

**BRUNO VINÍCIUS NISHIWAKI PASCHOAL
JOÃO VITOR OSÓRIO MISKO
LEANDRO MOLES DE MATOS
ROBERT ELIEL WERUTSKY**

**V3D - SISTEMA DE CONVERSÃO DE VÍDEOS 2D
PARA 3D**

Trabalho apresentado à Escola Politécnica da
Universidade de São Paulo para a Graduação
em Engenharia da Computação.

São Paulo
2010

**BRUNO VINÍCIUS NISHIWAKI PASCHOAL
JOÃO VITOR OSÓRIO MISKO
LEANDRO MOLES DE MATOS
ROBERT ELIEL WERUTSKY**

V3D - SISTEMA DE CONVERSÃO DE VÍDEOS 2D PARA 3D

Trabalho apresentado à Escola Politécnica da
Universidade de São Paulo para a Graduação
em Engenharia da Computação.

Área de concentração:
Engenharia da Computação

Orientador:
Prof^a. Dr^a. Regina Melo Silveira

São Paulo
2010

Bruno Paschoal: Dedico este trabalho aos meus pais, que sempre me deram apoio incondicional e são fundamentais em cada uma das minhas realizações. Também dedico aos meus amigos de infância e de Poli, que tornam a vida mais fácil, completa e divertida.

João Misko: Dedico esse trabalho aos meus pais, amigos e colegas de turma, que, do início ao fim, me apoiaram nessa árdua e desafiadora jornada.

Leandro Matos: Dedico este trabalho à minha família e aos amigos, que ajudaram nos momentos difíceis de mais uma etapa da vida que se conclui.

Robert Werutsky: Dedico a conclusão desse projeto aos meus amigos e colegas, minha família e namorada, que sempre estiveram presentes nessa difícil, mas recompensadora fase.

AGRADECIMENTOS

Agradecemos a todos que tornaram possível a realização deste Projeto de Formatura, em especial à Prof^a. Regina Melo Silveira, nossa orientadora, que participou ativamente conosco desde o início, e ao Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica da USP, que possibilitou a realização deste trabalho. Também merecem um agradecimento especial nossos colegas da turma de 2010 de Engenharia de Computação, pessoas com quem muito aprendemos e tivemos prazer de conviver nesses anos na Escola Politécnica. A todos os professores que fizeram parte da nossa graduação na Escola Politécnica da USP, de quem levamos, com muito orgulho, um pequeno pedaço, em forma de ensinamentos importantes, não somente para a engenharia mas também para a vida, também fica nossa gratidão.

RESUMO

O trabalho descreve o desenvolvimento de um software cujo objetivo é realizar a conversão de vídeo digital de 2D para 3D estereoscópico. Através do documento são apresentados os seguintes conceitos teóricos: padrões de cor digital, processo de codificação e decodificação de vídeo, principais características do padrão H.264, técnicas de extração de profundidade de vídeo através do movimento (motion-to-disparity), e outros conteúdos aplicados durante a realização do projeto. A implementação da ferramenta e suas características são detalhadas. Um estudo comparativo entre o conversor desenvolvido - V3D e ferramentas comerciais equivalentes é descrito, bem como a análise dos resultados obtidos.

Palavras-chave: estereoscopia, 3D, anáglifo, H.264, motion-to-disparity

ABSTRACT

This project describes a software development aiming the conversion of 2D digital video to stereoscopic 3D. Through the document are presented the following theoretical concepts: digital color patterns, video encoding and decoding process, H.264 standard main features, video depth extraction techniques through movement (motion-to-disparity) and additional content applied during the project. The tool implementation and its features are detailed. A comparative study between the developed converter - V3D and equivalent commercial tools is described, as well the analysis of the obtained results.

Keywords: stereoscopy, 3D, anaglyph, H.264, motion-to-disparity

LISTA DE FIGURAS

2.1	<i>Processo de codificação de um vídeo.</i>	24
2.2	<i>Quadro 1.</i>	25
2.3	<i>Quadro 2.</i>	25
2.4	<i>Quadro Residual.</i>	26
2.5	<i>Fluxo óptico do quadro 1 para o quadro 2.</i>	26
2.6	<i>Otimização da Busca por um Macrobloco.</i>	28
2.7	<i>Quadro 1.</i>	29
2.8	<i>Quadro 2.</i>	29
2.9	<i>Quadro Residual sem Compensação.</i>	30
2.10	<i>Quadro Residual Com compensação com Blocos de 16 x 16 Pixels.</i>	30
2.11	<i>Quadro Residual Com compensação com Blocos de 8 x 8 Pixels.</i>	30
2.12	<i>Quadro Residual Com compensação com Blocos de 4 x 4 Pixels.</i>	31
2.13	<i>Tratamento de imagens no padrão H.264 (RICHARDSON, 2003).</i>	32
2.14	<i>Estrutura Hierárquica dos Blocos (RICHARDSON, 2003).</i>	33
2.15	<i>Entradas e Saídas do Codificador H.264 ITU-T.</i>	35
2.16	<i>Entradas e Saídas do Decodificador H.264 ITU-T.</i>	35
2.17	<i>Fluxo simplificado do decodificador ITU-T.</i>	36
2.18	<i>Óculos 3D com Filtros, Ciano e Vermelho.</i>	36
2.19	<i>Exemplo de Imagem Anáglifa.</i>	37
4.1	<i>Diagrama de Casos de Uso.</i>	43
4.2	<i>Tela Principal do sistema.</i>	46
4.3	<i>Tela de configuração da conversão 3D.</i>	47

4.4	<i>Arquitetura do Sistema.</i>	48
4.5	<i>Fluxo principal do decodificador de referência do padrão H.264.</i>	49
4.6	<i>Geometria da captura de vídeo 3D com duas câmeras paralelas.</i>	49
4.7	<i>Geometria da captura de vídeo com uma câmera C de um objeto em deslocamento.</i>	50
4.8	<i>Fluxo modificado do decodificador para gerar o mapa de profundidade inicial.</i>	52
4.9	<i>Exemplo de quadro residual.</i>	53
4.10	<i>Ilustração das fases do algoritmo de definição de borda.</i>	54
4.11	<i>Resultado do preenchimento pela esquerda do algoritmo de definição de bordas.</i>	55
4.12	<i>Ilustração de preenchimento de lacuna no algoritmo de definição de bordas.</i>	55
4.13	<i>Processo de construção do vídeo anáglifo.</i>	57
4.14	<i>Ordem de análise dos pixels para deslocamento da camada vermelha.</i>	60
4.15	<i>Conversão para RBG dos pixels do bloco de origem e destino antes do deslocamento da camada vermelha.</i>	61
4.16	<i>Cálculo dos novos valores YCrCb para o bloco de destino.</i>	61
4.17	<i>Fluxo final do decodificador modificado para geração de vídeo 3D.</i>	62
5.1	<i>Quadro de exemplo do vídeo 1 de teste (Soccer).</i>	64
5.2	<i>Quadro de exemplo do vídeo 2 de teste (City).</i>	64
5.3	<i>Quadro de exemplo do vídeo 3 de teste (Ice).</i>	65
5.4	<i>Quadro de exemplo do vídeo 4 de teste (Bus).</i>	65
1.1	<i>Quadro original do vídeo 1 de teste (Soccer).</i>	74
1.2	<i>Mapa de profundidade inicial extraído do vídeo 1 de teste (Soccer).</i>	74
1.3	<i>Mapa de profundidade do vídeo 1 de teste (Soccer) após correção de movimento de câmera.</i>	74
1.4	<i>Mapa de profundidade do vídeo 1 de teste (Soccer) após segunda fase do algoritmo de definição de bordas.</i>	75
1.5	<i>Mapa de profundidade do vídeo 1 de teste (Soccer) após a conclusão do algoritmo de definição de bordas.</i>	75

I.6	<i>Mapa de profundidade do vídeo 1 de teste (Soccer) após a conclusão do algoritmo de diminuição da variação de profundidade.</i>	75
I.7	<i>Mapa de profundidade do vídeo 1 de teste (Soccer) após a conclusão do algoritmo de suavização do mapa.</i>	76
I.8	<i>Mapa de deslocamento da camada vermelha do vídeo 1 de teste (Soccer).</i>	76
I.9	<i>Quadro final estereoscópico do vídeo 1 de teste (Soccer).</i>	76
II.1	<i>Quadro original do vídeo 2 de teste (City).</i>	78
II.2	<i>Mapa de profundidade inicial extraído do vídeo 2 de teste (City).</i>	78
II.3	<i>Mapa de profundidade do vídeo 2 de teste (City) após correção de movimento de câmera.</i>	78
II.4	<i>Mapa de profundidade do vídeo 2 de teste (City) após segunda fase do algoritmo de definição de bordas.</i>	79
II.5	<i>Mapa de profundidade do vídeo 2 de teste (City) após a conclusão do algoritmo de definição de bordas.</i>	79
II.6	<i>Mapa de profundidade do vídeo 2 de teste (City) após a conclusão do algoritmo de diminuição da variação de profundidade.</i>	79
II.7	<i>Mapa de deslocamento da camada vermelha do vídeo 2 de teste (City).</i>	80
II.8	<i>Quadro final estereoscópico do vídeo 2 de teste (City).</i>	80
III.1	<i>Quadro original do vídeo 3 de teste (Ice).</i>	82
III.2	<i>Mapa de profundidade inicial extraído do vídeo 3 de teste (Ice).</i>	82
III.3	<i>Mapa de profundidade do vídeo 3 de teste (Ice) após segunda fase do algoritmo de definição de bordas.</i>	82
III.4	<i>Mapa de profundidade do vídeo 3 de teste (Ice) após a conclusão do algoritmo de definição de bordas.</i>	83
III.5	<i>Mapa de profundidade do vídeo 3 de teste (Ice) após a conclusão do algoritmo de diminuição da variação de profundidade.</i>	83
III.6	<i>Mapa de profundidade do vídeo 3 de teste (Ice) após a conclusão do algoritmo de suavização do mapa.</i>	83
III.7	<i>Mapa de deslocamento da camada vermelha do vídeo 3 de teste (Ice).</i>	84

III.8	<i>Quadro final estereoscópico do vídeo 3 de teste (Ice).</i>	84
IV.1	<i>Quadro original do vídeo 4 de teste (Bus).</i>	86
IV.2	<i>Mapa de profundidade inicial extraído do vídeo 4 de teste (Bus).</i>	86
IV.3	<i>Mapa de profundidade do vídeo 4 de teste (Bus) após correção de movimento de câmera.</i>	86
IV.4	<i>Mapa de profundidade do vídeo 4 de teste (Bus) após segunda fase do algoritmo de definição de bordas.</i>	87
IV.5	<i>Mapa de profundidade do vídeo 4 de teste (Bus) após a conclusão do algoritmo de definição de bordas.</i>	87
IV.6	<i>Mapa de profundidade do vídeo 4 de teste (Bus) após a conclusão do algoritmo de diminuição da variação de profundidade.</i>	87
IV.7	<i>Mapa de deslocamento da camada vermelha do vídeo 4 de teste (Bus).</i>	88
IV.8	<i>Quadro final estereoscópico do vídeo 4 de teste (Bus).</i>	88

LISTA DE TABELAS

3.1	Riscos	40
3.2	Matriz de Exposição aos Riscos	41
3.3	Cronograma	41
5.1	Vídeo 1: Soccer - Cena de um jogo de futebol com jogadores andando e correndo	63
5.2	Vídeo 2: City - Filmagem aérea de uma cidade com movimento de câmera rotacional em volta de um prédio	64
5.3	Vídeo 3: Ice - Cena de pessoas patinando no gelo	65
5.4	Vídeo 4: Bus - Filmagem de um ônibus em deslocamento	66
5.5	Resultado da pesquisa de comparação para o vídeo de teste 1 (Soccer)	67
5.6	Resultado da pesquisa de comparação para o vídeo de teste 2 (City)	67
5.7	Resultado da pesquisa de comparação para o vídeo de teste 3 (Ice)	68
5.8	Resultado da pesquisa de comparação para o vídeo de teste 4 (Bus)	68
I.1	Configuração utilizada para o vídeo 1 de teste (Soccer)	73
II.1	Configuração utilizada para o vídeo 2 de teste (City)	77
III.1	Configuração utilizada para o vídeo 3 de teste (Ice)	81
IV.1	Configuração utilizada para o vídeo 4 de teste (Bus)	85

LISTA DE ABREVIATURAS E SIGLAS

V3D Software desenvolvido pelo grupo para conversão de vídeos 2D para 3D

CODEC Codificador e decodificador

DPB Buffer de imagens decodificadas (Decoded Picture Buffer)

ITU-T International Telecommunication Union

MB Macrobloco

MV vetor de movimento (Motion Vector)

POC Numeração de sequência de exibição de quadros (Picture Order Count)

SUMÁRIO

1	Introdução	17
1.1	Objetivo	17
1.2	Estado da Arte	17
1.3	Motivação	18
1.4	Estrutura	18
2	Aspectos Conceituais	20
2.1	Vídeo Digital	20
2.1.1	Introdução	20
2.1.2	Quadros e Campos	20
2.1.3	Representação de Cores	20
2.2	Conceitos de Codificação de Vídeos	22
2.2.1	Introdução	22
2.2.2	CODEC de Vídeo	23
2.2.3	Modelo de Predição	24
2.3	Processo de Decodificação do padrão H.264 / MPEG-4 Part 10	31
2.3.1	Principais Conceitos	31
2.4	Sintaxe do Padrão H.264	32
2.4.1	Hierarquia de Blocos	32
2.4.2	Parâmetros de Sequência	33
2.5	CODEC H.264 ITU-T	34
2.5.1	Introdução	34

2.5.2	Codificador H.264 ITU-T	34
2.5.3	Decodificador H.264 ITU-T	35
2.6	Anáglifos	35
3	Planejamento e Metodologia	38
3.1	Metodologia	38
3.2	Planejamento	39
3.3	Riscos e Restrições	39
3.4	Cronograma	40
4	Especificação do Software	42
4.1	Descrição	42
4.2	Requisitos Funcionais	42
4.3	Requisitos Não Funcionais	43
4.4	Casos de Uso	43
4.5	Descrição da Interface do Usuário	46
4.6	Arquitetura	46
4.7	Implementação	48
4.7.1	Extração das Estimativas de Profundidade	48
4.7.2	Correção de Movimento de Câmera	51
4.7.3	Definição das Bordas dos Objetos	53
4.7.4	Diminuição da Variação de Profundidade	55
4.7.5	Suavização do Mapa de Profundidade	56
4.7.6	Conversão de Profundidade para Deslocamento da Camada Vermelha	57
4.7.7	Construção do Vídeo Estereoscópico	59
5	Testes e Avaliação	63
5.1	Cenários de Teste	63

5.2	Parâmetros de Avaliação	63
5.3	Resultados	66
5.4	Avaliação dos Resultados	66
6	Considerações Finais	69
6.1	Conceituais	69
6.2	Tecnológicas	69
6.3	Gerenciamento de Projeto	69
6.4	Conclusão	70
6.5	Contribuições	70
6.6	Trabalhos Futuros	70
	Referências Bibliográficas	71
	Apêndices	73
I	Características e Evolução dos Mapas de Profundidade: Vídeo de Teste 1 (Soccer)	73
II	Características e Evolução dos Mapas de Profundidade: Vídeo de Teste 2 (City)	77
III	Características e Evolução dos Mapas de Profundidade: Vídeo de Teste 3 (Ice)	81
IV	Características e Evolução dos Mapas de Profundidade: Vídeo de Teste 4 (Bus)	85

1 INTRODUÇÃO

1.1 Objetivo

O trabalho propõe avaliar as técnicas atualmente conhecidas para obtenção de profundidade em vídeos, com o intuito de propor melhorias. A avaliação será feita através do estudo e implementação das técnicas, avaliando o resultado final segundo critérios de percepção visual e comparação com soluções comerciais. O objetivo final é o desenvolvimento de um software para conversão de vídeos convencionais, com informações de duas dimensões, para vídeos anáglifos com efeito de profundidade, utilizando as técnicas que apresentarem os melhores resultados.

1.2 Estado da Arte

Atualmente, há poucas soluções comerciais com funcionalidade de conversão de vídeos 2D para 3D estereoscópico. A empresa Tridef (2010) comercializa o software Tridef 3D, que produz o efeito de profundidade em tempo real sem análise dos objetos na cena, gerando uma perspectiva tridimensional fixa. Outra solução comercial é o DepthGauge (2010), o qual realiza a conversão do vídeo e gera um novo arquivo em anáglifo. Diversos fatores, como quantidade de movimento de objetos, velocidade de movimento da câmera e luminosidade, afetam a qualidade do vídeo 3D gerado pelo software. Em alguns casos, há pouca ou nenhuma percepção de profundidade no resultado final. De forma geral, as soluções existentes apresentam resultados de baixa qualidade, com pouco efeito de profundidade e consomem muito tempo para o processo de análise e conversão do vídeo. Na área de pesquisa, há artigos cobrindo diversos aspectos da conversão de vídeos para três dimensões. De acordo com Ideses et al. (2007), Wu et al. (2008), Kim, Min e Sohn (2008), Pourazad, Nasiopoulos e Ward (2009), Feng, Jiang e Ipson (2006), Ideses, Yaroslavsky e Fishbain (2007) são descritas técnicas para obtenção de informações de profundidade, enquanto Fehn e Pastoor (2006) abordam a forma de transmissão e Son, Javidi e Kwack (2006), Smolic et al. (2009), Stern e Javidi (2006) detalham formas de

exibição de vídeo com profundidade.

1.3 Motivação

A análise de uma imagem de duas dimensões para extração da profundidade original da cena é um processo complexo, com diversas abordagens e soluções. Atualmente, nenhuma técnica é capaz de resolver essa questão satisfatoriamente. Portanto, há necessidade de comparação das técnicas existentes, análise de suas complementaridades e refinamento para gerar avanços e padronização nessa área. Do ponto de vista comercial, nota-se uma forte demanda por conteúdo em 3D estereoscópico. Filmes com grande bilheteria, tais como Avatar e Toy Story 3, ajudaram a popularizar a tecnologia 3D nos cinemas do mundo inteiro e aumentaram o faturamento da indústria, já que os ingressos para filmes com essa tecnologia são mais caros. Televisões capazes de reproduzir conteúdo em três dimensões, como a Samsung 3D TV, já são comercializadas desde o meio do ano de 2010. Dessa forma, há um mercado crescente por softwares que produzam 3D estereoscópico a partir de vídeos gravados de forma tradicional, para geração de conteúdo específico para esses novos produtos e serviços.

1.4 Estrutura

Este documento está organizado conforme descrito a seguir:

- Capítulo 1 - Introdução: apresenta os objetivos definidos para o projeto, o atual estado da arte da área de pesquisa relacionada ao trabalho e a motivação para sua realização.
- Capítulo 2 - Aspectos Conceituais: apresenta os principais conceitos utilizados como base para o desenvolvimento do projeto.
- Capítulo 3 - Planejamento e metodologia: descreve as atividades do projeto, bem como o cronograma de tarefas executadas.
- Capítulo 4 - Especificação do software: descreve os aspectos relacionados ao software V3D desenvolvido para o projeto, tais como as tecnologias utilizadas, casos de uso, requisitos funcionais e não funcionais, arquitetura e implementação.
- Capítulo 5 - Testes e Avaliação: apresenta cenários de teste propostos para o software V3D, assim como os resultados obtidos e análises extraídas desses resultados.

