VISÃO COMPUTACIONAL E MICROCONTROLADORES: APLICAÇÃO NA RESOLUÇÃO DO CUBO DE RUBIK

Aryane de Paula Cezario (UFSC) E-mail: aryanecezario@gmail.com
Sara Mayumi do Nascimento Miura (UFSC) E-mail: smn.miura@gmail.com

Resumo: O desenvolvimento desse trabalho é baseado na resolução do cubo de Rubik, sendo então produzido um protótipo de robô capaz de realizar as manipulações necessárias. Para isso foi elaborada uma estrutura física utilizando atuadores, um algoritmo baseado em processamento de imagem a fim de localizar cada uma das cores e um algoritmo de resolução do cubo. Dessa forma, esse documento apresenta todos os recursos que foram utilizados, bem como a infraestrutura necessária e os resultados obtidos a partir da implementação do algoritmo de Thistlethwaite para a resolução.

Palavras-chave: Cubo de Rubik, microcontrolador, Thistlethwaite, processamento de imagem.

COMPUTATIONAL VISION AND MICROCONTROLLERS: APPLICATION TO RUBIK'S CUBE RESOLUTION

Abstract: The development of this work is based on the resolution of the Rubik's cube, being then designed a robot prototype capable of performing the necessary manipulations. For this, a physical structure using actuators, an algorithm based on image processing in order to locate each color and a cube resolution algorithm will be developed. Thus, this document presents all the resources that were used, as well as the necessary infrastructure and the results obtained from the implementation of the Thistlethwaite algorithm for the resolution.

Keywords: Rubik's Cube, microcontroller, Thistlethwaite, image processing.

1. Introdução

O Cubo de Rubik ou cubo mágico, como também é conhecido o quebra-cabeça, possui 6 faces e cada uma delas é pintada de uma cor diferente, além disso elas são divididas em 9 partes, totalizando 26 cubos pequenos e o 27º é substituído pelo mecanismo que segura todos os outros. Os 26 cubos menores que constituem o cubo mágico são divididos em 6 centros fixos, 8 cantos, que possuem três cores cada e 12 arestas, de 2 cores cada. Com exceção dos centros, todas as peças podem ser comutadas dentro de suas especificações, tornando a resolução do cubo muito complexa quando se deseja realizar a menor quantidade de movimentos possível ou muito mais lenta quando deseja-se utilizar o algoritmo mais memorizável.

Devido a essa grande complexidade das possibilidades de comutação do cubo mágico, foram criados diversos algoritmos de resolução do mesmo, além de gerar uma busca pelo menor número de movimentos para atingir o objetivo final e a criação dos instrumentos necessários para fazê-lo no menor tempo possível, seja com o uso dos algoritmos ou inteligência artificial, além de incorporar atuadores e microcontroladores para realizar os movimentos mecânicos.

Como proposto no projeto de SIMÕES et al (2010), que realiza a análise de ambientes de desenvolvimento para automação do problema do cubo mágico para o robô lego, são inúmeras as formas de automatizar a resolução, mesmo dentro da plataforma LEGO NXT. Destaca-se como conclusão que os pontos como a familiaridade com a linguagem, facilidades do ambiente de desenvolvimento e tempo de entrega do projeto são fatores

fundamentais na escolha de uma plataforma de desenvolvimento para avanços na solução. Dessa forma, observou-se dificuldades de implementação em todas as linguagens, porém o resultado sempre foi atendido, tendo uma média de 60 movimentos do braço robótico e bandeja giratória.

A elaboração do projeto de SIMÕES et al (2010) foi facilitada pela modularidade do kit LEGO, além de já possuir todos os sensores e atuadores compatíveis, porém como os movimentos são realizados apenas pela cesta e pelo braço que tem o sensor de luz acoplado, é necessária uma maior quantidade de movimentos para realizar um único comando.

Dessa forma, este projeto propõe o desenvolvimento de um protótipo funcional de um robô para a resolução do cubo de Rubik utilizando seis motores de passo controlados por um Arduino Mega, para que cada comando gere apenas um movimento, diminuindo a quantidade total dos mesmo. Além disso visa a implementação do algoritmo de solução e de visão computacional através do processamento de imagem. com a utilização do software Matlab, devido a familiarização das autoras com o mesmo e pelas facilidades do ambiente.

2. Materiais e Métodos

O cubo mágico permite inúmeras possibilidades de movimentação, como o giro de duas camadas simultaneamente ou até mesmo a rotação deste por inteiro, porém para a o seguinte projeto serão considerados apenas os movimentos básicos, que estão apresentados na Figura 2. As faces do cubo são nomeadas da mesma forma que os movimentos realizados pelas mesmas. Ou seja, U é a face superior (up), D a base (down), R direita (right), L esquerda (left), F a face da frente (front) e B a de trás (back).

