

DISCIPLINA: Métodos Computacionais
PROFESSOR: Beatriz Regina Tavares Franciosi
SEMESTRE: 2014/I
AUTOR: Ramon Costi Fernandes


TUTORIAL

Para a realização do primeiro trabalho de Métodos Computacionais é recomendado o uso do editor gráfico GIMP[1] na marcação e extração dos pontos de controle que serão utilizados para a geração dos polinômios interpoladores. Por sua vez, estes pontos podem ser transcritos em um arquivo SVG[2], que utiliza o formato XML[3] em sua estrutura, possibilitando que os mesmos sejam facilmente extraídos e utilizados nos ambientes de computação científica.

Este breve tutorial tem por objetivo demonstrar como utilizar o GIMP para a obtenção de pontos em uma imagem qualquer e extrair o conteúdo de arquivos SVG, referente às coordenadas dos pontos definidos, através de um script que será fornecido juntamente com o Trabalho 1 da disciplina.

Utilizando o GIMP para extração de pontos em imagens

O editor de imagens GIMP encontra-se disponível nas instalações Linux atuais dos laboratórios, e portanto foi escolhido como ferramenta base para este tutorial.

Selecionando a ferramenta  *Paths Tool*, marque a opção *Polygonal* (figura 1a) e comece a marcar pontos na imagem, como se estivesse delimitando o contorno da forma desejada, demonstrado na figura 1b.

O próximo passo consiste em exportar as marcações realizadas anteriormente para um arquivo SVG. Considerando que os pontos marcados na imagem correspondem a um caminho (*path*), a ordem na qual estes pontos foram definidos será preservada.

Entretanto, antes de realizar a extração dos pontos, é necessário efetuar uma inversão vertical da imagem, como ilustrado na figura 2. Isto se deve a maneira pela qual o GIMP define a origem do eixo y da imagem.¹ Se este passo não for efetuado, o resultado final da letra gerada a partir do polinômio interpolador será invertido verticalmente e preciosas horas (que poderiam ser dedicadas para a escrita do artigo, por exemplo) serão gastas tentando entender porque o Maxima resolveu sabotar o nosso trabalho.

Após realizar a inversão da imagem, o último passo necessário para salvar as coordenadas em um arquivo SVG é selecionar a opção *Export Path* na aba *Paths*, conforme ilustrado na figura 3.

¹O GIMP define a origem do sistema de coordenadas no canto superior esquerdo da tela, enquanto o usual é utilizarmos o canto inferior esquerdo.

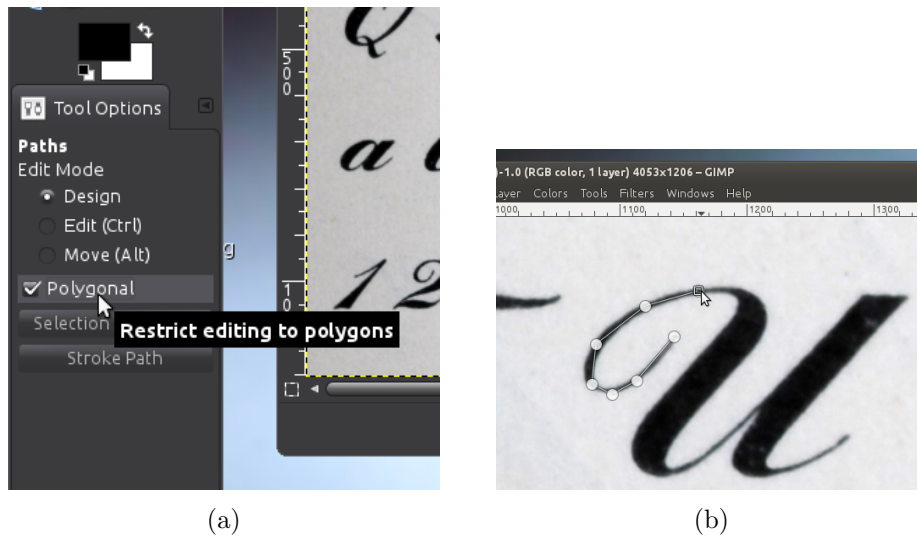


Figura 1: a) opção *Polygonal*, e b) traçando pontos de contorno da letra *U*.

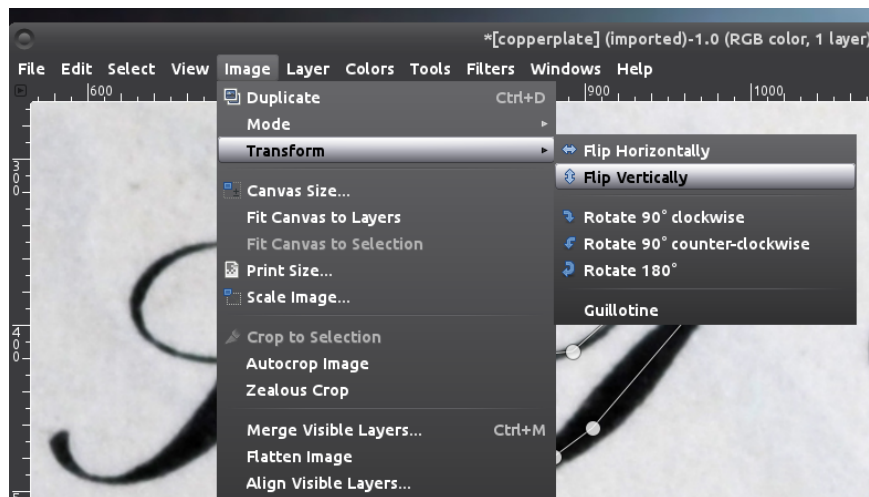


Figura 2: Inversão vertical da imagem.

