

Capítulo 3

Processamento de Vídeo

O seguinte capítulo aborda o mecanismo de representação de vídeos e os seus respetivos detalhes. O processo de reamostragem e redimensionamento de vídeo é analisado tendo em conta os diferentes algoritmos para o efeito e os seus resultados. Como conclusão deste capítulo são enunciadas ferramentas de processamento de imagem que são utilizadas atualmente para realizar este tipo de operações em meios de pós-produção de vídeo profissional.

3.1 Estrutura de um Vídeo

Um vídeo é uma sequência de imagens, usualmente designadas de frames. Cada frame é uma imagem estática que visualizada em sequência, em conjunto com outras frames, recriam uma imagem animada do vídeo. O número de frames que constituem um vídeo depende do número de imagens capturadas na sua gravação. Este valor refere-se à métrica *frame rate*. ~~Esta medida é o~~ número de frames apresentadas por segundo, ou FPS, na reprodução de um vídeo. A *frame rate* também pode ser designada de *frame frequency* e ser expressa em hertz ao invés de frames por segundo.

Na indústria de multimédia é considerado um valor de *frame rate* normalizado entre vinte e quatro, e trinta e um frames por segundo para captura digital de vídeos [20]. À semelhança das frames de um vídeo a visão humana captura imagens de cenários reais. A percepção de movimento da visão humana é criada por um processo de desfocagem das características entre imagens consecutivas. Este processo cria a sensação de continuidade de movimento entre duas imagens.

O valor mínimo de frames por segundo em que o processo de desfocagem acontece entre duas imagens é de vinte e quatro frames por segundo [21]. Um valor menor que vinte e quatro frames por segundo retira a percepção visual humana de continuidade entre imagens, enquanto um maior valor implica maiores custos de pós-produção do vídeo devido ao vídeo ser constituído por um número maior de imagens.

O processamento de um vídeo, como a aplicação do processo de reamostragem e redimensionamento que será abordado na seguinte seção, pode ser tomada como a aplicação do mesmo processo à sequência de imagens que constituem o vídeo. Por esta razão, a resolução do problema de processamento de um vídeo é considerado como a resolução do mesmo problema numa só imagem.

3.1.1 Representação de uma Imagem

As imagens podem ser distinguidas segundo o mecanismo de armazenamento gráfico utilizado na sua representação digital por imagens vetorizadas e imagens rasterizadas.

As imagens vetorizadas são geradas a partir de formulações matemáticas que definem primitivas geométricas, as suas posições relativas em relação a outras primitivas e as suas respetivas cores. As primitivas geométricas que constituem as imagens vetorizadas são representadas por pontos, linhas, curvas e formas geométricas básicas. A disposição e a escala destas primitivas são calculadas em função da resolução em que a imagem será apresentada. Por esta razão, é possível apresentar este tipo de imagens com diferentes dimensões sem que haja alteração da qualidade das suas características. Enquanto, as imagens rasterizadas são representadas digitalmente a partir de uma matriz de duas dimensões que armazena os valores de intensidade de cor para cada um dos pontos da imagem.

Como as imagens vetorizadas são constituídas a partir de primitivas geométricas, este tipo de imagens são inviáveis para a representação de imagens capturadas **a partir de um cenário real por não apresentarem detalhe suficiente das formas e cores das características de um cenário real.** Tendo em conta o contexto do problema desta dissertação serão apenas tidas em conta imagens rasterizadas.

3.2 Modelos de cor

Os pixels de uma imagem representam o valor de intensidade de cor de um ponto de uma imagem segundo um modelo de cor específico. Um modelo de cor descreve de forma normalizada como o valor de intensidade de cor é representado utilizando um sistema de coordenadas de três dimensões segundo o qual cada valor de intensidade de cor é representado por um único ponto desse sistema.

Os modelos de cor estão divididos em duas diferentes categorias: os modelos de cor específicos para representação de imagens e da qualidade das suas características, como os modelos RGB, CMY, HSI; e os modelos de cor específicos para a representação de imagens de um vídeo focando-se na redução do tamanho necessário para a sua representação em detrimento das qualidade das características das suas imagens, como os modelos YIQ, YUV, Y'CbCr.

No contexto do trabalho desta dissertação são apenas considerados os modelos YUV e Y'CbCr, pois os últimos são considerados standards quanto aos modelos de cor utilizados nos dispositivos de reprodução de vídeo. As designações dos modelos de cor YUV e Y'CbCr são consideradas

permutáveis neste trabalho, visto que ambos modelos são caracterizados pelas mesmas componentes de cor de uma imagem e diferem apenas para que tipo de dispositivo de reprodução de vídeos são direcionados. O modelo YUV é destinado para dispositivos de reprodução de vídeo analógico, enquanto o modelo Y'CbCr é direcionado para dispositivos de reprodução de vídeo digital [22].