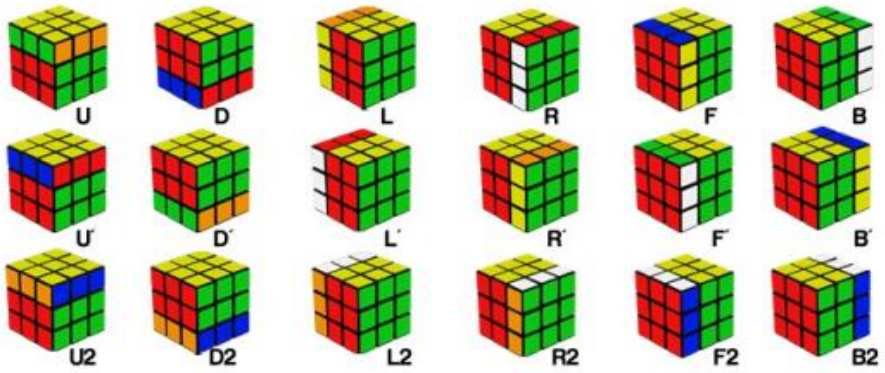


Figura 1 – Movimentos do Cubo (CINOTO,2020)

2.1. Processamento de Imagem

O processamento de imagem é um tipo de processamento de dados que consiste em utilizar imagens e vídeos como entrada de um sistema. De acordo com Cavalheiro (2017, p. 7, apud GOMES & VELHO, 2002), "O processamento de imagens tem como funções facilitar a visualização da imagem ou adequá-la para análises quantitativas através de correções de defeitos ou realces das regiões de interesse nas imagens; e a extração e tratamento de dados quantitativos, feitos pelo próprio computador". Conforme afirma Cao (2009, p3-993, apud SHEN) algumas das vantagens do processamento digital de imagens é a alta exatidão e boa reprodutibilidade, a facilidade de controle e a multiplicidade, pois se é necessário realizar uma mudança no processo, basta realizar uma adaptação no código. Entretanto também afirma que há algumas desvantagens em relação

á esse método, como a grande quantidade de informação e o alto custo de processamento. Ele é aplicado para diversas finalidades, como no reconhecimento de caracteres que de acordo com Nanda (2021, apud GOSWAMI), essa técnica "contribui para o avanço da automação dos processos e age como uma ponte entre homens e máquinas em aplicações de vários campos como inteligência artificial e redes neurais".

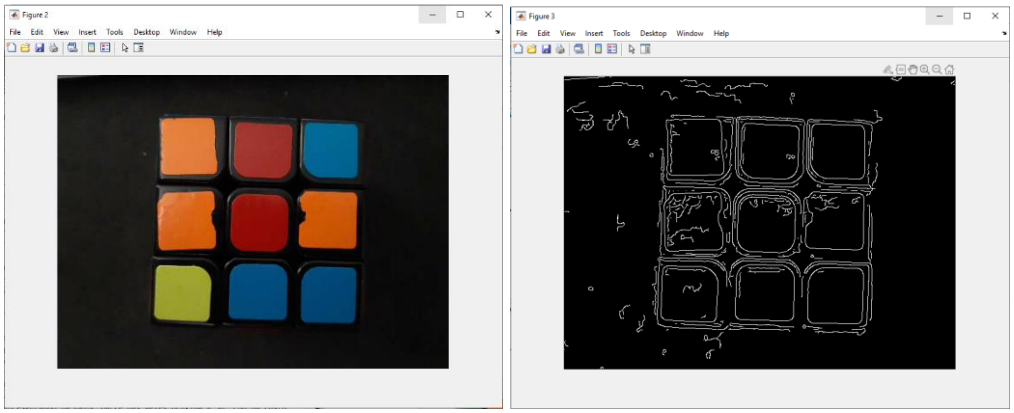
Assim, para a etapa de processamento de imagem foi utilizado o software Matlab e alguns de seus pacotes de ferramentas voltados ao Arduino e ao processamento de imagem, entretanto, devido a necessidade de realizar uma boa leitura das faces do cubo, foi utilizado também o software DroidCam.

Dessa forma, foi realizada a leitura da imagem utilizando a câmera de um celular conectado ao notebook através da rede pelo software DroidCam. Em seguida foi realizada a captura das imagens de cada uma das faces do cubo, que nesse caso deve ser realizada considerando sempre a ordem vermelho, azul, laranjado, verde, branco e amarelo. Além disso, o cubo também deve estar posicionado seguindo as determinações apresentadas no Quadro 1.

Tabela 1 – Tabela de Correlações

Face	Cor
F	Vermelho
R	Azul
B	Laranjado
L	Verde
U	Branco
D	Amarelo

Em seguida foram realizados uma série de procedimentos para o tratamento da imagem capturada, como a conversão para uma escala de cinza e a determinação da localização das bordas do cubo na imagem, para que assim seja realizado um corte e apresentado ao usuário a imagem final.



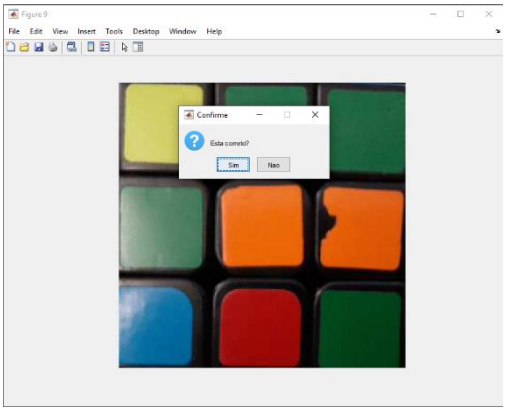


Figura 2 – Captura da imagem (a), determinação das bordas (b) e saída para a confirmação do usuário (c)

Assim, a partir da imagem lida e com o cubo centralizado corretamente, é possível determinar os parâmetros de altura e largura e a partir desses é possível determinar as linhas e colunas que dividem o cubo, que por sua vez permitem a determinação do quadrado central da face.

