

Molecule Viewer

CV Projeto 2

Resumo

O objetivo para este projeto consiste no desenvolvimento dum programa capaz de efetuar a deteção de fórmulas moleculares através da biblioteca OpenCV para C++ e gerar um ficheiro com determinado formato para a ilustração do modelo da molécula em 3D através duma plataforma web designada por *Molecule Viewer*.

Abstract

The objective for this project is to develop a program capable of detecting molecular formulas through the OpenCV library for C++ and generate a file with a certain format for the illustration of the model of the molecule in 3D through a web platform called Molecule Viewer.

Introdução

De modo a dar seguimento ao trabalho efetuado no primeiro projeto da unidade curricular de Computação Visual foi decidido para este projeto o desenvolvimento dum programa capaz de gerar ficheiros para a representação 3D da molécula através duma imagem duma fórmula molecular. Uma vez gerado o ficheiro por este programa, o cliente é capaz de fazer upload do gerado na plataforma web anteriormente criada para a eventual visualização do modelo e a sua manipulação.

Para o desenvolvimento deste trabalho é requerido e necessário o uso da biblioteca OpenCV para a linguagem C++. Está apresenta-nos ferramentas necessárias para dar maior resolução do desejado na imagem em interesse através da aplicação de diversos filtros, tais como Gaussian Blur, BGR to Gray Scale, Binary Threshold e outras ferramentas para a seleção do desejado, através de funções do tipo Find Contours. O processamento do programa é mencionado mais em frente em este documento.

Inicialmente o projeto divide-se em 3 faces, que são:

- Aplicação de filtros e deteção de objetos em interesse
- Determinação de Átomos correspondente às letras definidas na imagem e as suas correspondentes ligações.
- Processamento da informação recolhido para criação do ficheiro para a visualização em 3D do modelo.

Manual do utilizador

Para a execução do programa é necessário inicialmente executar os seguintes comandos:

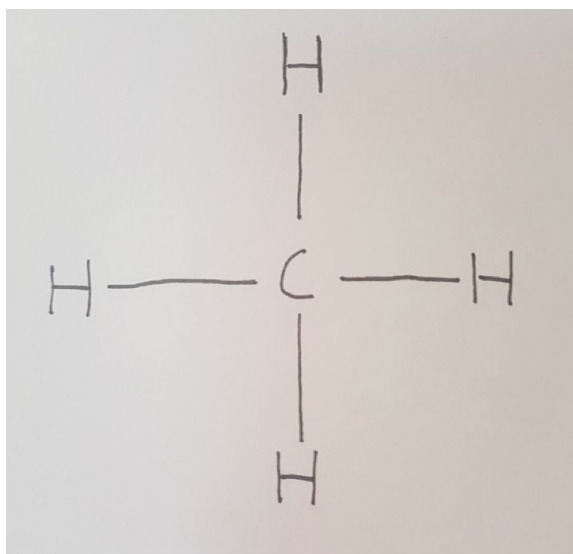
\$./compile.sh moleculedetection.cpp

\$./exec <nome do ficheiro da imagem>

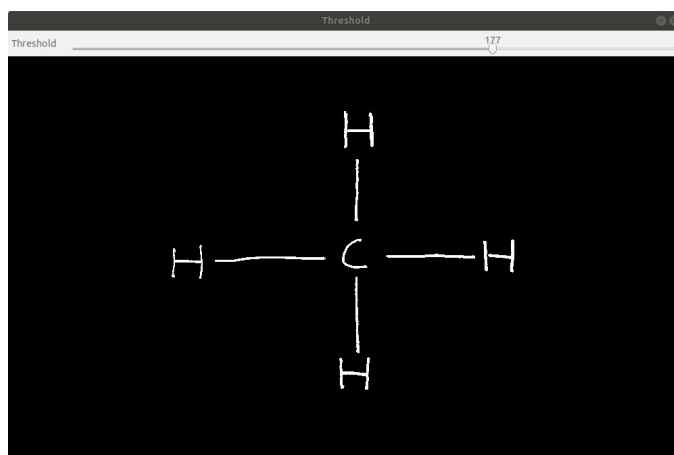
Para o utilizador executar o programa, é necessário escolher um ficheiro imagem com uma fórmula molecular simples desenhada em papel ou por software de desenho (paint, gimp e outras ferramentas).

De notar que os átomos reconhecidos pelo programas são os seguintes: {C, F, H, I, N, O, P, S}.

Na formula molecular podem existir ligações simples, duplas e/ou triplas.

