

Transformações Afins e Projetivas

Computação Visual

Professores: Joaquim Madeira e Paulo Dias

Filipe Tavares 72063

Tiago Henriques 73046

Resumo - Este projeto foi proposto na disciplina de Computação Visual e tem como objetivo principal aplicar as técnicas de OpenCV adquiridas nas aulas práticas, mais particularmente, criar uma aplicação que permita aplicar diferentes transformações/correções a várias imagens como por exemplo rotação, alteração uniforme e não-uniforme das dimensões da imagem. Outro requisito a cumprir consiste na disponibilização de funcionalidades de pré-e-pós processamento que permitam atenuar eventuais artefactos nas imagens resultantes. É ainda disponibilizada a possibilidade ao utilizador de guardar as imagens resultantes das transformações acima referidas.

Abstract - This project was proposed in the Visual Computation subject and its main purpose is to apply the multiple techniques of OpenCV learned through the practical classes. This project aims to create an application that allows a user to apply different transformations/corrections to multiple images such as rotation, uniform and non-uniform change of image dimensions. Another feature to conquer is to provide pre-and-pro capabilities that allow to mitigate any artefacts on the resulting images. It is also available to the user to save the resulting images from the transformations referred above.

I. INTRODUÇÃO

O trabalho proposto para o projeto da unidade curricular de Computação Visual é uma aplicação para processamento de transformações/correções a várias imagens, desenvolvida em OpenCV. A escolha deste tema foi baseada no facto de a manipulação de imagens ser um assunto bastante sério e usado por quase todos nós nos dias de hoje (por exemplo nas redes sociais) e também porque nos pareceu um desafio bastante interessante de realizar. Para isso, desenvolveu-se uma aplicação prática

com uma interface atrativa, com bastantes funcionalidades, e que se adapte a todos os utilizadores.

Como já foi referido, o nosso projeto foi desenvolvido com a biblioteca OpenCV e a linguagem de programação c++. O ambiente escolhido para a elaboração foi o Visual Studio 2015, uma vez que já tinha sido instalado para cadeiras anteriores frequentadas.

II. CONCEÇÃO/ORGANIZAÇÃO

O projeto desenvolvido contém pastas determinantes para o bom funcionamento desta aplicação. O ficheiro principal *.cpp* está situado na pasta *root* do projeto e é denominado *Main.cpp*. Este ficheiro contém todo o código desenvolvido para as diversas transformações/correções, embora pequenas partes dele tenham sido aproveitadas e reutilizadas das aulas práticas.

Contém ainda a pasta “Imagens”, onde deverão ser colocadas todas as imagens, pois só assim podem ser usadas pela aplicação. Esta pasta serve também como destino para as imagens resultantes guardadas das diversas operações permitidas pela aplicação.

Foram implementadas as transformações/correções solicitadas no enunciado do trabalho bem como algumas outras que foram acrescentadas ao projeto por iniciativa própria. Estas adições vão ser devidamente justificadas mais à frente.

III. INTERFACE DO UTILIZADOR

O interface do utilizador foi pensado e concebido para uma perceção fácil e rápida das funcionalidades que a nossa aplicação sustenta. Consiste num menu com várias opções que correspondem às funcionalidades do nosso sistema. A nossa aplicação começa por pedir ao utilizador

o nome da imagem que este pretende fazer alterações. Este nome deve conter a extensão da imagem solicitada, por exemplo: “lena.jpg”.

É lembrado que a imagem deve estar devidamente colocada na pasta “Imagens”. Aparecem ainda as opções de sair imediatamente do sistema através da introdução do comando ‘exit’ ou se o utilizador preferir listar todas as imagens que a nossa aplicação já contém, basta introduzir o comando ‘list’.

```
#####
Para sair escreva 'exit'
Para listar imagens existentes escreva 'list'
Nome da imagem (colocar extensao):
```

Figura 1 – Menu inicial

No caso do utilizador introduzir um nome de imagem inválido ou inexistente, este é alertado do sucedido e deve partir para a escolha de uma imagem válida (que se encontre na pasta ‘Imagens’). É recomendado o comando ‘list’ caso aconteça esta situação.

```
#####
Para sair escreva 'exit'
Para listar imagens existentes escreva 'list'
Nome da imagem (colocar extensao): lwna.jpg
Ficheiro nao foi aberto ou localizado !!
```