- Capítulo 6 - Considerações finais: apresenta lições aprendidas com a realização do projeto, dificuldades encontradas, contribuições, conclusão e aspectos que podem ser explorados em trabalhos futuros.

2 ASPECTOS CONCEITUAIS

2.1 Vídeo Digital

2.1.1 Introdução

Um vídeo digital é uma representação de uma cena do mundo real amostrada temporalmente. Cada amostragem dessa cena produz o que se denomina quadro (ou frame, em inglês), sendo que a exibição contínua dos quadros em um intervalo de tempo suficientemente pequeno, geralmente de um quadro a cada $1/25$ ou $1/30$ segundos, cria a impressão de movimento da cena original.

2.1.2 Quadros e Campos

Existem duas técnicas tradicionais para amostragem digital de vídeo: os métodos progressivo e entrelaçado. No método progressivo cada amostragem equivale a um quadro com todos os dados da cena, ou seja, uma imagem completa. Já no método entrelaçado cada amostragem produz um campo, que contém metade das informações de um quadro. Na prática, um campo contém todas as linhas pares de um quadro ou todas as linhas ímpares. A vantagem do método entrelaçado é que, com o mesmo tamanho de arquivo de vídeo, é possível ter o dobro de campos por segundo se comparado a quadros por segundo. Isso garante uma melhor qualidade de movimento na exibição do vídeo. Por exemplo, um vídeo com 50 campos por segundo apresenta melhor sensação de movimento do que um vídeo de mesmo tamanho com 25 quadros por segundo.

2.1.3 Representação de Cores

Para representar as cores de uma imagem são necessárias, no mínimo três, informações por pixel. O método utilizado para representar essas três informações é chamado sistema de cores, sendo que os mais utilizados são o RGB e o YCrCb.

2.1.3.1 RGB

O sistema de cores RGB (Red, Green and Blue) consiste em representar qualquer cor através de três valores numéricos que indicam a proporção das cores vermelha, verde e azul. O sistema se baseia no princípio de que essas três cores primárias, quando combinadas na proporção correta, podem formar qualquer cor. Dessa forma, uma tela colorida através de RGB compõe a imagem iluminando separadamente os canais vermelho, verde e azul de cada pixel com certa intensidade para cada componente. Assim, de uma distância adequada, os pixels aparentam ter uma cor real.

2.1.3.2 YCrCb

O sistema YCrCb é mais otimizado para o sistema visual humano, em relação ao RGB. O conceito no qual esse sistema se baseia é representar, de forma separada, a luminância e as informações sobre a cor. Utilizando-se do fato de que o olho humano é mais sensível à luminância do que às cores, o componente de luminância é representado com alta resolução, enquanto as informações sobre cor são representadas com resolução inferior.

A luminância Y pode ser calculada como uma média ponderada dos componentes R, G e B:

$$Y = i.R + j.G + k.B \quad (2.1)$$

onde i , j e k são fatores de peso.

Já as informações sobre a cor são representadas como as diferenças (crominância) entre as componentes R, G e B e a luminância Y. Descritas nas equações: 2.2, 2.3 e 2.4.

$$C_r = R - Y \quad (2.2)$$

$$C_b = B - Y \quad (2.3)$$

$$C_g = G - Y \quad (2.4)$$

Como $C_r + C_b + C_g$ é uma constante, é suficiente armazenar e transmitir apenas Y, C_r e C_b . A crominância C_g é sempre calculada a partir das outras duas.

No sistema RGB cada componente de cor tem a mesma importância e, portanto, utiliza a mesma resolução. A vantagem do sistema YCrCb é obter a mesma qualidade visual do RGB com uma quantidade menor de dados, pois representa a luminância Y com alta resolução e Cr e Cb com baixa resolução.

2.1.3.3 Formato de Resolução do Sistema YCrCb

Padrões de resolução do sistema YCrCb são representados na forma (Y:Cr:Cb), o que indica a quantidade de amostras de luminância e crominância presentes em cada pixel. Padrões de resolução comuns são 4:4:4, 4:2:2 e 4:2:0. Por exemplo: 4:4:4 indica que cada um dos componentes Y, Cr e Cb tem a mesma resolução, ou seja, para cada 4 amostras de luminância existem 4 amostras de Cr e 4 de Cb. No padrão 4:2:2, para cada 4 amostras de luminância existem 2 amostras de Cr e 2 de Cb. Já o padrão 4:2:0 indica que para cada 4 amostras de luminância teremos 1 de Cr e 1 de Cb (o padrão se chama 4:2:0 em vez de 4:1:1 por razões históricas).

2.2 Conceitos de Codificação de Vídeos

2.2.1 Introdução

Alguns conceitos apresentados a seguir são referentes ao padrão H.264/MPEG-4 Part10. Um vídeo digital em seu estado original, isto é, sem nenhum tipo de compressão, requer uma alta capacidade de transmissão de dados para que possa ser armazenado e transmitido com eficiência. Portanto, é necessário utilizar um processo de compressão de dados para viabilizar o armazenamento e distribuição de vídeos digitais.

A codificação consiste na conversão dos dados de um vídeo para uma forma reduzida através de um compressor (codificador). Posteriormente, é possível recuperar as informações, de forma que fiquem muito próximas dos dados originais, utilizando um descompressor (decodificador). O par codificador/decodificador é nomeado CODEC.

As grandezas 'grau de compressão' e 'qualidade dos dados decodificados' são inversamente proporcionais. Logo, quanto maior a compressão de vídeo utilizada, menor a qualidade final do vídeo, em geral.

Para alcançar a compressão de vídeo desejada, o CODEC elimina as redundâncias espaciais e temporais presentes no vídeo.

2.2.2 CODEC de Vídeo

Um codificador de vídeo é formado por três unidades funcionais: um modelo de predição, um modelo espacial e um codificador de entropia. O modelo de predição recebe vídeos sem compressão. Sua função é reduzir a redundância de informação que existe devido a semelhanças entre quadros consecutivos (predição temporal) ou amostras de imagens em um mesmo quadro (predição espacial). Para alcançar essa redução, primeiramente é gerada uma predição do quadro atual a partir de outros quadros (processo chamado de interpredição) ou a partir das amostras de imagens vizinhas no próprio quadro (processo chamado de intrapredição). Uma vez que a predição do quadro é construída, ela é subtraída do quadro atual original, gerando um quadro residual. A saída do modelo de predição é o quadro residual e um conjunto de parâmetros, chamados parâmetros de predição, que indicam como o processo de intra ou interpredição foi executado.

O quadro residual é a entrada do modelo espacial. Seu objetivo é reduzir a redundância das amostras no quadro residual. A redução, nesse caso, é alcançada aplicando uma transformada aos valores das amostras, convertendo-os para outro domínio em que os valores são representados pelos coeficientes da transformada. Esses coeficientes, chamados de coeficientes residuais, são então quantizados, resultando em uma representação compacta do quadro residual.

Finalmente, os parâmetros do modelo de predição e os coeficientes do modelo espacial são comprimidos pelo codificador de entropia. A última unidade funcional do codificador de vídeo remove as redundâncias estatísticas nos dados. Vetores e coeficientes de alta ocorrência, por exemplo, são representados por códigos binários curtos. A saída do codificador de entropia é um arquivo ou fluxo (stream) de bits eficientemente comprimido. Esse arquivo contém os parâmetros de predição, os coeficientes do quadro residual e as informações de cabeçalho. O processo é ilustrado na figura 2.1.

O decodificador de vídeo é formado, também, por três unidades funcionais que desfazem o processo de codificação. Primeiramente, o fluxo de bits ou arquivo comprimido é decodificado pelo decodificador de entropia. Em seguida, os coeficientes residuais são decodificados, gerando o quadro residual. Por último, o decodificador utiliza os parâmetros de predição e os quadros anteriormente decodificados (no caso de interpredição) para gerar um quadro de predição que é somado ao quadro residual para reconstruir o quadro original.

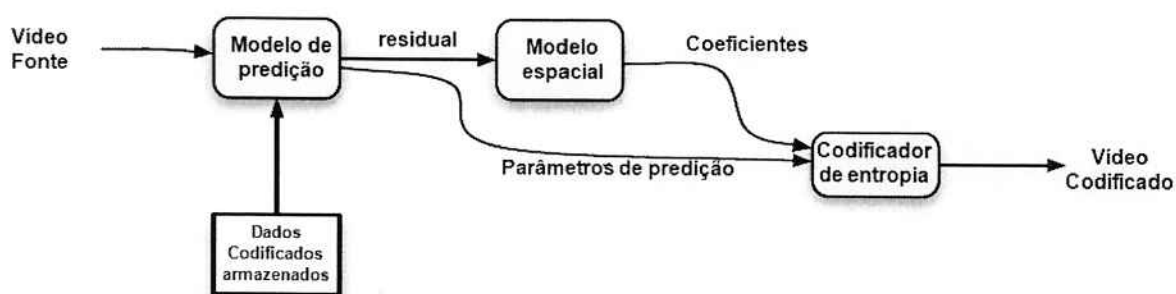


Figura 2.1: Processo de codificação de um vídeo.

2.2.3 Modelo de Predição

Como exposto anteriormente, o objetivo da unidade de modelo de predição é reduzir a redundância nos quadros de um vídeo sem compressão. Isso é feito através da construção de uma predição do quadro atual, usando como referência outros quadros do vídeo ou amostras de imagem do próprio quadro atual. Uma vez que a predição é gerada, ela é subtraída do quadro atual original, gerando um quadro residual. O quadro residual é codificado e enviado para o decodificador, o qual realiza o processo inverso: cria a mesma predição a partir dos parâmetros de predição que são enviados, e adiciona o quadro residual para reconstruir o quadro original. Para que o decodificador possa criar uma predição idêntica à do codificador, é necessário que o codificador utilize apenas dados também disponíveis ao decodificador.

2.2.3.1 Redundância Temporal

Redundância temporal é a semelhança entre quadros vizinhos de um vídeo. Por exemplo, em uma sequência onde há uma paisagem no fundo e personagens no primeiro plano, a paisagem não se move e pouco muda de um quadro para outro. Ou seja, os quadros desse vídeo apresentam alta redundância temporal devido à repetição do fundo da cena ao longo do tempo. Nesses casos, a informação que descreve a paisagem de fundo não precisa ser transmitida para cada quadro do vídeo, mas apenas em um quadro, desde que também se informe em quais quadros a paisagem se repete.

Quando a redundância temporal é utilizada no modelo de predição, o quadro de predição é construído a partir de um ou mais quadros passados ou futuros (que já foram codificados/decodificados) chamados quadros de referência. A forma mais simples de utilização dessa estratégia é definir o último quadro como predição do quadro atual. Nas figuras 2.2 e 2.3 é possível observar dois quadros sucessivos em que o quadro 1 é utilizado como quadro de predição do quadro 2. O quadro residual (gerado subtraindo o quadro de predição do quadro

2) está na figura 2.4. No quadro residual, as áreas cinzas representam regiões onde não há diferença entre as imagens. Já as áreas claras representam uma diferença positiva e as áreas escuras uma diferença negativa. Com essa abordagem, podemos notar que o quadro residual ainda contém muita energia (áreas claras ou escuras), o que indica que ainda há muita informação para ser codificada. A maior parte dessas informações são objetos em movimento na cena, por isso uma melhor codificação pode ser alcançada se o quadro de predição for construído com a compensação dos movimentos entre os dois quadros.



Figura 2.2: *Quadro 1.*



Figura 2.3: *Quadro 2.*

Para uma cena de um objeto em movimento, é possível estimar a trajetória de cada pixel do objeto entre os quadros da sequência de vídeo. As informações de trajetória de cada pixel em um quadro formam o que se denomina fluxo óptico. A figura 2.5 mostra o fluxo óptico da transição do quadro da figura 2.2 para o quadro da figura 2.3.



Figura 2.4: *Quadro Residual.*

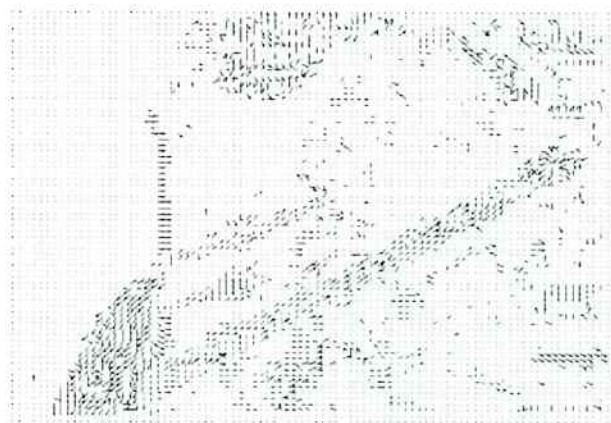


Figura 2.5: *Fluxo óptico do quadro 1 para o quadro 2.*

Utilizando-se o fluxo óptico de uma transição entre quadros, é possível construir um quadro de predição mais preciso. Basta mover os pixels do quadro de referência de acordo com os seus vetores, presentes no fluxo óptico. Porém, esse método de compensação de movimento não é eficiente. É necessário processamento intenso para gerar o fluxo óptico, já que são feitos cálculos para cada pixel. Além disso, um fluxo óptico deve ser transmitido junto com cada quadro, o que aumenta consideravelmente a quantidade de informações transmitidas.

Por isso, um método mais prático é a compensação de movimento por blocos retangulares de pixels, em vez da compensação de cada pixel individualmente. Nessa abordagem, para cada bloco de $M \times N$ pixels no quadro atual, é realizado o seguinte algoritmo:

- É realizada uma busca no quadro de referência por uma região $M \times N$ similar ao bloco do quadro atual. Essa busca procura pelo melhor resultado que satisfaz um determinado critério e é chamado de estimativa de movimento. Um critério comum é a menor energia residual resultante da subtração do bloco candidato pelo bloco procurado.

- O bloco selecionado é utilizado como predição do bloco atual e é subtraído do bloco $M \times N$ atual, gerando um bloco residual.
- O bloco residual é codificado e transmitido, juntamente com a diferença da posição do bloco atual e a posição do bloco candidato (vetor de movimento).

O decodificador, por sua vez, utiliza o vetor de movimento que foi transmitido para reconstruir a área de predição a partir do quadro anterior, já decodificado. Em seguida, o quadro residual é decodificado e adicionado à área de predição para gerar o bloco original.

2.2.3.2 Redundância Espacial

Redundância espacial é o nome dado a semelhanças entre diferentes partes de um mesmo quadro de vídeo. Por exemplo, em um quadro onde há um fundo monocromático, como uma parede branca, a maior parte dos pixels é da mesma cor, ou de cores muito próximas. Devido a essas semelhanças, é possível comprimir um quadro fazendo a predição da cor de um bloco a partir de um bloco vizinho e parâmetros auxiliares.

O método de intrapredição consiste na predição do bloco que está sendo codificado/decodificado, utilizando como base blocos do mesmo quadro codificados/decodificados anteriormente. Por exemplo, na codificação de um bloco, o codificador realiza uma busca no quadro atual por blocos já codificados que possuam semelhanças com aquele bloco. O bloco encontrado é utilizado como bloco de predição para o bloco atual. A predição é subtraída do bloco original, resultando em um bloco residual (processo semelhante ao utilizado na interpredição). O bloco residual codificado e as informações de geração do bloco de predição são transmitidos e o decodificador realiza o processo inverso para obter o bloco original.

2.2.3.3 Macrobloco

A unidade visual básica da técnica de predição por compensação de movimento é chamada de macrobloco na maioria dos padrões de codificação. O macrobloco representa um bloco de 16×16 pixels, e o processamento dos quadros é feito por meio dele.

2.2.3.4 Estimativa de Movimento

O processo de estimativa de movimento de um macrobloco consiste em encontrar uma região de 16×16 pixels no quadro de referência que seja similar ao macrobloco atual, de acordo com um determinado critério. O quadro de referência é um quadro da sequência de

vídeo já codificado que pode estar à frente ou atrás do quadro atual na exibição do vídeo. Para otimizar a busca, utiliza-se uma área restrita para busca ao invés do quadro inteiro. Como ilustrado na figura 2.6.

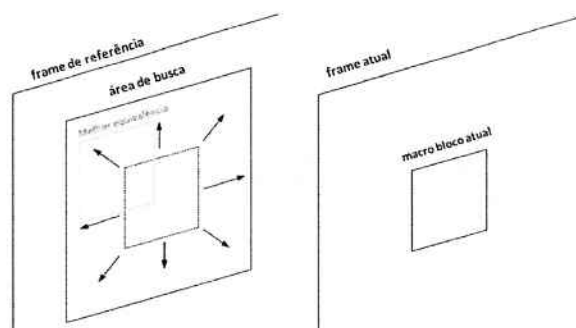


Figura 2.6: *Otimização da Busca por um Macrobloco.*

2.2.3.5 Compensação por Movimento

Os componentes luma e chroma da região encontrada são subtraídos do macrobloco atual, produzindo um macrobloco residual. O macrobloco residual é codificado e transmitido juntamente com o vetor de movimento, o qual descreve a posição do macrobloco de referência em relação ao macrobloco atual.

Quando há uma mudança significativa entre o quadro de referência e o quadro atual, por exemplo, em uma mudança de cena, é mais eficiente codificar os macroblocos sem compensação por movimento, pois nenhum, ou poucos, macroblocos da nova cena serão encontrados no quadro de referência. Nesse caso, o codificador pode definir a utilização de intrapredição, ao invés de interpredição. Além disso, raramente um objeto em movimento em uma cena segue corretamente as fronteiras de um bloco de 16×16 pixels, o que justifica a utilização de blocos de tamanho variável.

Os objetos também podem se deslocar por frações de pixel. Por exemplo, um deslocamento de 1,5 pixel em vez de 2 pixels. Uma melhor previsão pode ser gerada nesses casos interpolando o quadro de referência em frações de pixel antes de a busca ser realizada.

A figura 2.7 e a figura 2.8 representam dois quadros sucessivos de uma sequência de vídeo. O quadro 1 é subtraído do quadro 2, sem compensação por movimento, para produzir o quadro residual da figura 2.9. A energia residual é reduzida se utilizarmos compensação por movimento com macroblocos (16×16 pixels), conforme observado no quadro residual da figura 2.10. No caso de blocos de 8×8 pixels, a energia residual é ainda menor, como visto na figura 2.11. E por último, na figura 2.12, foram utilizados blocos de 4×4 pixels.

Esses exemplos mostram que quanto menor o bloco, melhor é o resultado da compensação por movimento. Porém, blocos pequenos aumentam a complexidade, pois exigem mais buscas e maior quantidade de vetores de movimento que devem ser transmitidos, o que reduz os benefícios de usar blocos pequenos. A melhor estratégia é adaptar o tamanho do bloco de acordo com a complexidade do quadro, ou seja, utilizar macroblocos para as áreas de poucos detalhes e blocos menores para as áreas de muitos detalhes e movimentos complexos.