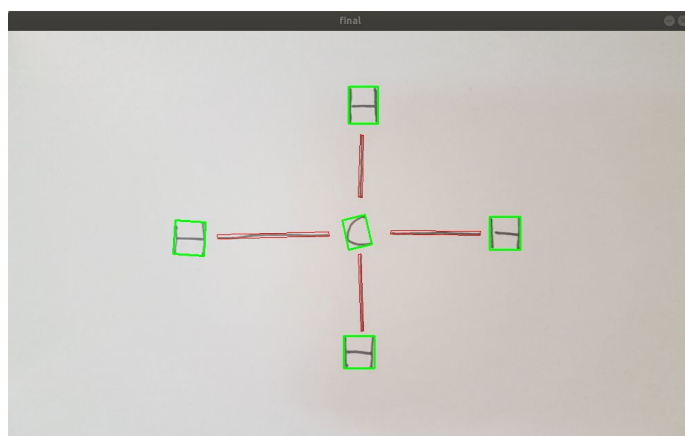


Uma vez executado o programa, é definido pelo utilizador o *threshold* a aplicar a operação de *Threshold Binary Inverted* a imagem submetida, através do *Trackbar* no topo da janela “*Threshold*”. Para avançar o utilizador deve clicar em qualquer tecla do teclado do computador.

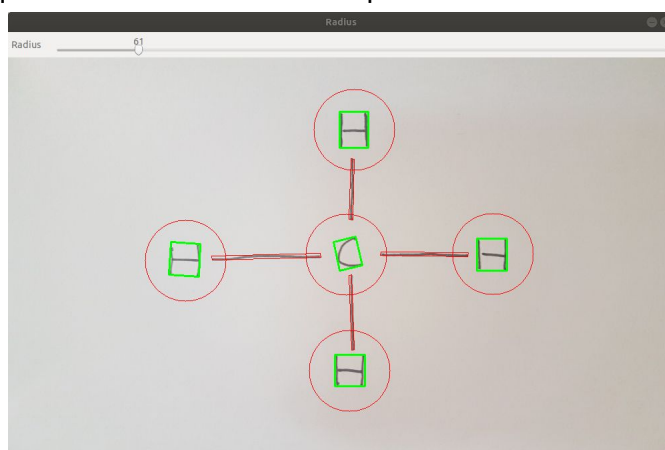


Uma vez escolhido o valor do threshold, é mostrado o que foi detectado pelo programa executado. Na seguinte figura é possível visualizar dois tipos de áreas de seleção com cor de contorno diferentes, cada cor tendo um significado, que é:

- Verde: O conteúdo selecionado pela área com a seguinte cor corresponde a uma letra;
- Vermelho: O conteúdo selecionado pela área com a seguinte cor corresponde a uma linha.



De modo a detetar de forma certa às correspondentes ligações a que a cada átomo lhe corresponde na fórmula molecular é ilustrado, em cada centro do átomo, um círculo a vermelho com tamanho variável em proporção ao valor atribuído ao *Trackbar* “Radius”. O pretendido é que a intersecção entre o círculo do correspondente átomo e linha, corresponde a ligação do átomo em consideração. Tomar em atenção que é necessário que não exista intersecção entre os círculos dos átomos, para o correto funcionamento do programa. Para avançar, tocar qualquer tecla do teclado do computador.



Finalmente, no final do programa, é gerado um ficheiro “output.txt”. Este ficheiro ao ser submetido na plataforma web “*Molecule Viewer*”, é visualizado o modelo 3D da molécula.

Deteção de letras e ligações

O programa desenvolvido para o trabalho designa-se por `moleculedetection.cpp`.

A função principal responsável pelo funcionamento geral do programa é a função com nome *detect* que tem como input arguments a string representante de um ficheiro imagem.

Uma vez inicializado o programa e a execução das instruções da função *detect*, inicialmente é lido um ficheiro imagem.

De modo a uniformizar a imagem, com boa prática, é primeiramente transformada a imagem numa escala em Gray, através da função `cvtColor` e a flag, que determina o tipo de modificação, `CV_BGR2GRAY`.

De seguida é aplicada a imagem resultante uma operação do tipo threshold, mais em específico do tipo ***Threshold Binary Inverted*** em que os pixels com intensidade maior ao valor do threshold definido pelo *Trackbar* já mencionado são seleccionados e apresentados com pixels de intensidade 255.

É aplicado uma transformação morfológica do tipo *Open* à imagem, sendo essa útil para remover *ruído* apresentado na imagem. O kernel a operar na operação morfológica é do tipo *Ellipse* sendo o mais apto para o pretendido.

É executada um tipo de transformação morfológica do tipo *Close*. De modo a fechar os pequenos buracos nos objetos definidos, neste caso buracos em letras e linhas representadas na imagem. O kernel a operar na operação morfológica é também do tipo *Ellipse* sendo o mais apto para o pretendido.

Tendo os objetos bem representados com pixels de intensidade equivalente a 255 queremos encontrar informação relativa aos seus contornos para a seleção dos objetos. Isto é possível através da utilização da função ***findContours*** da biblioteca de OpenCV.

