

## ESTUDO DO ALGORITMO RANSAC PARA IMPLEMENTAÇÃO EM HDL

BRUNA ARRUDA ARAUJO<sup>1</sup>; MATEUS BECK FONSECA<sup>2</sup>; MARCELO LEMOS ROSSI<sup>3</sup>

<sup>1</sup> Universidade Federal de Pelotas – bruna.arruda.araujo@hotmail.com

<sup>2</sup>Universidade Federal de Pelotas– mateus.fonseca@ufpel.edu.br

<sup>3</sup>Universidade Federal de Pelotas – marcelo.rossi@ufpel.edu.br

### 1. INTRODUÇÃO

O algoritmo RANSAC (RANdom SAmples Consensus) é um método iterativo para estimar os parâmetros de um modelo matemático baseado num conjunto de dados aumentando a probabilidade de encontrar um resultado razoável usando um número finito de iterações. (FISCHLER; BOLLES, 1981).

Devido a relativa simplicidade do algoritmo RANSAC em comparação com outros métodos de estimação robusta, ele se tornou o método padrão usado no desenvolvimento de várias tarefas de visão computacional, tal como a modelagem de imagens 3D a partir de imagens 2D apresentada por ROSSI (2018) e também em sistemas online de detecção e rastreamento de objetos em tempo real.

Esse algoritmo também é frequentemente utilizado em alguns sistemas embarcados, os quais possuem capacidade de processamento e armazenamento limitados; por isso, o aperfeiçoamento em hardware do RANSAC aumentaria a performance do processamento de imagens em tempo real em tais sistemas.

Na maioria das aplicações o RANSAC é utilizado para determinar a matriz fundamental ou os parâmetros que descrevem a transformação entre duas imagens, esse processo é conhecido como estimação geométrica da imagem. (TANG; SHAIKH-HUSIN; SHEIKH; 2013).

### 2. METODOLOGIA

De acordo com MATHWORKS (2018), o algoritmo RANSAC pode ser implementado seguindo os passos a seguir:

1. Aleatoriamente selecionar um subconjunto de dados do conjunto principal.
2. Ajustar um modelo ao subconjunto selecionado.
3. Determinar o número de pontos discrepantes.
4. Repetir os passos 1 a 3 para um número pré-determinado de iterações.

Os dados podem ser classificados em “*inliers*”, isto é, dados cuja distribuição pode ser explicada por algum conjunto de parâmetros do modelo, podem estar sujeitos à pequenos ruídos; e “*outliers*”, que são dados que não se encaixam no modelo. Os *outliers* podem vir, por exemplo, dos valores extremos do ruído, de medições erradas ou hipóteses incorretas sobre a interpretação dos dados (PAWAR; PATIL; 2014).

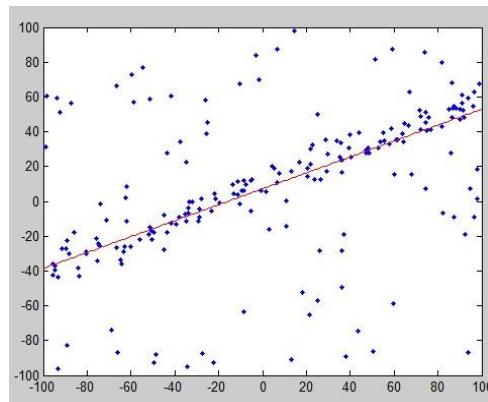


Figura 1 - Exemplo do uso do RANSAC para estimar a equação que melhor se adequa a um conjunto de pontos. Fonte: MATHWORKS, 2018.

Várias modificações do algoritmo RANSAC foram feitas com o objetivo de melhorar sua eficiência em termos de precisão, robustez e velocidade computacional. Esse trabalho foi baseado no projeto desenvolvido por TANG; SHAIKH-HUSIN; SHEIKH (2013) que propôs o uso de um sistema de estimativa geométrica por afinidade que utiliza um processo de cálculo de pontuação de ajuste do modelo de hipótese para todos os dados de entrada (*fitness score*).