Figura 2.7: *Quadro 1.*



Figura 2.8: *Quadro 2.*

2.2.3.6 Perfis e Níveis

Os perfis definem um conjunto de algoritmos e processos de codificação/decodificação que podem ser utilizados. Conseqüentemente, cada perfil define um nível de performance de compressão, devido à complexidade computacional dos algoritmos permitidos.



Figura 2.9: *Quadro Residual sem Compensação.*



Figura 2.10: *Quadro Residual Com compensação com Blocos de 16 x 16 Pixels.*



Figura 2.11: *Quadro Residual Com compensação com Blocos de 8 x 8 Pixels.*



Figura 2.12: *Quadro Residual Com compensação com Blocos de 4 x 4 Pixels.*

Um vídeo, aderente a um perfil em particular, deve ser codificado usando apenas o subconjunto de especificações definidas por aquele perfil. Um decodificador deve ser capaz de decodificar todas as características permitidas no subconjunto. Portanto, o perfil tem a função de limitar as capacidades requeridas para o decodificador. Os níveis definem a quantidade de dados que um decodificador deve processar, por isso têm função semelhante aos perfis. Cada nível especifica limites para tamanho do quadro, taxa de processamento (quantidade de quadros decodificados por segundo) e quantidade de memória alocada para decodificação.

2.3 Processo de Decodificação do padrão H.264 / MPEG-4 Part 10

2.3.1 Principais Conceitos

Para o processo de compressão de vídeo, o codec utiliza parâmetros que representam informações da imagem original com uma quantidade reduzida de bits. Esses parâmetros são elementos da sintaxe do padrão de vídeo, e são convertidos em cadeias de bits para transmissão e armazenamento.

A função básica de um decodificador de vídeo é interpretar essa cadeia de bits gerada pelo codificador. Através da identificação de elementos da sintaxe do padrão, é possível recompor as informações necessárias para reprodução do vídeo. Para cada macrobloco, o decodificador forma uma predição idêntica à original criada pelo codificador, através das técnicas de intrapredição e interpredição.

A Figura 2.13 apresenta uma visão geral do tratamento de imagens, com ênfase no

processo de decodificação. Nessa etapa existe um buffer para armazenamento de imagens já decodificadas (Decoded Picture Buffer - DPB). As imagens no DPB podem ser exibidas diretamente ou serem utilizadas em processos de interpredição.

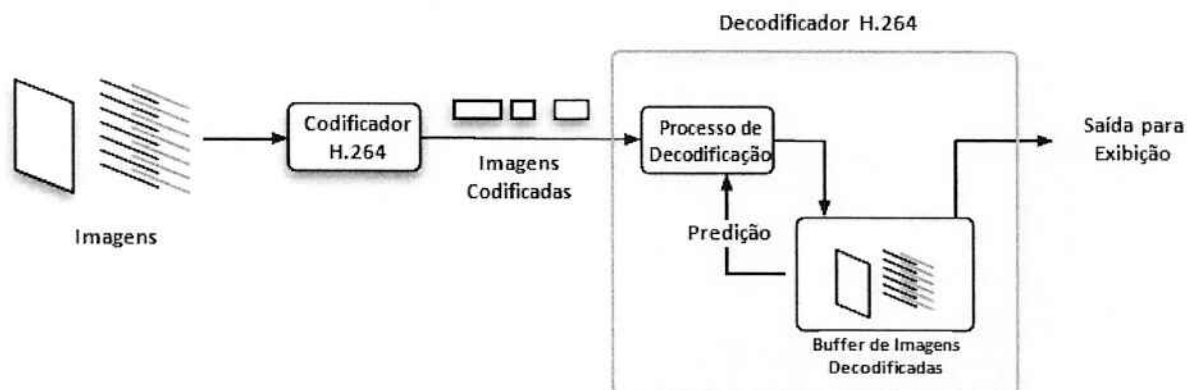


Figura 2.13: Tratamento de imagens no padrão H.264 (RICHARDSON, 2003).

2.3.1.1 Sequências de Imagens

Existem três numerações distintas de imagens:

- Sequência de decodificação: ordem em que as imagens são decodificadas da cadeia de bits.
- Sequência de exibição (POC - Picture Order Count): ordem em que as imagens devem ser enviadas para exibição.
- Sequência de referência: utilizada para o processo de interpredição de outras imagens. São implementadas duas listas (list0 e list1) para organização de duas sequências distintas de referência.

2.4 Sintaxe do Padrão H.264

2.4.1 Hierarquia de Blocos

O padrão H.264 define uma sintaxe para representação do vídeo codificado e de informações adicionais utilizadas no processo de decodificação. A organização do padrão segue uma estrutura hierárquica, conforme visto na Figura 2.14.

Um macrobloco (MB) pode ser de três tipos, definidos pelo parâmetro mb-type:

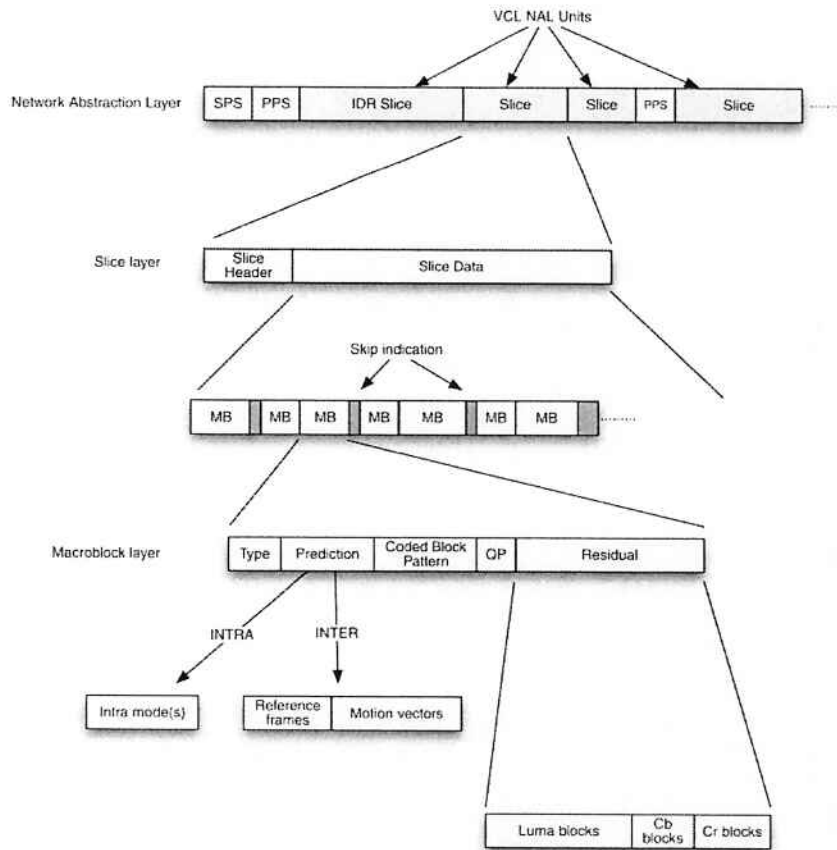


Figura 2.14: Estrutura Hierárquica dos Blocos (RICHARDSON, 2003).

- MB I: utiliza intrapredição. Não possui referências a outros slices.
- MB P: utiliza interpredição a partir de um único frame de referência.
- MB B: utiliza interpredição a partir de um ou dois frames de referência.

Macroblocos do tipo P e B contêm, em seu bloco *prediction*, apontadores para os *frames* utilizados como referência no processo de previsão e vetores de movimento.

2.4.2 Parâmetros de Sequência

A sequência de decodificação é definida pelo parâmetro *frame-num*, que se encontra no cabeçalho (header) da camada *slice*.

A sequência de exibição é informada pelos parâmetros *TopFieldOrderCount* e *BottomFieldOrderCount*, conhecidos coletivamente como *Picture Order Count* (POC), presentes no cabeçalho da camada *slice*. Os parâmetros *TopFieldOrderCount* e *BottomFieldOrderCount* podem ser utilizados de três maneiras diferentes:

- Tipo 0: cada cabeçalho da camada *slice* contém os bits menos significativos do POC.

- Tipo 1: a sequência de exibição é automática, subentendida através do ciclo definido no conjunto de parâmetros. O ciclo determina o intervalo entre os quadros utilizados para referência e pode ser alterado pelo parâmetro Delta, presente no cabeçalho da camada slice.
- Tipo 2: utiliza a mesma sequência de decodificação, definida pelo parâmetro frame-num. Já a sequência de referência é definida por uma lista (macrobloco tipo P) ou por duas listas de referência (macrobloco tipo B). Uma imagem utilizada como referência a longo prazo é ordenada pelo parâmetro LongTermPicNum e é armazenada por último nas listas.

2.5 CODEC H.264 ITU-T

2.5.1 Introdução

A ITU-T (International Telecommunication Sector) disponibiliza um software de referência em conjunto com a recomendação do padrão H.264. Esse software é um CODEC, cujo objetivo é auxiliar no estudo e demonstrar as capacidades do padrão.

O software da ITU-T foi utilizado como base para a ferramenta de conversão de vídeos 2D para 3D desenvolvida neste projeto. Portanto, nesse capítulo serão apresentados alguns detalhes do software de referência.

2.5.2 Codificador H.264 ITU-T

O CODEC da ITU-T é dividido logicamente em dois projetos, um codificador e um decodificador. O codificador será utilizado apenas como ferramenta de conversão de vídeo para o formato do padrão H.264 (extensão .264).

A entrada do codificador é composta de dois arquivos principais: um arquivo de configuração contendo parâmetros de codificação e um arquivo de vídeo com extensão .yuv (formato sem compressão com representação de cor no sistema YCrCb) e amostragem 4:2:0. Através do arquivo de configuração é possível ter total controle sobre o processo de codificação, o que é importante para criarmos cenários de teste personalizados para a ferramenta de conversão 2D para 3D desenvolvida. A figura 2.15 ilustra o processo.

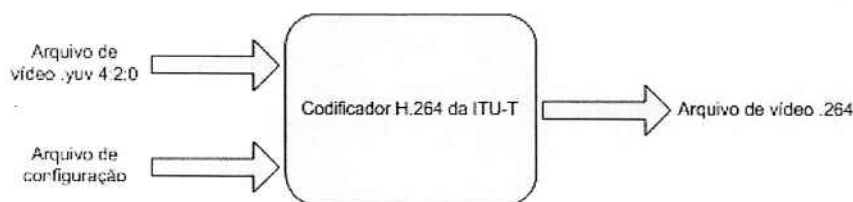


Figura 2.15: Entradas e Saídas do Codificador H.264 ITU-T.

2.5.3 Decodificador H.264 ITU-T

Já o decodificador presente no CODEC de referência da ITU-T aceita como entrada um arquivo no formato .264 e gera como saída o arquivo de vídeo decodificado no formato .yuv com amostragem 4:2:0. A figura 2.16 ilustra o processo.

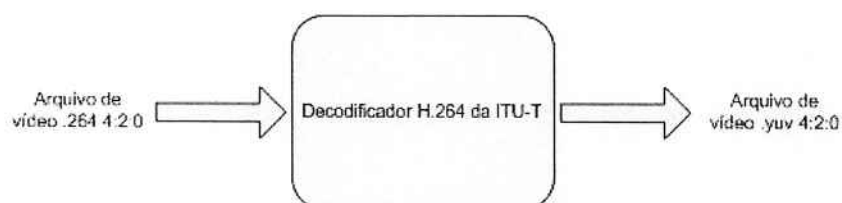


Figura 2.16: Entradas e Saídas do Decodificador H.264 ITU-T.

O fluxo lógico simplificado do decodificador de referência é apresentado na figura 2.17. Ao iniciar a decodificação, cada quadro tem todos os seus blocos decodificados. Uma vez que um quadro foi decodificado ele é armazenado no DPB (Decoded Picture Buffer) para ser utilizado como referência. Se após a inserção do quadro, o DPB estiver com um determinado limite de quadros armazenados, então o quadro com menor índice de ordem de exibição é escrito no arquivo de saída.

2.6 Anáglifos

Anáglifo é uma imagem ou vídeo formatado de maneira especial, para fornecer um efeito tridimensional estereoscópico quando visto com o auxílio de óculos de duas cores (cada lente com uma cor diferente. Tradicionalmente, vermelho para o olho esquerdo e ciano para o olho direito) ilustrado na figura 2.18. Nesse tipo de imagem existem duas camadas coloridas: uma camada filtrada com a cor vermelha e outra com as cores azul e verde (que formam a cor ciano), deslocadas horizontalmente para produzir o efeito de profundidade. O efeito 3D ocorre porque o olho associado ao filtro vermelho enxerga as partes vermelhas da imagem com a cor branca e as partes azuis e verdes com a cor preta. Já o olho associado ao filtro ciano percebe o efeito contrário. As áreas verdadeiramente brancas e pretas da imagem são

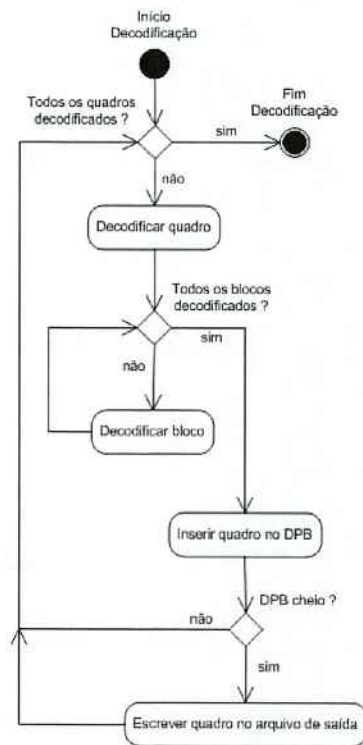


Figura 2.17: Fluxo simplificado do decodificador ITU-T .

igualmente percebidas por ambos os olhos. Assim, o cérebro une as imagens recebidas por cada olho e interpreta as diferenças como a profundidade da cena.

O conceito utilizado pelos anáglifos é a estereopsia, que é o processo no cérebro responsável por gerar a sensação de profundidade a partir de duas projeções de uma cena, ligeiramente diferentes, na retina de cada olho. Essa diferença das duas projeções é decorrente da diferença na posição dos olhos na cabeça e é chamada de disparidade horizontal ou disparidade binocular. Assim, os óculos com filtro de cores, usados em conjunto com um anáglifo, garantem que cada olho enxergará uma imagem diferente, sendo que essas imagens foram deslocadas para simular a disparidade binocular humana. O efeito é exemplificado na figura 2.19.

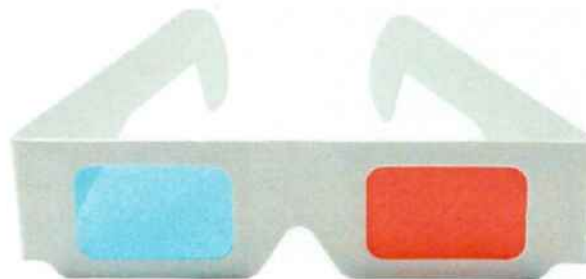


Figura 2.18: Óculos 3D com Filtros, Ciano e Vermelho.

Existem duas maneiras de criar um anáglifo. A primeira é a construção normal, em que a camada vermelha é deslocada para a direita, sendo que objetos mais distantes sofrem



Figura 2.19: *Exemplo de Imagem Anáglifa.*

menor deslocamento do que objetos próximos da câmera. Essa construção dá a sensação de que o objeto está saindo do plano da tela. A segunda construção é a invertida, onde a camada vermelha é deslocada para a esquerda, sendo que objetos mais distantes sofrem maior deslocamento do que objetos próximos à câmera. Por sua vez, essa construção gera a sensação de que os objetos estão atrás do plano da tela.

3 PLANEJAMENTO E METODOLOGIA

3.1 Metodologia

O desenvolvimento do projeto seguiu as atividades descritas a seguir:

- Estudo teórico: estudo de conceitos de vídeo digital, padrões de compressão e transmissão, formatos de exibição de vídeos 3D, estereoscopia
- Estudo de técnicas de conversão 3D: leitura de artigos acadêmicos sobre extração de profundidade em vídeos 2D, mapas de profundidade, cálculo de movimento de câmera, detecção de bordas de objetos.
- Estudo de CODECs: estudo da estrutura, configurações e características de CODECs comerciais, estudo de viabilidade de CODECs para adaptação às necessidades do projeto
- Documentação das pesquisas: documentação dos aspectos relevantes encontrados durante atividades de estudo
- Produção da monografia: definição da estrutura da monografia, desenvolvimento dos capítulos, formatação de acordo com padrão exigido
- Projeto do software: definição do escopo, casos de uso, interface, arquitetura, e descrição da implementação
- Implementação do software: codificação das funcionalidades previstas no escopo do projeto
- Realização de testes: avaliação subjetiva e comparação de vídeos anáglifos gerados pelo conversor do projeto e por ferramentas comerciais
- Análise de resultados: elaboração de conclusões a partir dos resultados obtidos

3.2 Planejamento

Tendo em vista a grande carga teórica associada ao tema do projeto, os primeiros meses foram dedicados ao estudo dos principais conceitos relacionados a vídeo digital e estereopsia. Paralelamente a essa atividade, foram realizadas pesquisas e leituras de papers com o intuito de compreender o processo de criação de mapas de profundidade e vídeos anáglifos, já com vistas à fase de implementação.

A partir dos conceitos e técnicas aprendidos, realizou-se o estudo de CODECs com o objetivo de selecionar aquele que possuísse as características necessárias à realização do projeto. As informações obtidas durante os estudos foram consolidadas na atividade de documentação das pesquisas.

A produção da monografia teve início na metade do ano e se estendeu até o final do projeto. A implementação do software conversor ocorreu durante o último trimestre do ano, culminando com a realização de testes e análise dos resultados obtidos.

No decorrer de todo o projeto foram realizadas reuniões com a professora orientadora para alinhamento do progresso realizado pelos integrantes do grupo e definição de novas etapas.

3.3 Riscos e Restrições

Os riscos aos quais o projeto está sujeito foram classificados nas categorias:

- Técnico: riscos ligados à tecnologia, ferramentas, implementação
- Gerência de projeto: riscos ligados ao planejamento e controle das atividades
- Externo: riscos externos ao escopo do projeto que afetam sua realização

A tabela 3.1 apresenta os riscos detectados para o projeto, sua probabilidade de ocorrência (Baixa, Média, Alta), o impacto previsto caso o risco se concretize, a exposição do projeto ao risco (produto entre probabilidade e impacto) e o plano de ação sugerido para sua resolução.

Com base nos riscos descritos na tabela 3.1, produz-se a matriz de exposição aos riscos descrita na tabela 3.2.

