

## COMPARAÇÃO DE DESEMPENHO EM OPERAÇÕES COM IMAGENS ENTRE PLACAS DE BAIXO CUSTO PARA SISTEMAS EMBARCADOS

B.H. S. FERREIRA<sup>1</sup>, L. BUONOCORE<sup>2</sup>

<sup>1</sup> Universidade Federal do Maranhão. Centro de Ciências Exatas e Tecnologia, Departamento de Engenharia Elétrica.

<sup>2</sup> Universidade Federal do Maranhão. Centro de Ciências Exatas e Tecnologia, Departamento de Engenharia Elétrica.

E-mail para contato: [brunohenri.sf@gmail.com](mailto:brunohenri.sf@gmail.com)

**RESUMO** – *A capacidade humana de avaliar características e propriedades dos ambientes se dá em função da possibilidade de adquirir informações através dos sentidos, sendo processadas em seguida no cérebro. Nesse contexto percebe-se que a visão é o sentido com maior capacidade de absorver informações, nos permitindo desenvolver atividades que envolvam precisão, agilidade, velocidade dentre outras habilidades. A utilização de câmeras está cada vez mais corriqueira no meio científico e tem sido muito utilizada em áreas como Engenharia, Biologia, Astronomia, Geografia, etc. Uma característica importante deve ser analisada para viabilizar a aplicação dessa tecnologia em meios onde a informação precisa ser tratada localmente através de microprocessadores de menor porte, como o caso de drones em tarefas de posicionamento espacial. O processo acontece com as seguintes etapas: aquisição, pré-processamento, segmentação, extração de características e interpretação. O tempo gasto para realizar essas etapas pode ser crucial na execução de tarefas que envolvam o processamento de imagens. Por isso, fez-se necessário a realização de um estudo do tempo de máquina em algumas operações básicas com imagens digitais nos sistemas embarcados Raspberry Pi 3 modelo B e o BeagleBone blue. Os resultados do tempo de processamento nos experimentos realizados mostraram-se mais favoráveis ao Raspberry Pi 3 modelo B em função principalmente da maior quantidade de recursos, como possuir quatro núcleos de processamento contra um do BeagleBone blue.*

**Palavras-chave:** *Processamento de Imagem. Raspberry Pi 3 modelo B. BeagleBone blue. Comparação de desempenho de sistemas.*

## 1. INTRODUÇÃO

Ao longo da história, as imagens tem sido um objeto de estudo e interesse da humanidade por representar ideias, sentimentos, crenças, etc. Um exemplo da busca por conhecimento nessa área é a câmara escura, que consistia em uma caixa com paredes totalmente opacas e um pequeno furo frontal. Quando um objeto era posicionado em frente a esse dispositivo a luz refletida por ele passava pelo pequeno furo, era invertida e projetada sobre a superfície interna formando uma imagem do objeto. Esse fenômeno físico foi observado pela primeira vez no século V a.C. pelo filósofo chinês Mozi<sup>1</sup>.

Define-se como processamento de imagens a capacidade de capturar, modificar, analisar e manipular imagens digitais. Esse processo baseia-se na execução de cálculos específicos para cada objetivo desejado, mostrando-se um ramo que envolve conhecimentos das mais diversas áreas como óptica, eletrônica, matemática, fotografia e computação.

O tempo de execução dos algoritmos que realizam esses processos varia de acordo com as entradas, as técnicas de implementação e o poder de processamento das máquinas utilizadas para a realização das tarefas envolvidas. Portanto, o tempo de processamento está diretamente relacionado aos *software* e *hardware* disponíveis. Para uma análise mais detalhada, uma métrica quantitativa foi aplicada sobre o tempo de processamento de dados e, assim, permitir a escolha de um sistema de baixo custo que melhor atende às necessidades da aplicação em questão.