A partir disso é possível determinar as cores de cada uma das divisões do cubo, uma vez que o centro possui uma cor predeterminada, basta realizar comparações para a determinação das outras cores. Entretanto, o processamento das cores é bem sensível: dependendo da iluminação, do foco da câmera ou da cor da base onde se encontra o cubo as cores podem ser trocadas, já que o espectro de frequência de branco e amarelo, vermelho e laranjado, azul e verde são bem parecidos. Dessa forma, decidiu-se por utilizar uma aplicação, desenvolvida por Joren Heit(2021), que possibilita a averiguação de cada uma das faces, assim, caso alguma das cores tenha sido lida de maneira errada, é possível realizar uma correção rápida manualmente.

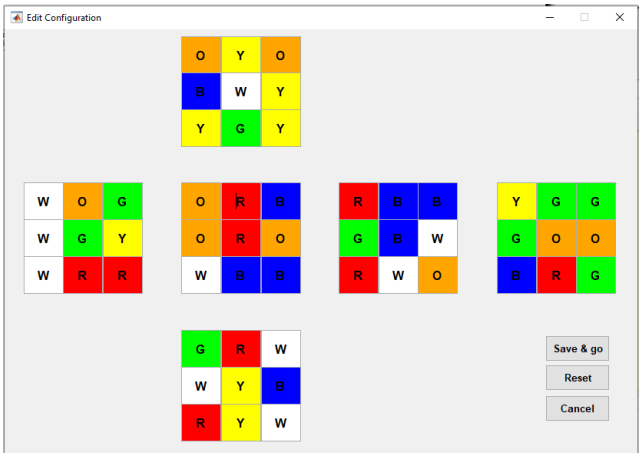


Figura 3 – Edição do estado das faces

2.2. Algoritmos de Resolução

Para iniciar a discussão sobre os algoritmos utilizados para a solução do cubo mágico primeiro é necessário considerar o que foi discutido sobre movimentações permitidas e abordar as possibilidades de combinações do mesmo. Se tratando de possibilidades de combinações em um cubo, considera-se o que foi abordado anteriormente em relação à anatomia do mesmo e utilizando de análise combinatória para calcular todas as comutações.

Análise combinatória segundo Nicholson (1818 apud OLIVEIRA, 2015, pg. 24) é “o

ramo da matemática que nos ensina a averiguar e expor todas as possíveis formas através das quais um dado número de objetos podem ser associados e misturados entre si", exemplificando, se existem 3 objetos (X, Y e Z) que podem ser permutados entre si, as possibilidades de combiná-los são 6 (XYZ, XZY, YXZ, YZX, ZXY e ZYX), porém é possível calcular sem a necessidade de obter todas as amostras. Para isso, utiliza-se da formula de arranjos simples Pn = n!, para o caso citado tem-se P3 = 3!.

O cálculo por análise combinatória das possibilidades de posições de cubo mágico se torna mais complexo devido as suas particularidades. Os centros, por serem fixos, não são considerados para esse calculo, já os 12 cubos menores de meio/aresta, podem ser combinados em P12 = 12! porém por possuir 2 cores cada, ou seja cada uma das peças pode apresentar duas posições diferentes dentro dessa combinação, também deve-se considerar 2^12. Da mesma forma, os 8 cubos de canto/vértice podem ser combinados em 8!*3^8, pois cada peça possui três cores diferentes.

Tendo esses valores, ainda tem que se levar em conta o fato de que ao posicionar as 7 peças de canto, a última só poderá ser posicionada de um maneira, evitando um caso impossível com o da Figura 4a, o mesmo com as peças de meio (Figura 4b). Por fim, considerasse a paridade do cubo, que apenas metade dessas posições serão possíveis através dos movimentos apresentados. Assim, as peças de canto podem ser misturadas em 8!*3^7 posições e as do meio em 12!*2^11. Dessa forma obtém-se um total de 43.252.003.274.489.856.000 (quarenta e três quintilhões) combinações.

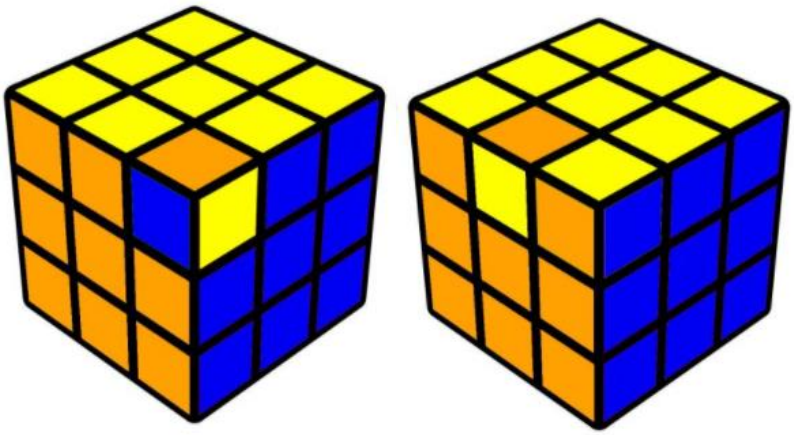


Figura 4 – Caso impossível para peça de canto (a) e peça de meio (b) (CINOTO,2020)

Existem diversos algoritmos utilizados para solucionar um cubo mágico, desde os mais simples e de fácil memorização até os mais complexos que só podem ser solucionados por máquinas. Quando se busca um método avançado e capaz de reduzir ainda mais a quantidade de movimentos necessários para a montagem, consequentemente diminuindo o tempo de execução, é necessário utilizar algoritmos computacionais, pois estão além da capacidade e rapidez humana. Com o objetivo de encontrar o número de Deus, que é a menor quantidade de movimentos que seriam necessários para montar qualquer configuração do cubo, diversos estudiosos e entusiastas começaram a desenvolver algoritmos que fossem capazes de atingir tal número.