O processo de *fitness score* se dá da seguinte forma: se os dados são de um valor não discrepante (*inlier*), o erro é a sua distância de desvio com o ponto de ajuste da hipótese. Se os dados foram de um ponto discrepante (*outlier*), um valor predefinido,  $d$  é usado como pontuação servindo como parâmetro para cada ponto discrepante. Um ponto é considerado discrepante se a distância absoluta entre o ponto de ajuste de hipótese e o ponto do dado for maior que a distância  $d$ . O *fitness score* final é a soma de todos os erros individuais de cada dado. Um perfeito ajuste dará um *fitness score* de 0.

O esforço computacional para o cálculo de *fitness score* aumenta com a quantidade de dados. Para acelerar o hardware foi utilizado 3 estágios de pipeline como mostrado na Figura 2.

A implementação desse trabalho em HDL será feita através da linguagem Verilog.

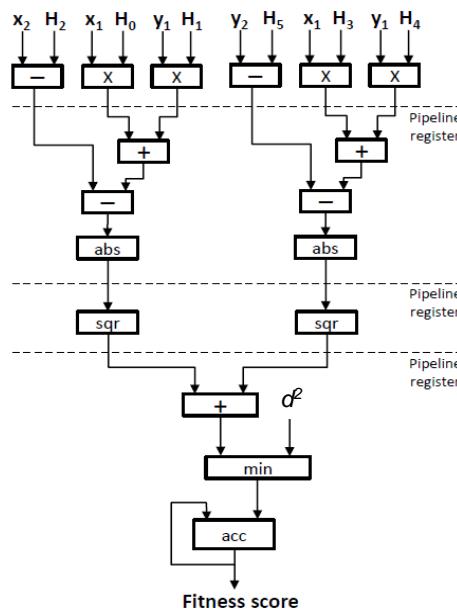


Figura 2 - Hardware para o fitness score.  
Fonte: adaptado de TANG; SHAIKH-HUSIN; SHEIKH (2013).

### 3. RESULTADOS E DISCUSSÃO

O algoritmo RANSAC mostra um desempenho satisfatório em velocidade quando um conjunto pequeno de dados com poucos pontos discrepantes é aplicado; no entanto, demanda muitos recursos, tais como processamento paralelo, prejudicando tarefas mais complexas do projeto no qual é aplicado. Já a aplicação de um conjunto maior com alta taxa de pontos discrepante irá resultar num resultado não satisfatório, assim como a limitação do número de iterações. Para aumentar o desempenho do algoritmo de forma eficiente, utilizando o mínimo de recursos computacionais, somente a tarefa mais intensa computacionalmente foi acelerada em hardware. Isso fornece uma solução balanceada em hardware.

Após concluído o estudo teórico do algoritmo RANSAC, dar-se-á início à implementação em HDL do hardware do fitness score em linguagem Verilog.

### 4. CONCLUSÕES

Durante o estudo do algoritmo RANSAC, percebeu-se a ampla capacidade desse método para estimar parâmetros matemáticos mostrando quão atuante ele pode ser em projetos de engenharia; também houve a oportunidade de entrar em contato com a linguagem Verilog que é amplamente utilizada na indústria para realizar descrição de hardware, aprimorando assim o conhecimento na área de processamento digital de sinais e microeletrônica.

## 5. REFERÊNCIAS BIBLIOGRÁFICAS

FISCHLER, M.A.; BOLLES, R.C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. **Communications of the ACM**. New York, v. 24, n. 6, p. 381-395, 1981.

ROSSI, M.L. **Modelagem 3D de áreas urbanas utilizando processamento digital de imagem**. 2018. Tese (Doutorado em Engenharia Elétrica) – Curso de Pós-graduação em Engenharia Elétrica, Universidade Federal de Uberlândia.

TANG, J.W.; SHAIKH-HUSIN, N.; SHEIKH, U.U. FPGA implementation of RANSAC algorithm for real-time geometry estimation. **2013 IEEE Student Conference on Research and Development**. Putrajaya, p. 290-294, 2013.

PAWAR, A.A.; PATIL, N.W.; Modified approach for recognition of 3-D faces with missing parts, **2014 IEEE International Conference on Computational Intelligence and Computing Research**. Coimbatore, p. 1-6, 2014.

MATHWORKS. **Using RANSAC for estimating geometric transforms in computer vision**. Acessado em 07 set. 2018. Online. Disponível em: <https://www.mathworks.com/discovery/ransac.html>