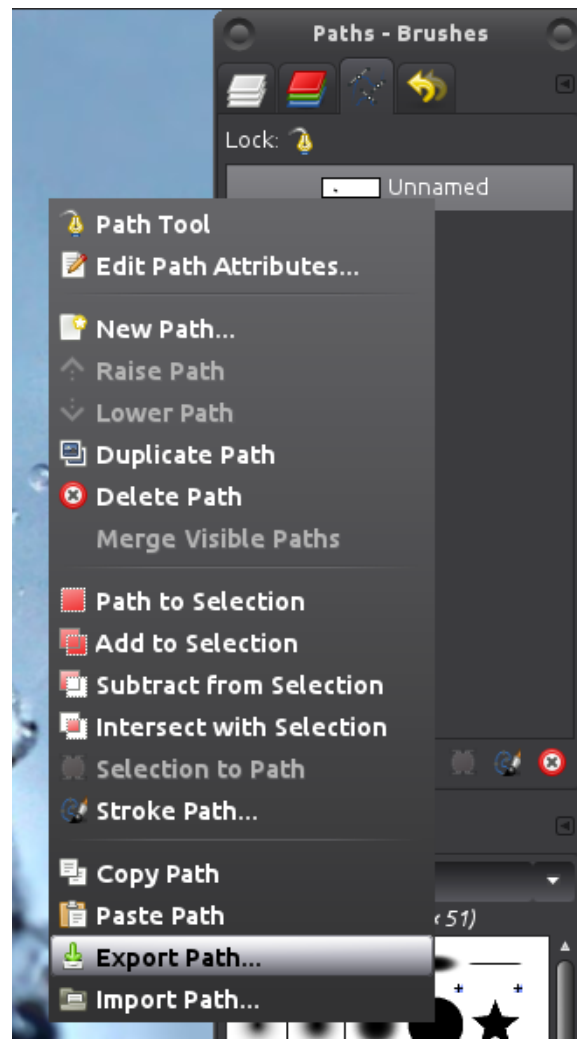


Figura 3: Como salvar o caminho selecionado em um arquivo SVG.

Interpretando o arquivo SVG

A segunda etapa deste tutorial demonstra como capturar as coordenadas individuais x e y dos pontos listados no arquivo SVG.

Conforme definido na especificação de arquivos SVG, o mesmo contém uma descrição de um caminho utilizado para representar a geometria do contorno de um objeto, isto é, um intervalo de pontos, em sequência, começando numa posição inicial e terminando no último ponto da série, sendo que os pontos podem ser interligados através de linhas, curvas, ou arcos. O resultado desta informação adicional, que descreve como os pontos são interligados (o caminho), é uma replicação da informação referente as coordenadas dos pontos. Como estamos interessados apenas em obter os pontos individuais (em ordem) contidos no arquivo SVG, é necessário filtrar a informação contida no mesmo.

Felizmente (para os alunos), também será fornecido com o Trabalho 1 da disciplina de Métodos Computacionais um script desenvolvido na linguagem *Python*[4] para interpretar a informação referente aos pontos em arquivos SVG.

As distribuições Linux dos laboratórios já possuem *Python* versões 2² e 3 instaladas. Para rodar o script, portanto, é necessário executar, através do ambiente shell (terminal) a seguinte linha de comando:

```
# python3 parse_svg.py <arquivo.svg>
```

o conteúdo referente aos pontos do arquivo SVG informado será salvo em um arquivo texto chamado "*output.txt*" ou, opcionalmente, o nome do arquivo de saída pode ser especificado:

```
# python3 parse_svg.py <arquivo.svg> <arquivo.de.saida>
```

Não considerem a existência deste script como um inibidor para que vocês desenvolvam uma solução própria. Utilizar expressões regulares para identificar elementos textuais de interesse (sejam estes em arquivos ou outros meios) é um tema recorrente na computação. Da mesma forma, não considerem este script como infalível. Dentro do limitado espaço de testes disponíveis para sua verificação, o mesmo se mostrou eficaz, mas não deixem de verificar se os resultados gerados estão coerentes com o esperado.

²O script fornecido foi desenvolvido em *Python* versão 3, utilizando comandos de sintaxe específicos para esta versão, e portanto não funcionará em interpretadores *Python* versão 2.

Referências

- [1] *GNU Image Manipulation Program*. <http://www.gimp.org/>.
- [2] *Scalable Vector Graphics*. <http://www.w3.org/TR/SVG/>.
- [3] *Extensible Markup Language*. <http://www.w3.org/XML/>.
- [4] *Python*. <https://www.python.org/>.