Em 2010, Tomas Rokicki, Herbert Kociemba, Morley Davidson e John Dethridge através de uma grande potência computacional, fornecida pela Google, conseguiram a prova final de que toda e qualquer posição do cubo poderia ser resolvida com no máximo 20 movimentos básicos. Nesse capítulo serão abordados os métodos de Thistlethwaite e o de

Kociemba.

2.2.1. Método Kociemba

Um dos algoritmos implementados para tentar atingir a solução com o número de Deus foi o de Kociemba, que define um objetivo intermediário e resolve o cubo em dois diferentes passos. Partindo do cubo embaralhado, ele busca a sequência de movimentos que atinja o segundo grupo, o qual possui movimentos mais restritos para a solução final. Com isso foi possível implementa-lo para resolver qualquer combinação de cubo com no máximo 30 movimento. Os grupos desse método são divididos em:

- G0: grupo inicial que permite todos os movimentos de 90º tanto no sentido horário como no anti-horário (R, R', L, L', U, U', D, D', F, F', B, B').

Através desses movimentos é possível atingir um estado do cubo que pode ser resolvido utilizando apenas os movimentos definidos para o Grupo 1.

- G1: grupo de combinações do cubo que podem ser resolvidas utilizando apenas os movimentos <U, U', D, D', L2, R2, F2, B2>.

Segundo SOUZA (2011) "o que caracteriza este subconjunto é que todas a peças estão orientadas e os meios BL,BR, FR e FL estão na sua combinação correta" conforme exemplificado na Figura 5.

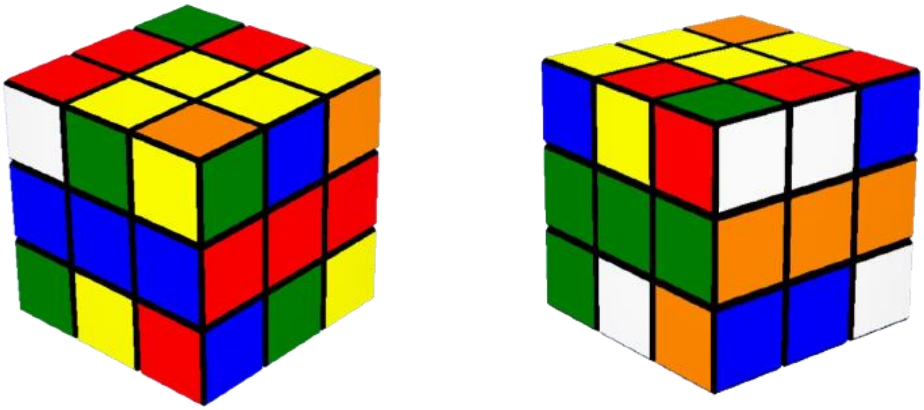


Figura 5 – Posições características do Grupo 1

2.2.2. Método Thistlethwaite

Um dos algoritmos que buscou atingir o menor número possível de movimentos para a solução do Cubo de Rubik foi o desenvolvido por Morwen Thistlethwaite, que é baseado na Teoria de Grupos e comprovado cientificamente. Diferente dos métodos amplamente conhecidos, que solucionam o cubo por partes, sendo por camadas ou primeiro as arestas depois os meio, segundo GRIMM (2016, p. 65) “Um dos motivos que torna o algoritmo de Thistlethwaite impressionante é o fato de ele não seguir nenhum destes dois padrões. Seu método consiste em resolver todos os cubinhos simultaneamente até que haja somente um movimento possível, que é justamente aquele que retorna o cubo à sua posição original, isto é, com todas as cores orientadas”.

Esse método de busca divide a solução em quatro etapas, que ficam cada vez mais restritas, gerando buscas mais rápidas, dividas nos quatro grupos a seguir:

- G0: grupo inicial que permite todos os movimentos de 90º tanto no sentido

horário como no anti-horário (R, R', L, L', U, U', D, D', F, F', B, B').

Através desses movimentos é possível atingir um estado do cubo que pode ser resolvido utilizando apenas os movimentos definidos para o Grupo 1.

- G1: definido pelo grupo de cubos que podem ser resolvidos utilizando movimentos de 90º em todas as faces, exceto a superior e inferior que só podem realizar movimentos de 180º, ou seja, são permitidos os movimentos (R, R', L, L', F, F', B, B', U2 e D2).

Utilizando esses movimentos, busca-se um estado presente no grupo G2.

- G2: nesse grupo as faces Front e Back são restringidas aos movimentos de 180º permitindo (R, R', L, L', F2, B2, U2 e D2).

- G3: ao chegar nesse grupo serão permitidos apenas movimentos de 180º em todas as faces (R2, L2, F2, B2, U2 e D2).

