



ISEL – INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
ADEETC – ÁREA DEPARTAMENTAL DE ENGENHARIA DE
ELECTRÓNICA E TELECOMUNICAÇÕES E DE COMPUTADORES

LEIM

LICENCIATURA EM ENGENHARIA INFORMÁTICA E MULTIMÉDIA
UNIDADE CURRICULAR DE PROJETO

PFC Rubik's Cube

(eventual) imagem ilustrativa do trabalho – *dimensão*: até 13cm x 4cm

Carlos Rosa 46348

João Cunha 45412

Miguel Matos 45423

Orientador

Professor [Doutor] António Teófilo

Setembro, 2021

Resumo

O nosso trabalho baseia-se em criar um website onde vários utilizadores podem aprender a fazer o cubo mágico e disputar entre si em competições de tempo.

As principais motivações para a realização deste trabalho foram os vários websites para a interacção com o cubo mágico e os vídeos de implementação do mesmo com o JavaScript. A partir dos websites nós adquirimos uma ideia básica do que o nosso website deve ou não possuir e qual deveria ser a interacção com o cubo mágico. Por outro lado, os vídeos de implementação do cubo mágico com o JavaScript ajudaram-nos a ter uma percepção do funcionamento do cubo mágico e concederam-nos novas funcionalidades que poderíamos implementar.

As funcionalidades chave que destacam o nosso projecto de outros já existentes é a possibilidade de o utilizador reproduzir o seu cubo mágico físico no cubo mágico virtual através de reconhecimento por câmara e a competição de tempo para a resolução do cubo mágico. Estas implementações permitem um aprendizado mais prático para o utilizador e um motivo para continuar a fazer uso do website após esse aprendizado estar completo.

Concluindo, o nosso projecto foi inspirado por outros websites existentes e destacasse dos mesmos através das novas funcionalidades que melhoram e aprimoram o aprendizado do utilizador.

Agradecimentos

Gostaríamos de agradecer o nosso Orientador António Teófilo por ternos ajudado ao longo do nosso projecto.

Índice

Resumo	i
Agradecimentos	iii
Índice	v
Lista de Tabelas	vii
Lista de Figuras	ix
1 Introdução	1
2 Modelo Proposto	3
2.1 Requisitos	3
2.2 Fundamentos	4
3 Implementação do Modelo	7
3.1 Acesso à câmara	7
3.2 Análise e transformações de cada frame do vídeo	7
3.3 Detecção do cubo	9
3.4 Detecção das cores	13
4 Validação e Testes	17
4.1 Ordenação dos contornos do cubo	17
4.2 Espaço de cores RGB e YUV	18
5 Conclusões e Trabalho Futuro	21
A Arquitectura geral	23

B	Arquitetura específica	25
C	Diagrama de sequência	27

Lista de Tabelas

4.1	Tabela das distâncias entre quadrados adjacentes horizontalmente	17
4.2	Tabela das distâncias entre quadrados adjacentes verticalmente	18

Lista de Figuras

3.1	Aumento do brilho da imagem.	8
3.2	Resultado das operações morfológicas.	9
3.3	Distâncias entre os cubos para diferentes tamanhos.	12
3.4	Contornos estimados	13
3.5	Espaço de cores YUV	14
3.6	Detecção das cores	15
4.1	Teste com o espaço de cores RGB	18
4.2	Teste com o espaço de cores YUV	19
A.1	Arquitetura geral.	23
B.1	Diagrama de sequência: Câmara.	25
B.2	Diagrama de sequência: Cubo 3D.	25
B.3	Diagrama de sequência: Cubo 2D.	26
C.1	Diagrama de sequência: Resolução do cubo.	27
C.2	Diagrama de sequência: Detecção do cubo com a câmara.	28

Capítulo 1

Introdução

Este trabalho baseia-se em criar um website onde é possível aprender a resolver o cubo mágico e aplicar este aprendizado para competir com outros utilizadores em concursos de tempo onde cada um tenta resolver o cubo mágico no menor tempo possível.

A principal motivação deste projecto é a de conferir uma melhor interacção com o utilizador através de ferramentas e funcionalidades que facilitem o seu aprendizado e explorem o interesse do mesmo em continuar a utilizar o website. Para atingir este objectivo foi conferida a possibilidade do utilizador implementar o seu cubo mágico físico no cubo mágico do website através do reconhecimento pela câmara ou pela configuração manual do utilizador e foi implementado um desafio de tempo onde cada utilizador tenta resolver o cubo mágico no menor tempo possível para competir com outros utilizadores. Estas funções distinguem este projecto de outros websites que também implementam o cubo mágico e funcionam como o maior destaque entre os seus competidores.

O desenvolvimento deste trabalho foi constituído em três partes que foram divididas pelos membros do grupo. A primeira parte encapsulava as regras e a lógica por de trás dos movimentos e da resolução do cubo mágico, que foram testadas inicialmente em um cubo 2D antes de ser implementada no projecto final. A segunda parte constituía o desenvolvimento do cubo 3D, configuração manual do cubo e o desafio de tempo, estas funcionalidades foram validadas consoante se o cubo apresentava o formato e as cores corretas, se a configuração manual implementava o que o utilizador pretendia e se o desafio de tempo guardava o tempo correto respectivamente. Por último,

a terceira parte baseava-se em configurar o reconhecimento do cubo pela câmara do utilizador, através da determinação das cores de cada face com o objectivo de reconstruir o cubo.

Concluindo, este relatório começará por demonstrar outros trabalhos relacionados este projecto, explicar o modelo sobre o qual ele foi construído e demonstrar a implementação e os testes executados.