Tabela 3.1: Riscos

Nº	Descrição do Risco	Probabilidade	Impacto	Exposição ao Risco	Ação Requerida
1	Técnico				
1.1	Dificuldade em implementar a técnica de conversão	B	A	BA	Aceitar - Buscar outra técnica de conversão mais fácil de implementar
1.2	Desempenho da técnica de conversão abaixo do esperado	B	M	BM	Mitigar - Otimizar o código
2	Gerência de Projeto				
2.1	Estimativa incorreta de cronograma	B	A	BA	Mitigar - Redefinir metas conforme prazos
2.2	Planejamento inadequado de atividades	M	A	MA	Mitigar - Aumentar a carga horária para não haver atraso
3	Externo				
3.1	Proprietário da técnica de conversão impede sua utilização	B	A	BA	Aceitar - Buscar e utilizar outra técnica de conversão

3.4 Cronograma

As atividades foram realizadas conforme a tabela 3.3. Algumas tarefas foram realizadas simultaneamente, conforme possibilidade e disponibilidade dos componentes do grupo.

Tabela 3.2: Matriz de Exposição aos Riscos

A			
M			2.2
B		1.2	1.1,2.1,3.1
Probabilidade / Impacto	B	M	A

Tabela 3.3: Cronograma

Atividade	Duração	Início	Fim
Estudo Teórico	3 Meses	Segunda 18/01/2010	Sexta 09/04/2010
Estudo de Técnicas	3 Meses	Segunda 15/02/2010	Sexta 07/05/2010
Estudo de CODECs	2 Meses	Segunda 05/04/2010	Sexta 28/05/2010
Documentação de Pesquisas	1 Mês	Segunda 31/05/2010	Sexta 25/06/2010
Produção da Monografia	6 Meses	Segunda 21/06/2010	Sexta 03/12/2010
Projeto e Implementação do Software	3 Meses	Segunda 30/08/2010	Sexta 19/11/2010
Realização de Testes	2 Semanas	Segunda 22/11/2010	Sexta 03/12/2010
Análise de Resultados	1 Semana	Segunda 29/11/2010	Sexta 03/12/2010

4 ESPECIFICAÇÃO DO SOFTWARE

4.1 Descrição

O sistema desenvolvido é um software de conversão de vídeos 2D para 3D anáglifo, ou seja, para ser visualizado com o auxílio de óculos com filtro de cores vermelho e ciano. A conversão deve ser automática, isto é, sem a intervenção do usuário para determinar a profundidade dos objetos em uma cena qualquer do vídeo original. As únicas entradas do sistema são o vídeo 2D que será convertido e um conjunto de parâmetros de configuração da técnica de conversão, que podem ser alterados pelo usuário. Como alternativa ao vídeo 2D, uma webcam poderá ser utilizada como entrada. As saídas do sistema são o vídeo 3D anáglifo e um conjunto de imagens que representam o mapa de profundidade dos objetos em cada quadro do vídeo, com a finalidade de avaliar o desempenho da técnica e ajudar o usuário na escolha dos parâmetros de configuração.

4.2 Requisitos Funcionais

Definir arquivo de vídeo 2D como entrada

O sistema deve aceitar arquivos de vídeo 2D selecionados pelo usuário como entrada no formato YUV (formato sem codificação) e 264 (formato do padrão H.264).

Definir diretório de saída

O sistema deve aceitar um diretório de saída definido pelo usuário onde será gravado o arquivo de vídeo 3D gerado.

Gravar vídeo 2D a partir de webcam

O sistema deve ser capaz de gravar, a partir de uma webcam, um vídeo 2D no formato aceito para conversão 3D.

Alterar configuração de conversão 3D

O sistema deve permitir que o usuário escolha quais passos da técnica de conversão 3D devem ser utilizados e também os valores dos diversos parâmetros da técnica.

Converter vídeo para 3D

O sistema deve converter um vídeo 2D definido como entrada ou um vídeo gravado através do sistema por uma webcam, para vídeo 3D anáglifo.

4.3 Requisitos Não Funcionais

Taxa de conversão

O tempo de conversão de um vídeo em HD (resolução de 1280 x 720) não poderá ultrapassar 5 segundos por quadro em um computador com processador de 2GHz, pois o tempo de conversão será considerado inviável.

4.4 Casos de Uso

Os casos de uso do software são apresentados primeiramente na figura 4.1 e depois descritos em detalhes no restante dessa seção.

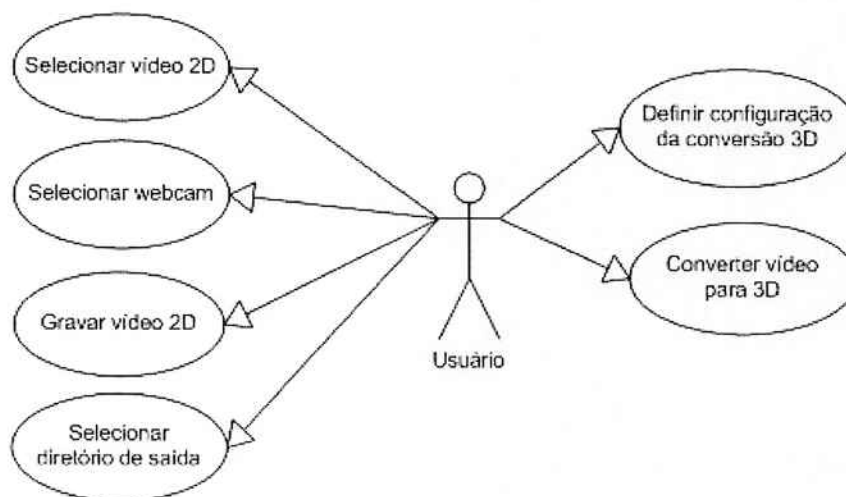


Figura 4.1: Diagrama de Casos de Uso.

Caso de Uso: Selecionar Vídeo 2D.

Pré-condição: Nenhuma.

Pós-condição: Vídeo 2D definido como entrada do sistema.

Ator: Usuário.

Fluxo Principal de Eventos:

- 1 - O usuário informa ao sistema que irá definir um arquivo de vídeo como entrada.
- 2 - O sistema apresenta os arquivos de vídeo disponíveis que pertençam ao conjunto de formatos aceitos.
- 3 - O usuário seleciona um vídeo e confirma.
- 4 - O sistema grava a escolha do usuário.

Fluxo Secundário de Eventos:

- 5 - Em 3 o usuário pode cancelar e o sistema retorna para a tela principal.

Caso de Uso: Selecionar Webcam.

Pré-condição: Nenhuma.

Pós-condição: Webcam definida como entrada do sistema.

Ator: Usuário.

Fluxo Principal de Eventos:

- 1 - O usuário informa ao sistema que irá definir uma webcam como entrada.
- 2 - O sistema apresenta uma lista com as webcams disponíveis.
- 3 - O usuário seleciona uma webcam.
- 4 - O sistema grava a escolha do usuário.

Fluxo Secundário de Eventos:

- 5 - Em 3 o usuário pode cancelar e o sistema retorna para a tela principal.

Caso de Uso: Gravar Vídeo 2D.

Pré-condição: Webcam definida como entrada do sistema.

Pós-condição: Vídeo 2D gravado da webcam disponível para conversão 3D.

Ator: Usuário.

Fluxo Principal de Eventos:

- 1 - O usuário informa que a gravação deve ser iniciada.
- 2 - O sistema persiste continuamente em arquivo de vídeo a gravação da webcam.
- 3 - O usuário informa que a gravação deve ser finalizada.
- 4 - O sistema interrompe a gravação e informa ao usuário que o vídeo gravado está pronto para ser convertido.

Fluxo Secundário de Eventos: Inexistente.

Caso de Uso: Selecionar Diretório de Saída.

Pré-condição: Nenhuma.

Pós-condição: Diretório de saída definido no sistema.

Ator: Usuário.

Fluxo Principal de Eventos:

- 1 - O usuário informa ao sistema que irá definir o diretório de saída (onde o arquivo de vídeo 3D deve ser salvo).
- 2 - O sistema apresenta os diretórios disponíveis.
- 3 - O usuário seleciona o diretório.
- 4 - O sistema grava a escolha do usuário.

Fluxo Secundário de Eventos:

- 5 - Em 3 o usuário pode cancelar e o sistema retorna para a tela principal.

Caso de Uso: Definir Configuração da Conversão 3D

Pré-condição: Nenhuma.

Pós-condição: Nova configuração da conversão 3D salva no sistema.

Ator: Usuário.

Fluxo Principal de Eventos:

- 1 - O usuário informa ao sistema que irá definir uma nova configuração da conversão 3D.
- 2 - O sistema apresenta os parâmetros disponíveis de configuração.
- 3 - O usuário define os valores da configuração.
- 4 - O sistema grava a nova configuração definida pelo usuário.

Fluxo Secundário de Eventos:

- 5 - Em 3 o usuário pode cancelar e o sistema retorna para a tela principal.

Caso de Uso: Converter Vídeo para 3D

Pré-condição: Arquivo de vídeo 2D indicado pelo usuário ou gravação da webcam realizada e diretório de saída definido.

Pós-condição: Vídeo 3D gerado e salvo no diretório de saída.

Ator: Usuário.

Fluxo Principal de Eventos:

- 1 - O usuário informa ao sistema que a conversão deve ser iniciada.
- 2 - O sistema converte o vídeo.
- 3 - O sistema grava o vídeo 3D no diretório de saída selecionado.
- 4 - O sistema informa ao usuário que o vídeo foi convertido.

Fluxo Secundário de Eventos:

- 5 - Em 2 o usuário pode cancelar a conversão, o sistema grava então o que já foi convertido

para vídeo 3D no diretório de saída.

4.5 Descrição da Interface do Usuário

A interface do sistema consiste de duas telas primárias, a tela principal e a tela de configuração da técnica de conversão 3D e três telas auxiliares, uma tela para seleção de arquivos ou diretórios, uma tela de progresso e uma tela de gravação da webcam.

A tela principal (figura 4.2) apresenta o modo de entrada selecionado (arquivo já existente ou webcam), as informações do arquivo de entrada ou da webcam, as informações do diretório de saída e disponibiliza botões para as principais funcionalidades do sistema.

Figura 4.2: Tela Principal do sistema.

Já a interface de configuração da conversão 3D (figura 4.3) permite selecionar quais passos serão utilizados e alterar os diversos parâmetros da técnica.

4.6 Arquitetura

A técnica de conversão de vídeo 3D adotada no projeto envolve o aproveitamento de recursos e informações disponibilizados na codificação e decodificação de vídeo no padrão de compressão H.264/MPEG-4 Part 10. A arquitetura do sistema naturalmente reflete essa característica, pois é necessário ter como parte do sistema um codificador e decodificador do padrão H.264. Como módulo codificador, o projeto x264 será usado por ser um dos poucos

Configuração

Corrigir movimento de câmera

Definir bordas de objeto

Sensibilidade:

Diminuir variação de profundidade

a:

b:

Sensibilidade:

Utilizar deslocamento variável

Profundidade:

Sensibilidade:

Suavizar mapa de profundidade

σ :

w:

Utilizar deslocamento por camadas

Número de camadas:

Diferença de profundidade:

Sensibilidade:

Deslocamento base:

Utilizar anáglifo inverso

Persistir em imagem os passos da conversão

Salvar

Figura 4.3: Tela de configuração da conversão 3D.

projetos de código aberto disponíveis atualmente e devido ao seu desempenho superior, em relação ao CODEC de referência do padrão H.264. Já como módulo decodificador, será utilizado o decodificador de referência do padrão, projeto de código aberto que acompanha os documentos de especificação do H.264 com finalidade educacional. Embora o decodificador do padrão apresente um desempenho muito abaixo do esperado, é um dos poucos decodificadores H.264 de código aberto e se mostrou mais adequado ao contexto do projeto devido à facilidade de compreensão do código.

A figura 4.4 representa a arquitetura do sistema. Os arquivos de vídeo no formato YUV (sem compressão) e amostragem 4:2:0 são enviados pela camada de controle para o módulo codificador x264, o qual irá codificar o vídeo com uma configuração especial que maximiza a qualidade do resultado da técnica de conversão 3D. Em seguida, o arquivo gerado com formato .h264 é enviado para o módulo decodificador, que se comunica com o módulo 3D para gerar

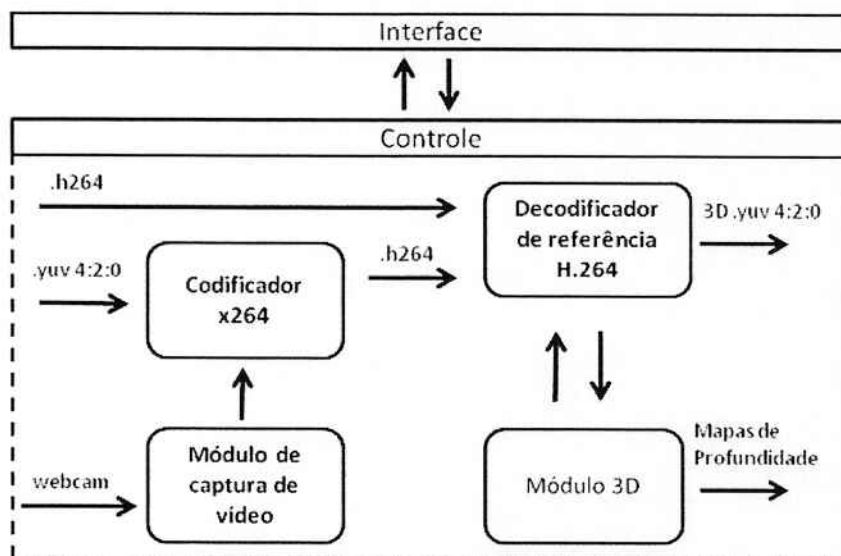


Figura 4.4: *Arquitetura do Sistema.*

o vídeo anáglifo com base nas informações disponíveis na decodificação. Como alternativa de entrada do sistema, existe o módulo de captura de vídeo. Toda comunicação entre os módulos é gerenciada pela camada de controle. Por fim, a camada de interface oferece ao usuário controle sobre o sistema.

4.7 Implementação

A técnica de conversão 3D adotada é baseada em diversos artigos recentes, Ideses et al. (2007), Pourazad, Nasiopoulos e Ward (2009), Zhang e Tam (2005) porém diversas modificações foram inseridas para o projeto. O objetivo inicial é extrair do vídeo 2D aproximações dos valores de profundidade dos objetos da cena, criando um mapa de profundidade para cada quadro do vídeo. Em seguida, o mapa gerado é corrigido e refinado, eliminando os valores inconsistentes das aproximações e melhorando a qualidade final do vídeo 3D. Por fim, com base no vídeo 2D original e os mapas de profundidade, é gerado o vídeo 3D anáglifo. Para alcançar o esse objetivo serão feitas interferências no fluxo do decodificador H.264 4.5 já comentado no capítulo 2.

4.7.1 Extração das Estimativas de Profundidade

A diferença na posição das imagens de um objeto do ponto de vista do olho esquerdo e direito é inversamente proporcional à distância do objeto. Portanto, um objeto distante sofrerá menor deslocamento relativo nas imagens do que um objeto próximo. Esse deslocamento, no

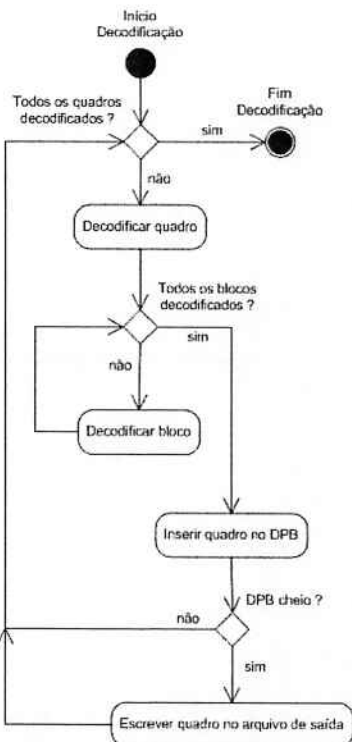


Figura 4.5: Fluxo principal do decodificador de referência do padrão H.264.

contexto de vídeos digitais estereoscópicos, é chamado de disparidade. A figura 4.6 ilustra esse fenômeno quando um objeto P é filmado por duas câmeras sincronizadas e paralelas C' e C . As coordenadas das imagens p' e p capturadas por C' e C são x' e x , respectivamente. Z é distância do objeto P (profundidade), t_c é a distância entre as câmeras e f a distância focal.

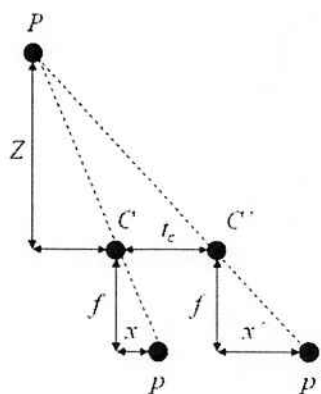


Figura 4.6: Geometria da captura de vídeo 3D com duas câmeras paralelas.

Assim, a disparidade d é definida como:

$$d = x - x' = \frac{ft_c}{Z} \quad (4.1)$$

No caso de um vídeo 2D gravado por uma única câmera, pode-se estimar a profundidade

Z de um objeto P utilizando os deslocamentos do seu movimento registrados nos quadros do vídeo. Essa situação é ilustrada na figura 4.7. Se um objeto se desloca de P_1 para P_2 , as imagens formadas em dois quadros consecutivos na câmera C são p_1 e p_2 respectivamente. Objetos próximos da câmera (profundidade Z pequena) geralmente possuem deslocamento D maior do que objetos distantes, então, como a distância d entre a posição das imagens p_1 e p_2 é proporcional a D e o tempo entre dois frames consecutivos é pequeno, podemos utilizar o valor d como uma aproximação da disparidade da configuração de duas câmeras apresentada anteriormente. Uma vez obtido o valor da disparidade, é possível calcular a profundidade do objeto a partir da fórmula 4.1. O deslocamento do objeto (disparidade) entre dois frames é obtido a partir das informações dos vetores de movimento presentes na codificação e decodificação do vídeo no padrão H.264.

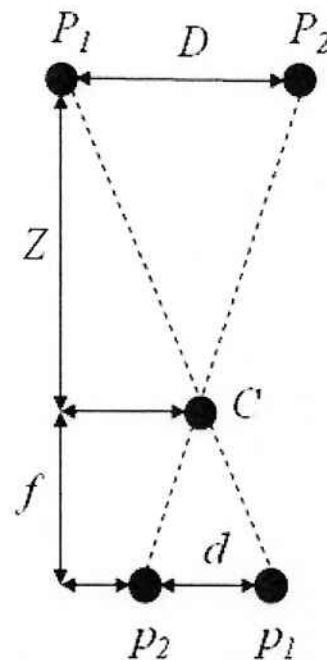


Figura 4.7: Geometria da captura de vídeo com uma câmera C de um objeto em deslocamento.