## 2. MATERIAIS E MÉTODOS

Nesta pesquisa foram padronizadas as etapas de sua execução com objetivo de se obter resultados confiáveis com base no desenvolvimento das atividades a seguir: a) pesquisa bibliográfica, para uma visão mais ampla e embasamento; b) configuração dos ambientes de trabalho, como instalação do sistema Raspbian no Raspberry Pi 3B e Debian no BeagleBone *blue*, através do *boot* em um cartão micro SD; c) instalação dos pacotes da biblioteca OpenCV (*Intel Open Source Computer Vision Library*) que possibilitam o processamento em visão computacional por possuir módulos de códigos específicos, estrutura de dados, álgebra linear e GUI (*Graphic User Interface*); d) implementação dos algoritmos em C++ baseados em construções já disponibilizadas nas comunidades e fóruns de programadores; e) execução dos códigos numa taxa definida para quantificar o tempo de execução (métrica); e f) avaliar os resultados obtidos para a comparação de ambas as plataformas.

Todos os materiais citados foram disponibilizados no LRC (Laboratório de Robótica Móvel e Comunicação Sem Fio), onde foi realizada esta pesquisa.

### 2.1. Raspberry Pi 3 modelo B

Um dos sistemas embarcados de baixo custo utilizado para a realização dos testes de desempenho foi o Raspberry Pi 3 modelo B<sup>2</sup> (ou 3B). Essa placa possui as seguintes características de *hardware*: processador Quad Core 1.2GHz Broadcom BCM2837 64bit, 1GB de RAM (*Random Access Memory*), conectividade a partir do BCM43438 Wireless LAN e Bluetooth Low Energy

<sup>1</sup> Site: <https://medium.com/@patricia.jones/hist%C3%B3ria-da-fotografia-27ec90487381>

<sup>2</sup> Site: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

(BLE) on board, quatro entradas USB, HDMI (*High-Definition Multimedia Interface*) e 40 GPIO (*General Purpose Input/Output*) distribuídos em pinos com funções especiais, como protocolo I2C (*Inter-Integrated Circuit*), SPI (*Serial Peripheral Interface*) e PWM (*Pulse-Width Modulation*). Na Figura 1 percebe-se a disposição física dos conectores de interface da parte superior da foto da placa.

Figura 1 - Raspberry Pi 3 modelo B



Fonte: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

## 2.2. Beaglebone Blue

Pensando no desenvolvimento de projetos voltados para a robótica, foi lançada a BeagleBone *blue*<sup>3</sup>, a qual possui interfaces para sensores, motores e bateria, além das seguintes características de *hardware*: processador AM335x 1GHz ARM® Cortex-A8 processor, 512MB DDR3 RAM (*Random Access Memory*), 4GB de armazenamento 8-bit eMMC flash storage (*Embedded Multimedia Card*), conectividade a partir da conexão *Wireless*: 802.11bgn e *Bluetooth* 4.1 BLE, interface para oito saídas de servo de V, quatro saídas para motores DC com entradas para *enconders* de quadratura, IMU (*Inertial Measurement Unit*) de nove eixos com barômetro, carregador de 2 células LiPo além das interfaces de GPS (*Global Position System*), DSM2 radio, UART (*universal Asynchronous Receiver Transmitter*), SPI, I2C, GPIOs e LEDs. Possui compatibilidade com os seguintes *software*: ROS, ArduiPilot, MATLAB, Simulink, LabVIEW, Cloud9 IDE on NODE.js, Python, OpenCV e muito mais.

Na Figura 2 verifica-se a disposição dos componentes eletrônicos bem como as interfaces da placa BeagleBone *blue*.

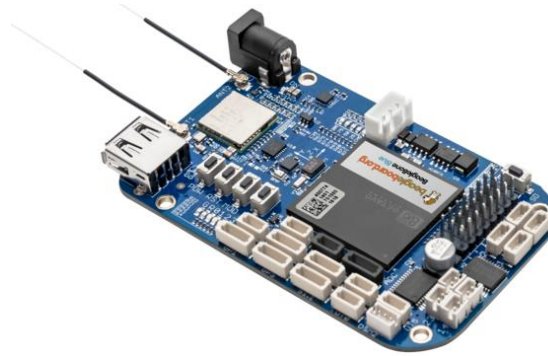
## 2.3. Logitech C270

Para a realização da captura das imagens nesta pesquisa, utilizou-se uma câmera Logitech C270<sup>4</sup> com as seguintes especificações (Figura 3): resolução máxima de 720p/30qps, foco fixo, lente padrão, microfone mono embutido e FoV (*Field of View*) de 60°.