Figura 2 – Imagem inválida

Caso o utilizador recorra ao comando ‘list’, todas as imagens contidas na pasta ‘Imagens’ são listadas. Note que este comando apenas está desenvolvido para o sistema operativo Windows, pelo que em Linux poderá não funcionar corretamente, aconselha-se comentar a linha 14, da linha 60 à linha 79 (inclusivamente) e a linha 95, para Linux.

```
#####
Para sair escreva 'exit'
Para listar imagens existentes escreva 'list'
Nome da imagem (colocar extensao): list
arestas.jpg
Calvin.jpg
deti.jpg
lena.jpg
lena_rodada.jpg
pontos.jpg
redimensionada.jpg
rodada.jpg
threshold.jpg
```

Figura 3 – Comando ‘list’

Depois do utilizador escolher a imagem que quer processar com sucesso, é-lhe mostrado todas as funcionalidades abrangidas pelo nosso sistema, pelo que o utilizador deverá escolher qual das transformações quer aplicar sobre a imagem. É então reproduzido um menu ao utilizador, sendo apresentado neste as funcionalidades da

nossa aplicação. Cada funcionalidade está associada a uma tecla diferente.

```
#####
Transformacoes Afins e Projetivas:
L - Mudar Imagem
K - Estilizacao da imagem
O - Aplicar efeito a lapis na imagem
D - Imagem a preto e branco
E - Espelhar Imagem
T - Definir Threshold
S - Alterar Tamanho
X - Rodar para direita
Y - Rodar para esquerda
B - Inverter branco com preto
A - Enevoar imagem
C - Fazer cartoon da imagem
F - Aplicar erosao na imagem
H - Aplicar dilatacao na imagem
Z - Adicionar moldura a imagem
G - Guardar Imagem Modificada
Esc- Sair
#####
```

Figura 4 – Menu principal

Após o utilizador ter escolhido a opção que quer executar sobre a imagem (seleccionando uma tecla específica sobre esta) poderão aparecer novos menus tendo em conta cada transformação/correção. Por exemplo se o utilizador escolher fazer uma rotação para a direita é-lhe pedido num menu secundário o valor dessa rotação.

Para cada operação, é aberta uma nova janela com a imagem resultante dessa alteração, podendo essa imagem modificada ser guardada no sistema, ‘operação’ essa que vai ser explicada detalhadamente mais à frente. Como já foi referido, imagens guardadas podem ser encontradas na pasta ‘Imagens’.

Sempre que o utilizador tente aplicar uma nova transformação/correção a uma imagem irá aparecer um menu que permite ao utilizador escolher a que imagem pretende aplicar essa transformação.

```
#####
Imagem que pretende alterar?
1 - Imagem Original
2 - Imagem a preto e branco
3 - Imagem estilizada
4 - Imagem Espelhada
5 - Imagem com Rotacao
6 - Imagem com Threshold
7 - Imagem redimensionada
8 - Imagem com cores invertidas
9 - Imagem enevoadada
10 - Imagem erodida
11 - Imagem dilatada
12 - Imagem com efeito a Lapis
13 - Cartoon
14 - Imagem com moldura
0 - Voltar atras
#####
Opcao:
```

Figura 5 – Menu secundário em que o utilizador pode escolher que imagem alterar

Pode aplicar o efeito à imagem original ou a uma imagem já com um outro efeito aplicado, ficando 3 janelas abertas (imagem original, imagem com 1 efeito aplicado, imagem com 2 efeitos aplicados), podendo o mesmo ser repetido para outras transformações.

O utilizador pode aplicar, no máximo, a uma imagem 14 transformações/correcções.

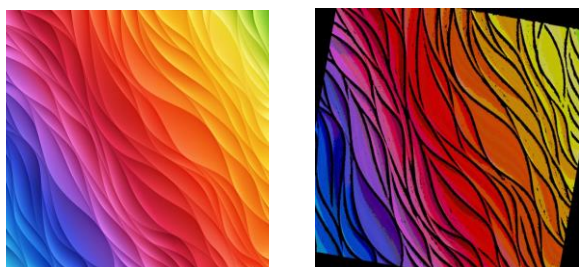


Figura 6 – À esquerda, imagem original, à direita, imagem cartoonizada e posteriormente rodada

IV. FUNCIONALIDADES

A. Mudar imagem

Esta opção é carregada após o utilizador premir a tecla ‘l’ no menu principal. Todas as janelas com imagens abertas são fechadas. O utilizador pode agora escolher outra imagem para efetuar transformações ou correcções.

