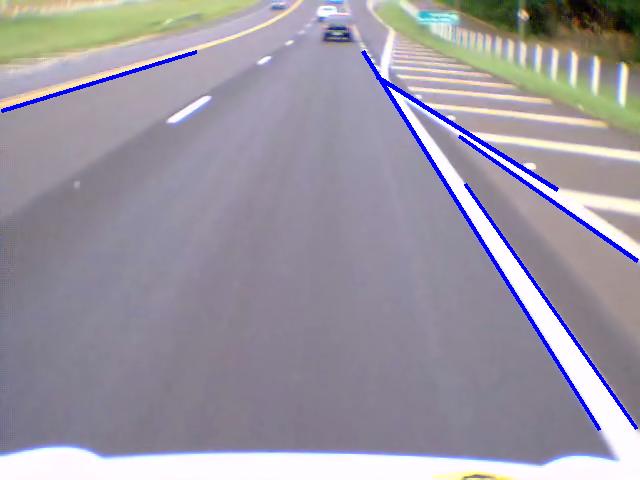


Figura - Exemplo Transformada de Hough



Figura - Exemplo Transformada de Hough



Figura - Exemplo Transformada de Hough



Figura - Exemplo Transformada de Hough

Os resultados apresentados da Figura 21 a Figura 24 em geral mostram que o algoritmo de Hough encontra de maneira eficiente as faixas desejadas, mas possuem alguns erros como no caso da Figura 23 em que a sombra atrapalha a detecção. Esse tipo de detecção é parte do resultado da baixa confiabilidade dos pontos de borda encontrados para o algoritmo no ambiente urbano, a Transformada de Hough diminui muito essa incerteza ao eliminar grande parte do ruído das imagens, mas ainda assim existem falsas detecções que são inerentes ao fato da detecção se basear na diferença entre a intensidade de nível de cinza, e como é possível observar na imagem a região de sombra possui um grande contraste com a parte iluminada.

Existia a opção de se trabalhar com a sobreposição de imagens antes de aplicar a Transformada de Hough, mas essa ideia acabou não se mostrando tão eficiente, pois qualquer tipo de objeto encontrado na pista era realçado, e no caso de uma movimentação lateral do carro a incerteza da localização da faixa aumentava muito. Assim, a Transformada de Hough aplicada a cada frame de modo isolado se mostrou a melhor opção para o desenvolvimento desse projeto.

# 6 - CONCLUSÃO

O trabalho foi concluído cumprindo todos os objetivos do plano inicial de trabalho. Pode-se ratificar a superioridade do operador de Canny sobre os outros detectores testados, e chegou-se à conclusão que a Transformada de Hough é o melhor método para fazer a descrição geométrica dos segmentos de retas que serão considerados como faixas ou sarjetas das ruas. Isso ocorreu principalmente pela maior robustez dessa operação sobre a técnica de IPM, que é muito suscetível a falhas de detecção devido a alterações nas condições de aquisição da imagem.

Pode-se concluir também que o algoritmo genético se mostrou eficaz em encontrar os valores de maior otimização para os detectores de borda, entretanto os resultados de falsos positivos continuaram altos em grande parte das situações analisadas. Sendo assim, pode-se dizer que a técnica de AG não foi a melhor escolha para encontrar a solução desse problema, já que grande parte do tempo dispendido na realização do projeto foi para o desenvolvimento dessa técnica e os resultados não foram tão bons quanto o esperado inicialmente, principalmente devido a problemas inerentes da detecção de bordas em imagens.

# 7 – EVOLUÇÃO DO PROJETO

Este item visa comparar o cronograma e o plano de trabalho propostos com o desenvolvimento final do projeto. As tarefas inicialmente propostas, o cronograma de execução (Tabela 13), 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 foi concluído e teve grande importância no desenvolvimento da metodologia adotada na execução do projeto. Atividade concluída

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

Essas técnicas foram intensamente estudadas e muitos testes foram realizados utilizando-as, todos os métodos foram compreendidos e implementado na busca da solução do problema, realizando atividades práticas paralelas aos estudos teóricos de modo a favorecer um maior aprendizado. Atividade concluída

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

Com a câmera instalada na plataforma de testes, foram obtidas 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 o 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:

Todas as rotinas foram desenvolvidas e implementadas em diferentes condições ambientais como é possível observar com os resultados apresentados. Atividade concluída