As diferenças entre os dispositivos de reprodução de vídeo analógico e digital são mais relevantes em relação à representação de cores de uma imagem. Tendo em conta os diferentes dispositivos alvo dos modelos de cor YUV e Y'CbCr, ambos têm diferentes valores de coeficientes no cálculo das suas amostras de crominância. Nas equações seguintes é possível observar as diferenças nos valores de crominância presentes na conversão entre modelos de cor de vídeo RGB e YUV, e RGB e Y'CbCr [23], respetivamente:

$$Y = 0.299 * R + 0.587 * G + 0.114 * B \quad (3.1)$$

$$U = -0.147 * R - 0.289 * G + 0.436 * B \quad (3.2)$$

$$V = 0.615 * R - 0.515 * G - 0.1 * B \quad (3.3)$$

$$Y' = 0.299 * R + 0.587 * G + 0.114 * B \quad (3.4)$$

$$Cb = -0.169 * R - 0.331 * G + 0.499 * B + 128 \quad (3.5)$$

$$Cr = 0.499 * R - 0.418 * G - 0.0813 * B + 128 \quad (3.6)$$

3.2.1 Modelo YUV

O modelo de cor YUV é um modelo que tira proveito da fraca percepção da visão humana às variações de cor em relação às variações de luminosidade de uma imagem [24]. O valor de cor de um pixel é constituído por três ~~diferentes~~ componentes que correspondem à componente luma, o brilho de uma imagem, e duas outras componentes de crominância correspondentes às componentes de cor segundo uma projeção de cor azul e vermelha, respetivamente.

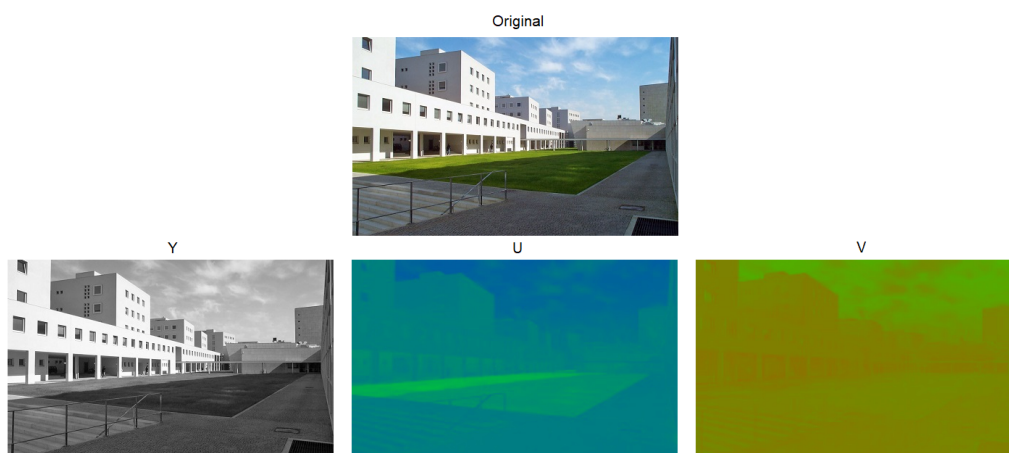


Figura 3.1: Decomposição de uma imagem nas componentes do modelo de cor YUV.

Segundo este modelo, a cor de um ponto de uma imagem é igual ao resultado da junção dos valores de intensidade de cada uma das três componentes referidas. Pode ser visualizado na figura anterior, a figura 3.1, a decomposição de uma imagem nas diferentes componentes de luma e croma.

De acordo com este modelo é possível tirar partido da fraca percepção da visão humana às alterações das componentes de croma de uma imagem através da redução do número de amostras que as constituem. Esta prática é designada de subamostragem de croma.

3.2.1.1 Subamostragem de Croma

A subamostragem de croma é a prática que consiste em reduzir o número de amostras das componentes de croma de uma imagem segundo o modelo YUV. Esta prática permite reduzir o tamanho utilizado na representação de uma imagem através da partilha de valores das componentes de croma por vários blocos de pixels [7]. A subamostragem de croma é definida por diferentes tipos que podem ser observados na seguinte figura 3.2:

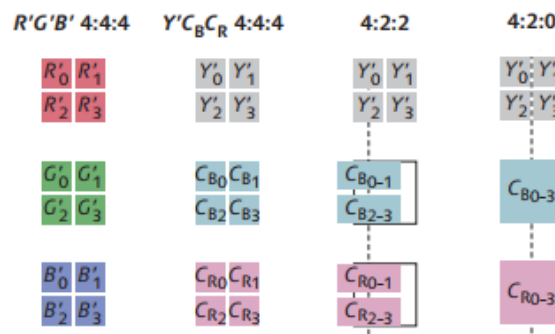


Figura 3.2: Diferentes tipos de subamostragem de croma [7].

Segundo uma subamostragem do tipo 4:4:4, cada pixel é constituído por um valor próprio de cada uma das componentes de cor do modelo YUV. Com uma subamostragem do tipo 4:2:2, cada dois pixels horizontais consecutivos partilham o mesmo valor das componentes de croma, mantendo apenas cada um o seu respetivo valor de luma. Por último, com uma subamostragem do tipo 4:2:0, cada bloco de quatro pixels adjacentes partilham os mesmos valores das componentes de croma.