Capítulo 2

Modelo Proposto

Neste capítulo vai ser demonstrado o modelo sobre o qual este trabalho foi construído, os casos de utilização, os fundamentos teóricos e tecnológicos e os principais contributos para o desenvolvimento deste projecto.

2.1 Requisitos

Os requisitos essenciais para este trabalho incluem a interacção com o cubo mágico, a configuração manual e através da câmara do cubo mágico, resolução automática e o desafio de tempo.

A interacção com o cubo mágico deve ser simples e fácil de aprender a utilizar. Esta função é a mais importante do projecto, pois permite ao utilizador interagir com o cubo mágico sem a necessidade de possuir um ele mesmo. A maioria dos utilizadores deste site farão uso dele para aprender a resolver o cubo mágico, por consequente esta função melhorará a experiência dos utilizadores uma vez que a mesma permite que eles experimentem as suas ideias e vejam os resultados das mesmas quando interagem com o cubo mágico.

A configuração manual e através da câmara do cubo mágico devem ser fáceis de aprender a usar e adaptáveis para qualquer ambiente que o utilizador se encontrar. Estas funções permitem ao utilizador adaptar o cubo mágico para qualquer configuração que o mesmo desejar. Caso um utilizador procurasse reconstruir o cubo mágico para uma configuração específica de forma a que pudesse receber conselhos ou determinar a resolução do cubo, ele fará uso destas funções que melhoram a interacção entre o utilizador e o

website e fortalecem a aprendizagem do mesmo.

A resolução automática deve funcionar para todas as configurações possíveis do cubo e estar no completo controlo do utilizador para que o mesmo possa acompanhar a resolução. Esta função ajudará o utilizador a perceber a lógica por de trás da resolução do cubo mágico de um forma simples e eficaz. Um utilizador que esteja interessado em utilizar o nosso website procurará aprender a resolver o cubo mágico, caso o mesmo não possua conhecimento para o fazer, e fará uso da resolução automática para aprender passo por passo a solução para resolver o cubo.

O desafio de tempo deve guardar um vídeo do utilizador a resolver o cubo e guardá-lo no servidor para que possa ser posteriormente analisado. Um utilizador experiente na resolução do cubo mágico não estará interessado nas funções previamente mencionadas, por consequente esta função foi criada para entreter esses utilizadores. O nosso website fornece um sistema de classificação onde é possível verificar os utilizadores que conseguiram completar o cubo mágico no menor tempo possível, qualquer utilizador experiente que esteja interessado em entrar para os melhores do website e competir com outros utilizadores vai procurar utilizar esta função para demonstrar as suas habilidades.

Concluindo, estes são os requisitos mínimos que o nosso projecto deve implementar para estar considerado completo.

2.2 Fundamentos

Aqui o sustento formal (teórico / tecnológico) do trabalho realizado ... Para implementar os requisitos mencionados na secção 2.1 foi necessário estudar algumas tecnologias que fossem capaz de executar o que era pretendido.

Na interacção com o cubo mágico e a resolução automática foi necessário estudar várias formulas de resolução para o cubo mágico de forma a que fosse possível implementá-las para qualquer que fosse a configuração do cubo. Por outro lado, existiu a necessidade de criar uma estrutura onde todas as informações do cubo, face, posição e cor, pudessem guardadas e apresentadas em um ambiente 2D e 3D.

A configuração manual e detecção por câmara do cubo mágico devem acessar a estrutura de dados referida anteriormente de forma a implementar

a configuração do utilizador ou o cubo detectado pela câmara. A configuração manual vai alterar o cubo quadrado por quadrado até o mesmo ficar com a composição desejada pelo utilizador. Por outro lado, a detecção por câmara deve reconhecer o cubo do utilizador através de várias transformações de imagens que tornam a detecção adaptável para qualquer ambiente que o utilizador se encontrar.

Capítulo 3

Implementação do Modelo

3.1 Acesso à câmara

Para obter acesso à câmara, é necessário que o utilizador forneça a sua permissão para o programa obter os dados recebidos por ela. Com estes dados, é utilizado a função `cv.VideoCapture` para transformar as frames recebidas pela câmara em um objecto `Mat` que é o objecto base usado em todas as funções do `OpenCV.js` para a transformações de imagem, este objecto guarda os valores dos pixels da frame em vários formatos de dados para serem posteriormente analisados por diferentes funções.

3.2 Análise e transformações de cada frame do vídeo

Após os valores das frames recebidas pela câmara do computador serem obtidas, é necessário analisar e aplicar algumas transformações nas mesmas para que possam ser utilizadas nas páginas da aplicação web e para a detecção do cubo e das cores.

A primeira transformação é o aumento do brilho e do contraste entre as cores da face do cubo. Para facilitar a detecção de cada cor na face de um cubo mágico foi implementado a função “`convertScaleAbs`” que permite manipular o brilho e o contraste das cores na frame através dos valores β e α . O valor β é somado ou subtraído, dependendo do valor do mesmo, a cada pixel da frame de forma a aumentar ou subtrair os seus valores, caso o valor do pixel fique fora da zona de valores entre 0 e 255 ele será saturado e

o seu valor passará a ser 255. O valor α é utilizado para definir o nível de contraste entre as várias cores da frame, um valor inferior a 1 fará com que as cores possuam menos contraste, o que não é desejável, e um valor superior a 1 aumentará o contraste das cores. Esta função foi utilizada para aumentar consideravelmente o contraste entre as cores do cubo para facilitar a detecção dos contornos das mesmas para ambientes com uma iluminação variada.



Figura 3.1: Aumento do brilho da imagem.