<sup>3</sup> Site: <https://beagleboard.org/blue>

<sup>4</sup> Site: <https://www.logitech.com/pt-br/product/hd-webcam-c270>

Figura 2 - BeagleBone blue



Fonte: <https://beagleboard.org/blue/>

Figura 3 - Logitech C270

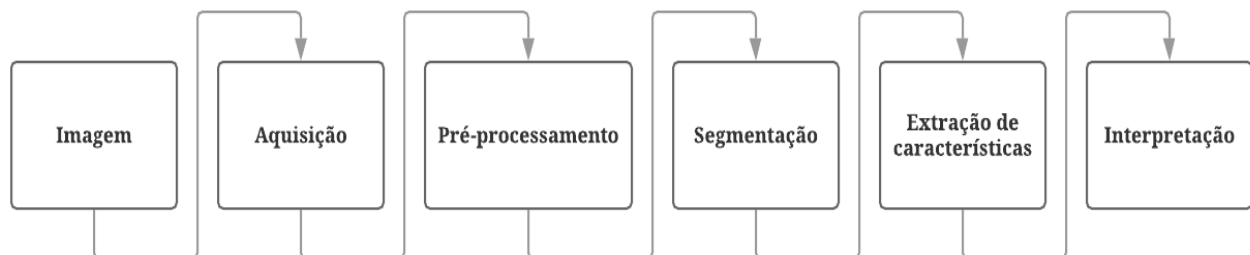


Fonte: <https://www.amazon.ca/Logitech-C270-HD-Webcam-960-000621/dp/B003SDDR74>

## 2.4. Processamento de imagens

Na Figura 4 observam-se todas as etapas envolvidas no processamento de imagens (BACKES e SÁ Jr., 2016). Mas para a comparação de desempenho preocupou-se apenas até o estágio de pré-processamento e segmentação, uma vez que a partir dessa fase o processamento realiza análises extras das imagens para obter características que permitam abstrair determinados padrões que não estão previstas para serem exploradas neste trabalho.

Figura 4 - Etapas do processamento de imagens



Fonte: Autor

Com o intuito de explicitar regiões de interesse, agrupando os pixels da imagem de forma a destacar regiões ou ainda suprimir características, são utilizadas diversas técnicas e operações matemáticas sobre as imagens digitais obtidas e já pré-processadas. Os olhos humanos conseguem facilmente realizar processos de reconhecimento e interpretação de objetos a partir de um conjunto de informações contidas no cérebro (MARQUES FILHO e VIEIRA NETO, 1999).

A abordagem computacional se difere em parte por basear-se em realizar operações matemáticas para, só então, comparar com um banco de dados previamente construído pelo programador. Em função disso algumas problemáticas devem ser contornadas como, por exemplo: ambientes bem controlados com relação à iluminação, movimentação do objeto e regiões de interesse devem ser pensados. Considerando esses fatores, é possível se obter diferentes aspectos da imagem, perceber limites e bordas, etc (MARQUES FILHO e VIEIRA NETO, 1999).

Foram realizadas as seguintes operações para explorar a dinâmica e o tempo de processamento das placas estudadas para os seguintes algoritmos: filtro de mediana, filtro Gaussiano, filtro bilateral, Sobel, Laplaciano, Canny, erosão, dilatação e fechamento. Esses algoritmos foram utilizados por possibilitarem tanto na etapa de pré-processamento como na de segmentação o destaque de características importantes para aplicações na engenharia, como por exemplo, detecção de bordas, suavização de transições, isolamento de objetos de interesse, preenchimento de pequenas lacunas nas aplicações com robôs seguidores de linha etc.

### 3. RESULTADOS E DISCUSSÃO

Nesta seção serão descritos os experimentos e os resultados obtidos com relação ao tempo requerido no processamento de imagens em algumas operações comumente realizadas, com o intuito de avaliar o desempenho dessas tarefas nos sistemas embarcados microprocessados Raspberry Pi 3B e BeagleBone *blue*.

