

# Resumo do Projeto Final

*Vinícius Schnaider Zolet, Wilson Felipp dos Santos, Pedro F.  
Departamento Acadêmico de Informática - DAINF*

Universidade Tecnológica Federal do Paraná – Curitiba, PR – Brasil

**Resumo:** *Este artigo demonstra a participação de cada membro, os principais problemas encontrados e como eles foram superados para a elaboração dos programas solicitados no projeto prático final e também detalhes adicionais sobre os programas, de modo a explicar conceitos ou métodos que não ficaram explícitos nos comentários do programa.*

## 1 - O problema:

O projeto consistia na criação de um algoritmo capaz de contar e categorizar veículos gerados aleatoriamente através de um código previamente escrito pelo professor Bogdan.

## 2 - A solução:

O grupo se reuniu ao longo das semanas, dentro e fora de sala, para discutir quais rumos tomar na elaboração do projeto, e foi decidido, em primeira mão, pela seguinte abordagem: Transformar a imagem apenas para o cinza, filtrar, aplicar a limiar e então a rotulação, fazendo a categorização da mesma nas 4 diferentes categorias de veículos apresentadas no enunciado do projeto.

### 2.1. Os Problemas encontrados:

A filtragem e o limiar ocorreram sem problemas, visto que foram aplicados os mesmos códigos do Pré-projeto Final.

O primeiro problema ocorreu na tentativa de rotulação de componentes conexos, em primeira instância, o programa estava agindo de forma anômala, considerando cada pixel como um objeto independente, variando de 1 a 255, o problema foi solucionado primeiramente ignorando as bordas da imagem, visto que a mesma não continha nenhuma informação relevante ao projeto, e era impossível de veículos serem gerados naquela área, após isso, o problema ainda persistia.

#### 2.1.1. Tentativa 1:

A primeira tentativa de classificação foi sem o uso da rotulação, a ideia era pegar as dimensões conhecidas de cada tipo de veículo e especifica-las no programa com uma margem de erro entre 10 e 20% para altura e para largura. Após isso, percorrer toda a imagem procurando elementos diferentes de 0, quando fossem encontrados, o programa compararia com as dimensões conhecidas e diria o tipo de veículo.

Infelizmente não foi elaborado um método de fazer o programa considerar cada objeto na imagem como algo individual, pelo fato de todos possuírem a mesma cor (255), no momento que a imagem era percorrida, todos os pixels brancos naquela linha e coluna eram considerados como um único objeto. A criação de uma distância mínima de pixels pretos entre os pixels brancos foi tentada, sem sucesso. Com isso, a solução 1 foi descartada.

#### 2.1.2. Tentativa 2:

Era uma variação da solução um, pensando no mesmo princípio, porém percorrendo a matriz nos 2 eixos simultaneamente utilizando 4 for, novamente, procurando por pixels diferentes de 0 e comparando-os com o tamanho pré-definido.

Por conta da filtragem e do limiar, alguns objetos estavam englobando a faixa no background, fazendo com que ficassem acima do tamanho estipulado, na tentativa de remover isso, diminuimos a janela da filtragem e o tamanho do limiar, porém, a distorção causada nas imagens ficou acima de níveis toleráveis. Para solucionar este problema foi feito com que o filtro e a limiar rodassem duas vezes, porém perdia-se a contagem de alguns carros que estava mais para a direita da imagem da mesma forma, sendo então também descartada.

### 2.1.3. Tentativa 3:

Utilizando a imagem rotulada, a ideia era percorrer a matriz procurando pelo primeiro rótulo, após encontrado, verificar quantos iguais a esse estavam presentes na imagem, dependendo da quantidade, a classificação do veículo mudaria.

O processo de rotulação se mostrou inconsistente, por motivos desconhecidos, a mesma imagem possui diversos rótulos diferentes, mesmo utilizando o *forward pass* e o *backward pass* diversas vezes, o problema se mantém, ademais, o *backward pass* se mostrou dispensável, visto que os valores se alteraram apenas com o uso do *forward pass*. Todas as tentativas de consertar resultaram em travamentos do programa. Por conta disso, a solução 3 foi posta em aberto.

### 2.1.4. Tentativa 4:

Na tentativa 4 foi tentada uma variação da tentativa 3, buscando uma aproximação diferente com valores inversos (se tornando 0 para os carros e 255 para o background, mudando então a forma da rotulação).

A mesma se mostrou não efetiva, não rotulando nada da imagem, ficando inalterada com relação à limiar anterior.

Ainda assim foram tentadas outras várias formas de resolver a rotulação e a contagem sem ela, mas nada se mostrou realmente efetivo.

## 3. As participações:

Todos do grupo foram desenvolvendo separadamente cada parte das funções do programa, verificando então qual foi a melhor ideia e implementando-a. Nas discussões foram consideradas velocidade de processamento e eficiência dos códigos.