Para cada contorno encontrado na submetida imagem, é calculado o retângulo de limite superior direito de um conjunto de pontos do contorno. Com a função ***minAreaRect***, com o argumento um determinado contorno, é calculado informação relativa da mínima área retangular que forma o contorno. De notar que as áreas retangulares podem ter uma inclinação. A informação é guardada numa variável de classe ***RotatedRect***, sendo auto-explicativa o tipo de informação que o objeto irá conter.

Através duma série de testes foi verificado que as áreas retangulares calculadas para as linhas possuem uma relação entre largura e espessura muito elevado. Sendo que quando a linha se encontra na vertical, a altura da área retangular é geralmente bastante maior do que a largura. Quando a linha se encontra na horizontal, a largura é bastante maior do que a altura. No caso das letras, este tipo de relação não se verifica com muita frequência, logo

opta-se como forma prática e simples na detecção de linhas que representam a ligação entre átomos este tipo de padrão mencionado.

Se a área retangular calculada representa uma linha, então é desenhada os contornos da área retangular a vermelho que o representa na imagem. Juntamente é procedido a determinação das pontas da reta, sendo inseridas os correspondentes ponto num vetor que representa a linha. Esta é juntamente inserida num vetor que contém todas as ligações da molécula.

Identificação de letras

Se a área retangular tem um padrão equivalente a de uma letra, então é desenhado os contornos da área retangular, representados em verde. É então procedido a determinação do tipo de letra que este representa. No entanto, inicialmente procede-se a determinação do centro do objeto em consideração. Antes da identificação da letra que a área retangular representa é necessário uma imagem exclusiva da letra, da molécula, a determinar. A imagem da molécula pode encontrar-se inclinada com um certo ângulo, portanto antes da efetuação do crop da imagem para a letra a considerar, é endireitado a região de interesse e depois é efetuado o crop que irá resultar numa imagem duma letra. Estas instruções são executadas na função **insertROIs**. A imagem cortada é retornada. Tendo a imagem da letra selecionada, esta é identificada a partir da função **FindLetters**.

Na função **FindLetters**, inicialmente é carregado todas as letras dos átomos a comparar com a imagem de entrada. As imagens das letras do átomos a comparar com a imagem da letra em consideração encontra-se na pasta “atoms” do projeto. Cada uma das letras de átomos registadas no programa é comparada com a imagem de entrada. Mas para isso tanto a largura e a altura das duas imagens que representam uma letra devem ser equivalentes. Se a letra registada no programa tiver uma escala maior do que a imagem da letra de entrada, então é efetuado uma modificação do tamanho da imagem de entrada de forma a ter um tamanho equivalente a da letra registada.

Caso contrário, se a letra de entrada tiver uma escala maior do que a da letra registada então é efetuado a mudança de tamanho da imagem da letra registada no programa, de modo a ter um tamanho equivalente a imagem do átomo de entrada. É efetuado operações threshold binário nas duas imagens, que sempre quando a intensidade dos pixels for superior a 50 passam a ser equivalentes a um valor de intensidade de 255.

A forma que é determinado se a letra de entrada corresponde a letra em consideração do programa, é através da operação de subtração entre estas duas imagens, após ser aplicado a operação de threshold anteriormente mencionado.

Uma vez efetuada a subtração é calculado o número de pixels com intensidade equivalente 255, como também o número de pixels com mesma intensidade antes da operação da subtração.

Com o número de pixels com intensidade a 255 da imagem resultante e o seu número total de pixels, é determinado o número de pixels com intensidade diferente a 255, neste caso com intensidade equivalente a 0. Quanto mais próximo o número de pixels com intensidade nula ao número total de pixels da imagem, maior é a probabilidade da letra de entrada ser a letra registada no programa.

De modo a reforçar a semelhança entre as imagens das letras em comparação é efetuada uma relação entre o número total de pixels com intensidade 255 que passaram, depois efetuada a operação de subtração, para valor 0 e o número total de pixels com maior intensidade possível antes da operação mencionada ser efetuada na imagem da letra a determinar.

É feita uma média entre estas duas percentagens/relações calculadas. Se a média calculada for superior a maior média da percentagem de semelhança entre a letra de entrada e uma determinada letra registada anteriormente comparada no programa, então a nova letra com maior semelhanças a letra em consideração é a atual.

Após a deteção de todas as letras e ligações é iniciada o estabelecimento de ligações entre átomos que compõem a molécula.