B. Estilização de imagem

Esta transformação pode ser feita após o utilizador premir a tecla ‘k’ no menu principal.

Para utilizar esta técnica de transformação foi necessário incluir uma nova biblioteca do OpenCV - *opencv2/photo.hpp*.

Esta técnica pretende produzir imagens digitais com uma grande diversidade de efeitos não focados em fotorrealismo. Usa filtros de detecção de arestas e ignora regiões de baixo contraste enquanto preserva, ou evidencia características da imagem com alto contraste.



Figura 6 – Imagem original e imagem estilizada

C. Aplicar efeito a lápis na imagem

O utilizador pode processar esta operação premindo a tecla ‘o’ no menu principal. Como o próprio nome da transformação indica, consiste em aplicar um efeito a lápis na imagem. Após a escolha desta opção, é pedido ao utilizador para escolher a imagem a que quer aplicar o efeito. O utilizador pode escolher também se pretende gerar, com o efeito a lápis, uma imagem a preto e branco ou a cores.



Figura 7 – Imagem original e imagem com efeito a lápis a preto e branco

D. Espelhar a imagem

Se o utilizador desejar espelhar a imagem tudo o que tem a fazer é escolher a opção correspondente, tecla ‘e’, quando este se encontra no menu principal. Após escolher esta opção, tem ainda de escolher se deseja que a imagem seja espelhada em relação ao eixo x (0), em relação ao eixo y (1), ou então, em relação a ambos os eixos (2). A imagem resultante da ação de espelhamento será apresentada numa nova janela. Esta operação está associada ao módulo “[flip](#)” que tem o intuito de girar um array de entrada numa de três formas diferentes (vertical (flipCode = 0), horizontal (flipCode > 0) ou ambas (flipCode < 0)).

```
0 - Efeito espelhamento em relacao ao eixo x
1 - Efeito espelhamento em relacao ao eixo y
2 - Efeito espelhamento em relacao a ambos os eixos [x/y]
Opcao:
```

Figura 8 – Menu secundário para ação de espelhamento.

Se escolhermos a opção de girar em ambos os eixos, é suposto termos uma imagem resultante que consiste numa operação de flip horizontal seguida de uma operação de flip vertical.



Figura 9 – Imagem original e resultante após espelhamento em relação a ambos os eixos

E. Operação de Threshold

Para fazer threshold de uma imagem, é necessário selecionar a opção ‘t’ quando no menu principal. O threshold é o método de segmentação mais simples. A função threshold, a partir de um limiar, filtra uma imagem executando um dos seus cinco operadores. De acordo com cada tipo de threshold, é possível selecionar parte da imagem e decidir descartar ou substituir por um valor. Para selecionar os píxeis relevantes, é realizada uma comparação de cada valor de intensidade do píxel com um valor de threshold (limite).

Depois de selecionada esta transformação, surge um menu secundário com as seguintes opções para a execução do treshold:

```
#####
0 - Treshold Binario
1 - Treshold Binario Invertido
2 - Treshold Truncado
3 - Treshold para zero
4 - Treshold para zero invertido
Opcao:
```

Figura 10 – Menu secundário Treshold

- **Binário:** É requerido ao utilizador um valor de threshold e um valor máximo para os píxeis da imagem. Se o valor do pixel for maior do que o valor de threshold, então o novo valor do pixel passa a ser o valor máximo. Caso contrário, o valor é colocado a zero;



Figura 11 – Imagem Treshold binário com valor máximo 220

- **Binário Invertido:** É requerido ao utilizador um valor de threshold e um valor máximo para os píxeis da imagem. Se o valor do píxel for maior do que o valor de threshold, então o novo valor do píxel passa a ser 0. Caso contrário, passa a ser o valor máximo;



Figura 12 – Imagem Treshold binário invertido com valor máximo 200

- **Truncado:** É pedido ao utilizador um valor máximo. Se o valor de um píxel ultrapassar o máximo definido pelo utilizador, então o seu valor passa a ser 0. Caso contrário, o seu valor não é alterado.