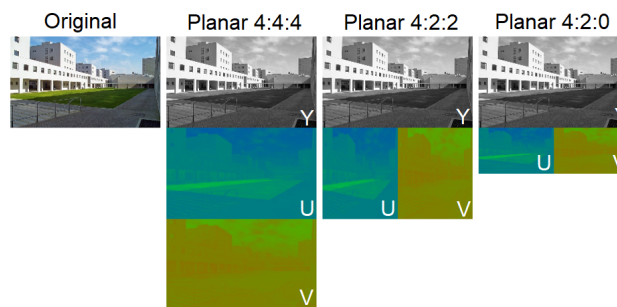


Figura 3.3: Subamostragem de croma segundo o modelo YUV.

Este mecanismo de subamostragem de crominâncias é a principal razão da utilização frequente do modelo de cor YUV na área de multimídia para representar conteúdos de vídeo. Com a subamostragem de crominâncias é possível reduzir o tamanho de representação dos vídeos assim como diminuir o esforço computacional de processamento dos mesmos [22]. Entre os dois tipos de subamostragem 4:4:4 e 4:2:0, por exemplo, existe uma redução de 50% de amostras utilizadas na representação da imagem o que por consequência indica uma diminuição do seu tamanho e dos recursos computacionais para o seu processamento [25].

3.2.1.2 Profundidade de Cor

A profundidade de cor é o número de bits utilizados na representação do valor de cada componente de cor de um pixel. Quanto maior o valor da profundidade de cor maior será o intervalo de valores que cada componente pode representar e, por consequência, maior será a variedade de tonalidades de cor que um pixel pode tomar [25]. A figura 3.4 demonstra as diferenças das tonalidades de cor presentes numa imagem com diferentes valores de profundidade de cor, respectivamente: 8, 4, 2 e 1 bit(s).

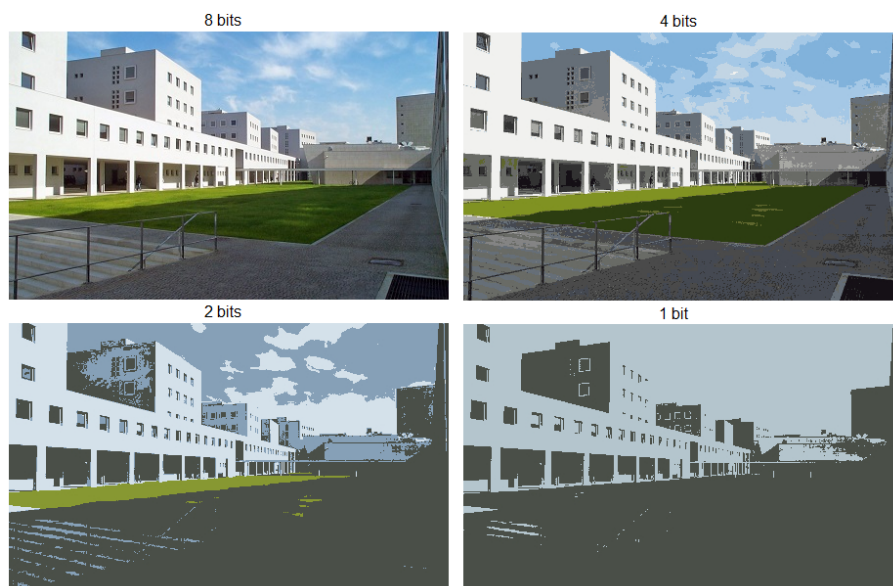


Figura 3.4: Variação de tonalidades de cor em função do valor de profundidade de cor.

3.3 Reamostragem e Redimensionamento

A reamostragem e redimensionamento de vídeo é o processo de alteração das dimensões das frames que o constituem e é um processo realizado na maioria dos dispositivos de reprodução de vídeo. As dimensões de um vídeo são expandidas ou reduzidas de modo a que o mesmo seja ajustado a uma determinada área de exibição ou para satisfazer uma restrição imposta de tamanho de imagem.

A reamostragem de sinal é o processo de alteração da taxa de amostragem de um sinal discreto baseado no sinal contínuo subjacente [26]. A alteração da taxa de amostragem de um sinal **refere-se mudança** do número de amostras que o constitui. Em específico ao tema deste trabalho, a alteração da taxa de amostragem de um vídeo **refere-se à mudança** do número de pixels que constituem cada frame. A inserção de novos pixels numa frame implica o aumento das suas dimensões, ou *upscaling*, enquanto a diminuição das dimensões de uma frame, ou *downscaling*, implica a remoção de parte dos pixels.

A reamostragem ou redimensionamento não é apenas realizada com a alteração do número de pixels de uma frame mas, também, é necessário a correção dos valores da intensidade de cor dos pixels das frames reamostradas para que as características visuais sejam mantidas o quanto possível. Para realizar este processo podem ser utilizados uma variedade de algoritmos que balanceiam a qualidade dos resultados e o tempo de execução da aplicação do processo.

Como com qualquer algoritmo de reamostragem de sinal, a alteração das dimensões de um vídeo induz artefactos como desfocagem dos detalhes e inserção de ruído na imagem, assim como distorção das características visuais apresentadas [27].