- G4: Contém apenas um elemento, que corresponde ao cubo solucionado, ou seja, não há necessidade de realizar nenhum movimento.

A transição de G0 para G1 envolve apenas a peças de aresta/meio. Estas são consideradas "ruins" se no processo de coloca-las no lugar correto forem realizados quantidades impares de movimentos de 90º nas faces U ou D, porém esses movimentos podem rapidamente converte-las em peças ditas "boas".

De G1 para G2, considera-se que as peças de aresta já iniciam na orientação correta, então realiza-se o mesmo processo para os cantos, porém apesar de estes não possuem uma orientação natural devem ser alinhados, notando-se que essas peças sempre possuirão elementos ou da face L ou da R. O algoritmo de busca deve encontrar um número máximo de 10 movimentos para atingir o objetivo de orientar os cantos. Nota-se que apenas movimentos de 90º nas faces F e B alteram o giro das peças de canto e os outros demais movimentos não surtem efeito. Nessa transição também deve-se utilizar no máximo 4 movimentos para obter uma posição do cubo onde as peças FU, FD, BU, BD estão todas nas camadas U e D.

De G2 para G3 é o estágio mais complicado, dividido em 2 fases, a primeira tem a função de colocar os cantos em suas orbitas naturais, a segunda permuta os cantos em suas orbitas até obter uma das 96 permutações para cantos do grupo G3 e coloca as arestas em suas devidas camadas. Do grupo G3 para a solução final são realizados apenas movimentos duplos, colocando os cantos nas posições certas em apenas 4 movimentos e restaurando a posição correta com meios.

Esse processo é mais rápido que os métodos manuais e possui um custo computacional menor do que se fosse realizada uma busca geral de todas as possibilidades de combinações do cubo. Através desse método Thistlethwaite conseguiu comprovar que o mínimo de movimentos possível para resolver o pior caso do cubo, aquele que demanda mais movimentos é 52.

Dessa forma esse foi o método utilizado nesse trabalho, pois apresenta uma quantidade consideravelmente menor de movimentos quando comparado aos métodos de camadas e de Friedrich. Sua implementação é mais fácil e rápida quando comparada ao de Kociemba, porém apesar de ter um número máximo de movimentos igual a 52,

comparado a 30 de Kociemba, a média prática acaba se apresentando na maior parte das vezes próxima a 30, com um custo computacional mais reduzido.

Após o processamento digital de imagem é gerada uma matriz R, que contém a posição de cada uma das peças do cubo, a partir desta, é aplicado o algoritmo de solução. O primeiro passo consiste em obter a orientação e permutação das peças de meio/aresta. Esta é dita "boa" em dois casos, o primeiro se a cor da frente pertencer as faces F ou B e o segundo se a cor da frente pertencer as faces U ou D e a cor lateral pertencer as faces L ou R. As peças que são consideradas "ruins" são comparas as posições dos meios do cubo resolvido e é realizada uma varredura para encontrar a melhor sequência de movimentos que torne uma peça "ruim" em "boa" e considera se será necessário realizar uma orientação e/ou permutação. O mesmo é realizado para as peças de canto.

Ao analisar esses movimentos também deve-se atualizar as posições do cubo, para assim finalizar a primeira fase e passar para o próximo grupo de movimentos, que consiste em mover as peças da parcela LR para as UD e orientar os cantos. A terceira fase consiste em arrumar os meios em cada uma das camadas de uma face do cubo utilizando movimentos pares e arrumar os meios em orbita com permutações pares. Já a última fase é a que finalmente soluciona o cubo, utilizando apenas movimentos pares em todas as faces. Todos os movimentos calculados são salvos em uma matriz solução v que é utilizada como base para o algoritmo de movimentação dos motores utilizados.

2.2.3. Eletrônica e Atuadores

Na parte eletrônica desse projeto foi utilizada uma placa de Arduino MEGA 2560. A placa foi criada com o intuito de prover uma plataforma eletrônica open source de baixo custo capaz de interagir com sensores e atuadores. A plataforma conta com vários modelos de placas de Arduino, dentre elas as mais conhecidas estão a Uno e a Mega 2560. Esse último, apresentado na Figura 6, conta com um microcontrolador ATmega 2560, regulador de 5V e 3.3V, quatro portas seriais, uma porta I2C, uma porta SPI, dezesseis portas analógicas de conversor ADC (analógico-digital), doze portas PWM (modulação por largura de pulso) de 16 bits, trinta e duas portas digitais, um LED para TX0 e um para RX0 (conexões seriais 0) e um LED conectado ao pino D13.