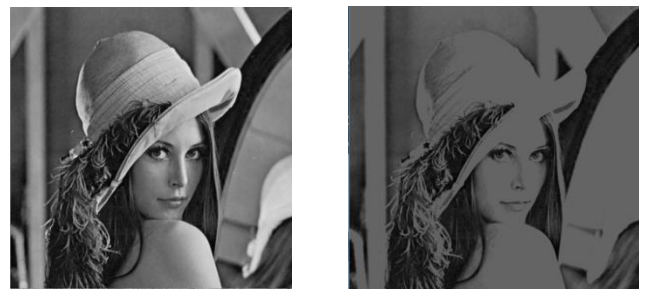


Figura 13 – Imagem Treshold Truncado com valor 100

- **Threshold para Zero:** É pedido ao utilizador um valor máximo. Se o valor de um píxel for menor do que o máximo, então o seu valor passa a ser zero. Caso contrário, o seu valor mantém-se.



Figura 14 – Imagem Treshold para zero com valor 100

- Threshold para Zero Invertido: É pedido ao utilizador para inserir um valor máximo. Se o valor de um píxel for maior do que o máximo, então o seu valor passa a ser zero. Caso contrário, mantém-se.



Figura 15 – Imagem Treshold para zero invertido com valor 100

F. Redimensionar imagem

É possível alterar as dimensões da imagem uniformemente ou não uniformemente, tal como pedido no enunciado do projeto. Para isso, após escolher a opção de alterar as dimensões da imagem (premindo a tecla 's' no menu principal), é pedido ao utilizador se deseja fazer um redimensionamento uniforme ou não.

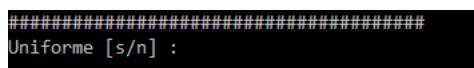


Figura 16 – Menu secundário alterar dimensões imagens

Se o utilizador escolher o redimensionamento uniforme, tem de indicar a percentagem de proporção com que pretende fazer o redimensionamento sendo permitidos valores superiores a 100, que significam que irão ser aumentadas as dimensões da imagem de acordo com o valor introduzido.

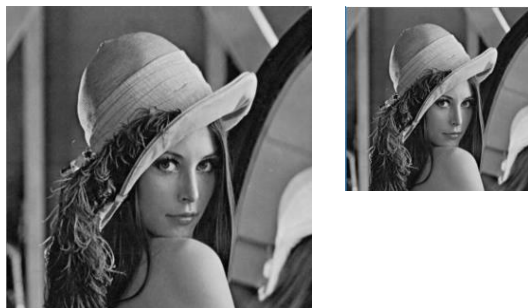


Figura 17 – Imagem Original e imagem redimensionada uniformemente com proporção de 50%

Por outro lado, se o utilizador desejar alterar as dimensões da imagem de forma não uniforme, deve introduzir os valores de largura e altura que pretender.



Figura 18 – Imagem Original e imagem redimensionada não-uniformemente com largura superior à altura

É importante referir que são utilizados dois modos diferentes de interpolação: INTER_AREA quando é diminuída a imagem e INTER_CUBIC quando a imagem é aumentada.

G. Rotação de imagens

A aplicação permite ainda que a imagem sofra uma rotação de valor desejado pelo utilizador. É possível executar rotação à direita ou à esquerda. Uma rotação à direita é solicitada quando o utilizador premir a tecla 'x' no menu principal. Por outro lado, uma rotação à esquerda é solicitada quando o utilizador premir a tecla 'y' também no menu principal. Após a escolha do tipo de rotação a ser feita, é necessário introduzir o valor (em graus) da rotação que se pretende executar. Este valor vai ser introduzido na função *getRotationMatrix2D*.

O utilizador pode voltar a rodar a imagem (para a direita ou para a esquerda), para isso só tem de repetir o processo anterior, sendo o novo valor introduzido somado (para a direita), ou subtraído (para a esquerda) ao valor anterior.



Figura 19 – Imagem Original e imagem rodada 45° à direita

H. Tornar imagem preto e branco

Foi implementada também a operação de tornar a imagem lida numa imagem a preto e branco. O módulo usado foi [“cvtColor”](#) que tem o intuito de converter uma imagem com uma gama de cores para outra, usando neste caso a conversão ‘CV_RGB2GRAY’ se a imagem tiver 3 canais e ‘CV_RGBA2GRAY’ se a imagem tiver 4 canais. magens que já se encontrem a preto e branco não são alteradas.