A dimensão com que as imagens são criadas têm em conta a aparência das suas características ao serem representadas numa determinada resolução. A alteração da dimensão de uma imagem provoca um desalinhamento entre a disposição original dos seus pixels e as novas posições dos pixels da imagem redimensionada. A imagem da figura 3.5 representa o desalinhamento consequente da reamostragem de uma imagem em relação à disposição original dos seus pixels:

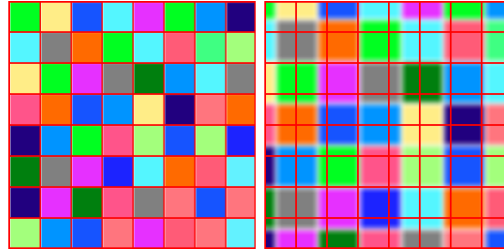


Figura 3.5: Desalinhamento entre pixels reamostrados e a sua disposição original.

Considerando uma imagem original I_s de dimensões W_s de largura e H_s de altura, e uma imagem reamostrada I_t de dimensões W_t de largura e H_t de altura, os fatores de reamostragem horizontal e vertical são, respetivamente, R_h e R_v . Estes fatores, formulados pelas equações 3.7 e 3.8, são o valor da razão entre as taxas de amostragem da imagem reamostrada e da imagem original.

$$R_h = \frac{W_t}{W_s} \quad (3.7)$$

$$R_v = \frac{H_t}{H_s} \quad (3.8)$$

Assume-se as funções $I_s(x, y)$ e $I_t(x, y)$ como formulações do mapeamento do valor de intensidade de cor um pixel na coluna x e na linha y da imagem original e da imagem reamostrada, respetivamente. Como as imagens I_s e I_t apresentam diferentes taxas de amostragem devido ao

número de pixels que as constituem, a posição de um pixel da imagem reamostrada $I_t(x, y)$ é calculada a partir de:

$$I_t(x, y) = I_s\left(\frac{x}{R_h}, \frac{y}{R_v}\right) \quad (3.9)$$

Por essa razão, a posição de um pixel da imagem reamostrada pode ser mapeada, dependendo do fator de amostragem, a uma posição contínua da imagem original I_s .

Pela definição de reamostragem, um certo sinal amostrado é reconstruído de modo a gerar um sinal contínuo de dados. O último é amostrado a uma taxa de amostragem diferente do sinal inicial. Desta forma, é possível calcular o valor da intensidade de cor de um pixel reamostrado que se encontra numa posição contínua da imagem original. Esta definição descreve o comportamento de interpolação [28].

Assim, a operação de reamostragem pode ser formulada como uma convolução que define o princípio geral de interpolação [29], apresentada em 3.10. As variáveis u e v correspondem à posição de um pixel da imagem reamostrada numa posição da imagem original e são dados a partir de $u = x/R_h$ e $v = y/R_v$, respetivamente. O *kernel* de interpolação $k(x)$, também designado de filtro de reconstrução, é o método utilizado para interpolar os dados da imagem original de modo a reconstruir o sinal contínuo da mesma. A variável a é o valor de apoio de pixels do *kernel* de interpolação e define o número de amostras originais a ter em conta na operação de interpolação.

$$I_t(x, y) = \sum_{i=\lfloor u \rfloor - a + 1}^{\lfloor u \rfloor + a} \sum_{j=\lfloor v \rfloor - a + 1}^{\lfloor v \rfloor + a} I_s(\lfloor u \rfloor, \lfloor v \rfloor) * k(u - i) * k(v - j) \quad (3.10)$$

3.3.1 Filtros de Reconstrução

Os filtros de reconstrução são o aspeto fundamental das operações de interpolação, visto que os últimos apresentam diferentes complexidades de reconstrução do sinal contínuo a partir de amostras. Os filtros de reconstrução são caracterizados pelo seu valor de apoio de pixels, o tempo de execução do *kernel* de interpolação e a qualidade dos resultados relativamente ao filtro de reconstrução ideal, o filtro *sinc*.

O filtro *sinc* é um filtro do tipo IIR, ou Infinite Impulse Response, o que indica que este filtro tem uma resposta de impulso infinito [30]. A aplicação deste filtro na operação de reamostragem de imagens seria uma abordagem muito complexa, visto que o valor de intensidade de cor de todos os pixels da imagem reamostrada seriam calculados a partir da interpolação de todos os pixels da imagem original [31].

A aplicação do filtro de reconstrução ideal *sinc* na operação de reamostragem de imagens tem uma complexidade temporal de $(W_t H_t)^{W_s H_s}$, porém esta complexidade pode ser reduzida para $(W_t H_t)^{a^2}$ utilizando uma abordagem por regiões de processamento. Esta última abordagem refere-se ao processamento de um pixel da imagem reamostrada através de um filtro de reconstrução considerando apenas a amostras da imagem original.

O filtro de reconstrução é o método utilizado para atribuir um peso ao valor de intensidade de cor de uma amostra da imagem original no cálculo de um pixel da imagem reamostrada. O peso calculado pelo filtro de reconstrução é obtido em função da distância entre as posições da amostra da imagem original e da posição do pixel da imagem reamostrada mapeada na imagem original.