Criação do ficheiro com a molécula para a modelação 3D

Para a criação do ficheiro com a molécula no formato desejado para a modelação 3D usamos a função “make_molecule()”. Esta função recebe como argumentos uma lista de tuplos, cada tuplo contendo o símbolo químico de um átomo e o seu centro, uma lista de listas de pontos com as coordenadas das extremidades de cada ligação e recebe o raio determinado usando a trackbar. Esta função vai primeiro encontrar todas as ligações que existem entre átomos na molécula e vai adicioná-las a uma lista de tuplos triplos, com os índices dos dois átomos ligados e por último o tipo de ligação, que pode ser simples, dupla ou tripla. Depois de determinar as ligações, a função irá começar a executar o algoritmo de determinação de posições dos átomos na modelação 3D. Para isto ele prioriza átomos com mais de 2 ligações. Um átomo é sempre posicionado em relação a um átomo ao qual ele está ligado, exceto quando é o primeiro átomo a ser posicionado e nesse caso ele fica na origem. Um átomo posicionado uma vez já não pode ser mais reposicionado. Depois de todos os átomos com mais de duas ligações terem uma posição os restantes átomos serão posicionados nas aberturas que ainda existirem. No final todos os átomos têm posições e todas as ligações estão constituídas, e será então possível agora escrever o ficheiro que possibilitará a modelação 3D.

Este ficheiro tem um formato específico. Nas linhas iniciais são descritas os átomos existentes nas moléculas. Primeiro escreve-se o símbolo químico do átomo e depois os 3 valores das coordenadas. Por exemplo, para um átomo de carbono que ficaria na origem a linha seria: “C 0 0 0”. Cada átomo tem a sua própria linha.

Depois de estarem todos os átomos escritos escreve-se uma linha com o símbolo "/" e de seguida as várias ligações e o seu tipo. Para simbolizar os átomos usa-se os índices pela ordem em que aparecem no ficheiro. Uma linha poderia ser "0 2 3" e isto quereria dizer que o átomo com índice 0 se liga ao átomo com índice 2 com uma ligação tripla.

Um exemplo de um ficheiro completo seria:

```
H 1 0 0
H 0 0 1
C 0 0 0
H -1 0 0
H 0 0 -1
/
2 4 1
2 3 1
1 2 1
0 2 1
```

A função `write_molecule()` aceita como argumento a lista com os átomos, a lista com as suas posições e a lista com as ligações e usa-a para escrever um ficheiro no formato referido acima.

Desafios:

Algoritmo para posicionar átomos - Um dos grandes desafios deste trabalho depois de conseguirmos detectar e identificar letras e ligações foi reorganizar estes de modo a criar o ficheiro final que queríamos para a simulação. Várias hipóteses foram pensadas e um dos maiores desafios era como saber que átomos estavam relacionados. A nossa solução acabou por ser que iríamos ter informação sobre os limites de cada ligação e também sobre o centro de cada letra, então saberíamos que dois átomos estariam ligados quando ambos estiverem na vizinhança de um dos limites de uma ligação. Este conceito de vizinhança foi bastante trabalhoso pois para imagens diferentes a distância, em pixels, dos centros das letras aos limites das ligações era diferente e não parecia haver um valor ótimo. Para resolver este problema implementamos uma trackbar na qual o utilizador poderia escolher este valor da vizinhança visualizando o raio de alcance do centro de cada letra na imagem da molécula. Embora ainda haja alguns problemas quando os tamanhos das ligações não são regulares na mesma imagem, esta solução teve bastante êxito nos testes efetuados por nós.

Algoritmo de identificação de letras - Outro grande desafio deste trabalho foi como identificar as letras que foram detectadas. Várias hipóteses foram ponderadas desde comparar imagens pixel a pixel até usar bibliotecas externas. No entanto a nossa solução foi ter um modelo para cada letra para a qual poderíamos ter correspondência e depois para todos esses modelos disponíveis subtraímos a letra detectada às letras modeladas. No entanto a maneira como interpretamos os resultados dessa subtração também foram bastante discutidos, havendo duas interpretações diferentes. A primeira era depois da subtração fazer a razão entre o número de pixels pretos (ou seja os pixels onde as duas imagens eram similares) e os pixels totais. Isto dava-nos um coeficiente de semelhança que

depois era analisado e a letra com maior coeficiente de semelhança era a escolhida. A segunda interpretação era fazer a razão entre o número de pixels que mudaram de branco para preto e o número de pixels que estavam a branco na imagem modelo. Deste modo o coeficiente de semelhança levava apenas em conta os pixels não a preto das duas imagens e assim não “diluía” o coeficiente de semelhança. No final a solução à qual chegamos foi fazer um misto das duas maneiras, para obter os melhores resultados.

Conclusão:

Com este trabalho conseguimos complementar o trabalho do primeiro projeto, concretizando a ideia de levar uma simples molécula desde o modelo científico até à sua modelação 3D. Também conseguimos avançar bastante os nossos conhecimentos sobre processamento de imagem, que nos permitiu realizar a detecção e identificação de letras e outros símbolos no nosso trabalho. O trabalho trabalhou também sobre ideias interessantes e às quais ambos pensamos que nos vamos debruçar novamente sobre.