# 6 – Testes pós-processsados dos algoritmos:

Os algoritmos de detecção de borda foram primeiramente otimizados com o auxílio de um algoritmo genético e então puderam ser testados e avaliados para que o melhor método fosse utilizado no projeto. Além disso foram realizados testes utilizando a Transformada de Hough e a Transformada Inversa da Perspectiva . Atividade concluída

# 7 – Elaboração de relatório parcial e final:

O relatório parcial foi desenvolvido e aceito e este é o relatório final que está sendo entregue dentro do prazo estipulado. Atividade concluída

# 8 – Redação de trabalhos e publicações:

Um artigo foi aceito para XIX Congresso Brasileiro de Automática (CBA) [NUNES, 2012] e para o evento de iniciação científica da USP [SIICUSP, 2012]. Atividade concluída

É importante frisar que o término do projeto aconteceu com um ano de diferença do esperado inicialmente porque o aluno participou do programa Ciência Sem Fronteiras e estudou por um ano no Instituto Politecnico di Torino, em Turim na Itália. Por esse motivo a bolsa de iniciação científica foi paralisada por um total de 14 meses e assim as atividades foram concluídas num total de 10 meses de trabalho.

Tabela 13 - Cronograma proposto inicialmente

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Mês 1 | Mês 2 | Mês 3 | Mês 4 | Mês 5 | Mês 6 | Mês 7 | Mês 8 | | Mês 9 | Mês 10 | Mês 11 | Mês 12 | Andamento |
| #1 |  | | | | | | | | | |  |  |  | Concluído |
|  | | | | | | | | | | | |  |
| #2 |  | | |  |  |  |  |  |  | |  |  |  | Concluído |
|  | | | |  |  |  |  |  | |  |  |  |
| #3 |  |  | |  |  |  |  |  |  | |  |  |  | Concluído |
|  |  | | |  |  |  |  |  | |  |  |  |
| #4 |  |  | |  |  |  |  |  |  | |  |  |  | Concluído |
|  |  | | |  |  |  |  |  | |  |  |  |
| #5 |  |  |  |  | | | | | | | | |  | Concluído |
|  |  |  |  | | | | | | | | | |
| #6 |  |  |  |  |  |  | | | | | | |  | Concluído |
|  |  |  |  |  |  |  |  | | | | | |
| #7 |  |  |  |  |  | |  |  |  | |  |  | | Concluído |
|  |  |  |  |  |  |  |  |  | |  |  |  |
| #8 |  |  |  |  |  | |  |  |  | |  |  | | Concluído |
|  |  |  |  |  |  |  |  |  | |  |  |  |

# 8 – 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: 03 dez. 2013.

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](http://www.fenabrave.org.br/principal/pub/Image/20110614100628anuario2010.pdf%20) >. Acesso em: 03 dez. 2013.

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: 03 dez. 2013.

FORD. (2011). “Lane Keeping System Helps Fusion Drivers Stay Alert and Between the Lines”. Disponível em: < <http://www.at.ford.com/news/cn/Pages/Ford%20Lane%20Keeping%20System%20Helps%20Fusion%20Drivers%20Stay%20Alert%20and%20Between%20the%20Lines.aspx> >. Acesso em: 03 dez. 2013.

GOLDBERG, D. (1989). “Genetic Algorithms in Search, Optimization and Machine Learning”. Assison-Wesley.

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: 03 dez. 2013.

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.

NUNES, R. F.; HERNANDES A. C.; BECKER M. (2012). “Detecção de Faixas com Câmera Monocular e Otimização Através de Algoritmo Genético”. Anais do XIX Congresso Brasileiro de Automática, CBA, Campina Grande, pp. 3534-3540

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

OMS. (2013). “Global Plan for the Decade of Action for Road Safety 2011-2200”. 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: 03 dez. 2013.

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://www3.eesc.usp.br/sena/url/pt/index.php> >. Acesso em: 03 dez. 2013.

SIICUSP. (2012). “20º Simpósio Internacional de Iniciação Científica” Disponível em: < <https://uspdigital.usp.br/siicusp/cdOnlineTrabalhoObter?numeroInscricaoTrabalho=2896&numeroEdicao=20&print=S>>. Acesso em: 03 dez. 2013

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