#### 3.1. Tempo de processamento das operações em imagens nos dois sistemas embarcados

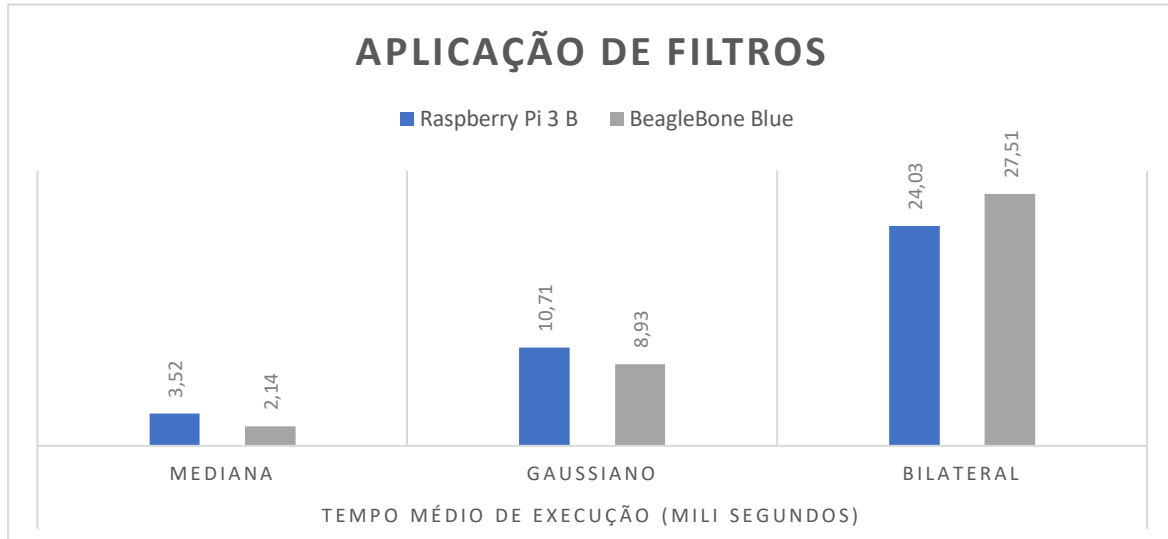
Para se mensurar o tempo requisitado na execução de um trecho do algoritmo, fez-se uso da função '*clock( )*' disponível na biblioteca *time.h*. Esse código retorna o tempo de CPU decorrido desde o início da execução do programa até o momento em que ele foi chamado, sendo o tempo medido em ciclos do relógio interno, onde cada CLOCKS\_PER\_SEC ciclos equivale a um segundo.

Portanto, é necessário capturar um valor de tempo inicial e em seguida capturar um valor de tempo final. A partir da diferença entre eles, se obtém o intervalo de tempo decorrido em milissegundos para execução das linhas de código que se situam entre essas duas chamadas à '*clock( )*'.

• Tempo de execução - Aplicação de Filtros: a partir da execução dos três algoritmos de filtragem de ruídos no Raspberry Pi 3B e do BeagleBone *blue*, gerou-se uma tabela para construção do gráfico da Figura 5.

Os parâmetros das funções utilizadas devem ser considerados, uma vez que suas alterações podem modificar os resultados visuais obtidos e, conseqüentemente, o tempo requerido no processamento da imagem de entrada. Além disso, a riqueza de detalhes esperados na operação com a imagem também pode modificar os resultados desses tempos.