## 4. As funções:

### 4.1. Escala de cinza e Eliminação de fundo:

Para esta parte foi idealizado o uso de uma Imagem auxiliar (\*aux) que tem largura e altura idênticas a Imagem "img", porém com apenas 1 canal. Recebendo então apenas os dados do primeiro canal de "img", transformando "aux" em uma escala de cinza, por possuir esse único canal.

"aux" recebe, ainda, o módulo da diferença entre os dados de "img" e "bg", eliminando então detalhes extras e deixando aux também já pronto para o próximo passo.

### 4.2. Filtragem e Limiar:

Neste caso foi verificado de forma empírica que seria necessário fazer um filtro menor do que o pensado inicialmente (21x21), pois o mesmo acabou por pegar partes da faixa da estrada. Ficando então um filtro da média, que se trata da soma de todas as casas em um espaço em volta do ponto atual e então a divisão do mesmo pelo total de casas, de espaço 11x11, ganhando então o formato desejado.

Durante a limiar foi novamente encontrada o ponto ideal através de testes, encontrado um valor final ideal para qualquer situação, sendo assim se o valor do espaço atual da varredura for menor que a limiar, ele se torna 0, ou 255 caso seja maior. Fazendo com que os carros se tornem brancos e o fundo preto.

Foi feito esse processo duas vezes, através de um laço, para eliminar buracos em carros de cores mais claras que estavam dando problemas inicialmente.

#### *4.2.1. Borda e Inversão:*

Neste momento foi percebido uma borda extra que ficou da imagem inicial, sendo assim foi criada uma segunda Imagem auxiliar (aux2), pois novas imagens geradas automaticamente vinham zeradas (zero em todas as casas da matriz).

No final do limiar, que é o final do laço de Filtragem+Limiar, as informações de “aux” é passado para “aux2”, ficando assim sem bordas.

Foi criado um laço para varrer ela por completo, então, passando novamente de aux2 para aux, eliminando assim a borda com valores errôneos.

A inversão foi criada com dois laços também varrendo do começo ao fim, verificando que caso o valor fosse 255 se tornasse 0 e vice-versa, para a utilização no teste 4, gerando o “negativo” da imagem.

### *4.3. Rotulação:*

#### *4.3.1. Rotulo de ida:*

Para o rótulo foi utilizado várias técnicas, onde todas elas se demonstraram erradas. As técnicas utilizadas foram, inicialmente, utilizando um laço e uma variável indicadora, depois 2 laços com essa variável indicadora e então vários condicionais.

A de vários condicionais é a que mais atende ao requisito explicado em sala para uma rotulação, mas dos 3 é o que menos gera alguma rotulação de qualquer tipo. A primeira rotulava alguns, mas ignorava algumas linhas de carros e a com dois laços estava rotulando vários objetos com o mesmo valor e vários valores no mesmo objeto.

#### *4.3.2. Rotulo de volta:*

A lógica aplicada no rótulo de volta é o mesmo, porém invertendo os sinais para que pegasse os valores exemplificados em sala anteriormente.

### *4.4. Contagem:*

A ideia seria de pegar cada um dos valores preenchidos pela rotulação e colocar na variável vetor “contador”, com 256 posições (de 0 até 255) para somar +1 toda vez que o rótulo naquele espaço aparecesse, sendo assim, caso o dado na casa 0, i, j fosse x, no contador[x] seria adicionado 1.

#### *4.4.1: Testes empíricos:*

Para gerar, então, uma média dos valores rotulados seriam feitos diversos testes com diversas imagens diferentes até encontrar uma consistência dos valores rotulados, encontrado assim um valor a,b,c e d, onde são, respectivamente, o menor valor encontrado anteriormente para o rótulo de uma moto, a dos carros, a de um ônibus ou caminhão pequeno e a dos caminhões grandes.

#### ***4.5. Classificação:***

A parte de classificação não foi utilizada, mas a ideia geral seria de fazer vários condicionais dentro de um laço que percorre o vetor “contador”, sendo assim 4 condicionais, um para cada tipo de veículo.

Inicialmente é zerado o vetor “contagem” e entrado nos laços. Os condicionais seguem a ideia dos valores a,b,c e d encontrados anteriormente, sendo que se fosse  $\geq a$  mas  $< b$ , aumentaria 1 em contagem[0], se fosse  $\geq b$  e  $< c$  ele seria contado em +1 em contagem[1], e assim por diante.

Esta classificação nunca foi efetivamente testada pois a parte da contagem estava dando problema, uma vez que o rótulo nunca teve êxito.

### **5. Considerações Finais:**

Apesar de todas as tentativas de classificar os veículos, nenhuma obteve êxito, logo, não foi possível fazer com que o trabalho fosse executado de maneira completa, resultando, assim, em apenas uma finalização parcial do mesmo, ainda que com a ideia geral da finalização após-rotulação explícita neste relatório e em forma de comentários dentro do arquivo “.c”.