3.3.1.1 Box Filter

O *box filter*, também designado de interpolação de *nearest neighbor*, é um filtro de interpolação com um apoio de pixel de $a = 2$, o que indica que durante a interpolação apenas quatro amostras originais são utilizadas para obter o valor de intensidade de cor do pixel da imagem reamostrada.

Com este filtro, o valor de intensidade de cor de um pixel da imagem reamostrada é igual ao valor do pixel mais próximo da imagem original. Este tipo de interpolação tem um reduzido tempo de execução, visto que não é necessário cálculos aritméticos complexos mas apenas uma atribuição do valor da intensidade de cor da imagem original ao pixel da imagem reamostrada.



Figura 3.6: Qualidade de resultados da interpolação com um *box filter*.

Porém, a qualidade dos resultados obtidos por esta interpolação apresentam uma acentuada ocorrência de *aliasing* [32], diferenças abruptas entre os gradientes de cor das características visuais, como pode ser visualizado na figura anterior. O *box filter* é formulado matematicamente segundo a seguinte equação:

$$k(x) = \begin{cases} 1 & |x| < 0.5 \\ 0 & \text{senão} \end{cases} \quad (3.11)$$

3.3.1.2 Filtro Linear

O filtro de reconstrução linear, também designado de interpolação bilinear ou *tent filter*, é um filtro de interpolação com valor de apoio de pixels $a = 2$. Segundo este filtro, o valor de intensidade de cor de um pixel da imagem escalada será igual à média ponderada dos quatro pixels originais mais próximos.

O filtro linear é uma interpolação mais sofisticada em relação ao *box filter*, pois são utilizadas operações de interpolação linear para calcular o valor de intensidade de cor de um pixel da imagem

reamostrada. Quanto maior for a proximidade de um pixel original ao pixel da imagem escalada, maior será a sua influência no valor de intensidade de cor a calcular.



Figura 3.7: Qualidade de resultados da interpolação com um filtro linear.

Este filtro introduz novos valores de intensidade de cor que poderiam não existir na imagem original por consequência das interpolações realizadas. Por esta razão as imagens reamostradas recorrendo a este filtro de reconstrução apresentam características desfocadas especialmente em locais da imagem em que existem grandes discrepâncias de intensidade de cor, como pode ser observado na figura anterior. O filtro de interpolação linear apresentado tem a seguinte formulação matemática:

$$k(x) = \begin{cases} 1 - |x| & |x| < 1 \\ 0 & \text{senão} \end{cases} \quad (3.12)$$

3.3.1.3 Spline Filter

Os resultados do filtro de reconstrução por *spline* apresentam uma melhor qualidade do que os filtros anteriores [32], pois o valor de intensidade de cor de cada pixel da imagem reamostrada é obtido pela interpolação cúbica dos dezasseis pixeis da imagem original mais próximos de acordo com o valor de apoio de pixeis de $a = 4$. A boa qualidade dos resultados deste filtro é contrabalançada com o moroso tempo de execução da sua aplicação.

O filtro por *spline* tem uma variedade de formulações que determinam a presença e intensidade de artefactos de imagem [33]. Os artefactos de imagem mencionados são a desfocagem e distorção das características visuais, e erros no cálculo de valores interpolados em transições abruptas de gradiente, também designado de *ringing*.

$$k(x) = \frac{1}{6} \begin{cases} (12 - 9B - 6C)|x|^3 + (-18 + 12B + 6C)|x|^2 + (6 - 2B) & |x| < 1 \\ (-B - 6C)|x|^3 + (6B + 30C)|x|^2 + (-12B - 48C)|x| + (8B + 24C) & 1 \leq |x| < 2 \\ 0 & \text{senão} \end{cases} \quad (3.13)$$

A família de filtros de reconstrução por *spline* é formulado segundo a equação apresentada em 3.13, onde B e C regulam a qualidade dos resultados e a presença dos artefactos de imagem mencionados.



Figura 3.8: Qualidade de resultados da interpolação com um filtro por *spline*.

Segundo Mitchell e Netravali, o filtro de reconstrução por *spline* com o valor de B igual a 0 e C igual a 0.6, produz resultados próximos dos que seriam esperados com o filtro de reconstrução ideal *sinc* apesar da indução de artefactos de imagem *ringing* [33]. Os resultados e a formulação deste filtro em específico podem ser observados, respetivamente, pela figura 3.8 e a seguinte equação:

$$k(x) = \begin{cases} 1.4|x|^3 - 2.4|x|^2 + 1 & |x| < 1 \\ 0.6|x|^3 + 3|x|^2 - 4.8|x| + 2.4 & 1 \leq |x| < 2 \\ 0 & \text{senão} \end{cases} \quad (3.14)$$

3.3.1.4 Visão Geral

Na figura 3.9 são apresentados os resultados de cada um dos filtros onde é possível visualizar o **o detalhes** das imagens reamostradas e redimensionadas, e a presença dos diferentes artefactos de imagem. É possível verificar as transições abruptas de gradiente nas arestas dos detalhes da imagem pela interpolação *Nearest Neighbor*, a desfocagem das características visuais do filtro linear e o *ringing* do filtro por *spline* de Mitchell e Netravali.



