

UNIVERSIDADE FEDERAL DE SERGIPE

PROCESSAMENTO DE IMAGENS

FRANCISCO BRAZ DE SOUZA
RAUL MIGUEL SANTOS NASCIMENTO
SAMUEL AUGUSTO SOUZA ALVES SANTANA
TALES SILVA LEOPOLDINO

Generic Stuff

SÃO CRISTÓVÃO
2023

1. Implementação

A linguagem escolhida pelo grupo para efetivar o código foi Python. Utilização do VsCode para ambiente de desenvolvimento.

Funções codificadas: dilatação adaptada ao requisitado, preenchimento de região externa, extração de componentes conexos, captura de objetos e buracos, captura e carregamento da imagem em formato pbm, extensão de borda branca na matriz/imagem.

2. Técnicas aplicadas

Baseando-se em conhecimentos de morfologia visto em sala de aula, foram utilizados os métodos de:

Dilatação por elementos estruturantes.

Preenchimento de regiões interseccionado ao complemento do conjunto de objetos, necessário para o isolamento de buracos na imagem por meio de vizinhança 4.

Extração de componentes conexos por meio de vizinhança 8, para captura dos objetos.

3. Solução

O programa recebe a imagem em formato pbm e carrega a mesma em uma matriz para que seja modificada. O primeiro passo se resume em acrescentar uma borda branca na imagem, para tratar de componentes posicionados na borda. Separadamente, após o primeiro passo, as duas operações morfológicas são aplicadas: primeiramente a extração de componentes conexos, e consecutivamente, preenchimento de região (externa ao objeto). Dessa forma, a imagem é manipulada para identificação das posições de cada objeto e furo presente. Enfim, o programa devolve as imagens manipuladas em suas respectivas funções, junto a impressão da quantidade de objetos e furos, suas posições e suas relações (se o furo pertence ao objeto ou não).

4. Executar o projeto

Para executar o projeto será necessário ter o python na versão 3.7 ou superior e clonar o repositório do código para uma máquina. Para isso, é fundamental seguir os passos abaixo:

- Em um terminal execute:
 - `git clone https://github.com/talessl/Processamento-de-Imagem.git`
- Entre na pasta “Processamento-de-Imagem”
- Execute o arquivo `rast.py`

- Escolha uma das imagens dentro da pasta “Imagens” e passe o caminho do arquivo sem a extensão. Por exemplo: Imagens/teste_1_PI, Imagens/teste_2_PI, etc.

5. Potencial de uso do algoritmo

Com base na implementação e na proposta sobre a qual o algoritmo foi desenvolvido, alguns cenários foram imaginados para sua aplicação. A primeira ideia está presente na questão de identificação de furos em peças que se encontram em ambientes de produção. Seria possível identificar buracos em objetos mal construídos ou então que sofreram algum tipo de falha durante o desenvolvimento. Como o algoritmo trabalha com imagens em preto e branco, torna-se necessário, nesse caso, implementar uma conversão da imagem colorida para o formato esperado.

O segundo cenário diz respeito à identificação de locais desmatados dentro de áreas florestais. Ao se imaginar grandes agrupamentos de vegetação com buracos dispersos dentro da região, poderia-se procurar por alterações na cobertura florestal, ou seja, regiões que não possuam os forros de mata. Imaginando que esses dispositivos estariam em satélites, uma problemática que surgiria, seria a questão de nuvens que poderiam ofuscar a área de visão dos instrumentos, prejudicando a acuracidade. Cabe salientar, que assim como na primeira proposta, as imagens também precisariam ser convertidas para preto e branco.

O terceiro e último cenário proposto, seria a identificação de buracos em vias urbanas. Utilizando câmeras de monitoramento, poderia-se inspecionar as imagens com o intuito de analisar os dados e identificar estas fendas. Um obstáculo que surgiria, seria o tráfego rotineiro, já que ao selecionar imagens para se inspecionar, é provável que em determinadas capturas, algum veículo estivesse em cima do buraco, impossibilitando a identificação daquela região. Assim, seria necessário realizar uma pré-operação, com o objetivo de separar as imagens sem obstáculos nas capturas, por exemplo, poderia-se selecionar imagens capturadas em horários de menor intensidade de tráfego. Por fim, em semelhança com as ideias anteriores, as imagens também precisariam ser convertidas para preto e branco, em consonância com o funcionamento do algoritmo.

6. Relação com o conceito de ética

Em relação ao conceito de ética, foram separados alguns pontos que detalham a relação do algoritmo proposto com a definição deste juízo:

- Privacidade: se o sistema captura imagens de pessoas ou ambientes privados, é importante garantir que as imagens sejam tratadas com respeito à privacidade e que apenas as informações necessárias sejam coletadas.

- **Viés:** é importante garantir que o algoritmo não tenha um viés em relação a certos tipos de objetos ou grupos de pessoas. Isso pode ser especialmente importante se o sistema for usado para tomar decisões importantes, como em processos de seleção ou contratação.
- **Segurança:** é importante garantir que o sistema seja seguro e não permita que pessoas não autorizadas acessem as informações coletadas ou possam fazer alterações indevidas no sistema.
- **Responsabilidade:** se o sistema for usado para tomar decisões importantes, como em processos de seleção ou contratação, é importante garantir que as decisões sejam justas e precisas. Isso pode envolver a realização de testes regulares para garantir que o sistema esteja funcionando corretamente e tomando decisões justas.
- **Transparência:** é importante garantir que as pessoas que estão sendo avaliadas pelo sistema entendam como o sistema funciona e como as decisões são tomadas. Isso pode envolver a criação de documentação clara e acessível sobre o sistema e seus algoritmos.
- **Vigilância e controle de massa:** o sistema pode ser usado para monitorar e controlar as atividades de grupos de pessoas, como manifestantes ou dissidentes políticos. Essa aplicação pode ser usada para limitar a liberdade de expressão e os direitos humanos, além de permitir a repressão do Estado.