Após o aumento do contraste estar efectuado, a frame é transformada para níveis de cinzento para que se possa aplicar o algoritmo de canny para a detenção de contornos. O algoritmo de canny suaviza a frame que analisa e a passa por um filtro de forma a determinar a intensidade dos gradientes dos pixels da mesma, desta maneira este algoritmo localiza as bordas da face do cubo mágico eficientemente.

Por fim, antes que os contornos sejam analisados são efectuadas transformações morfológicas com o objectivo de eliminar ou delimitar contornos desnecessários. A estrutura elementar destas transformações será uma cruz, pois é pretendido determinar os contornos dos quadrados de cada face do cubo mágico e a cruz permitirá realçar esses contornos e eliminar aqueles que não cumprem este critério. A primeira transformação é uma operação de fecho para fechar os contornos que o algoritmo da canny não fechou, de seguida é efectuada uma dilatação para aumentar os contornos que sobreviveram à primeira transformação, depois é executado mais uma vez a operação de fecho para fechar os novos contornos dilatados e por último é realizada

uma erosão para eliminar eventuais ruídos que tenham sobrevivido às transformações anteriores. Com estas transformações é possível preparar a frame para ser analisada e realçar os contornos mais importantes para essa análise.



Figura 3.2: Resultado das operações morfológicas.

3.3 Detecção do cubo

A detecção do cubo será efectuada depois que as frames tenham sido transformadas e preparadas para a análise. Esta detecção basear-se-á em extrair as características dos contornos detectados em cada frame e usá-los para determinar se fazem parte do cubo mágico, a sua posição na face e a sua respectiva cor.

O primeiro passo a ser executado é a eliminação de contornos que não representam a face do cubo mágico. O algoritmo de detecção de contornos de canny irá localizar todos os contornos presentes na frame e não apenas os contornos presentes no cubo mágico, consequentemente, é necessário separar os contornos que representam as faces do cubo mágico dos outros contornos. Este objectivo é atingido ao efectuar a verificação de se o contorno possui características quadradas e se possui um contorno pai. A verificação de se o contorno possui características quadradas é efectuada ao extrair a área, A , e o perímetro, P , do contorno e aplicar a fórmula $P^2/16$ e comparar o seu resultado com a área. Esta fórmula advém da característica do cubo de que todos os seus lados possuem o mesmo comprimento, ou seja, a fórmula da área e do perímetro podem ser traduzidas da seguinte forma:

$$A = l * l = l^2 \quad (3.1)$$

$$P = l + l + l + l = 4l \quad (3.2)$$

Como se pode verificar, ambas as fórmulas podem ser simplificadas ao ponto de apenas precisarem de apenas uma variável l , lado. Portanto podemos fazer uso da fórmula anteriormente mencionada para igualar o valor do perímetro ao valor da área:

$$\frac{P^2}{16} = \frac{(4l)^2}{16} = \frac{16l^2}{16} = l^2 \quad (3.3)$$

A partir do resultado desta equação é apenas preciso compará-lo com o valor da área e verificar se ele se encontra dentro de uma margem de erro de ± 200 . Esta margem de erro foi criada devido ao fato de que os contornos raramente formam um quadrado perfeito. A necessidade de verificar se o contorno possui um contorno pai advém do fato de que a verificação de se o contorno possui características quadradas não é perfeita e pode capturar outros objectos quadrados que o utilizador possa possuir no ambiente, portanto esta verificação reforça a detecção das faces do cubo mágico em relação a outros objectos que possam existir na imagem. Um contorno pai é um contorno que possui dentro de si um ou mais contornos. Todos os cubos mágicos são envolvidos por bordas pretas que separam cada quadrado presentes nas faces do cubo, portanto todos os contornos que representam os quadrados na face do cubo mágico devem possuir um contorno pai.

Após os contornos que podem representar a face do cubo mágico estarem delimitados, será necessário extrair as coordenadas das posições desses contornos e a área que os mesmos ocupam. Estas informações serão necessárias para ordenar os contornos consoante as suas posições na face do cubo mágico para as diferentes distâncias a qual o cubo mágico possa se encontrar da câmara do utilizador e permitirão a estimação das posições dos quadrados na face do cubo mágico que não foram detectas pelo algoritmo de detecção de contornos de canny. Por outro lado, nesta fase também é determinado a cor que os contornos já detectados ocupam na face do cubo mágico através do algoritmo de detecção de cores que será explicado posteriormente.

Quando os contornos que representam o cubo são detectados, na maior parte das vezes eles se encontraram desorganizados, que por consequente cria

a necessidade de organizá-los para que a cor detectada pelo contorno esteja na localização correta. Para organizar os contornos detectados é inicialmente estimado os maiores valores das coordenadas x e y , as coordenadas que representariam o canto inferior direito do cubo mágico.

De seguida, são comparadas as coordenadas dos contornos detectados com os maiores valores das coordenadas x e y para determinar a posição de cada contorno. Caso a diferença entre as coordenadas, x ou y seja, inferior a 10, o contorno estará localizado na última coluna ou última linha do cubo respectivamente. Por outro lado, se a diferença for superior a 10 é verificado a área do contorno e dependendo desse valor é efectuado uma comparação para observar se o contorno pertence à linha do meio ou à coluna do meio. Este algoritmo foi configurado desta forma para que o mesmo fosse adaptável às várias distâncias possíveis que o utilizador poderia colocar o cubo da câmara. Se o valor da diferença for superior ao valor determinada pela área, 66 se a área for inferior a 2000, 77 se for superior a 2000 e inferior a 3000 e 86 se for superior a 3000 e inferior a 4000, o contorno será colocado na primeira linha ou na primeira coluna. Por fim, os valores mencionados foram estimados depois de terem sido efectuados vários testes com o valor da área dos contornos e a distância entre as coordenadas.