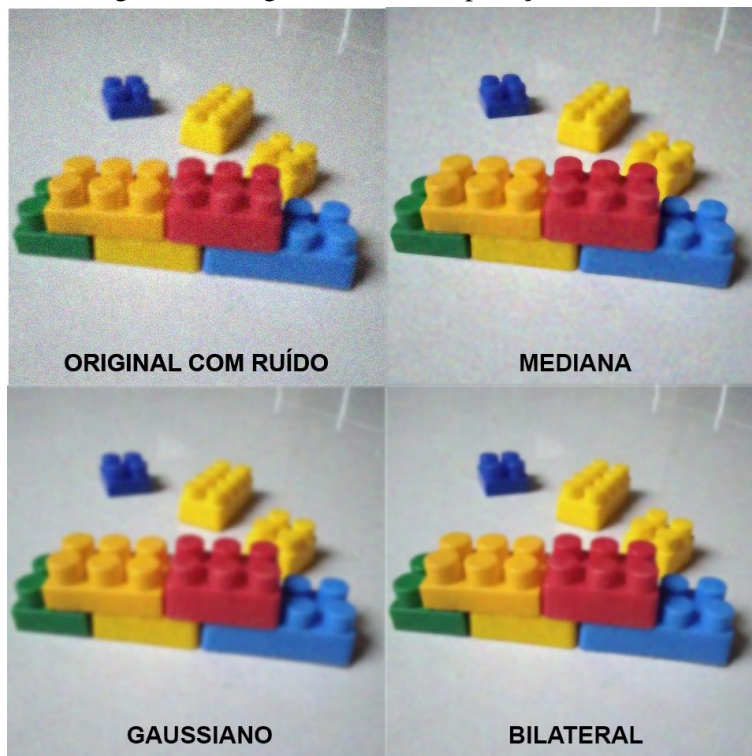
Figura 5 – Tempo de execução com aplicações de filtros



Fonte: Autor

Portanto, todos os testes foram realizados sem muitas alterações na imagem de entrada, a fim de se manter as mesmas características das imagens obtidas. Inseriu-se ruído na imagem original como se pode verificar na Figura 6.

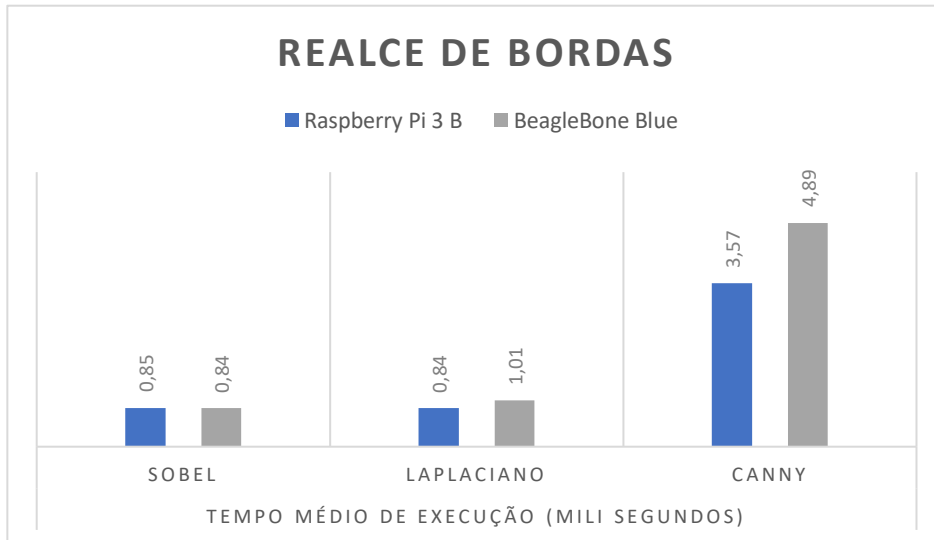
Figura 6 – Imagens obtidas na aplicação dos filtros



Fonte: Autor

• Tempo de execução - Realce de Bordas: mais uma vez, a partir da execução dos três algoritmos de realce de bordas no Raspberry Pi 3B e no BeagleBone blue, gerou-se uma tabela para construção do gráfico da Figura 7.