Durante a decodificação de um bloco, o vetor de movimento e a posição do bloco são extraídos e enviados para o módulo 3D, que decompõe o vetor de movimento em deslocamento horizontal e vertical e armazena essas componentes em um mapa, de acordo com a posição do bloco. Esse mapa é chamado de mapa de disparidade e será transformado no mapa de profundidade, de acordo com a fórmula 4.2, quando todos os blocos do quadro forem decodificados.

$$P(i, j) = \sqrt{MVx(i, j)^2 + MVy(i, j)^2} \quad (4.2)$$

Onde P é a profundidade do pixel na posição (i,j) do quadro de vídeo, $MV_x(i,j)$ é a componente horizontal do vetor de movimento e $MV_y(i,j)$ é a componente vertical. Ou seja, as componentes do vetor de movimento de um bloco são utilizadas para calcular uma única profundidade que é atribuída para cada pixel dentro do bloco. Porém, o quadro de referência do bloco atual que está sendo decodificado nem sempre é o quadro anterior em ordem de exibição, pois a ordem de codificação e decodificação é geralmente diferente da ordem de exibição dos quadros. De fato, um quadro pode ter múltiplos quadros de referência. Logo, o vetor de movimento não pode ser usado nesses casos, pois levaria a uma profundidade maior do que a real. Para corrigir o problema é necessário aplicar as fórmulas 4.3 e 4.4.

$$MV'_x(i,j) = \frac{MV_x(i,j)}{poc_{atual} - poc_{ref}} \quad (4.3)$$

$$MV'_y(i,j) = \frac{MV_y(i,j)}{poc_{atual} - poc_{ref}} \quad (4.4)$$

Onde MV' é o valor corrigido do vetor de movimento, poc_{atual} é o número de exibição do quadro atual e poc_{ref} é o número de exibição do quadro de referência do bloco atual. Outro problema são os quadros do tipo B, que são quadros que possuem blocos com dois vetores de movimento, possivelmente cada um com um quadro de referência diferente. Nesses casos, a estratégia adotada é utilizar o vetor de movimento que possui o quadro de referência mais próximo em ordem de exibição e também aplicar as fórmulas 4.3 e 4.4. O último problema encontrado é que blocos do tipo I não possuem vetor de movimento, logo não é possível extrair estimativas de profundidade. Para evitar mapas de profundidade incompletos, a abordagem adotada é de forçar o codificador H.264 a não utilizar esse tipo de bloco em quadros do tipo P ou B. No caso dos quadros do tipo I, é utilizado o mapa de profundidade do quadro anterior em ordem de exibição.

4.7.2 Correção de Movimento de Câmera

Um problema de qualquer técnica de extração de profundidade a partir dos deslocamentos dos objetos em cena é relativo a movimentações de câmera, as quais levam a estimativas incorretas de profundidade. Por exemplo: se um objeto e a câmera estão em movimentos opostos, o deslocamento do objeto será maior e conseqüentemente sua profundidade calculada será menor do que é de fato. Para corrigir esse tipo de problema é necessário determinar o valor do deslocamento da câmera e compensar todo o movimento do quadro.

A estratégia para identificar o movimento da câmera é criar um histograma dos valores dos vetores de movimento dos blocos do tipo *Skip Mode*, os quais são blocos especiais de 16x16 pixels do padrão H.264 que geralmente contém áreas do fundo da cena do vídeo. Dessa forma, o valor do vetor de movimento mais freqüente no histograma é igual ao movimento da câmera. Portanto, durante a extração dos valores de vetor de movimento, o histograma do movimento de câmera é alimentado e, ao fim da decodificação de um quadro, o valor mais freqüente do histograma é subtraído de cada valor do mapa de disparidade gerado anteriormente.

A figura 4.8 ilustra parte do fluxo do decodificador, modificado para gerar o mapa de profundidade inicial com correção de movimento de câmera.

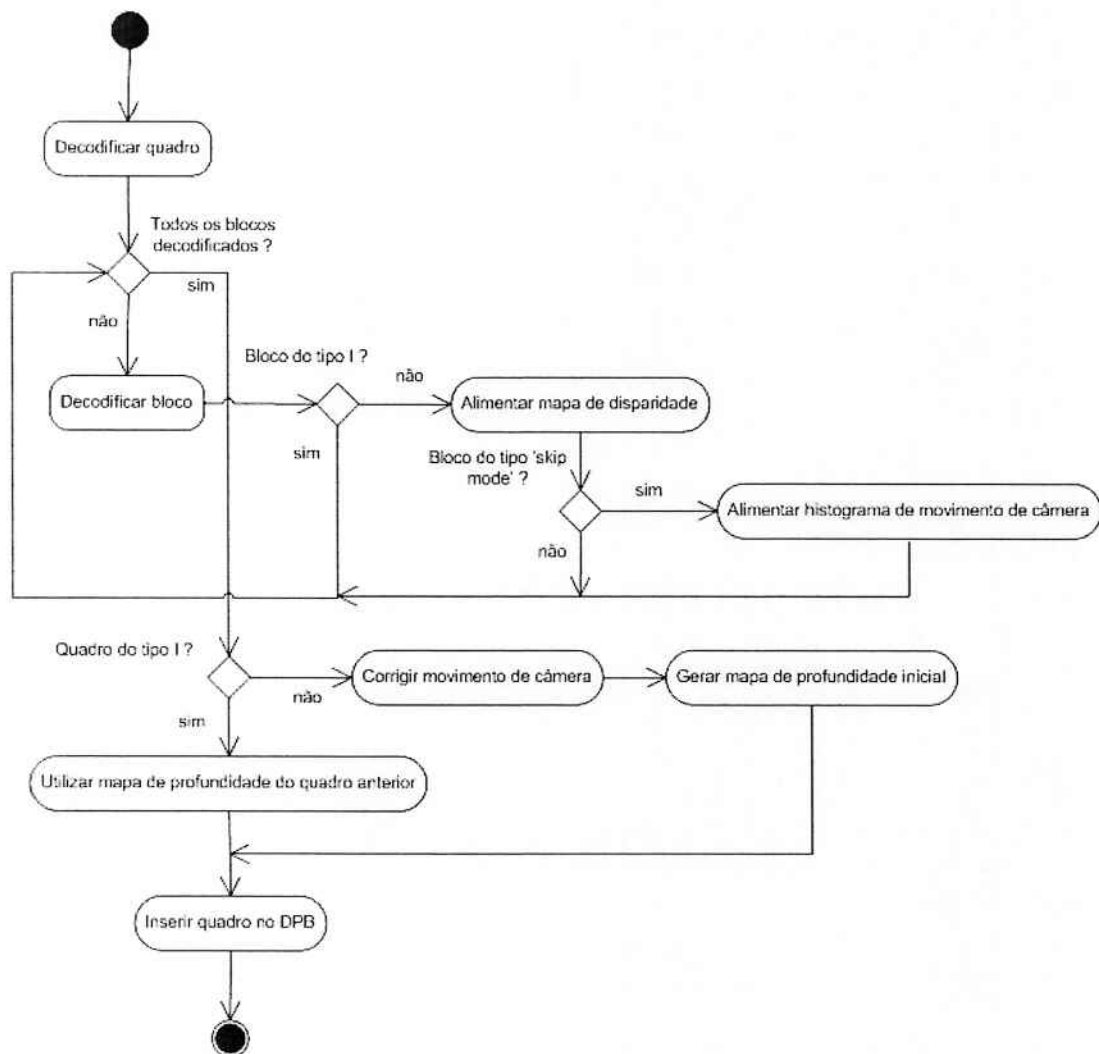


Figura 4.8: Fluxo modificado do decodificador para gerar o mapa de profundidade inicial.

4.7.3 Definição das Bordas dos Objetos

Como os vetores de movimento são utilizados para blocos de tamanho mínimo de 4x4 até no máximo 16x16 pixels e a profundidade calculada é atribuída para cada pixel do bloco, teremos como resultado mapas de profundidade com bordas de objetos mal definidas. Para melhorar a qualidade final do contorno dos objetos no vídeo estereoscópico é necessário um refinamento.

A estratégia adotada é composta de três fases. O objetivo da primeira fase do algoritmo é identificar as bordas dos objetos em movimento. Para isso, a camada de luminância do quadro atual é subtraída da camada de luminância do quadro anterior, gerando o que é chamado de quadro residual. No quadro residual, as bordas dos objetos em movimento aparecem com maior brilho, como pode ser observado na figura 4.9.

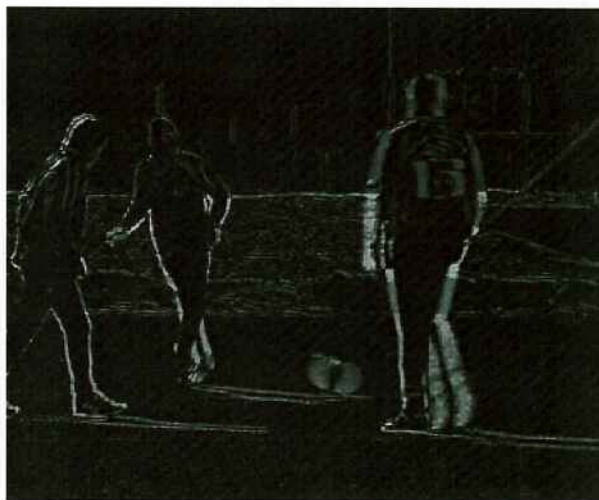


Figura 4.9: Exemplo de quadro residual.

A segunda fase do algoritmo consiste em marcar cada pixel dentro de um bloco como pixel objeto ou de plano de fundo. Para isso é calculada a média da luminância do bloco correspondente no quadro residual. Então, verifica-se para cada pixel do bloco atual se a luminância do pixel correspondente no quadro residual é menor do que a média. Se esse for o caso, ele é classificado como pixel de plano de fundo. Caso contrário, como pixel objeto. Por fim, o valor de profundidade dos pixels marcados como plano de fundo é zerado, enquanto para pixels objetos o valor é mantido.

A figura 4.10 ilustra as três fases do algoritmo de definição de bordas. O item A da figura representa uma parte do quadro residual gerado na primeira fase, sendo que a linha branca é a borda de um objeto em movimento, semelhante ao que pode ser observado na figura 4.9. O item B da figura 4.10 mostra as divisões dos blocos nessa região do quadro do

vídeo e identifica como o objeto está distribuído nos blocos, sendo que o bloco 1 não contém nenhuma parte do objeto, o bloco 2 contém parte da fronteira do objeto e o bloco 3 contém parte do interior do objeto. O item C é a profundidade inicial em escala de cinza, calculada pela técnica de extração de profundidade apresentada anteriormente, sendo que as regiões mais claras são objetos mais próximos da câmera. Como pode ser observado, os blocos 2 e 3, que contém partes do objeto tem profundidade mais clara e o fundo da cena representado pelo bloco 1 está mais distante, ou seja, com profundidade mais escura. O item D ilustra o resultado da segunda fase do algoritmo de definição de bordas, onde os pixels marcados como plano de fundo tem sua profundidade zerada (cor preta) e os pixels marcados como borda de objeto têm profundidade inicial mantida. O item E representa a terceira fase do algoritmo, explicada em seguida, e tem como objetivo preencher as áreas de profundidade zero criadas pela segunda fase, resultando na profundidade final do item F.

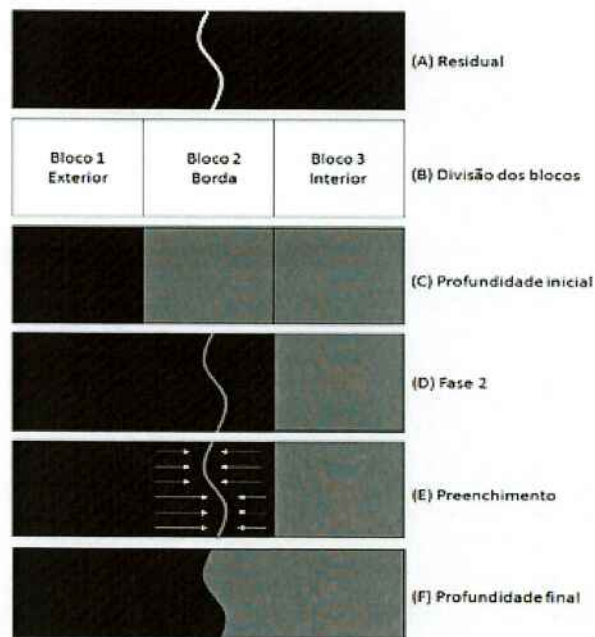


Figura 4.10: Ilustração das fases do algoritmo de definição de borda.

Após alguns testes concluiu-se que se todos os blocos do quadro de vídeo passarem pela classificação da fase 2 haverá perda de informação de profundidade. Mas de fato, apenas os blocos que contenham o contorno principal dos objetos devem ser submetidos à fase 2. Por exemplo, na figura 4.9, busca-se definir apenas o contorno principal dos três jogadores e não o contorno dos detalhes nas roupas ou do gramado. Logo, para filtrar quais blocos devem ser submetidos à fase 2, é calculada a variância da luminância no bloco correspondente no quadro residual. O bloco é utilizado somente se a variância calculada for maior do que um limite estabelecido pelo usuário, o qual é denominado sensibilidade do algoritmo de definição de bordas.

A terceira fase do algoritmo tem como objetivo preencher as áreas de profundidade zero criadas anteriormente. Para isso, se um bloco contém pelo menos um pixel objeto, então cada linha de pixel dentro do bloco é iterada e preenchida da esquerda para a direita com valores de profundidade até o primeiro pixel objeto da linha. O valor de profundidade do preenchimento é a profundidade do pixel à esquerda do início da linha, ou seja, do bloco à esquerda do bloco analisado. Se não há pixel objeto em uma linha, então ela não é preenchida. O resultado do preenchimento inicial pela esquerda é ilustrado na figura 4.11.



Figura 4.11: Resultado do preenchimento pela esquerda do algoritmo de definição de bordas.

Em seguida, o mesmo procedimento é realizado pela direita e por fim no interior do bloco para corrigir situações como da figura 4.12, onde existem duas ou mais linhas de borda de objeto em um mesmo bloco no quadro residual. Nesse último caso, é verificado em cada linha de pixel se existem lacunas de profundidade zero entre grupos de pixels objeto. Se existirem, são preenchidas com o valor de profundidade dos pixels objeto do bloco.

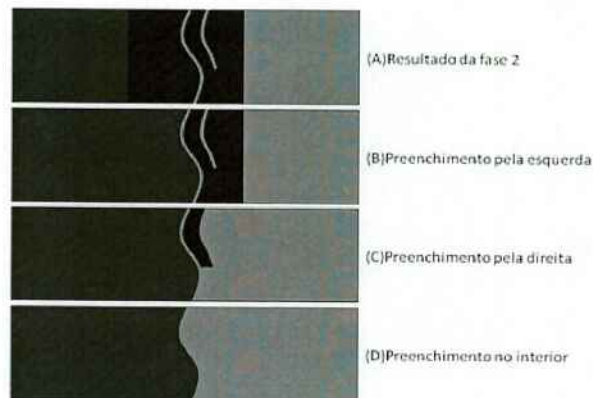


Figura 4.12: Ilustração de preenchimento de lacuna no algoritmo de definição de bordas.

Para finalizar, o algoritmo de preenchimento é aplicado também verticalmente, ou seja, por cima, por baixo e procurando lacunas verticalmente.

4.7.4 Diminuição da Variação de Profundidade

Um problema da técnica de conversão 3D adotada é que objetos complexos muitas vezes possuem diversos valores distintos de profundidade para cada ponto do objeto, ainda que

todos esses pontos estejam no mesmo nível de profundidade na cena real do vídeo. Por exemplo, no caso de uma pessoa caminhando, as pernas se movimentam mais do que o corpo e conseqüentemente terão maiores valores estimados de profundidade. Além disso, existem outras fontes de variação de profundidade dentro dos objetos ligadas à própria natureza da codificação no padrão H.264, pois este visa obter a melhor compressão possível dos dados e não um mapa de vetores de movimento que leve a um bom mapa de profundidade. Portanto, é necessário um mecanismo para diminuir o efeito da variação de profundidade nos objetos, que causa desconforto visual e qualidade reduzida do vídeo 3D anáglifo.

A solução adotada consiste em recalcular cada valor de profundidade de acordo com a fórmula 4.5.

$$P' = b.P^a \quad (4.5)$$

Onde, P' é a nova profundidade do pixel, P é a profundidade original, 'a' é o fator de redução da variação de profundidade e 'b' uma constante de normalização.

Um efeito colateral dessa estratégia é a distorção da profundidade real dos objetos e a redução do número de camadas no mapa de profundidade. Portanto, é necessário utilizar valores adequados das constantes 'a' e 'b' para reduzir a variação de profundidade nos objetos, sem comprometer a percepção da diferença de profundidade dos objetos na cena do vídeo.

Em testes preliminares verificou-se que se os pixels do fundo da cena não forem submetidos à fórmula 4.5, então os objetos em movimento terão um maior destaque na cena, produzindo uma melhor percepção 3D. Para isso, se o valor de profundidade do pixel for menor do que um limite estabelecido pelo usuário, classificado como sensibilidade de detecção do fundo da cena, então não é aplicada a fórmula 4.5 ao pixel.

4.7.5 Suavização do Mapa de Profundidade

Para melhorar a qualidade visual final do vídeo 3D, é aplicado um algoritmo de suavização do mapa de profundidade. Para isso, o filtro gaussiano abaixo é utilizado.

$$P'(i, j) = \frac{\sum_{v=-\frac{w}{2}}^{\frac{w}{2}} \sum_{u=-\frac{w}{2}}^{\frac{w}{2}} \{P(i-u, j-v)g(u, \sigma)g(v, \sigma)\}}{\sum_{v=-\frac{w}{2}}^{\frac{w}{2}} \sum_{u=-\frac{w}{2}}^{\frac{w}{2}} \{g(v, \sigma)g(u, \sigma)\}} \quad (4.6)$$

$$g(x, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{\sigma^2}} \quad (4.7)$$

Onde, P' é a nova profundidade do pixel (i,j) , w é o tamanho janela do filtro definido pelo usuário e σ o desvio padrão.

4.7.6 Conversão de Profundidade para Deslocamento da Camada Vermelha

Para construir o vídeo anáglifo é necessário gerar as duas imagens do par estereoscópico. Para isso, a camada vermelha é retirada do quadro original e deslocada com base no mapa de profundidade, sendo utilizada como uma das imagens do par. A segunda imagem é o quadro original com as camadas verde e azul. Se o anáglifo é do tipo normal, a imagem da camada vermelha é deslocada para a direita. Porém, se o anáglifo é do tipo invertido, a imagem da camada vermelha é deslocada para a esquerda. A figura 4.13 ilustra o processo descrito de construção do vídeo 3D anáglifo.

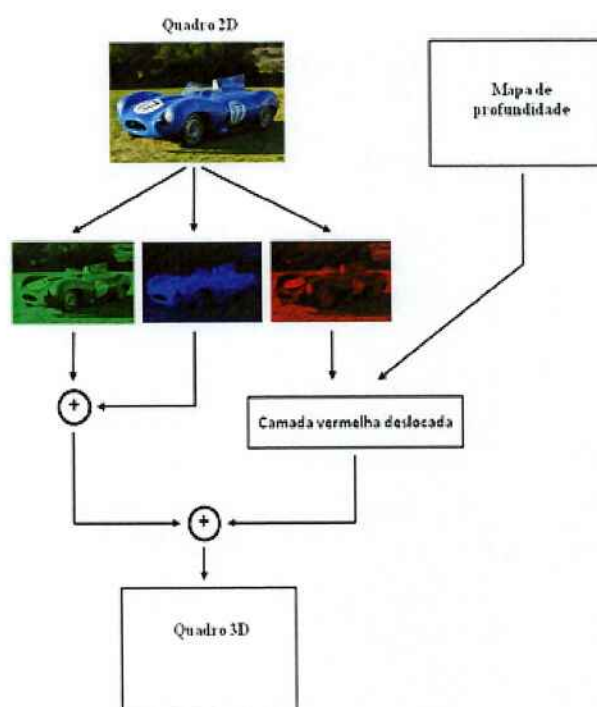


Figura 4.13: Processo de construção do vídeo anáglifo..