Por último, foi adicionado uma função à detecção do cubo para que ela fosse capaz de estimar a posição de quadrados que não foram detectados na face do cubo mágico. Por vezes o algoritmo da detecção do cubo pode falhar e não ser capaz de detectar a face completa do cubo mágico, portanto esta função foi desenvolvida para auxiliar o algoritmo a resolver este problema. Esta estimação apenas é executada se o algoritmo for capaz de obter no mínimo 7 contornos que podem representar o cubo mágico, pois ela é dependente da função para organizar os contornos na face do cubo mágico. A função de organização necessita que pelo menos uma linha completa e uma coluna completa sejam detectadas para que a mesma saiba a localização dos valores extremos na face do cubo mágico, por consequente é preciso que sejam detectados no mínimo 7 contornos numa face uma vez que isso garante que pelo menos uma linha e uma coluna completa são detectadas. A função de estimação faz uso desta característica da função de organização para determinar as posições que precisam ser determinadas e que valores deve usar para a determinação. A estimação começa por procurar o quadrado na face



Figura 3.3: Distâncias entre os cubos para diferentes tamanhos.

que precisa ser estimado, ao encontra – lo ela aplica um de dois métodos consoante as condições da situação para encontrar a localização do quadrado. O primeiro método faz uso dos contornos que se encontram na mesma linha do quadrado a ser estimado, neste caso a posição y seria a média dos outros dois contornos e o cálculo da posição x vai depender do lugar que está a ser estimado. Por outro lado, existe a possibilidade de um dos contornos localizados na mesma linha que o quadrado a ser estimado também precisar de ser estimado, neste caso é feito uso dos contornos pertencentes à mesma coluna que o quadrado a ser estimado. Quando são usados os contornos pertencentes à mesma coluna em vez dos contornos pertencentes à mesma linha, a posição x é a média dos dois contornos e o cálculo da posição y vai depender do lugar

que está a ser estimado. Por fim, quando todos os quadrados em falta forem estimados é aplicado o algoritmo de detecção de cor para determinar a cor das localizações estimadas.

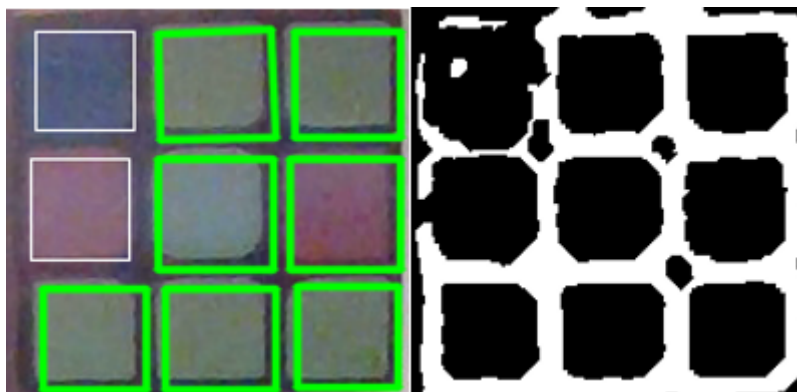


Figura 3.4: Contornos estimados

3.4 Detecção das cores

O algoritmo de detecção das cores é executado durante a primeira determinação dos contornos que fazem parte do cubo e durante a estimação dos contornos em falta. Este algoritmo permitirá que seja possível determinar a cor de cada quadrado na face do cubo mágico a partir dos contornos que representam os mesmos.

Antes que o algoritmo seja executado, logo após o utilizador começar a detecção do cubo, é pedido que seja mostrado cada uma das cores do cubo mágico na câmara do computador do utilizador. Este passo é de extrema importância para a detecção das cores, pois permite que o algoritmo possa se adaptar a qualquer ambiente em que se encontre. Durante esta fase é disponibilizado ao utilizador algumas ferramentas para alterar a luminosidade das imagens recebidas para que o algoritmo possa funcionar em ambientes mais escuros onde a distinção entre as cores não é tão clara. Quando uma cor é mostrada à câmara do utilizador, o algoritmo faz uma média dos pixels que representam essa cor em dois espaços de cores, RGB e YUV. O espaço de cores RGB é utilizado porque é o espaço que o utilizador vai visualizar quando for mostrar as cores do seu cubo para a câmara e ajustar a luminosidade da

mesma, porém este espaço de cores sozinho é sujeito a erros devido ao facto de que o olho humano é mais sensível à luminância do que à cromaticidade das cores, que por consequente dificulta a detecção delas quando o utilizador está a alterar a luminosidade da imagem para tornar as cores mais perceptíveis na sua visão, pois pode diminuir o contraste das mesmas nos canais RGB. Porém, o espaço de cores YUV tem em consideração a percepção humana, ou seja, favorece mais a luminância do que a cromaticidade e esta característica torna este espaço de cores uma mais – valia para a detecção das mesmas, pois quando as cores se tornam mais distintas no olhar do utilizador, elas também se tornam mais distintas neste espaço de cores. Após o algoritmo obter os resultados das médias dos valores dos pixels nos dois espaços de cores, ele guarda esses valores para serem usados posteriormente durante a detecção do cubo.



Figura 3.5: Espaço de cores YUV

O primeiro passo que o algoritmo executa é a determinação do rectângulo que delimita toda a área encobrida pelos contornos detectados e estimados com o objectivo de aplicar a função “roi” do OpenCV que devolve a secção da imagem onde o rectângulo delimitador se encontra. A partir desta secção de imagem é possível calcular a média de todos os pixels presentes nela e obter o valor médio da cor do contorno a ser correntemente analisado. Este valor médio, tal como na tarefa de mostrar as cores do cubo mágico à câmara do utilizador, é calculado nos espaços de cor RGB e YUV.