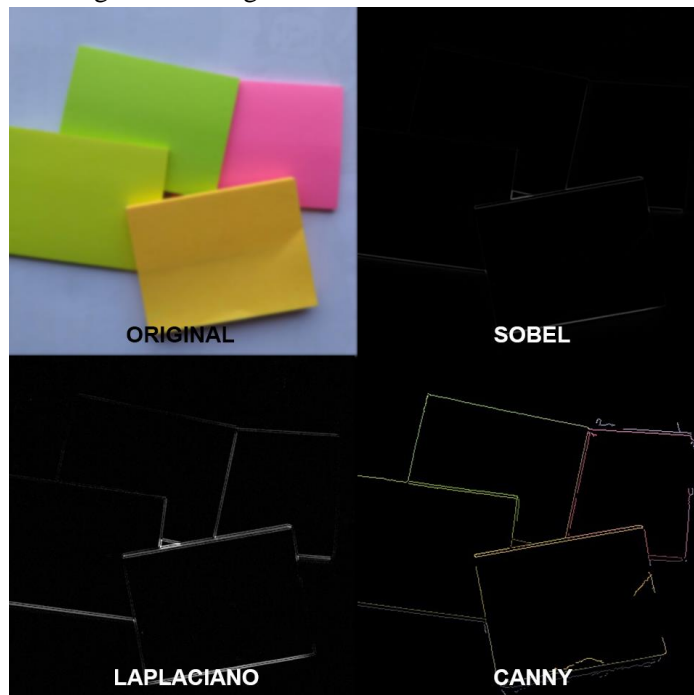
Figura 7 – Tempo de execução com realce de bordas



Fonte: Autor

Na Figura 8 têm-se os resultados de realce de bordas. Verifica-se que o *Canny* apresentou uma melhor delimitação das regiões de transição (bordas).

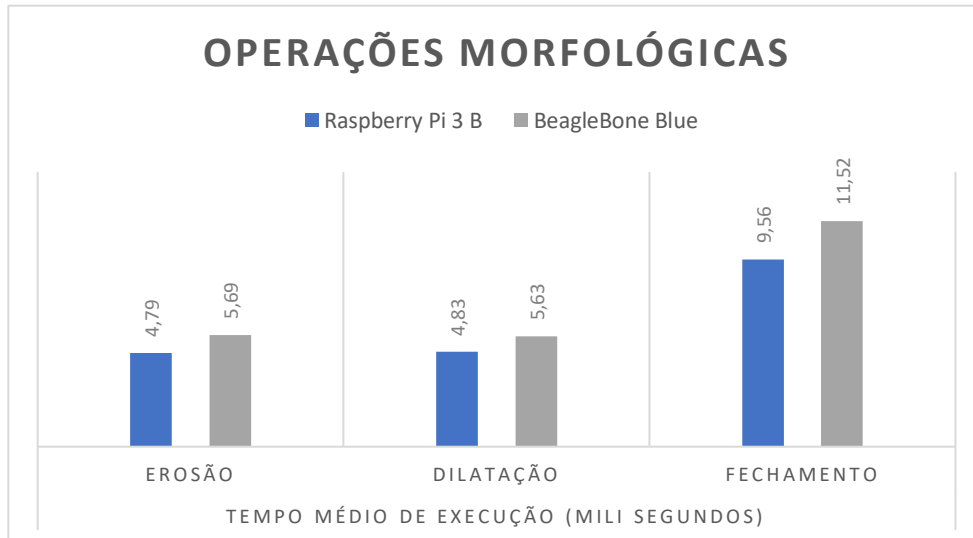
Figura 8 – Imagens obtidas com realce de bordas



Fonte: Autor

• Tempo de execução - Operações Morfológicas: por último os algoritmos de operações morfológicas foram executados no Raspberry Pi 3B e BeagleBone blue para construção do gráfico da Figura 9.

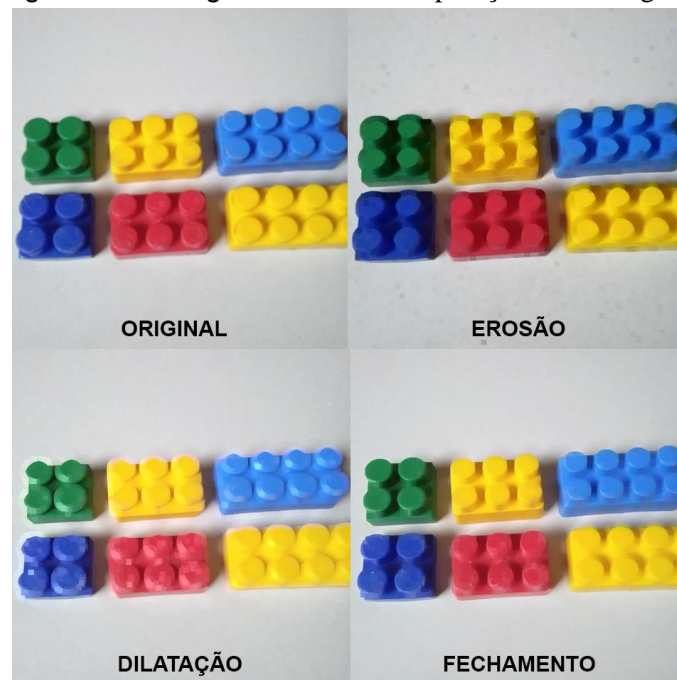
Figura 9 - Tempo de execução com operações morfológicas