Figura 3.9: Comparação de resultados entre diferentes filtros de reconstrução.

3.4 Formatos de Pixels

De modo a implementar uma solução de processamento de uma imagem, é necessário entender as diferentes peculiaridades de representação de um pixel na componente de memória de um computador. Na seguinte secção serão abordados os diferentes formatos de representação interna de pixels do modelo de cor YUV. Os formatos de pixels especificam o número de bits utilizados na representação de cada componente de cor de um pixel e a sua organização de acordo com o tipo de subamostragem de crominâncias da imagem.

3.4.1 Formatos Entrelaçados

Os formatos entrelaçados de pixels são caracterizados pela representação consecutiva de pixels. Este tipo de formatos são alocados a uma região contígua de memória onde as diferentes componentes de cor são representadas de acordo com o tipo de subamostragem de crominâncias.

Plano A

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
U0	U0	U0	U0	U0	U0	U0	U0	Y0	Y0	Y0	Y0	Y0	Y0	Y0	Y0	V0	V0	V0	V0	V0	V0	V0	V0	Y1	Y1	Y1	Y1	Y1	Y1	Y1	Y1

Figura 3.10: Representação interna do formato de pixel UYVY.

O formato de pixels UYVY é uma das representações mais utilizadas entre os formatos entrelaçados. Segundo o formato UYVY cada componente é representada com uma profundidade de cor de 8 bits. Também, utiliza uma subamostragem de crominâncias do tipo 4:2:2, o que indica a partilha dos valores de crominância entre cada dois pixels consecutivos.

Na figura 3.10 é apresentada a representação interna do formato de pixel UYVY na componente de memória de uma máquina. Os formatos entrelaçados são caracterizados pela representação de todos os pixels e as suas componentes de cor numa região contígua de memória, designada na figura como *Plano A*. Este formato segue um padrão de representação das componentes de cor de cada par de pixels consecutivos: componente crominância U, seguida da componente luma do primeiro pixel do par, a componente de crominância V e, por último, o valor de luma do segundo pixel do par.

3.4.2 Formatos Planares

Os formatos planares de pixels são caracterizados pela representação individual das componentes de cor de pixels. Cada componente de cor dos pixels de uma imagem, segundo este formato, são alocadas a uma região independente de memória.

Os seguintes formatos de pixels, respetivamente, YUV422p e YUV420p são exemplos de formatos planares. As diferentes componentes de cor, a componente de luma e as duas componentes de crominâncias, são organizadas cada uma de forma individual em relação às restantes.

Na figura anterior é apresentada a representação em memória dos pixels de uma imagem segundo o formato de pixels YUV422p. Este tipo de formato é específico a subamostragem de

Processamento de Vídeo

Plano A

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Y0	Y0	Y0	Y0	Y0	Y0	Y0	Y0	Y1	Y1	Y1	Y1	Y1	Y1	Y1	Y1	Y2	Y2	Y2	Y2	Y2	Y2	Y2	Y2	Y3	Y3	Y3	Y3	Y3	Y3	Y3	Y3
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Y4	Y4	Y4	Y4	Y4	Y4	Y4	Y4	Y5	Y5	Y5	Y5	Y5	Y5	Y5	Y5	Y6	Y6	Y6	Y6	Y6	Y6	Y6	Y6	Y7	Y7	Y7	Y7	Y7	Y7	Y7	Y7

Plano B

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
U0	U0	U0	U0	U0	U0	U0	U0	U2	U2	U2	U2	U2	U2	U2	U2
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
U4	U4	U4	U4	U4	U4	U4	U4	U6	U6	U6	U6	U6	U6	U6	U6

Plano C

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
V0	V0	V0	V0	V0	V0	V0	V0	V2	V2	V2	V2	V2	V2	V2	V2
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
V4	V4	V4	V4	V4	V4	V4	V4	V6	V6	V6	V6	V6	V6	V6	V6

Figura 3.11: Representação interna do formato de pixel YUV422p.

crominâncias do tipo 4:2:2 e, por essa razão, cada uma das componentes de crominância apresentam metade das amostras da componente de luma. Também, este formato é distinto a componentes de cor com profundidade de 8 bits.

Como já referido, as componentes de cor dos formatos de pixels planares são representadas de forma individual das restantes componentes como pode ser observado na figura 3.11 a divisão pelos três diferentes planos. O plano A corresponde à representação da componente de luma dos pixels e, o plano B e C correspondem à representação da componente de crominância segundo uma projeção azul e vermelha, respetivamente.