O segundo passo que o algoritmo executa é o cálculo da distância entre os valores obtidos na tarefa executada pelo utilizador e os valores obtidos na detecção do cubo. A fórmula da determinação das cores do cubo mágico

funciona como um cálculo da distância euclidiana entre os resultados observados, onde os valores do espaço de cor RGB são calculados usando apenas os resultados pertencentes a esse espaço de cor e os valores do espaço de cor YUV são calculados da mesma maneira. O cálculo da distância euclidiana é efectuado com o objectivo de estimar de forma directa o quão longe a cor detectada se encontra das cores guardadas, basicamente o raciocínio por detrás deste processo é de que os valores que representam as mesmas cores vão possuir uma distância menor do que os valores que representam cores diferentes. Este cálculo é efectuado usando os canais de cada espaço de cores, red, blue e green para o RGB e luma, blue projection e red projection para o YUV, como se fossem coordenadas de um ponto num espaço 3D, desta maneira é possível estimar a distância entre pontos detectados e os pontos guardados. Por fim, após as distâncias terem sido calculadas é efectuada uma média entre a distância obtida para o espaço de cor RGB e a distância para o espaço de YUV, este processo é efectuado para todas as cores guardadas e a cor que obter o menor valor entre todas as outras é a escolhida pelo o algoritmo para representar o contorno detectado ou estimado.

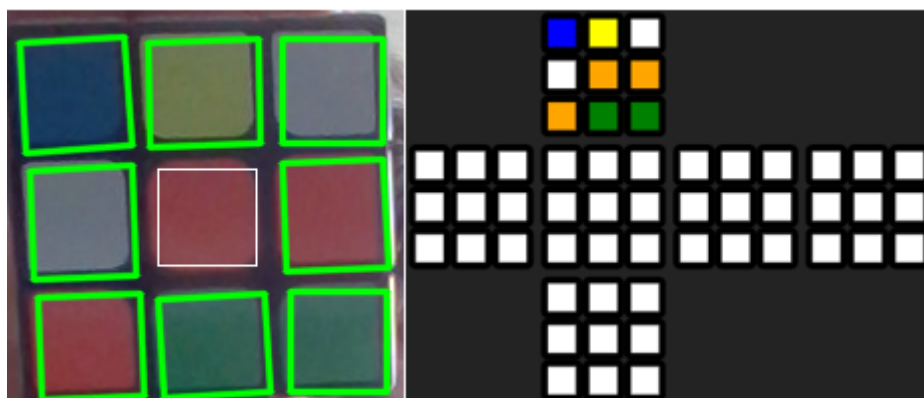


Figura 3.6: Detecção das cores

Capítulo 4

Validação e Testes

4.1 Ordenação dos contornos do cubo

Durante a explicação sobre o funcionamento da detecção do cubo, 3.3, foi referido a importância da ordenação dos contornos que representam a face do cubo mágico e o método utilizado para realizar esta tarefa, a estimação das coordenadas do canto inferior direito da face e a distância entre os quadrados da mesma.

A distância entre os quadrados de uma face é uma informação importante a se ter durante esta ordenação, pois permite estimar a posição dos outros contornos com base no quadrado localizado no canto inferior direito da face que foi determinado a partir dos maiores valores x e y dos contornos detectados. Afim de adquirir esta informação, foram efectuados alguns testes para descobrir a distância máxima a qual quadrados adjacentes horizontalmente e verticalmente se encontram para várias áreas de forma a que a detecção do cubo fosse adaptável à sua distância da câmara do utilizador.

<i>Área</i>	< 2000	< 3000	< 4000
<i>max x</i>	55	66	79
<i>max y</i>	1	2	5

Tabela 4.1: Tabela das distâncias entre quadrados adjacentes horizontalmente

Como se pode verificar, tanto a distância vertical, y , para quadrados na mesma linha como a distância horizontal, x , para quadrados na mesma coluna

<i>Área</i>	< 2000	< 3000	< 4000
<i>max x</i>	2,5	6	7
<i>max y</i>	56	68	80,5

Tabela 4.2: Tabela das distâncias entre quadrados adjacentes verticalmente

são sempre inferiores a 10 para todas as áreas e para uma área inferior a 2000 todas as distâncias são inferiores a 66, para uma área inferior a 3000 todas as distâncias são inferiores a 77 e para uma área inferior a 4000 elas são inferiores a 86, reforçando a explicação sobre os valores utilizados na secção 3.3.

4.2 Espaço de cores RGB e YUV

Na secção 3.4 foi explicado o processo pelo o qual o algoritmo de detecção de cores executa as suas funções através do uso do espaço de cores YUV em vez do espaço de cores RGB devido às características do YUV que o torna mais favorável que o RGB. Nesta secção será apresentado os testes que demonstram a vantagens da utilização do YUV.

O espaço de cores RGB favorece a crominância, a mudança no valor das cores, porém como o olho humano não é muito perceptível à variância das cores esta característica não possui muito impacto e pode atrapalhar a detecção das mesmas quando existem cores parecidas na mesma face.