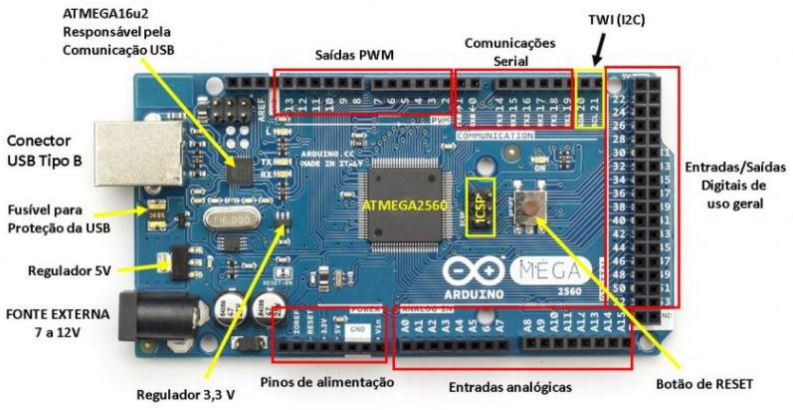


Figura 6 – Recursos do arduino Mega 2560 (SOUZA, 2014)

O microcontrolador ATmega2560 com clock de 16MHz. O 2560 possui uma arquitetura RISC avançada com até 16 MIPS, 256KB de memória flash, além de 8KB de memória estática SRAM, que são utilizados para fazer a inicialização de um sistema operacional,

caso necessário, (bootloader) e 4KB de memória não volátil EEPROM. O microcontrolador conta ainda com dois contadores de 8 bits, dois de 16 bits e um real time, um conversor ADC de 10 bits com 16 canais, quatro canais PWM de 8 bits e 12 canais PWM de 16 bits, quatro portas seriais e uma porta I2C, dentre outros recursos.

Dessa forma, escolheu-se utilizar uma placa da plataforma Arduino nesse projeto devido à sua relativa simplicidade de utilização, se comparado com uma placa Raspberry Pi por exemplo, e ao ótimo custo benefício, uma vez que ela atende de maneira satisfatória os propósitos do trabalho. Dentre os diferentes modelos da plataforma, escolheu-se trabalhar com o Arduino Mega 2560 devido a quantidade de portas e recursos disponíveis e ao fato de que seu processador possui melhores especificações, comparando com o Uno.

Em relação aos atuadores, tendo em mente a precisão necessária para o manejamento do cubo mágico e a sua relativa delicadeza, uma vez que se os movimentos forem realizados de forma imprecisa o cubo pode se desmontar, escolheu-se utilizar um motor de passo para realizar a rotação de cada uma das faces.

Esse tipo de motor é caracterizado pela existência de polos magnéticos, o que elimina a necessidade de utilização de escovas e possibilita que a rotação seja feita em pequenos passos. Assim, esse motor é composto por um número fixo de polos magnéticos, que varia de acordo com o modelo, e o acionamento de cada um desses polos através de pulsos possibilita o movimento. Dessa forma, esse tipo de motor também necessita de um circuito através do qual seja possível realizar o controle desses passos.

O motor de passo escolhido para esse projeto foi o 28BYJ-48, pois possui características de construção adequadas para os objetivos citados anteriormente, além de ser pequeno, podendo se adaptar à estrutura que será criada facilmente, e possuir custo acessível. Esse modelo ainda é unipolar, propriedade que irá possibilitar o movimento das faces do cubo de Rubik nos sentidos horário e anti-horário sem a necessidade de acoplar uma ponte H a cada um dos motores utilizados. Juntamente com esse motor será utilizado também o driver ULN2003, com a finalidade de realizar o controle adequadamente.

2.2.4. Montagem do Protótipo

Na montagem, os motores são alimentados aos pares, para que haja tensão e corrente suficientes passando por cada um deles, o que gera torque e velocidade para rotacionar as faces do cubo. Eles foram fixados nos suportes utilizando parafusos de cabeça chata phillips 4.0x50, dois dos suportes laterais são fixos e os outros dois possuem um sistema de afastamento possibilitando que seja feita a retirada do cubo e o travamento do mesmo. O suporte superior é deslocado utilizando uma dobradiça e fixado utilizando parafusos e borracha.

Como o controle é realizado no Matlab, houve a necessidade de manter o arduino conectado ao computador durante todo o processo de rotação dos motores, para isso foi dedicada uma porta USB, especificada no código principal, a partir da qual foram inicializadas todas as portas digitais utilizadas para conectar os motores. Dessa forma, foi possível montar uma sequência condicional baseada no vetor de strings (v) fornecido pelo algoritmo de solução, que representa os movimentos necessários para solucionar o cubo, movimentando os atuadores em \$90º no sentido horário, anti-horário e em 180º, cuja rotação em ambos sentidos atingem o mesmo resultado.

Os motores são acionados pelo driver ULN2003 e para alterar o sentido de rotação, foram utilizadas sequências opostas na alimentação dos pinos de entrada do driver, por acionamento a passo pleno, ou seja, os pinos de entrada são alimentados aos pares, apresentando sequências opostas para o sentido horário e anti-horário.

A estrutura que comporta todos os elementos utilizados é composta por peças de madeira pinus, contando com uma base, 4 pés, 4 suportes laterais para os motores e um suporte superior, ainda foi necessário realizar adaptações para melhorar o encaixes com duas peças adicionadas ao suporte superior. Os encaixes dos motores no cubo, foram moldados nos mesmos e confeccionados com o uso de Durepoxi Loctite, o que proporcionou uma junção única para cada uma das faces permitindo uma melhor rotação. A obtenção das especificações para o tamanho das peças utilizadas na estrutura do protótipo foram feitas baseadas nas medidas do Cubo Mágico Profissional 3x3x3 Moyu Mf3 Moyu Mofangjiaoshi, utilizado em competições oficiais e modelado no software SketchUp. Na Figura 7 é possível observar o protótipo desenvolvido para a realização dos testes de resolução do cubo de Rubik.