Fonte: Autor

A partir do diâmetro dos círculos superiores na Figura 10, podem-se observar as mudanças realizadas pelos algoritmos de operações morfológicas.

Figura 10 – Imagens obtidas com operações morfológicas



Fonte: Autor

### 3.2. Comparação de desempenho dos processamentos de imagens nos dois sistemas embarcados

Uma comparação direta entre os dois sistemas embarcados mostrou-se difícil de ser feita, em função das diferenças de *hardware* que as plataformas apresentam. Como já citado, o Raspberry Pi 3B conta com um processador Quad Core 1.2GHz e 1GB de RAM, enquanto o BeagleBone *blue* possui um processador AM335x 1GHz ARM® Cortex-A8 processor e 512MB DDR3 RAM.

Logo, há uma desvantagem de recursos de *hardware* entre as plataformas pesquisadas, o que já oferece uma desvantagem para o BeagleBone *blue*. Portanto, a análise realizada baseia-se nessa diferença de poder de processamento.

A lógica básica dos algoritmos empregada nos testes na forma de fluxograma está representada na Figura 11. Inicia-se com a captura da imagem através de uma câmera USB, onde o tempo de obtenção da imagem não foi considerado por mostrar-se consideravelmente pequeno, mesmo se a imagem fosse obtida a partir de um diretório (já registrada) ou pela câmera USB (ambiente).

O Raspberry Pi 3B apresentou melhor desempenho em uma grande parcela dos códigos testados, com exceção dos algoritmos de aplicação de filtros, onde em média o BeagleBone *blue* obteve um desempenho melhor.

Em realce de bordas e operações morfológicas o Raspberry Pi 3B obteve melhor desempenho, com um tempo de execução menor. No entanto, apesar de uma diferença de *hardware* considerável, o BeagleBone *blue* mostrou-se bem competitivo, uma vez que os tempos obtidos de execução são próximos (máximo de 36% na operação de realce de bordas no método *Canny*).

O processo de contagem de tempo é computado 100 vezes para que uma média aritmética da métrica seja realizada. Na Figura 11 a operação de testes na forma de fluxograma é detalhada. Cabe ressaltar que os tempos obtidos são resultados dos parâmetros das funções do pacote OpenCV sobre as operações empregadas (filtros, realce de bordas e morfológicas) escolhidos e ainda das imagens utilizadas nesta pesquisa. Dessa forma, quaisquer variações de parâmetros podem mudar os resultados obtidos.