Para calcular o valor do deslocamento dos pixels da camada vermelha a partir do mapa de profundidade, adotou-se dois métodos distintos. O usuário da ferramenta de conversão pode optar por um dos dois, dependendo do tipo de controle desejado sobre as camadas de profundidade.

No primeiro método, chamado de deslocamento variável, o deslocamento de um pixel do quadro é calculado através da fórmula 4.8.

$$d = a.p + b \quad (4.8)$$

Onde 'd' é o deslocamento da camada vermelha, 'p' a profundidade do pixel e 'a' e 'b' são constantes calculadas a partir de um dos dois sistemas lineares a seguir, dependendo do tipo de anáglifo definido pelo usuário, normal ou invertido.

Se a construção do anáglifo é normal, então quanto mais próximo o objeto está da câmera (maiores valores de 'p'), maior o deslocamento 'd' da camada vermelha. Logo, o sistema linear é:

$$d_{max} = a.p_{max} + b$$

$$0 = a.0 + b$$

Sendo que d_{max} é o deslocamento máximo do anáglifo, definido pelo usuário. E p_{max} é a profundidade máxima considerada do mapa de profundidade, chamado sensibilidade do algoritmo de deslocamento variável, também definida pelo usuário.

Já no caso da construção do anáglifo do tipo invertido, quanto mais próximo o objeto está da câmera, menor o deslocamento da camada vermelha. Logo, o sistema linear é:

$$0 = a.p_{max} + b$$

$$d_{max} = a.0 + b$$

No segundo método, chamado deslocamento por camadas, são criadas faixas de valores de profundidade, sendo que o número de faixas 'n' é definido pelo usuário. Os limites das faixas são calculados a partir do incremento 'i', de acordo com a fórmula 4.9, sendo que P_{max} , nesse caso, é chamado de sensibilidade do algoritmo de deslocamento por camadas.

$$i = \frac{P_{max}}{n} \quad (4.9)$$

Cada faixa de profundidade possui uma diferença constante, definida pelo usuário, no valor de deslocamento. E, semelhante ao método de deslocamento variável, se a construção do anáglifo é normal à primeira faixa de profundidade (objetos mais distantes), é associado o valor mínimo de deslocamento. Se a construção é de anáglifo invertido, então o valor associado é máximo.

4.7.7 Construção do Vídeo Estereoscópico

O formato de vídeo definido como entrada do sistema de conversão é o YUV, um formato sem codificação, onde cada pixel do quadro é representado nas suas componentes de luminância e cromaticidade. Portanto, para manipular a componente vermelha e criar a imagem deslocada é necessário converter cada pixel para o sistema de cores RGB. Para isso, são utilizadas as fórmulas 4.10, 4.11 e 4.12, sendo R, G e B as componentes no sistema RGB e Y, U, V as componentes no sistema YCrCb.

$$R = \text{mnimo}((9535 * (Y - 16) + 13074 * (V - 128)) \gg 13, 255) \quad (4.10)$$

$$G = \text{mnimo}((9535.(Y - 16) - 6660.(V - 128) - 3203.(U - 128)) \gg 13, 255) \quad (4.11)$$

$$B = \text{mnimo}((9535.(Y - 16) + 16531.(U - 128)) \gg 13, 255) \quad (4.12)$$

Após o deslocamento, os pixels devem ser reconvertidos para o sistema YCrCb utilizando as fórmulas 4.13, 4.14 e 4.15.

$$Y = \text{mnimo}(\text{mdulo}(R.2104 + G.4130 + B.802 + 4096 + 131072) \gg 13, 235) \quad (4.13)$$

$$U = \text{mnimo}(\text{mdulo}(R.(-1214)+G.(-2384)+B.3598+4096+1048576) \gg 13, 240) \quad (4.14)$$

$$V = \text{mnimo}(\text{mdulo}(R.3598+G.(-3013)+B.(-585)+4096+1048576) \gg 13, 240) \quad (4.15)$$

Outra característica importante do vídeo de entrada é a amostragem, que deve ser 4:2:0, a mais comum atualmente. Esse tipo de amostragem implica que não se pode deslocar a componente vermelha de cada pixel individualmente, pois haveria distorção nas cores dos outros três pixels do bloco de amostragem. Portanto, a estratégia de deslocar o grupo de quatro pixels do bloco de amostragem é adotada, ao invés do pixel individual.

O processo de construção do vídeo 3D é executado através da iteração sobre cada bloco de 4 pixels do quadro, na ordem da figura 4.14.

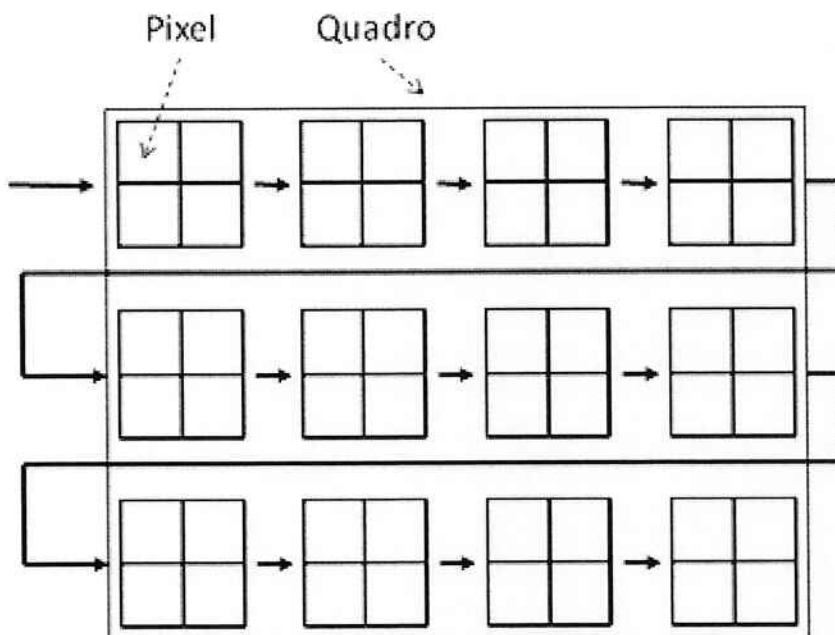


Figura 4.14: Ordem de análise dos pixels para deslocamento da camada vermelha.

Para cada bloco de pixels iterado, há quatro valores de luminância, um de cromaância C_r e um de cromaância C_b . O bloco iterado é chamado de origem e a partir do valor 'd' do deslocamento calculado para o primeiro pixel do bloco, é determinado qual o bloco de destino, ou seja, o bloco com 'd' pixels de distância do bloco de origem. Os valores no sistema YCrCb de cada pixel do bloco de origem e do bloco de destino são convertidos para RGB, conforme a figura 4.15.

Os valores da componente vermelha dos pixels de destino são então substituídos pelos da origem e são recalculadas as componentes YCrCb para o primeiro pixel. Já para os demais, apenas a componente de luminância Y é recalculada, conforme a figura 4.16, sendo que Y' é o novo valor de luminância e C' os novos valores de cromaância.

Com isso, é concluída a conversão do vídeo para 3D. Os mapas de profundidade são salvos em imagem, para que o usuário possa verificar o impacto da escolha dos parâmetros de conversão. A figura 4.17 ilustra o fluxo final do decodificador modificado.

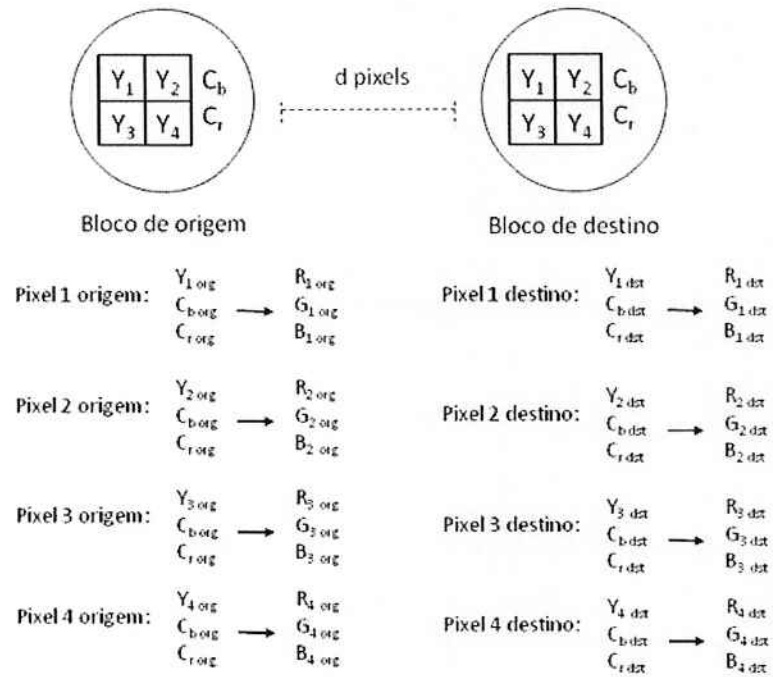


Figura 4.15: Conversão para RGB dos pixels do bloco de origem e destino antes do deslocamento da camada vermelha.

$$\begin{aligned} R'_{1\text{ dst}} &= R_{1\text{ orig}} \\ R'_{2\text{ dst}} &= R_{2\text{ orig}} \\ R'_{3\text{ dst}} &= R_{3\text{ orig}} \\ R'_{4\text{ dst}} &= R_{4\text{ orig}} \end{aligned}$$

$$\begin{matrix} R'_{1\text{ dst}} \\ G_{1\text{ dst}} \\ B_{1\text{ dst}} \end{matrix} \rightarrow \begin{matrix} Y'_{1\text{ dst}} \\ C'_{b\text{ dst}} \\ C'_{r\text{ dst}} \end{matrix}$$

$$\begin{matrix} R'_{3\text{ dst}} \\ G_{3\text{ dst}} \\ B_{3\text{ dst}} \end{matrix} \rightarrow \begin{matrix} Y'_{3\text{ dst}} \\ C'_{b\text{ dst}} \\ C'_{r\text{ dst}} \end{matrix}$$

$$\begin{matrix} R'_{2\text{ dst}} \\ G_{2\text{ dst}} \\ B_{2\text{ dst}} \end{matrix} \rightarrow \begin{matrix} Y'_{2\text{ dst}} \\ C'_{b\text{ dst}} \\ C'_{r\text{ dst}} \end{matrix}$$

$$\begin{matrix} R'_{4\text{ dst}} \\ G_{4\text{ dst}} \\ B_{4\text{ dst}} \end{matrix} \rightarrow \begin{matrix} Y'_{4\text{ dst}} \\ C'_{b\text{ dst}} \\ C'_{r\text{ dst}} \end{matrix}$$

Figura 4.16: Cálculo dos novos valores YCrCb para o bloco de destino.

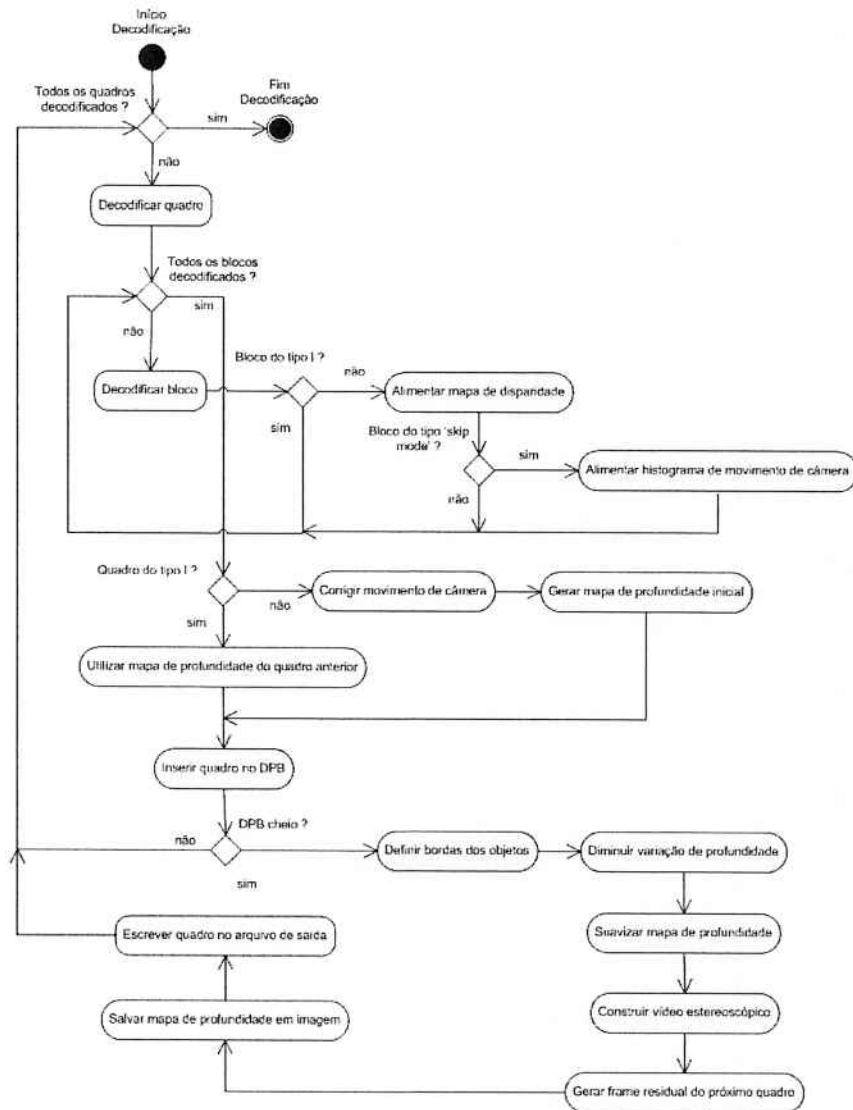


Figura 4.17: Fluxo final do decodificador modificado para geração de vídeo 3D.

5 TESTES E AVALIAÇÃO

5.1 Cenários de Teste

Para avaliar o desempenho do sistema de conversão V3D, foram adotados quatro vídeos de teste frequentemente utilizados em demonstrações de padrões de codificação, por serem de uso livre para projetos sem fins comerciais. Para cada vídeo foram geradas três versões para avaliação: uma versão convertida pelo software do projeto V3D, uma versão convertida pelo software Depthgauge e uma versão gerada em tempo real pelo software Tridef. As características dos vídeos podem ser observadas nas tabelas: 5.1, 5.2, 5.3 e 5.4.

Tabela 5.1: Vídeo 1: Soccer - Cena de um jogo de futebol com jogadores andando e correndo

Característica	Valor
Resolução:	352x288
Quadros por segundo:	30
Movimento de câmera:	Horizontal
Distância do plano de fundo:	Média
Distância dos objetos principais:	Próxima
Complexidade do plano de fundo:	Média
Complexidade dos objetos principais:	Alta
Velocidade do movimento dos objetos principais:	Baixa e Média
Imagem Ilustrativa:	Figura: 5.1

5.2 Parâmetros de Avaliação

A pesquisa comparativa dos vídeos consistiu na avaliação subjetiva da qualidade visual de cada versão produzida. Os entrevistados foram instruídos a atribuírem uma nota geral numa escala de 0 a 5 para avaliar duas características:

- **Percepção de Profundidade:** sensação adequada da diferença de profundidade entre os objetos da cena, gerando o efeito 3D. Na escala, uma nota 0 indica nenhuma percep-



Figura 5.1: Quadro de exemplo do vídeo 1 de teste (Soccer).

Tabela 5.2: Vídeo 2: City - Filmagem aérea de uma cidade com movimento de câmera rotacional em volta de um prédio

Característica	Valor
Resolução:	352x288
Quadros por segundo:	30
Movimento de câmera:	Rotacional
Distância do plano de fundo:	Distante
Distância dos objetos principais:	Média
Complexidade do plano de fundo:	Alta
Complexidade dos objetos principais:	Média
Velocidade do movimento dos objetos principais:	Inexistente
Imagem Ilustrativa:	Figura: 5.2



Figura 5.2: Quadro de exemplo do vídeo 2 de teste (City).

ção de profundidade, enquanto 5 corresponde à visualização de uma grande profundidade no vídeo.

Tabela 5.3: Vídeo 3: Ice - Cena de pessoas patinando no gelo

Característica	Valor
Resolução:	352x288
Quadros por segundo:	30
Movimento de câmera:	Inexistente
Distância do plano de fundo:	Próxima
Distância dos objetos principais:	Média
Complexidade do plano de fundo:	Baixa
Complexidade dos objetos principais:	Alta
Velocidade do movimento dos objetos principais:	Baixa
Imagem Ilustrativa:	Figura: 5.3



Figura 5.3: Quadro de exemplo do vídeo 3 de teste (Ice).



Figura 5.4: Quadro de exemplo do vídeo 4 de teste (Bus).

- **Qualidade do Vídeo:** manutenção das características originais do vídeo, tais como a definição do contorno dos objetos e inexistência de artefatos. Nota 0 corresponde a

Tabela 5.4: Vídeo 4: Bus - Filmagem de um ônibus em deslocamento

Característica	Valor
Resolução:	352x288
Quadros por segundo:	30
Movimento de câmera:	Horizontal
Distância do plano de fundo:	Média
Distância dos objetos principais:	Média
Complexidade do plano de fundo:	Alta
Complexidade dos objetos principais:	Média
Velocidade do movimento dos objetos principais:	Média
Imagem Ilustrativa:	Figura: 5.4

grande perda visual, inviabilizando seu entendimento. Nota 5 indica nenhuma degradação visual observada.

5.3 Resultados

Embora seja possível utilizar uma configuração padrão da técnica de conversão 3D, para cada vídeo de teste procurou-se uma configuração que trouxesse a melhor percepção possível de profundidade, sem comprometer a qualidade do vídeo. As configurações utilizadas e os mapas de profundidade de cada fase da técnica podem ser observados nas imagens dos apêndices I a IV.

5.4 Avaliação dos Resultados

Analisando os mapas de profundidade e os vídeos 3D anáglifo gerados, nota-se que a técnica de conversão 3D adotada é incapaz de gerar mapas de profundidade bem definidos para todos os quadros do vídeo. Além disso, quadros consecutivos muitas vezes apresentam grandes variações de profundidade. Esses dois fatores levam a uma redução da qualidade final do vídeo 3D. Porém, a técnica se mostrou suficiente para gerar vídeos 3D com uma razoável percepção de profundidade.

O resultado da pesquisa pode ser observado nas tabelas 5.5 - 5.8.