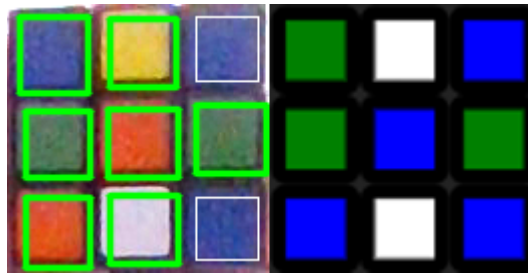


Figura 4.1: Teste com o espaço de cores RGB

Como se pode observar, as cores mais escuras como o laranja, azul e verde são confundidas como sendo a mesma cor, da mesma forma as cores mais claras como o amarelo e o branco também são detectadas como sendo

da mesma cor. O espaço de cores YUV favorece a luminância tal como o olho humano e por consequente consegue distinguir melhor as cores que se apresentam mais distintas nos olhos do utilizador.

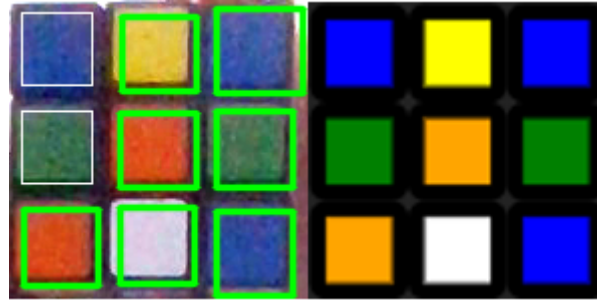


Figura 4.2: Teste com o espaço de cores YUV

Tal como é apresentado, neste espaço as cores são correctamente distinguidas e não existe nenhuma confusão por parte do algoritmo de detecção de cores.

Capítulo 5

Conclusões e Trabalho Futuro

Nós conseguimos realizar todas as nossas ambições iniciais para o projecto e a maioria dos objectivos que foram surgindo ao longo do percurso.

O nosso projecto é capaz de conferir o conteúdo dos outros trabalhos do qual foi inspirado e fornecer algo mais para ajudar e entreter os utilizadores. Apreendemos a fazer uso de tecnologias recentes, tecnologias que já havíamos aprendido mas aplicadas para a linguagem JavaScript e como combiná-las para atingir os nossos objectivos.

Concluindo, apreendemos fazer uso de ferramentas já conhecidas de novas formas e pretendemos aplicar este conhecimento em futuros trabalhos.

Apêndice A

Arquitectura geral

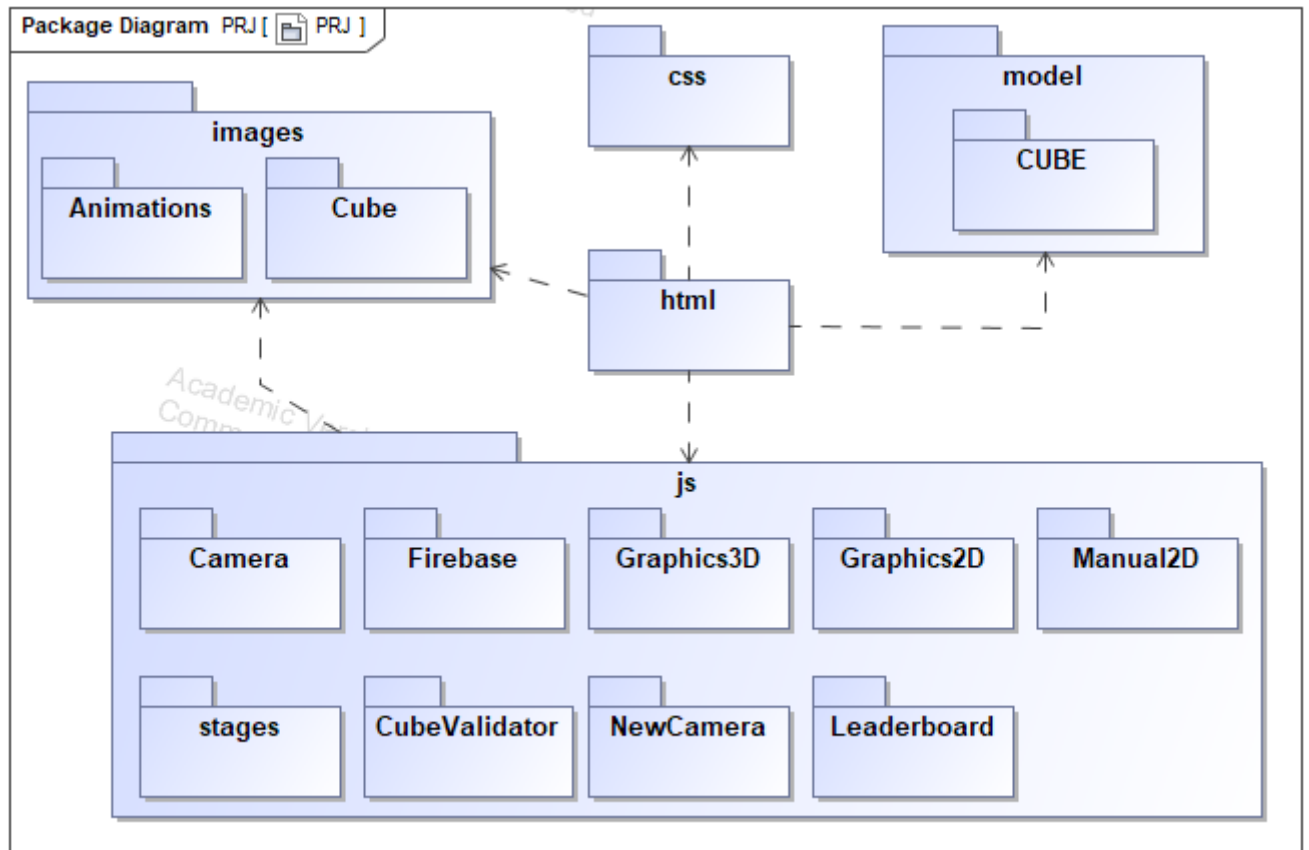


Figura A.1: Arquitectura geral.