Plano A

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Y0	Y0	Y0	Y0	Y0	Y0	Y0	Y0	Y1	Y1	Y1	Y1	Y1	Y1	Y1	Y1	Y2	Y2	Y2	Y2	Y2	Y2	Y2	Y2	Y3	Y3	Y3	Y3	Y3	Y3	Y3	Y3
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Y4	Y4	Y4	Y4	Y4	Y4	Y4	Y4	Y5	Y5	Y5	Y5	Y5	Y5	Y5	Y5	Y6	Y6	Y6	Y6	Y6	Y6	Y6	Y6	Y7	Y7	Y7	Y7	Y7	Y7	Y7	Y7

Plano B

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
U0	U0	U0	U0	U0	U0	U0	U0	U2	U2	U2	U2	U2	U2	U2	U2

Plano C

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
V0	V0	V0	V0	V0	V0	V0	V0	V2	V2	V2	V2	V2	V2	V2	V2

Figura 3.12: Representação interna do formato de pixel YUV420p.

O formato YUV420p apresenta as mesmas características de representação de pixels do formato YUV422p. Contudo, o formato YUV420p é destinado à representação de pixels de imagem que seguem uma subamostragem de crominâncias do tipo 4:2:0, o que difere do formato YUV422p. Essa peculiaridade pode ser observado pelo menor número de amostras das componentes de crominância na representação do mesmo número de pixels na figura 3.12.

3.4.3 Formatos Semi-Planares

Os formatos de pixels semi-planares consideram parte das características de representação de pixels de uma imagem de **ambos tipo** de formatos mencionados nas seções anteriores. Isto é, os modelos semi-planares integram a característica planar da representação individual de parte das componentes de cor e o entrelaçamento dos seus valores segundo, respetivamente, os formatos planares e entrelaçados.

Plano A																															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Y0	Y0	Y0	Y0	Y0	Y0	Y0	Y0	Y1	Y1	Y1	Y1	Y1	Y1	Y1	Y1	Y2	Y2	Y2	Y2	Y2	Y2	Y2	Y2	Y3	Y3	Y3	Y3	Y3	Y3	Y3	Y3
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Y4	Y4	Y4	Y4	Y4	Y4	Y4	Y4	Y5	Y5	Y5	Y5	Y5	Y5	Y5	Y5	Y6	Y6	Y6	Y6	Y6	Y6	Y6	Y6	Y7	Y7	Y7	Y7	Y7	Y7	Y7	Y7

Plano B																															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
U0	U0	U0	U0	U0	U0	U0	U0	V0	V0	V0	V0	V0	V0	V0	V0	U2	U2	U2	U2	U2	U2	U2	U2	V2	V2	V2	V2	V2	V2	V2	V2

Figura 3.13: Representação interna do formato de pixel NV12.

O formato de pixels NV12 é um dos formatos semi-planares de representação de pixels de uma imagem do modelo de cor YUV. Como pode ser observado na figura anterior, à semelhança dos formatos planares a componente de luma é representada de forma independente das restantes componentes de crominância. Contudo, a organização dos valores das componentes de crominâncias dos pixels de uma imagem seguem as características dos formatos entrelaçados.

O formato NV12 é destinado à representação de pixels de uma imagem cujo tipo de subamostragem de crominâncias é o tipo 4:2:0, com uma profundidade de cor de cada componente de 8 bits.

3.4.4 Outros Formatos

Os formatos de representação de pixels de uma imagem segundo o modelo de cor YUV não estão restritos às características dos formatos mencionados. Os formatos de pixels entrelaçados, planares e semi-planares detalham a maioria dos formatos utilizados segundo o modelo de cor YUV. Contudo, outros formatos especificam diferentes tipos de representação de pixels de uma imagem.

O formato V210 especifica as características de representação de pixels de imagens em que cada valor de intensidade de cor de uma componente é representada com uma profundidade de 10 bits e segundo o tipo de subamostragem de crominâncias 4:2:2.

Devido à profundidade de cor de 10 bits deste formato e à arquitetura binária das máquinas, existe um desalinhamento entre a representação das componentes de cor de um pixel e o tipo de dados utilizado para a sua representação. Como se pode observar na figura 3.14, a representação de três componentes de cor não utilizam totalmente o bloco de memória de 32 bits reservado ao efeito. A não utilização eficaz do espaço de memória reservado para o efeito é uma das peculiaridades deste tipo de formato de representação de pixels.

Plano A																															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
U0	U0	U0	U0	U0	U0	U0	U0	U0	U0	Y0	Y0	Y0	Y0	Y0	Y0	Y0	Y0	Y0	Y0	V0	V0	V0	V0	V0	V0	V0	V0	V0	V0	-	-
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Y1	Y1	Y1	Y1	Y1	Y1	Y1	Y1	Y1	Y1	U2	U2	U2	U2	U2	U2	U2	U2	U2	U2	Y2	Y2	Y2	Y2	Y2	Y2	Y2	Y2	Y2	Y2	-	-
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
V2	V2	V2	V2	V2	V2	V2	V2	V2	V2	Y3	Y3	Y3	Y3	Y3	Y3	Y3	Y3	Y3	Y3	U4	U4	U4	U4	U4	U4	U4	U4	U4	U4	-	-
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
Y4	Y4	Y4	Y4	Y4	Y4	Y4	Y4	Y4	Y4	V4	V4	V4	V4	V4	V4	V4	V4	V4	V4	Y5	Y5	Y5	Y5	Y5	Y5	Y5	Y5	Y5	Y5	-	-

Figura 3.14: Representação interna do formato de pixel V210.