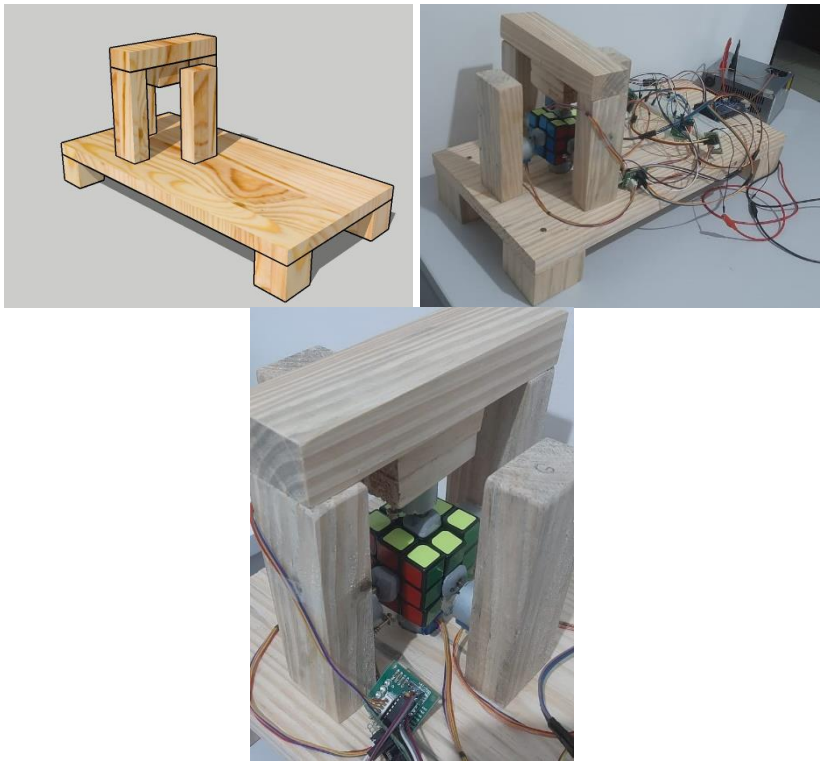


Figura 7 – Modelo tridimensional (a), protótipo desenvolvido (b) e posicionamento do cubo (c)

3. Conclusões

O desenvolvimento do projeto apresentou diversas dificuldades no processamento de imagem e montagem física. Em relação ao processamento de imagem, o maior problema está relacionado a quantidade de luz necessária para que todas as cores sejam devidamente identificadas, pois quando há pouca incidência de luz as cores vermelho e laranjado apresentam frequências muito semelhantes e com muita luz o mesmo acontece com o branco e amarelo. Dessa forma, mesmo a olho nu essa comparação é muito complicada, gerando erros. Isso foi corrigido utilizando um código para imprimir e permitir a configuração manual caso houvesse alguma incoerência.

Na montagem física, o problema está relacionado aos motores de passo, que foram inadequados para o projeto, pois apresentam uma imprecisão considerável, que gera o travamento do cubo depois de alguns movimentos. Quando a velocidade foi reduzida, o torque aumentou, porém não solucionou o problema. Para melhorar o desempenho, a tensão foi alterada de 5V para 12V, dessa forma, os motores apresentaram maior velocidade e torque.

A partir dos resultados obtidos e ao longo do desenvolvimento do trabalho notou-se alguns pontos em que poderiam ser realizadas algumas melhorias caso o projeto seja retomado futuramente. Uma delas é o desenvolvimento de uma placa de circuito impresso para a soldagem dos componentes eletrônicos, medida que não foi tomada pois os elementos utilizados foram emprestados da universidade. Outra medida seria a utilização de motores de passo com um torque maior que o utilizado, como o modelo NEMA 17 por exemplo, que permitiria maior liberdade e melhor movimentação das faces do cubo.

Além disso, poderia ainda ser utilizada impressão 3D tanto para a montagem da base dos motores quanto para a peça que realiza a conexão entre eles e o cubo. Isso porque com uma estrutura assim cada uma das medidas seriam mais precisas do que se pode conseguir no corte da madeira e o conjunto ainda seria mais leve, o que poderia facilitar no ajuste de velocidade e torque dos motores.

Tendo em vista a finalização do projeto, algumas considerações podem ser feitas. A primeira delas é notar que as melhorias discutidas anteriormente, como a troca do modelo dos motores, podem fazer diferença significativa no resultado final, entretanto adotar essas mudanças requer um investimento financeiro relativamente alto. Notou-se ainda, que o custo computacional do processo como um todo é elevado, o que fez afetou o desempenho do software e por isso foi possível concluir que a utilização de algum método de busca de solução mais avançado, como o de Kociemba, não seria viável.

Além disso, é necessário ressaltar ainda as contribuições que o projeto teve para as proponentes academicamente. Ao longo do desenvolvimento foi necessário a aplicação de conceitos nunca antes vistos, como o processamento de imagem e o funcionamentos dos algoritmos aplicados. Além disso, foi possível colocar em prática conceitos estudados em classe, como circuitos elétricos, algoritmos, programação e o entendimento de motores e microcontroladores.

Referências

BRITO, SABRINA. *Por que o cubo mágico retornou às lojas ainda mais popular***.** 2021. Disponível em: https://veja.abril.com.br/cultura/por-que-o-cubo-magico-retornou-as-lojas-ainda-mais-popular/. Acesso em 24 fevereiro de 2021.