Apêndice B

Arquitectura específica

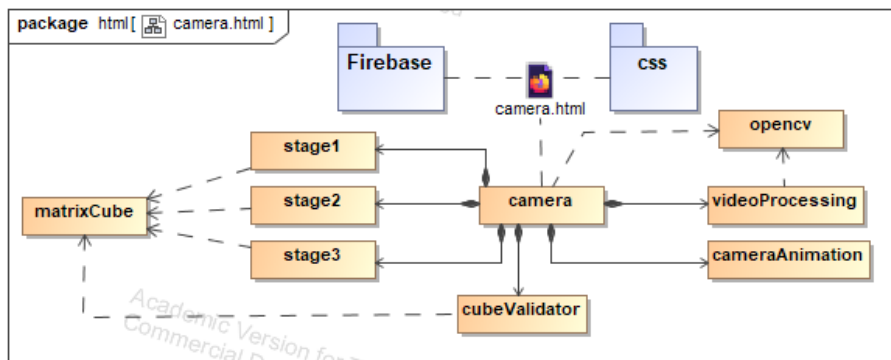


Figura B.1: Diagrama de sequência: Câmera.

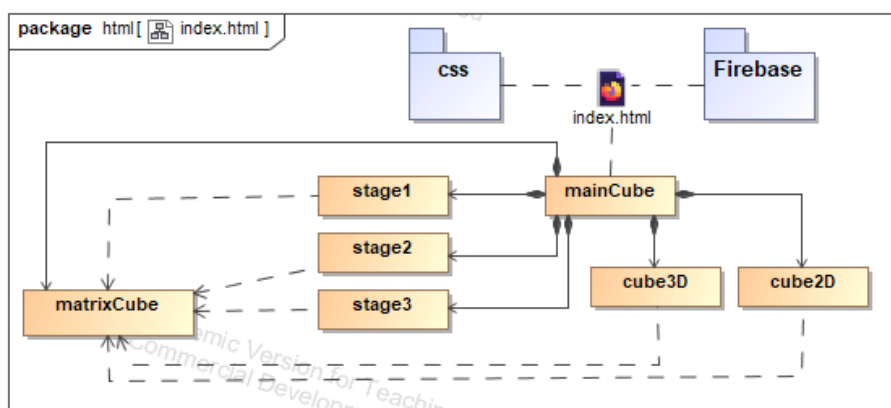


Figura B.2: Diagrama de sequência: Cubo 3D.

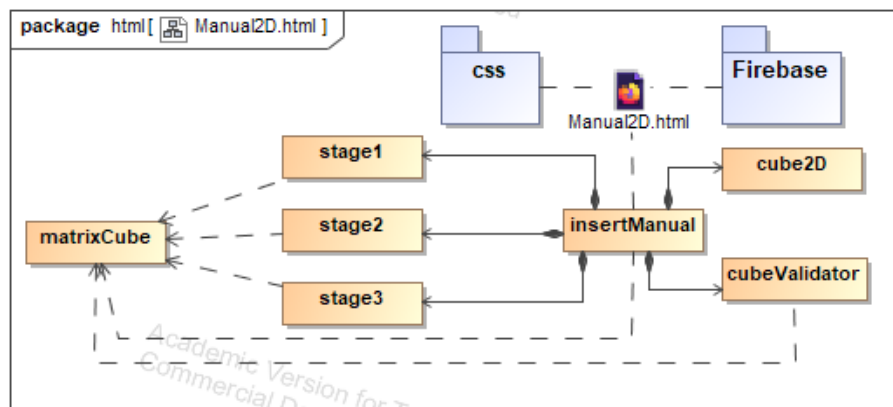


Figura B.3: Diagrama de sequência: Cubo 2D.

