



# UM PROJETO DE HARDWARE PARA A SOLUÇÃO DE SISTEMAS DE EQUAÇÕES LINEARES NA ESTIMAÇÃO DE MOVIMENTO AFFINE DO PADRÃO VVC

DENIS MAASS<sup>1</sup>; MARCELLO MUÑOZ<sup>1</sup>; MURILO PERLEBERG<sup>1</sup>;  
LUCIANO AGOSTINI<sup>1</sup>; MARCELO PORTO<sup>1</sup>

<sup>1</sup>Universidade Federal de Pelotas – Video Technology Research Group (ViTech)  
denismaass7@gmail.com, {mmunoz, mrperleberg, agostini, porto} @inf.ufpel.edu.br

## 1. INTRODUÇÃO

Nos últimos anos, impulsionado pela popularização da internet e dos serviços de *streaming*, observamos um substancial aumento na demanda por vídeos digitais (MUÑOZ et al., 2023). Além disso, à medida que as resoluções de vídeo crescem, a necessidade de lidar com volumes cada vez maiores de dados para representar esse conteúdo torna-se evidente. Assim, para armazenar e transmitir vídeos digitais de maneira eficaz, torna-se necessário recorrer a técnicas de compressão de dados. Neste cenário, o padrão Versatile Video Coding (VVC) (BROSS et al., 2021), lançado em 2020, destaca-se como o estado da arte em compressão de vídeo.

O padrão VVC aplica inúmeras técnicas que visam diminuir a quantidade de dados necessários para representar um vídeo. Uma das mais importantes consiste em explorar as redundâncias temporais, isto é, a semelhança entre os quadros do vídeo. Esta técnica é chamada de Estimação de Movimento (EM), e consiste em empregar algoritmos de busca para encontrar similaridades entre os blocos do quadro que está sendo codificado e os blocos dos quadros de referência. Onde o bloco com a melhor correspondência, ou bloco predito, é referenciado a partir de um vetor de movimento (VM) que caracteriza o deslocamento translacional do bloco predito em relação ao bloco atual (MUÑOZ et al., 2023).

A EM já foi empregada em diversos padrões de codificação anteriores ao VVC, no entanto, essa técnica se limita apenas à representação de movimentos translacionais. Neste contexto, a novidade no padrão VVC fica por conta da implementação da Estimação de Movimento *Affine* (EMA), que permite representar movimentos mais complexos, como rotação e *zoom* (LI et al., 2018). A EMA pode ser realizada nos modos de 4 ou 6 parâmetros, onde na configuração de 4 parâmetros são utilizados dois Vetores de Movimentos (VMs), conforme ilustrado na Figura 1 (a). Já na configuração de 6 parâmetros, a EMA faz uso dos VMs de três pontos de controle, como exemplificado na Figura 1 (b).

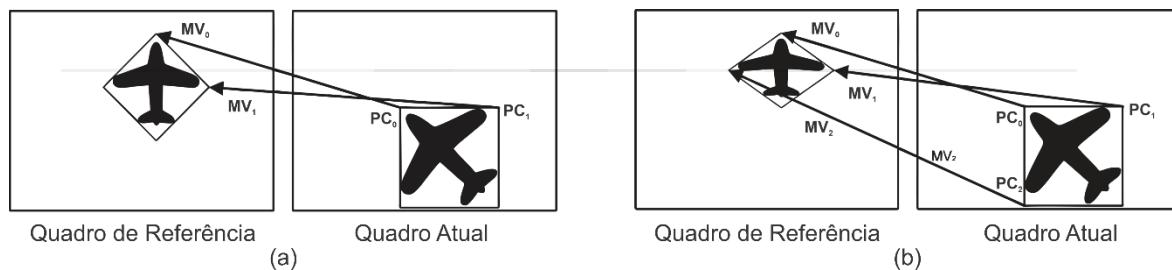


Figura 1: Modos Affine: (a) 4 parâmetros e (b) 6 parâmetros

Para encontrar o conjunto de VMs que melhor descreve o movimento na AME, o VVC emprega um algoritmo iterativo chamado *Affine Merge-Mode*, que refina um conjunto inicial de VMs, baseado nos valores de erro e de gradiente do bloco gerado a cada iteração (MARTINA, 2022). Neste processo, para gerar o próximo

conjunto de MVs a ser testado, é necessária a resolução de um sistema de equações lineares, que dará origem a um conjunto de  $\Delta$ MVs para a próxima iteração. A resolução deste sistema é feita a partir da aplicação do método de Eliminação Gaussiana com Pivotamento Parcial (RODRIGUES DA SILVA, 2018), que consiste em transformar a matriz associada ao sistema em uma forma triangular superior. Para isso são aplicados os seguintes passos:

1. Pivotamento Parcial: Escolher o elemento com o maior valor absoluto na primeira coluna da matriz, trocar as linhas para posicionar o pivô na parte superior da coluna.
2. Eliminação: Utilizar o pivô para eliminar os coeficientes abaixo dele na coluna, tornando-os zero. Isso é feito por meio de operações de linha, subtraindo ou adicionando múltiplos da linha do pivô às outras linhas.
3. Repetição: Repetir os passos 1 e 2 para as colunas subsequentes, indo da esquerda para a direita, escolhendo novos pivôs em cada coluna e eliminando os coeficientes abaixo deles.
4. Resolução: Após a eliminação em toda a matriz, obtemos um sistema triangular superior. A resolução começa pela última linha, onde encontramos o valor da última variável e substituímos esse valor nas linhas superiores para encontrar as variáveis restantes.

## 2. METODOLOGIA

O projeto da arquitetura baseou-se na análise do *software* de referência do padrão VVC, conhecido como VTM (BROWNE, 2022). O algoritmo empregado no *software* foi adaptado para a linguagem de descrição de *hardware* VHDL. Para validar a arquitetura desenvolvida, 100k dados reais de uma codificação de vídeo foram inseridos na arquitetura proposta, e as saídas da arquitetura foram comparadas com as saídas do *software*, a fim de avaliar sua acurácia. Visto que no processo de resolução do sistema são realizadas operações de divisão, as quais resultam em valores fracionários, foram implementadas e comparadas duas versões da arquitetura, uma utilizando 8 e outra 16 bits fracionários. Esse processo foi realizado utilizando um *testbench* e o *software Model/Sim*.

Em uma perspectiva de alto nível, ambas as arquiteturas compartilham a mesma topologia, conforme ilustrado na Figura 2. A entrada do circuito consiste em um conjunto de quatro ou seis equações, dependendo do modelo *Affine*, com cada termo da equação possuindo 32 bits de tamanho. A saída, por sua vez, consiste em quatro ou seis valores de 19 bits.

Internamente, o sistema conta com cinco módulos: Um módulo de controle, que faz o direcionamento dos dados; Módulos de Pivotamento, Eliminação e Resolução, para lidar com cada uma das etapas da Eliminação Gaussiana; E um módulo de arredondamento, que leva em conta o tamanho do bloco sendo processado, para arredondá-lo para um valor inteiro. Para executar todos estes passos e gerar o novo conjunto de VMs podem ser necessários até 103 ciclos de *clock*.