CAO, H., e Y. SHEN. *Application of MATLAB image processing technology in sewage monitoring system*. 9th International Conference on Electronic Measurement \& Instruments, Beijing, China, 2009, pp. 3-993-3-995, doi: 10.1109/ICEMI.2009.5274144. Acesso em 03 de março de 2021.

CAVALHEIRO, GUILHERME SEHNEM. *Sistema Embarcado para Processamento de Imagens com Aquisição de Características Aplicada a Aquicultura.* 2017. 43. f. Disponível em: https://www.politecnica.pucrs.br/conclusao/files/20172-guilherme-sehnem-cavalheiro-VOLUME-2617.pdf. Acesso em 03 de março de 2021.

CINOTO, RAFAEL. *Coisas que não acontecem no cubo mágico.* 2021. Disponível em: https://cinoto.com.br/cubomagico/coisas-que-nao-acontecem-no-cubo-magico/. Acesso em 25 de fevereiro de 2021.

EDDINS, S. L., e M. T. ORCHARD. *Using MATLAB and C in an image processing lab course.* Proceedings of 1st International Conference on Image Processing, vol. 1, IEEE Comput. Soc. Press, 1994, p. 515–19. DOI.org (Crossref), doi:10.1109/ICIP.1994.413367. Acesso em 03 de março de 2021.

FEIS, UNESP. *Motor de Passo. 2013.* Disponível em: https://www.feis.unesp.br/Home/departamentos/engenhariaeletrica/aula3-motor-de-passo-2013-1-13-03-2013-final.pdf. Acesso em 26 de fevereiro de 2021.

GRIMM, Luis Gustavo Hauff Martins. Cubo Mágico: propriedades e resoluções envolvendo álgebra e teoria de grupos. 2016. 81 f. Dissertação (Mestrado) - Curso de Matemática, Instituto de Geociências e Ciências Exatas, Universidade Estadual Paulista "júlio de Mesquita Filho", Rio Claro, 2016. Disponível em: https://repositorio.unesp.br/bitstream/handle/11449/144192/grimm_lghm_me_rcla_int.pdf?sequence=4&isAllowed=y. Acesso em 11 fevereiro de 2021.

THISTLETHWAITE, MORWEN. *Thistlethwaite's 52-move algorithm*, 1981. Disponível em https://www.jaapsch.net/puzzles/thistle.htm#:~:text=Thistlethwaite%20is%20a%20mathematician%20who,Cube%20in%20remarkably%20few%20moves.&text=Instead%20it%20works%20on%20all,and%20the%20cube%20is%20solved. Acesso em 08 de abril de 2021.

NANDA, PRADEEP K.; GOSWAMI, LAXMI. *Image processing application in character recognition*. Materials Today: Proceedings, [S.L.], abr. 2021. Elsevier BV. http://dx.doi.org/10.1016/j.matpr.2021.03.697. Acesso em 25 de setembro de 2021.

NAYYAR, ANAND, PURI, ER. VIKRAM. *A Review Of Arduino Board's, Lilypad's \& Arduino Shields*. 2021. Disponível em: https://ieeexplore.ieee.org/document/7724514. Acesso em 25 de fevereiro de 2021.

OLIVEIRA, CARLOS ALBERTO LOPES DOS SANTOS DE. *Análise Combinatória: Raciocínio recursivo e processos de enumeração.* 2015. 103 f. Dissertação (Mestrado) - Curso de Matemática, Centro de Ciência e Tecnologia. Laboratório de Ciências Matemáticas, Universidade Estadual do Norte Fluminense Darcy Ribeiro, Campos dos Goytacazes, 2015. Disponível em: https://uenf.br/posgraduacao/matematica/wp-content/uploads/sites/14/2017/09/16092015Carlos-Alberto-Lopes-dos-Santos-de-Oliveira.pdf. Acesso em 18 fevereiro 2021.

SILVA, JOSÉ VINÍCIUS DO NASCIMENTO. *Uma proposta de aprendizagem usando o Cubo Mágico em Malta - PB.* 2015. 71 f. Dissertação (Mestrado) - Curso de Matemática, Pró-Reitoria de Pós-Graduação e Pesquisa, Universidade Estadual da Paraíba, Campina Grande, 2015. Disponível em: http://tede.bc.uepb.edu.br/jspui/bitstream/tede/2390/2/PDF%20-%20Jos%C3%A9%20Vin%C3%ADcius%20do%20Nascimento%20Silva.pdf. Acesso em 08 fevereiro de 2021.

SOUZA, WALTER PEREIRA RODRIGUES DE. *Soluções Eficientes para o Cubo Mágico*. 2011. 22 f. Monografia (Especialização) - Curso de Matemática, Departamento de Ciência da Computação, Universidade de São Paulo, São Paulo, 2011. Disponível em: https://www.ime.usp.br/~cef/mac499-11/monografias/walter/monografia.pdf. Acesso em 12 fevereiro 2021.

YAURI-MACHACA, MELISSA, et al. *Design of a Vehicle Driver Drowsiness Detection System Through Image Processing using Matlab.* 2018 IEEE 38th Central America and Panama Convention (CONCAPAN XXXVIII), IEEE, 2018, p. 1–6. DOI.org (Crossref), doi:10.1109/CONCAPAN.2018.8596513. Acesso em 03 de março de 2021.