Através dos resultados da pesquisa, conclui-se que todas falharam em prover ao mesmo tempo um vídeo 3D com boa qualidade e percepção de profundidade real dos objetos em cena. Tanto a ferramenta Depthgauge como o Tridef, geraram vídeos 3D com poucas alterações da qualidade dos vídeos originais, mas a percepção de profundidade foi avaliada, em todos os

Tabela 5.5: Resultado da pesquisa de comparação para o vídeo de teste 1 (Soccer)

	Percepção de Profundidade	Qualidade do Vídeo	Nota Geral
Conversor Desenvolvido	3	2,5	3
Conversor Depthgauge	0,5	4	1
Conversor Tridef	2	3,5	2,5

Tabela 5.6: Resultado da pesquisa de comparação para o vídeo de teste 2 (City)

	Percepção de Profundidade	Qualidade do Vídeo	Nota Geral
Conversor Desenvolvido	3,5	2	3
Conversor Depthgauge	1	4	1
Conversor Tridef	1,5	4	2,5

testes, como pior em relação à do conversor desenvolvido - V3D. A conclusão final da pesquisa é que o conversor desenvolvido gera perdas na qualidade visual dos vídeos, mas produz boa percepção de profundidade em comparação com as outras duas ferramentas, que priorizam uma manutenção da qualidade do vídeo original. Existe, portanto, uma dependência entre a produção de noção de profundidade e a manutenção da qualidade do vídeo original.

Tabela 5.7: Resultado da pesquisa de comparação para o vídeo de teste 3 (Ice)

	Percepção de Profundidade	Qualidade do Vídeo	Nota Geral
Conversor Desenvolvido	3,5	3	3
Conversor Depthgauge	1	5	1
Conversor Tridef	1,5	3,5	2

Tabela 5.8: Resultado da pesquisa de comparação para o vídeo de teste 4 (Bus)

	Percepção de Profundidade	Qualidade do Vídeo	Nota Geral
Conversor Desenvolvido	1,5	3	2,5
Conversor Depthgauge	0	5	1
Conversor Tridef	0,5	4,5	1

6 CONSIDERAÇÕES FINAIS

6.1 Conceituais

Uma das maiores dificuldades encontradas no projeto foi a definição de escopo, pois exigiu a assimilação de muitos conceitos de codificação de vídeo para estimar a capacidade de produção e contribuição do trabalho. Inicialmente, não havia uma especificação de requisitos clara do software conversor, conseqüentemente muitos conceitos pesquisados e desenvolvidos pelo grupo não foram necessários para a implementação do software.

De toda forma, os integrantes do grupo adquiriram bons conhecimentos na área de vídeos digitais, em complementação à formação tradicional em engenharia de computação.

6.2 Tecnológicas

Em relação à tecnologia, foram enfrentadas dificuldade em modificar o código do decodificador de referência do padrão H.264, pois a documentação é bastante restrita, uma vez que o público alvo é composto por pessoas já familiarizadas com os conceitos e práticas da área. Outro ponto de destaque foi o baixo desempenho do codificador de referência, o que inviabilizou seu uso na versão final do software. Porém, o problema foi contornado com a utilização de um codificador H.264 comercial.

6.3 Gerenciamento de Projeto

O planejamento definido pelo grupo subestimou a bagagem teórica necessária para realização do projeto. Foi gasto mais tempo do que o estimado para pesquisa e estudo, até que os componentes do grupo estivessem seguros para definir o escopo e executar o trabalho. As reuniões realizadas com a professora orientadora do projeto serviram para definir as atividades seguintes e verificar se o andamento do projeto era satisfatório.

6.4 Conclusão

A técnica de conversão de vídeo 3D adotada e implementada no software conversor desenvolvido no projeto foi considerada insuficiente para a criação de vídeos 3D de alta qualidade. Porém, a percepção de profundidade gerada foi avaliada como superior em comparação com duas ferramentas comerciais de conversão.

Conclui-se então que a técnica é adequada para uma aproximação inicial do mapa de profundidade, sendo necessários outros mecanismos de refinamento, a fim de garantir a qualidade final do vídeo.

6.5 Contribuições

A maior contribuição do projeto são os resultados da implementação da técnica de extração de profundidade a partir do deslocamento dos objetos em cena, utilizando o padrão H.264. Os testes mostraram que a técnica é viável, porém é necessário ainda o desenvolvimento de novos algoritmos para garantir a qualidade final do vídeo 3D.

6.6 Trabalhos Futuros

Como trabalhos futuros, propõe-se o estudo de técnicas de refinamento da determinação de profundidade, de melhorias na área de definição do contorno dos objetos e de estratégias de controle de variações no mapa de profundidade entre os quadros do vídeo. O formato de exibição também pode ser adaptado para outras opções além do anáglifo com óculos coloridos. Uma sugestão é a implementação de óculos polarizados.

REFERÊNCIAS BIBLIOGRÁFICAS

- DEPTHGAUGE. www.3dcombine.com - Depthgauge (Software Conversor de Videos 2D - 3D). 2010.
- FEHN, C.; PASTOOR, R. Interactive 3-DTV-concepts and key technologies. *Proceedings of the IEEE*, IEEE, v. 94, n. 3, p. 524–538, 2006. ISSN 0018-9219.
- FENG, Y.; JIANG, J.; IPSON, S. A shape-match based algorithm for pseudo-3D conversion of 2D videos. **Image Processing, 2005. ICIP 2005. IEEE International Conference on**. [S.l.], 2006. v. 3. ISBN 0780391349.
- IDESES, I.; YAROSLAVSKY, L.; FISHBAIN, B. Real-time 2D to 3D video conversion. *Journal of Real-Time Image Processing*, Springer, v. 2, n. 1, p. 3–9, 2007. ISSN 1861-8200.
- IDESES, I.; YAROSLAVSKY, L.; FISHBAIN, B.; VISTUCH, R. 3D from compressed 2D video (Proceedings Paper). 2007.
- KIM, D.; MIN, D.; SOHN, K. A Stereoscopic Video Generation Method Using Stereoscopic Display Characterization and Motion Analysis. *Broadcasting, IEEE Transactions on*, IEEE, v. 54, n. 2, p. 188–197, 2008. ISSN 0018-9316.
- POURAZAD, M.; NASIOPOULOS, P.; WARD, R. An H. 264-based scheme for 2D to 3D video conversion. *Consumer Electronics, IEEE Transactions on*, IEEE, v. 55, n. 2, p. 742–748, 2009. ISSN 0098-3063.
- RICHARDSON, I. *H. 264 and MPEG-4 video compression*. [S.l.]: Wiley Online Library, 2003. ISBN 0470848375.
- SMOLIC, A.; MUELLER, K.; MERKLE, P.; KAUFF, P.; WIEGAND, T. An overview of available and emerging 3D video formats and depth enhanced stereo as efficient generic solution. **Picture Coding Symposium, 2009. PCS 2009**. [S.l.], 2009. p. 1–4.
- SON, J.; JAVIDI, B.; KWACK, K. Methods for displaying three-dimensional images. *Proceedings of the IEEE*, IEEE, v. 94, n. 3, p. 502–523, 2006. ISSN 0018-9219.
- STERN, A.; JAVIDI, B. Three-dimensional image sensing, visualization, and processing using integral imaging. *Proceedings of the IEEE*, IEEE, v. 94, n. 3, p. 591–607, 2006. ISSN 0018-9219.
- TRIDEF. www.tridef.com - Tridef 3D (Software Conversor de Videos 2D - 3D em Tempo Real). 2010.
- WU, C.; ER, G.; XIE, X.; LI, T.; CAO, X.; DAI, Q. A novel method for semi-automatic 2D to 3D video conversion. **3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, 2008**. [S.l.], 2008. p. 65–68.

ZHANG, L.; TAM, W. Stereoscopic image generation based on depth images for 3D TV. *Broadcasting, IEEE Transactions on, IEEE*, v. 51, n. 2, p. 191–199, 2005. ISSN 0018-9316.

Apêndice I – CARACTERÍSTICAS E EVOLUÇÃO DOS MAPAS DE PROFUNDIDADE: VÍDEO DE TESTE 1 (SOCCER)

Tabela I.1: Configuração utilizada para o vídeo 1 de teste (Soccer)

Parâmetro	Atribuição
Corrigir Movimento de Câmera	Sim
Definir as Bordas dos Objetos	Sim
- Sensibilidade	15
Diminuir Variação de Profundidade	Sim
- a	0.5
- b	2.5
- Sensibilidade	5
Utilizar Deslocamento Variável	Sim
- Profundidade	10
- Sensibilidade	25
Suavizar Mapa de Profundidade	Sim
- σ	3
- w	9
Utilizar Deslocamento por Camadas	Não
- Número de Camadas	-
- Diferença de Profundidade	-
- Sensibilidade	-
Deslocamento Base	5
Construir Anáglifo Inverso	Não



Figura I.1: *Quadro original do vídeo 1 de teste (Soccer).*

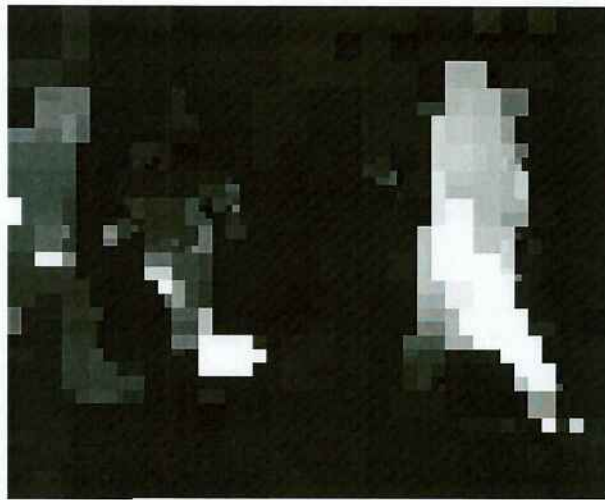


Figura I.2: *Mapa de profundidade inicial extraído do vídeo 1 de teste (Soccer).*

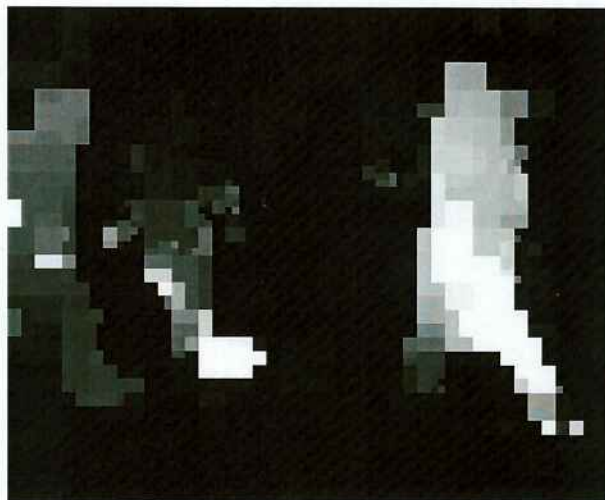


Figura I.3: *Mapa de profundidade do vídeo 1 de teste (Soccer) após correção de movimento de câmera.*

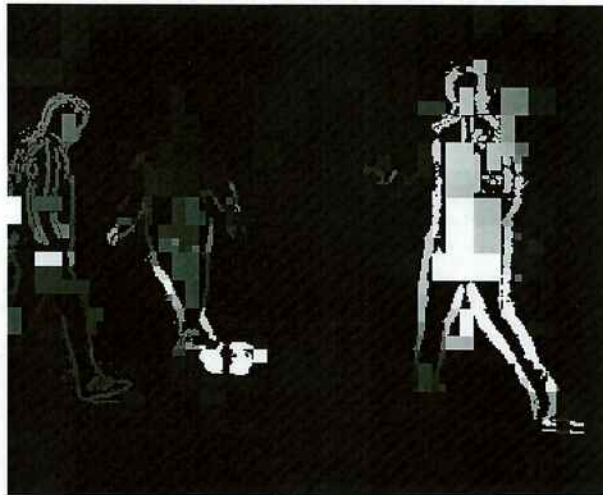


Figura 1.4: Mapa de profundidade do vídeo 1 de teste (Soccer) após segunda fase do algoritmo de definição de bordas.

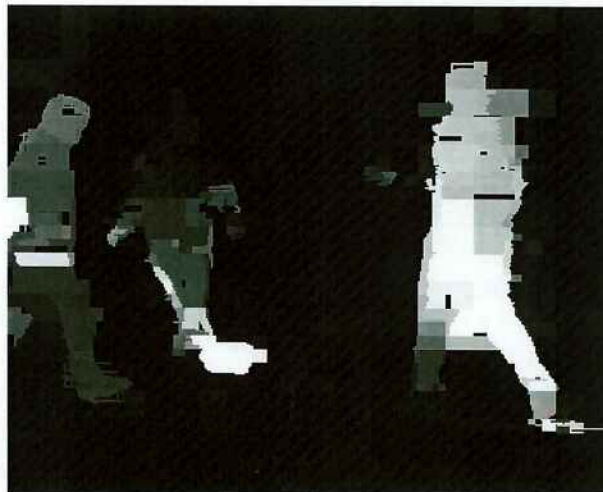


Figura 1.5: Mapa de profundidade do vídeo 1 de teste (Soccer) após a conclusão do algoritmo de definição de bordas.

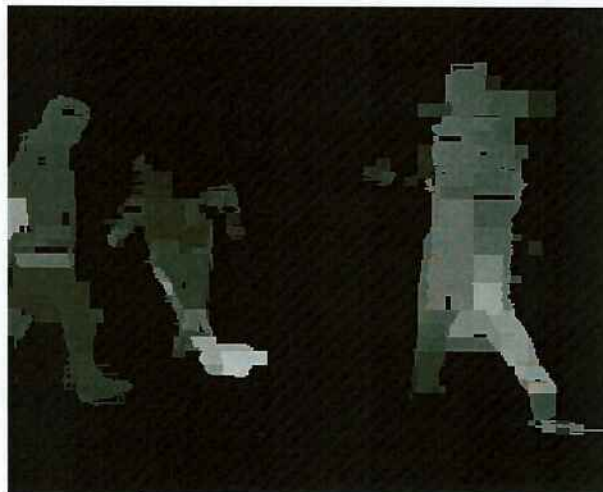


Figura 1.6: Mapa de profundidade do vídeo 1 de teste (Soccer) após a conclusão do algoritmo de diminuição da variação de profundidade.

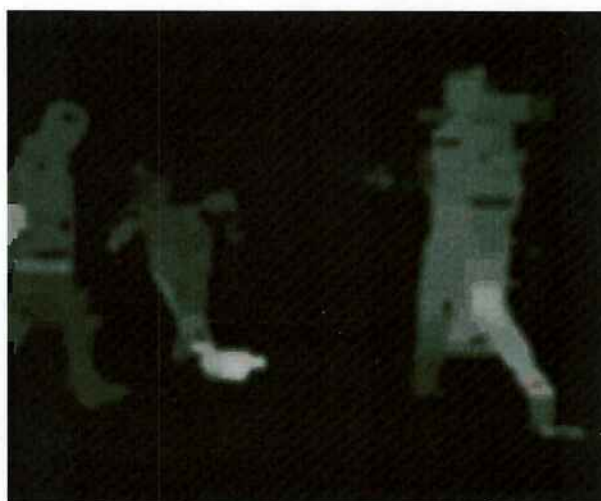


Figura I.7: Mapa de profundidade do vídeo 1 de teste (Soccer) após a conclusão do algoritmo de suavização do mapa.



Figura I.8: Mapa de deslocamento da camada vermelha do vídeo 1 de teste (Soccer).



Figura I.9: Quadro final estereoscópico do vídeo 1 de teste (Soccer).

Apêndice II – CARACTERÍSTICAS E EVOLUÇÃO DOS MAPAS DE PROFUNDIDADE: VÍDEO DE TESTE 2 (CITY)

Tabela II.1: Configuração utilizada para o vídeo 2 de teste (City)

Parâmetro	Atribuição
Corrigir Movimento de Câmera	Sim
Definir as Bordas dos Objetos	Sim
- Sensibilidade	10
Diminuir Variação de Profundidade	Sim
- a	0.5
- b	2.5
- Sensibilidade	2
Utilizar Deslocamento Variável	Sim
- Profundidade	15
- Sensibilidade	40
Suavizar Mapa de Profundidade	Não
- σ	-
- w	-
Utilizar Deslocamento por Camadas	Não
- Número de Camadas	-
- Diferença de Profundidade	-
- Sensibilidade	-
Deslocamento Base	5
Construir Anáglifo Inverso	Sim



Figura II.1: *Quadro original do vídeo 2 de teste (City).*

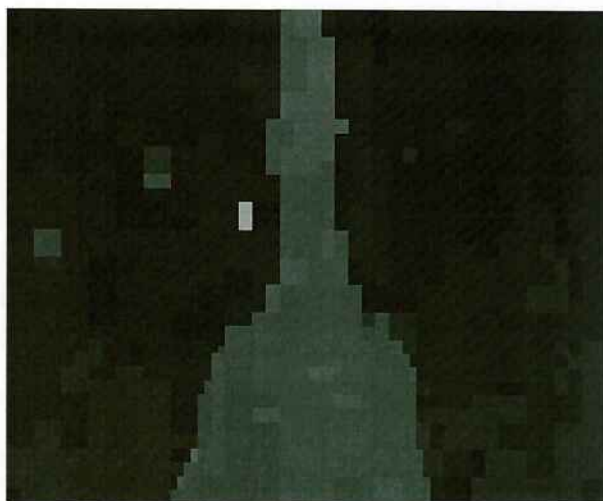


Figura II.2: *Mapa de profundidade inicial extraído do vídeo 2 de teste (City).*

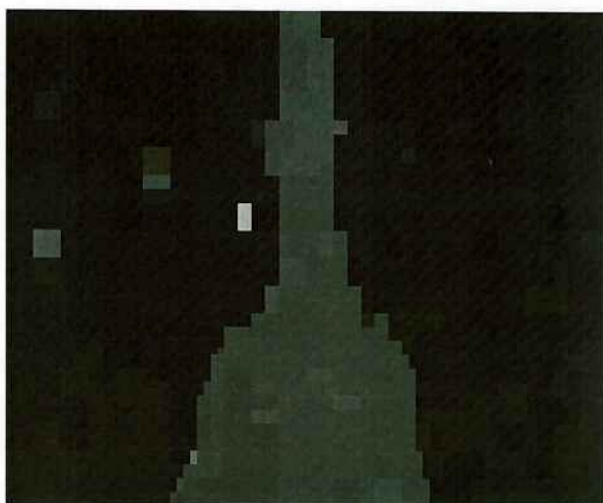


Figura II.3: *Mapa de profundidade do vídeo 2 de teste (City) após correção de movimento de câmera.*

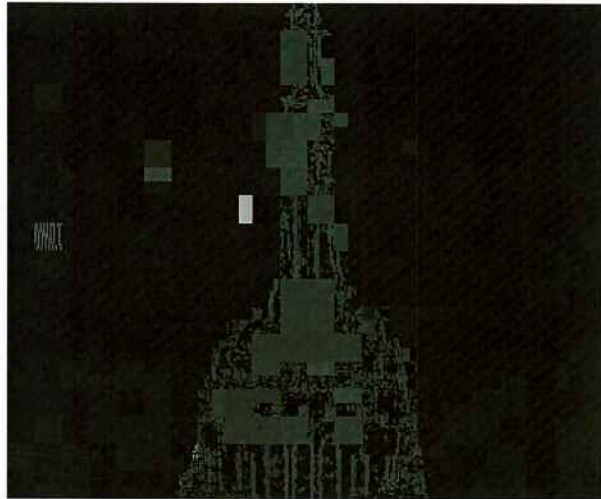


Figura II.4: Mapa de profundidade do vídeo 2 de teste (City) após segunda fase do algoritmo de definição de bordas.