Na figura anterior é possível observar a organização dos valores das componentes de cor segundo o formato de pixels V210 de uma imagem constituída por seis pixels. É de notar a semelhança do padrão de representação das componentes com o formato entrelaçado V210, no sentido em que existe uma repetição do padrão: componente crominância U, seguida da componente luma do primeiro pixel do par, a componente de crominância V e, por último, o valor de luma do segundo pixel do par.

3.5 FFmpeg

A ferramenta FFmpeg é um projeto *open source* que fornece um conjunto de bibliotecas e programas para processamento de vídeo, áudio, *streams* e outros tipos de conteúdos multimédia. O FFmpeg é maioritariamente utilizado para transcodificação, codificação e descodificação de vídeo, edição de conteúdos multimédia, aplicação de operações de pós-produção de vídeo e para a introdução de conformidade nos conteúdos quanto aos standards utilizados pelos mesmos, ou também designado de *standard compliance*.

Várias entidades líderes de mercado da área de processamento gráfico como a Intel, NVidia e AMD fornecem circuitos integrados específicos para a realização de operações de codificação e descodificação de vídeo. Porém, estes circuitos têm interfaces específicas e não apresentam características comuns entre si das particularidades das suas implementações e arquiteturas dos circuitos. O FFmpeg suporta como abstração as várias interfaces das entidades referidas e distribui de forma normalizada a capacidade da sua utilização.

A ferramenta FFmpeg permite a orquestração de todo um sistema de processamento e pós-produção de conteúdos multimédia a partir da própria implementação de protocolos de comunicação predominantemente utilizados por software de pós-produção de conteúdos multimédia. Em conjunto com o extenso suporte do FFmpeg aos principais *muxers*, *codecs*, modelos de cor e respetivos formatos de pixels, o FFmpeg é uma das bibliotecas mais utilizadas na manipulação de conteúdos multimédia pois encontra-se na base de software como o VLC Media Player, Blender, Plex, Youtube, Google Chrome, Mozilla Firefox, entre outros.

Os *muxers* são modelos que determinam a forma de representação de todos os dados e metadados de diferentes *codecs* de vídeo e áudio numa só especificação. Os *codecs* definem as propriedades de representação dos dados de um conteúdo multimédia, pormenorizam a codificação e descodificação de sinais que podem ser realizados em hardware ou em software. Os conteúdos multimédia são definidos pelo tipo de *codec* que utilizam, pois o último indica a organização e representação do seu sinal [34].

3.6 Conclusão

Como já referido no capítulo ~~introdutório~~ 1, o objetivo prático deste trabalho é a otimização do processo de reamostragem e redimensionamento de vídeo sem compressão. Devido à característica digital dos vídeos, o modelo de cor YUV é o mais apropriado para a sua representação e processamento. Por essa razão, apenas o último modelo foi analisado e utilizado neste trabalho.

Os filtros de reconstrução *Nearest Neighbor*, linear e por *spline* são os filtros que expõem os conceitos gerais e indicam o comportamento esperado dos resultados do processo de reamostragem e redimensionamento de vídeo. Contudo, existem outros filtros de reconstrução como os filtros *Lanczos*, *Catmull-Rom* e *Sinc2* que não foram abordados na secção 3.3.1. Os últimos são semelhantes aos filtros apresentados, diferindo apenas em pequenos detalhes de formulação. Por essa razão, não foram considerados relevantes para o desenvolvimento deste trabalho.

De igual forma, os formatos de pixeis apresentados na secção 3.4 não cobrem todos os formatos de pixeis do modelo de cor YUV. Contudo, apenas foram mencionados os formatos referidos pois mostram de forma geral as características de representação de cada tipo de formato de pixel mais comum na área de pós-produção de conteúdos multimédia.

Como comparação dos resultados obtidos neste trabalho, o FFmpeg foi considerado como referência. Software como o Adobe Premiere, Sony Vegas, MainConcept e Elecard são ferramentas bastante utilizadas para a aplicação de processos de pós-produção. Contudo, a sua utilização está dependente de licenciamento o que torna inviável a sua utilização como referência neste projeto.

As ferramentas e bibliotecas *open source* de processamento de imagem que poderiam ter sido utilizadas neste projeto como o ImageMagick ou OpenCV, não integram nativamente o modelo de cor YUV e por essa razão foram desconsideradas neste trabalho, pois a sua utilização implicaria um nível de implementação adicional já presente com a ferramenta FFmpeg. No caso da biblioteca OpenCV, parte dos métodos de processamento de certos *codecs* de vídeo são baseados na ferramenta FFmpeg.

O tema desta dissertação foi proposta por uma empresa da área de multimédia que utiliza a ferramenta FFmpeg como o principal recurso de automatização do seu produto de pós-produção de conteúdos multimédia. Como o resultado deste trabalho irá substituir, idealmente, a implementação atual do processo de reamostragem e redimensionamento de vídeo do produto da empresa, a comparação mais correta deve ser realizada entre a ferramenta FFmpeg e a solução implementada neste trabalho.