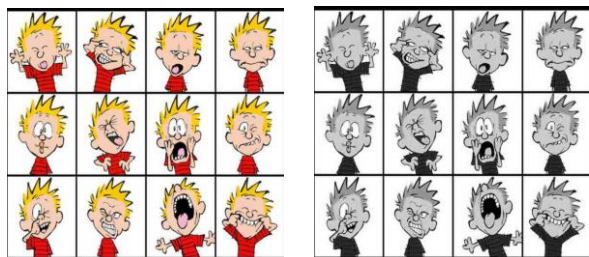


Figura 20 – Imagem Original e imagem tornada a preto e branco

I. Inverter preto com branco

Esta operação pode ser executada a partir do menu principal após o utilizador premir a tecla ‘b’. Implementa o módulo [“bitwise_not”](#) que tem o intuito de inverter cada bit de um array, neste caso, inverte cada píxel da imagem de entrada.



Figura 21 – Imagem Original e imagem invertida preto com branco

J. Tornar Imagem enevoada

Esta operação está acessível após a tecla ‘a’ ser premida no menu principal pelo utilizador. Optámos por usar e implementar o módulo de aplicação de filtros de mediana já utilizado nas aulas práticas, [“medianBlur”](#).

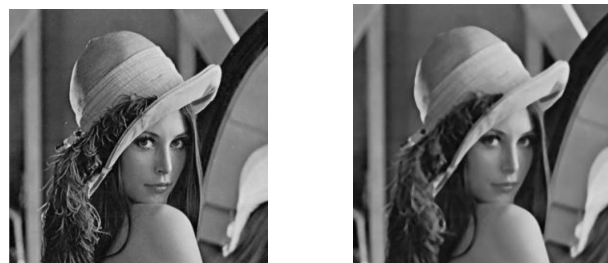


Figura 22 – Imagem Original e imagem com filtro de mediana

K. Aplicar filtro de cartoon na Imagem

Esta operação está acessível após a tecla ‘c’ ser premida no menu principal pelo utilizador. Esta é talvez a transformação mais complexa, pois é aquela que envolve um maior número de sequências de transformações até chegar ao resultado final. É aplicado o filtro de mediana ([medianBlur](#)) já referido, o módulo de [“adaptiveTreshold”](#) e o módulo de [“bitwise_and”](#) que calcula a conjunção bitwise por elemento de dois arrays (neste caso, duas imagens, a imagem original sem transformações e a imagem com as transformações anteriores aplicadas) ou de um array e um escalar.



Figura 23 – Imagem original e imagem com filtro de cartoon

L. Adicionar moldura à imagem

Esta operação está acessível após a tecla ‘z’ ser premida no menu principal pelo utilizador. Nesta transformação, é adicionada uma moldura com dimensões constantes à imagem sendo que o utilizador pode escolher uma de duas opções possíveis:

1. Adicionar à imagem uma moldura com uma cor estática, sendo que essa cor muda de imagem para imagem. (*borderType = BORDER_CONSTANT*);
2. Adicionar à imagem uma moldura formada pela réplica dos píxeis das bordas da imagem original. (*borderType = BORDER_REPLICATE*);



Figura 24 – À esquerda, imagem com moldura de cor constante, à direita, imagem original

M. Aplicar erosão na imagem

É possível aplicar erosão à imagem premindo a tecla ‘f’ no menu principal.

Esta correção pode ser aplicada diretamente à imagem original e/ou após lhe ser aplicadas transformações, permitindo assim atenuar eventuais artefactos nas imagens resultantes.

A principal utilidade da função ‘*erode*’ é eliminar ruídos numa imagem, desgastando-os de forma que não afete outras regiões (necessariamente maiores), com conteúdo relevante.

Foi escolhido a forma ‘*MORPH_ELLIPSE*’ para o nosso kernel.

O utilizador pode escolher que valor pretende para o tamanho do kernel, ou seja, sempre que esta opção for escolhida, é pedido ao utilizador um valor para o *size*.

Esse valor entra na seguinte função:

```
getStructuringElement(erosion_type,
Size(2*size+1,2*size+1),Point(size,
size));
```

N. Aplicar dilatação na imagem

É possível aplicar ‘*dilate*’ à imagem premindo a tecla ‘h’ no menu principal.