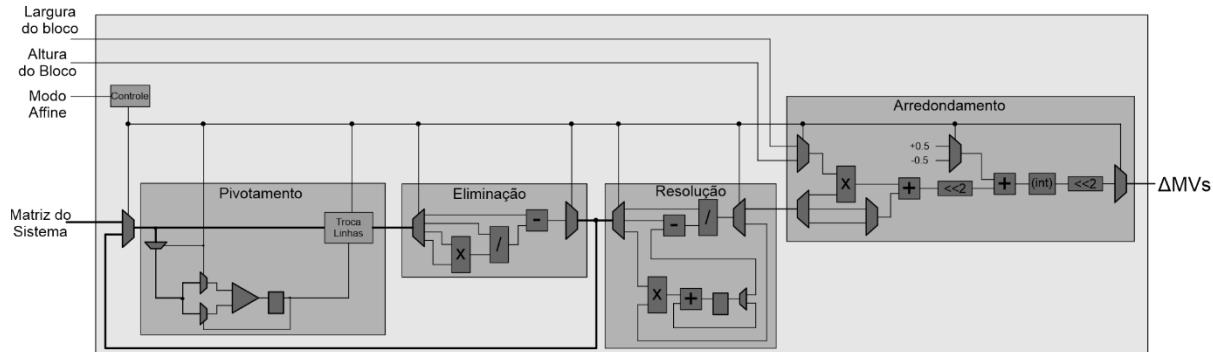


Figura 2: Arquitetura Proposta

As duas versões da arquitetura foram sintetizadas em ASIC utilizando o software *Cadende RTL Compiler* e a biblioteca de células-padrão de 40 nm fornecida pela TSMC. A frequência alvo para o design do ASIC foi definida em 55,08 MHz, garantindo a capacidade de processar blocos com tamanho de  $128 \times 128$  amostras em resolução Ultra High Definition (UHD) 4K ( $3840 \times 2160$ ) a 120 quadros por segundo.

### 3. RESULTADOS E DISCUSSÃO

A Tabela 1 mostra os resultados obtidos para as duas versões da arquitetura proposta. Podemos notar que a versão utilizando 16 bits fracionários possui uma elevada acurácia, gerando a mesma saída que o software de referência em 99,95% das ocorrências, enquanto que a versão de 8 bits alcança uma acurácia de 89,36%. Para atingir esses valores de acurácia, a versão de 16 bits demanda mais recursos em termos de área e maior dissipação de potência em comparação com a versão de 8 bits. Especificamente, a versão de 16 bits registra um aumento de 56,70% na utilização de área e um aumento de 47,72% na dissipação de potência em relação à versão de 8 bits.

Tabela 1: Resultados das duas versões da arquitetura

<b>Bits Fracionários</b>	<b>8</b>	<b>16</b>
Acurácia	89,36%	99,95%
Potência Dissipada (mW)	3,64	5,38
Área (Kgates)	142,20	222,84

### 4. CONCLUSÕES

Este trabalho apresentou uma proposta de *hardware* dedicado para calcular os valores  $\Delta MV$ , necessários para processar a Estimação de Movimento *Affine* do padrão de codificação de vídeo VVC. Até onde se sabe, esse é o primeiro trabalho da literatura propondo um projeto de *hardware* dedicado para essa etapa de codificação. Duas versões diferentes da arquitetura foram avaliadas, levando em consideração a resolução fracionária. Os resultados da análise e síntese indicam que a versão de 16 bits oferece maior precisão, porém, isso acarreta um aumento significativo na utilização de recursos e no consumo de energia. Por outro lado, a versão de 8 bits apresenta uma solução mais eficiente em termos de recursos, com uma pequena redução na precisão.



Trabalhos futuros têm como objetivo aprimorar essa arquitetura, especialmente na paralelização de operações para reduzir o número de ciclos de *clock* necessários para resolver cada sistema de equações e, assim, aumentar a vazão da arquitetura. Além disso, visa-se preparar essa arquitetura para ser incorporada em um projeto de *hardware* para realizar toda etapa da EMA.

## 5. REFERÊNCIAS BIBLIOGRÁFICAS

MUÑOZ, M. et al. Efficient Hardware Design for the VVC Affine Motion Compensation Exploiting Multiple Constant Multiplication. **2023 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)**, p.1-6, 2023.

BROSS, B. et al. Overview of the Versatile Video Coding (VVC) Standard and its Applications. **IEEE Transactions on Circuits and Systems for Video Technology**, v.31, n.10, p.3736-3764, 2021.

LI, L. et al. An Efficient Four-Parameter Affine Motion Model for Video Coding. **IEEE Transactions on Circuits and Systems for Video Technology**, v.28, n.8, p.1934-1948, 2018.

MARTINA, M. **Simplified Affine Motion Estimation algorithm and architecture for the Versatile Video Coding standard**. 2022. Tese de Doutorado. Politecnico di Torino.

RODRIGUES DA SILVA, N.; PEREIRA DE SOUZA, F.; ROMANINI, E. **MÉTODO DE FATORAÇÃO LU PARA SOLUÇÃO DE SISTEMAS LINEARES**. Colloquium Exactarum. ISSN: 2178-8332, [S. I.], v. 9, n. 4, p. 41–47, 2018. Disponível em: <https://revistas.unoeste.br/index.php/ce/article/view/2267>. Acesso em: 22 set. 2023.

BROWNE, A., Y. Ye, and S. Kim, “**Algorithm description for versatile video coding and test model 17 (vtm 17)**,” JVET-Z2002, July 2022.