Apêndice C

Diagrama de sequência

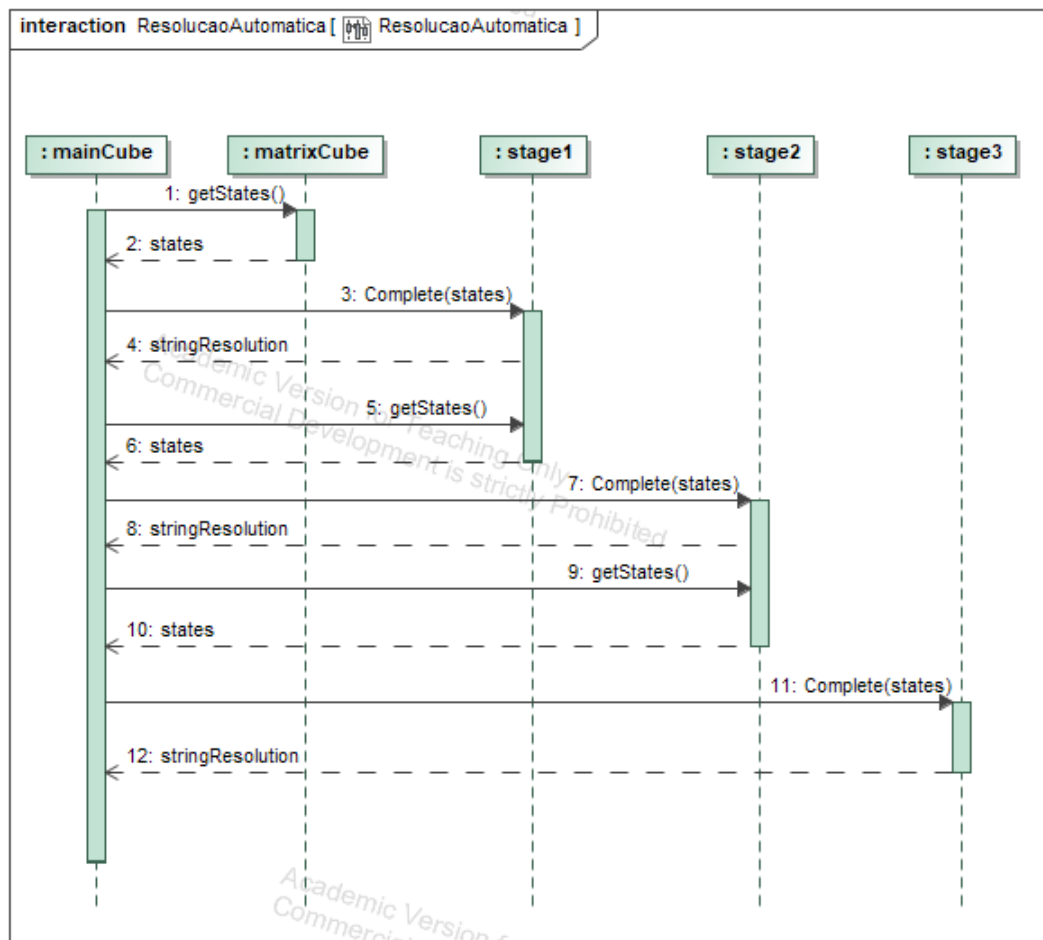


Figura C.1: Diagrama de sequência: Resolução do cubo.

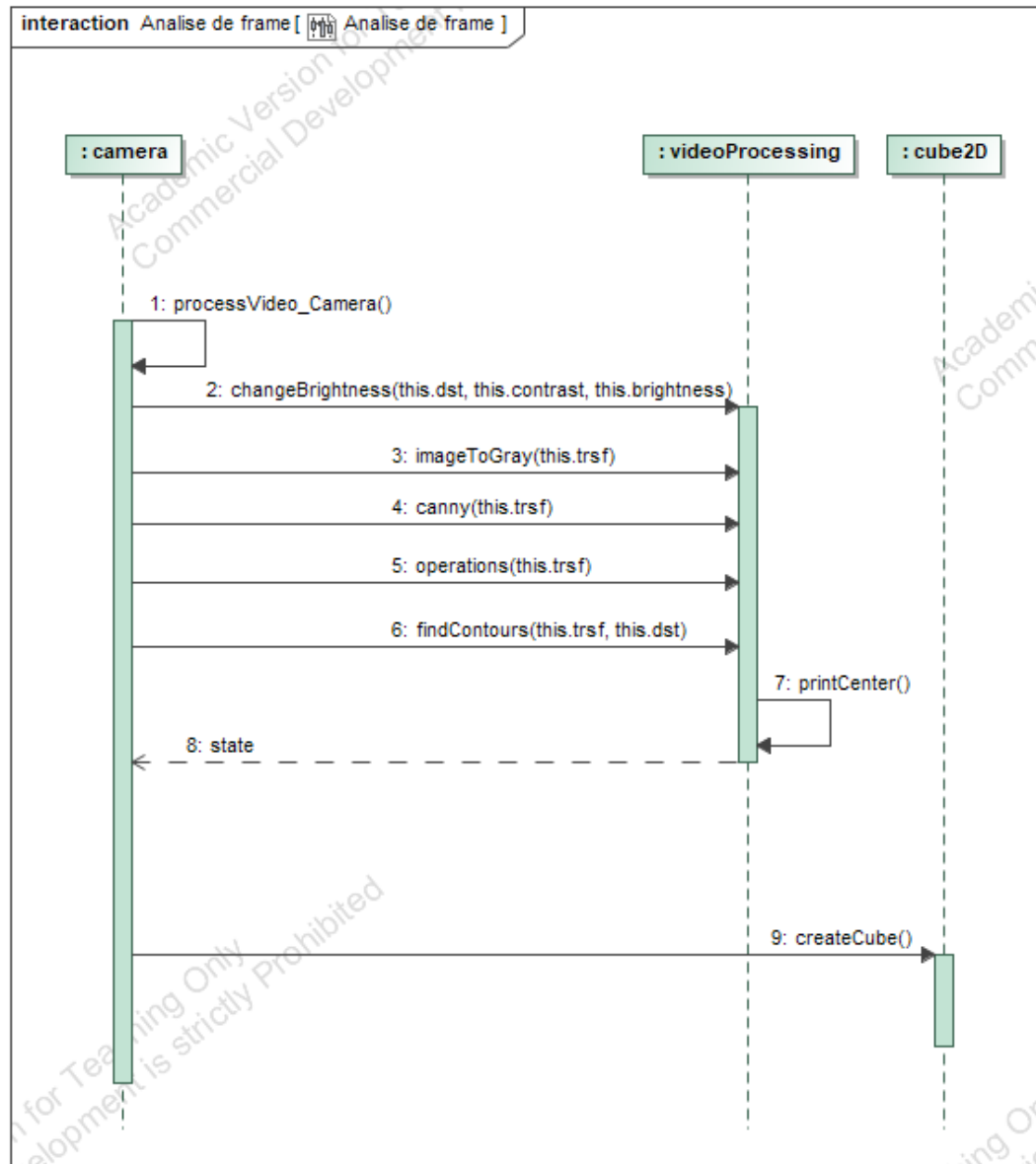


Figura C.2: Diagrama de sequência: Detecção do cubo com a câmara.