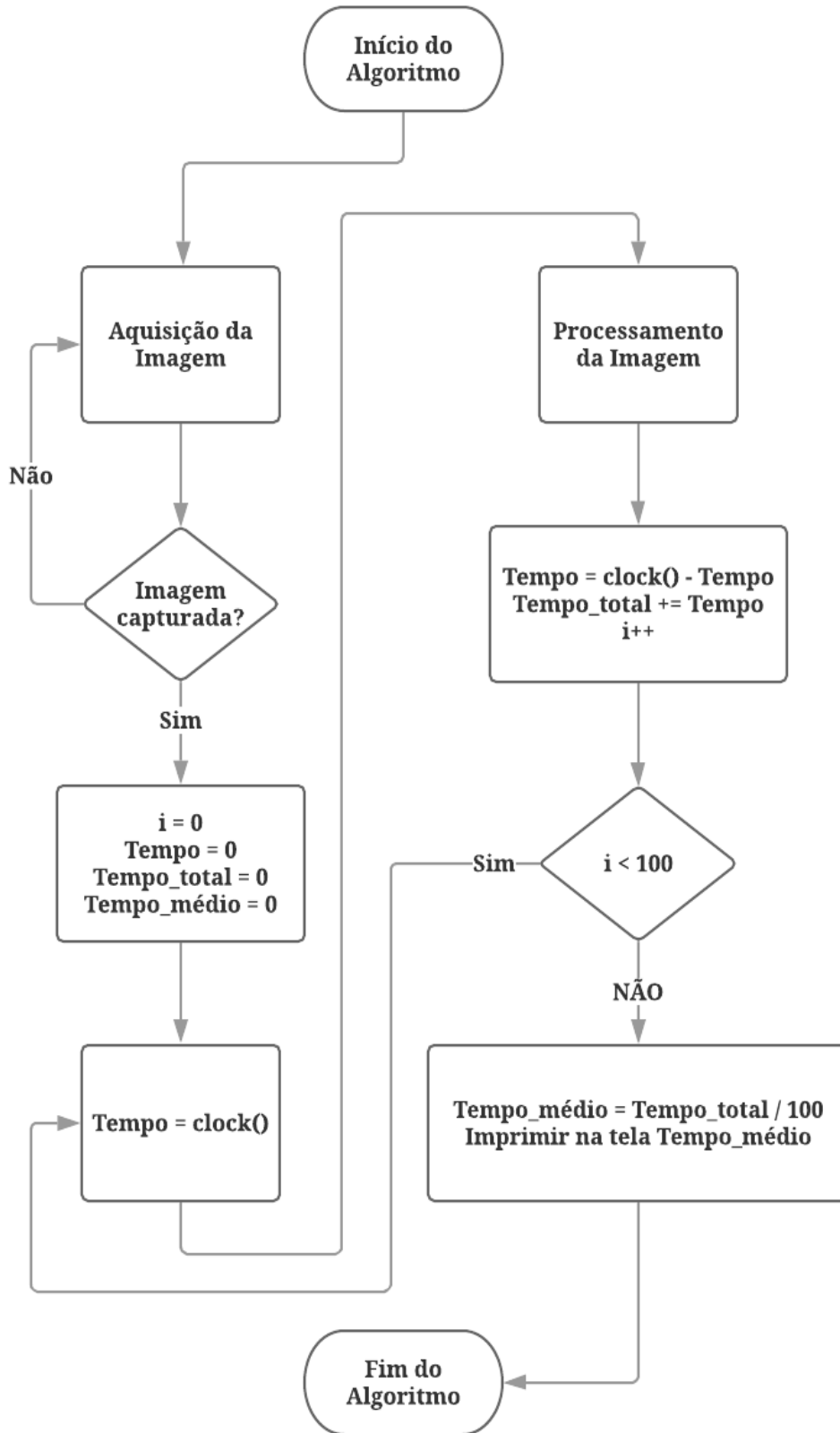
## 4. CONCLUSÃO

Com relação ao tempo de execução dos códigos, pode-se concluir que o Raspberry Pi 3B apresenta um melhor desempenho e, portanto, é mais recomendado para aplicações onde essa métrica é fundamental, como por exemplo, o posicionamento espacial de drones através de câmeras.

Entretanto, outra vertente pode ser considerada, pois o BeagleBone *blue* apresenta uma gama de sensores e interfaces para controle de motores e leitura de variáveis externas tais como: velocidade de rotação de motores, pressão, giros, aceleração etc. Logo, possui uma aplicabilidade consideravelmente alta em sistemas robóticos autônomos, sendo totalmente contrário ao Raspberry Pi 3B que apresenta muitas limitações quanto ao controle direto desses dispositivos externos, requisitando implementações de *hardware* externas.

Portanto, cabe ao projetista que necessita aplicar o processamento de imagens, realizar um estudo da velocidade de resposta que seu sistema requisita para então comparar com os valores médios obtidos nesta pesquisa. Uma margem pequena de diferença em desempenho foi observada nos testes realizados com imagens que pode ser justificada pela diferença de recursos computacionais entre as placas estudadas.

Figura 11 – Fluxograma geral do processo



Fonte: Autor