Esta correção pode ser aplicada diretamente à imagem original e/ou após lhe ser aplicadas transformações, permitindo assim atenuar eventuais artefactos nas imagens resultantes.

A principal utilidade da função ‘*dilate*’ é juntar os píxeis desconexos em áreas próximas com o mesmo tom. A função aumenta o brilho da imagem, fundindo os pontos adjacentes numa única forma do mesmo tom.

O algoritmo funciona de forma análoga à função ‘*erode*’, utilizando também um kernel, com uma forma e tamanho fixo.

Tal como no módulo ‘*erode*’, também foi escolhido a forma ‘*MORPH_ELLIPSE*’ para o nosso kernel.

O utilizador pode escolher que valor pretende para o tamanho do kernel, ou seja, sempre que esta opção for escolhida é pedido ao utilizador um valor para *size*. Esse valor entra na seguinte função:

```
getStructuringElement(erosion_type,
Size(2*size+1,2*size+1),Point(size,
size));
```

O. Guardar Imagem resultante

A última operação a ser implementada, e embora não seja propriamente uma operação sobre imagens, é a opção de guardar as imagens resultantes. Foi optado por colocar esta opção pois achámos importante para comparar imagens, tirar apontamentos, bem como fazer a análise dos resultados das diversas operações implementadas.

Portanto, se o utilizador quiser guardar alguma imagem gerada, tem de premir a tecla ‘g’ no menu principal.

Ao seleccionar esta opção é perguntado ao utilizador qual a das imagens resultantes, este pretende guardar.

```
#####
Imagem que pretende guardar?
1 - Imagem Original
2 - Imagem a preto e branco
3 - Imagem estilizada
4 - Imagem Espelhada
5 - Imagem com Rotacao
6 - Imagem com Threshold
7 - Imagem redimensionada
8 - Imagem com cores invertidas
9 - Imagem enevoadada
10 - Imagem erodida
11 - Imagem dilatada
12 - Imagem com efeito a Lapis
13 - Cartoon
14 - Imagem com moldura
0 - Voltar atras
#####
```

Figura 25 – Menu secundário da imagem a ser guardada

Se a imagem em questão não tiver sido gerada (verificação por operação), esta ação não é permitida e é apresentado ao utilizador uma mensagem de erro.

```
#####
Opcao: 4
Nome com que pretende guardar (s/extensao): lena_invalida
Imagem nao foi aberta, impossivel guardar
```

Figura 26 – Mensagem erro, imagem não gerada

Se a imagem que o utilizador pretende guardar tiver de facto sido gerada, é então proposto ao utilizador a escolha de um nome para esta (s/extensão).

No caso da escolha do nome para imagem coincidir com um nome já existente na pasta ‘Imagens/’, é perguntando ao utilizador se deseja mesmo substituir a imagem/ficheiro existente ou se pretende abortar a substituição e escolher outro nome para a imagem.

No fim, se tudo estiver correto, a imagem é guardada com sucesso e pode ser encontrada na pasta 'Imagens'.

V. CONCLUSÃO

O principal objetivo do projeto foi atingido pois foi desenvolvida uma aplicação capaz de aplicar diferentes transformações/correções a imagens bem como funcionalidades de pré-e-pós-processamento que permitem atenuar eventuais artefactos nas imagens resultantes.

Optámos por adicionar algumas operações que não eram inicialmente solicitadas no guião como Threshold, Grayscale, aplicação de efeitos de cartoon e lápis, espelhamento de imagens, etc, pois concluímos que eram operações bastante convenientes e seguiam o objetivo do nosso projeto.

Foi um projeto bastante interessante de realizar pois permitiu-nos utilizar os conhecimentos adquiridos ao longo do semestre e expandir os nossos conhecimentos sobre processamento de imagens.

VI. REFERÊNCIAS

http://docs.opencv.org/2.4/modules/imgproc/doc/geometric_transformations.html#geometric-image-transformations

http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/warp_affine/warp_affine.html#affine-transformations

http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html#canny-edge-detector

<http://docs.opencv.org/2.4/doc/tutorials/imgproc/threshold/threshold.html#basic-thresholding-operations>

http://docs.opencv.org/2.4/doc/tutorials/introduction/load_save_image/load_save_image.html

http://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html#cvtColor