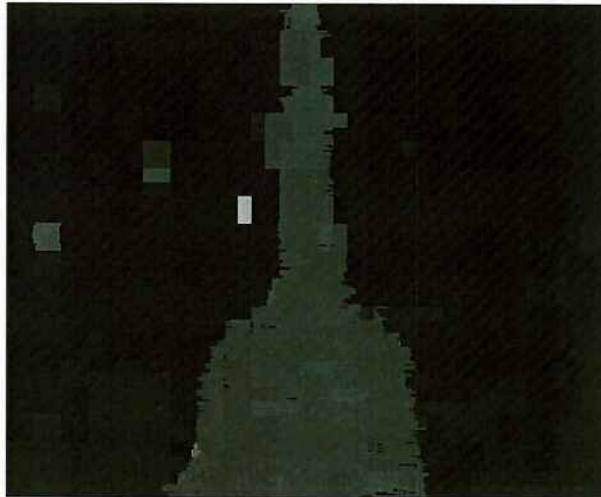


Figura II.5: Mapa de profundidade do vídeo 2 de teste (City) após a conclusão do algoritmo de definição de bordas.

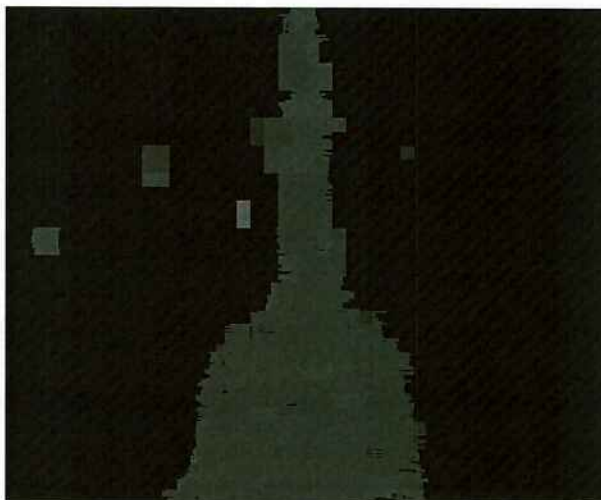


Figura II.6: Mapa de profundidade do vídeo 2 de teste (City) após a conclusão do algoritmo de diminuição da variação de profundidade.

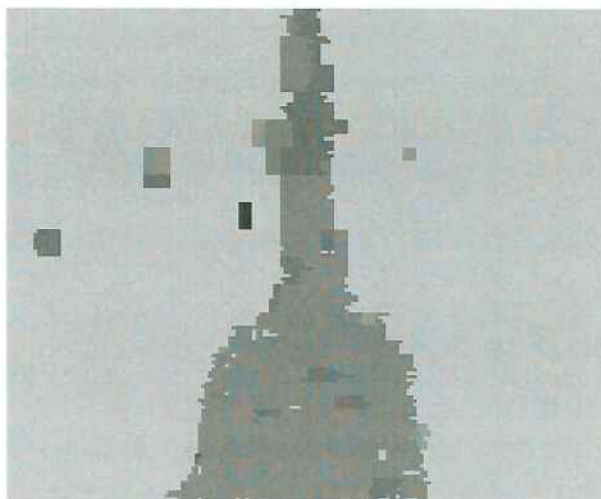


Figura II.7: Mapa de deslocamento da camada vermelha do vídeo 2 de teste (City).

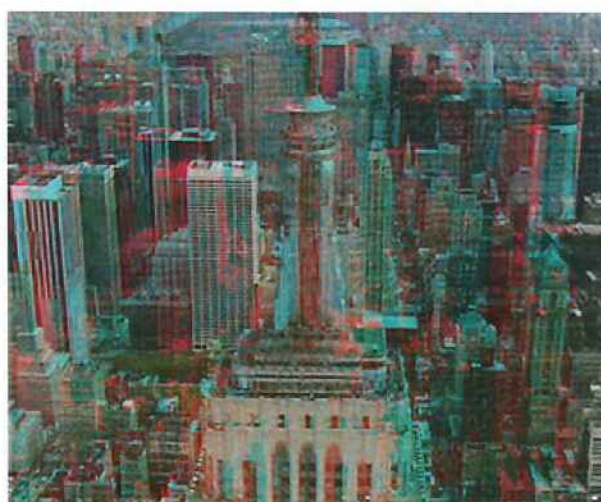


Figura II.8: Quadro final estereoscópico do vídeo 2 de teste (City).

Apêndice III – CARACTERÍSTICAS E EVOLUÇÃO DOS MAPAS DE PROFUNDIDADE: VÍDEO DE TESTE 3 (ICE)

Tabela III.1: Configuração utilizada para o vídeo 3 de teste (Ice)

Parâmetro	Atribuição
Corrigir Movimento de Câmera	Não
Definir as Bordas dos Objetos	Sim
- Sensibilidade	10
Diminuir Variação de Profundidade	Sim
- a	0.5
- b	2.5
- Sensibilidade	2
Utilizar Deslocamento Variável	Sim
- Profundidade	15
- Sensibilidade	25
Suavizar Mapa de Profundidade	Sim
- σ	3
- w	9
Utilizar Deslocamento por Camadas	Não
- Número de Camadas	-
- Diferença de Profundidade	-
- Sensibilidade	-
Deslocamento Base	5
Construir Anáglifo Inverso	Sim



Figura III.1: *Quadro original do vídeo 3 de teste (Ice).*

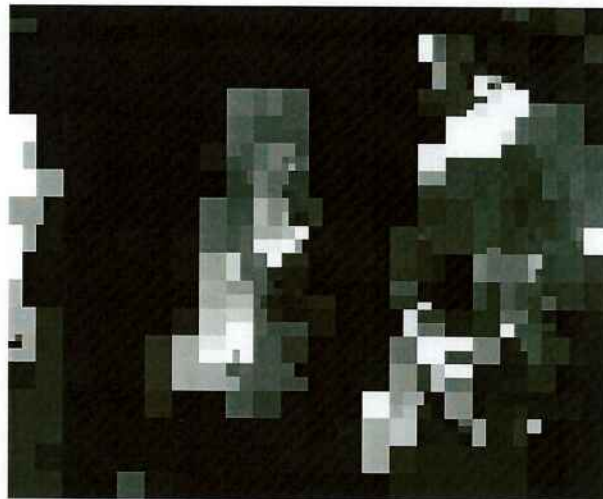


Figura III.2: *Mapa de profundidade inicial extraído do vídeo 3 de teste (Ice).*

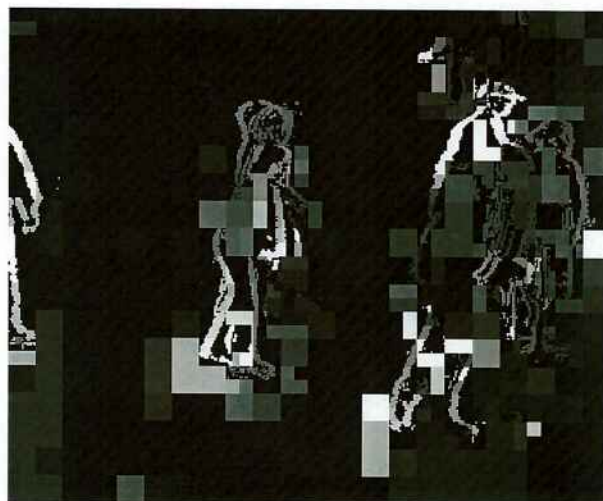


Figura III.3: *Mapa de profundidade do vídeo 3 de teste (Ice) após segunda fase do algoritmo de definição de bordas.*

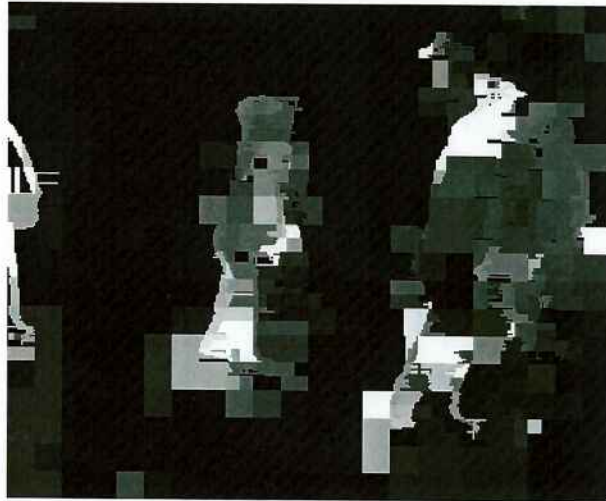


Figura III.4: Mapa de profundidade do vídeo 3 de teste (Ice) após a conclusão do algoritmo de definição de bordas.

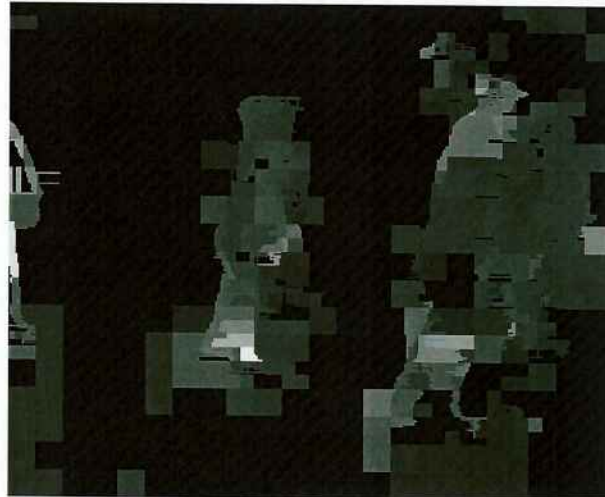


Figura III.5: Mapa de profundidade do vídeo 3 de teste (Ice) após a conclusão do algoritmo de diminuição da variação de profundidade.



Figura III.6: Mapa de profundidade do vídeo 3 de teste (Ice) após a conclusão do algoritmo de suavização do mapa.

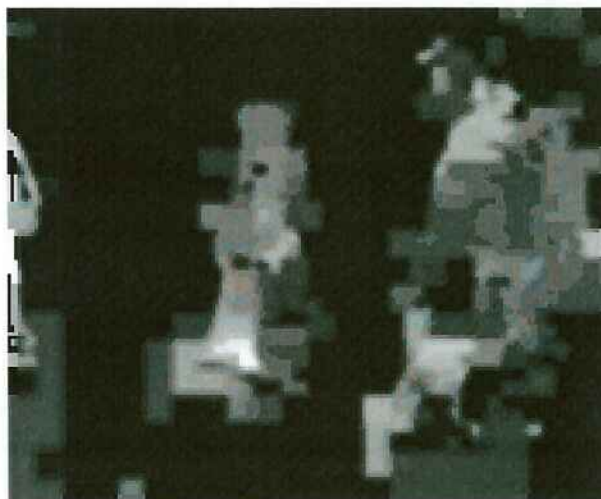


Figura III.7: Mapa de deslocamento da camada vermelha do vídeo 3 de teste (Ice).



Figura III.8: Quadro final estereoscópico do vídeo 3 de teste (Ice).

Apêndice IV – CARACTERÍSTICAS E EVOLUÇÃO DOS MAPAS DE PROFUNDIDADE: VÍDEO DE TESTE 4 (BUS)

Tabela IV.1: Configuração utilizada para o vídeo 4 de teste (Bus)

Parâmetro	Atribuição
Corrigir Movimento de Câmera	Sim
Definir as Bordas dos Objetos	Sim
- Sensibilidade	15
Diminuir Variação de Profundidade	Sim
- a	0.5
- b	2.5
- Sensibilidade	5
Utilizar Deslocamento Variável	Sim
- Profundidade	15
- Sensibilidade	25
Suavizar Mapa de Profundidade	Sim
- σ	3
- w	9
Utilizar Deslocamento por Camadas	Não
- Número de Camadas	-
- Diferença de Profundidade	-
- Sensibilidade	-
Deslocamento Base	5
Construir Anáglifo Inverso	Sim



Figura IV.1: *Quadro original do vídeo 4 de teste (Bus).*

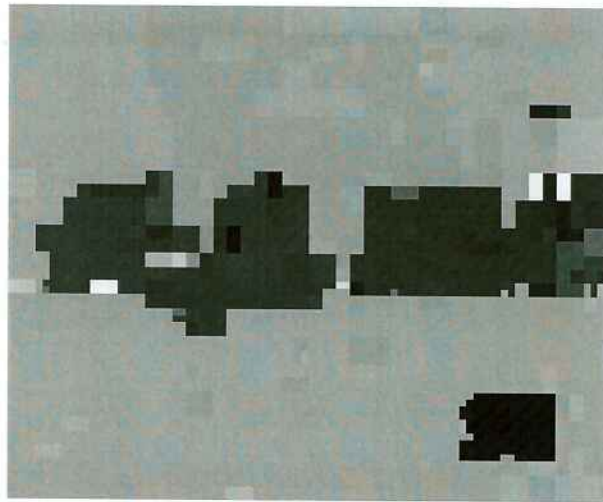


Figura IV.2: *Mapa de profundidade inicial extraído do vídeo 4 de teste (Bus).*

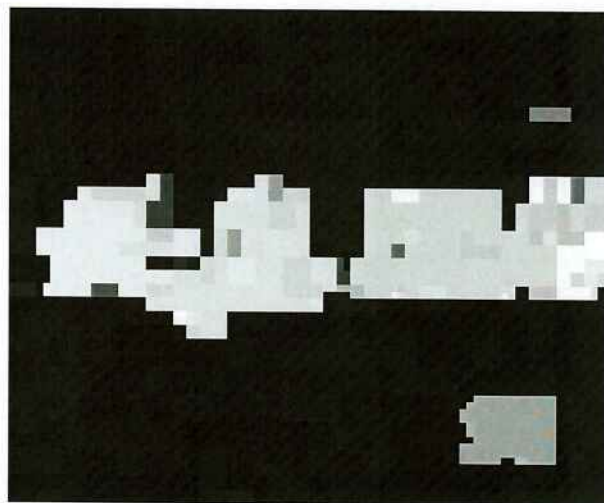


Figura IV.3: *Mapa de profundidade do vídeo 4 de teste (Bus) após correção de movimento de câmera.*

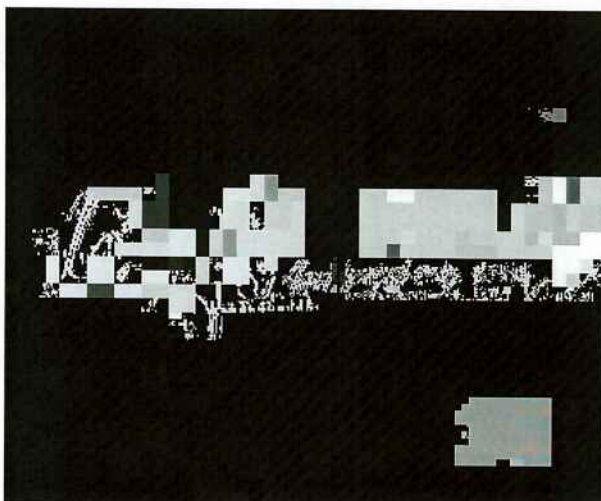


Figura IV.4: Mapa de profundidade do vídeo 4 de teste (Bus) após segunda fase do algoritmo de definição de bordas.

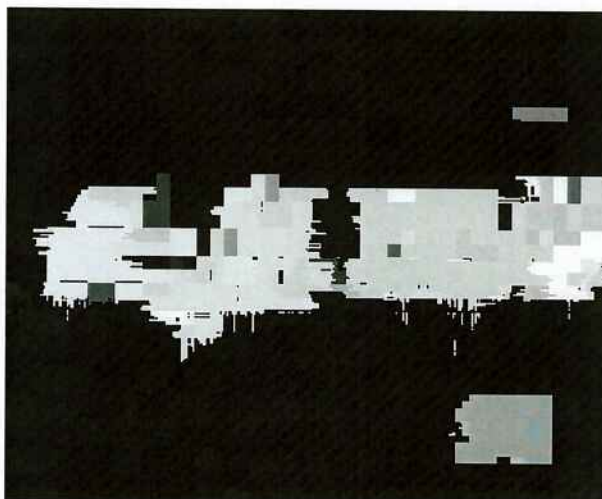


Figura IV.5: Mapa de profundidade do vídeo 4 de teste (Bus) após a conclusão do algoritmo de definição de bordas.

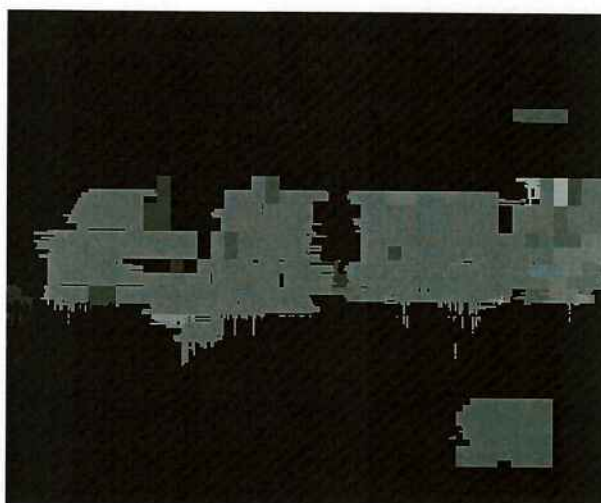


Figura IV.6: Mapa de profundidade do vídeo 4 de teste (Bus) após a conclusão do algoritmo de diminuição da variação de profundidade.

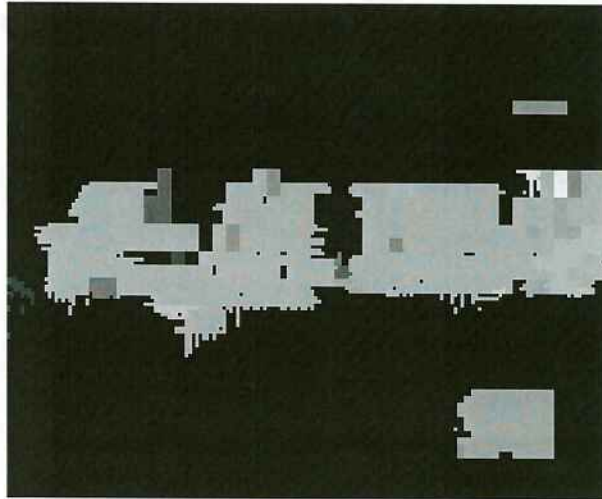


Figura IV.7: Mapa de deslocamento da camada vermelha do vídeo 4 de teste (Bus).



Figura IV.8: Quadro final estereoscópico do vídeo 4 de teste (Bus).