

UNIVERSIDADE FEDERAL DO PAMPA

Gustavo Lopes Tamiosso

Implementação de um modelo  
fenomenológico para geração de imagens  
CRT

Alegrete  
2023

Gustavo Lopes Tamiosso

# Implementação de um modelo fenomenológico para geração de imagens CRT

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Marcelo Resende Thielo

Alegrete  
2023

**Gustavo Lopes Tamiosso**

**Implementação de um modelo fenomenológico para a geração de imagens CRT**

Trabalho de Conclusão de Curso apresentado ao Curso de Ciência da Computação da Universidade Federal do Pampa como requisito parcial para obtenção do Título de Bacharel em Ciência da Computação.

Monografia defendida e aprovada em: 04 de julho de 2023.

Banca examinadora:

---

Prof. Dr. Marcelo Resende Thielo

Orientador

UNIPAMPA

---

Prof. Dr. Fabio Paulo Basso

UNIPAMPA

---

Prof. Me. Jean Felipe Patikowski Cheiran

UNIPAMPA



Assinado eletronicamente por **MARCELO RESENDE THIELO, PROFESSOR DO MAGISTERIO SUPERIOR**, em 14/07/2023, às 02:48, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **JEAN FELIPE PATIKOWSKI CHEIRAN, PROFESSOR DO MAGISTERIO SUPERIOR**, em 14/07/2023, às 09:42, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **FABIO PAULO BASSO, PROFESSOR DO MAGISTERIO SUPERIOR**, em 14/07/2023, às 13:13, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



A autenticidade deste documento pode ser conferida no site [https://sei.unipampa.edu.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **1172045** e o código CRC **52B9983B**.

---

Este trabalho é dedicado aos meus pais e irmãos,  
por todo incentivo, tempo e dedicação empenhada.

## **Agradecimentos**

Neste momento, gostaria de agradecer a minha família, meus amigos, colegas e professores que contribuíram para meu desenvolvimento, sabedoria e saúde. Todos tiveram participação ativa e significativa em minha graduação.

Primeiramente, agradeço ao meu pai, Juliano Silva Tamiosso; minha mãe, Tatiana Lopes Tamiosso; meu irmão, Leonardo Lopes Tamiosso; e minha irmã, Laura Lopes Tamiosso. Eles sempre me apoiaram nas minhas decisões, me ensinaram a ser justo e resiliente e representam meu porto seguro nos dias difíceis. Mesmo à distância, foram de suma importância para superar meus desafios e estiveram presentes em todas as fases da minha vida e graduação.

Também gostaria de expressar minha gratidão às pessoas maravilhosas que encontrei durante a graduação: Douglas Vinicius Ledur Dullius, Máximo Antero Corrêa Leguissamo Filho, Mariana Toledo Costa, Uéslei Bervanger Brandt, Rafael de Oliveira Jarczewski, Vinicius de Souza da Silva e Vitor Xavier Siqueira. Por meio dessas amizades, tornei-me uma pessoa melhor, mantive-me no caminho correto e aprendi inúmeras coisas. Competimos de forma saudável durante as disciplinas do curso e isso possibilitou um grande progresso. Eles estiveram sempre lá para me auxiliar, mesmo nos dias mais difíceis.

Por fim, mas não menos importante, gostaria de agradecer ao meu professor orientador e amigo, Marcelo Resende Thielo. Com ele, adquiri mais conhecimento do que um dia imaginei possível, não apenas no âmbito acadêmico, mas também em termos de vida pessoal. Ele sempre me ofereceu apoio e alimentou minha curiosidade.

“A lógica é apenas o princípio da sabedoria,  
e não o seu fim.”

(Spock. Leonard Nimoy, Star Trek)

## Resumo

Neste trabalho, abordamos o problema de exibir corretamente imagens de baixa resolução em telas de alta resolução, desenvolvendo e validando um modelo para simular a renderização típica de imagens de baixa resolução em um monitor de tubo de raios catódicos. O modelo, que inclui uma geometria do tipo “*slot mask*” e um *pipeline* de etapas de processamento foi criado e aplicado em imagens para simular os vários efeitos de um monitor Tubo de Raios Catódicos (CRT). Essas imagens de entrada também foram renderizadas em um televisor CRT real e posteriormente fotografadas para validar empiricamente os resultados do modelo proposto. É importante enfatizar que o modelo não consiste em uma simples máscara de imagem de sombreamento como a usada em muitos emuladores de videogame, mas inclui várias etapas de tratamento e filtragem de entrada no intuito de replicar os processos físicos da forma mais próxima possível. Como os resultados obtidos sugerem, existe uma semelhança visual considerável entre as imagens reais e as sintetizadas utilizando nosso método.

**Palavras-chave:** Tubos de raios catódicos, gráficos de baixa resolução, jogos, experiência visual, renderização.

## Abstract

In this work, we address the problem of properly displaying low resolution images on high resolution displays by developing and validating a model to simulate the typical rendering of low resolution images on a Cathode Ray Tube (CRT) monitor. A model that includes a slot mask-type geometry and a pipeline of processing steps for simulating the various effects due to a CRT display was created and applied on the source images. These raw source images were also rendered on a real CRT device and subsequently photographed to empirically validate the results of the proposed model. It is important to stress that our model does not consist of a simple shader image mask as used in many video game emulators, but includes multiple steps of input treatment and filtering in order to replicate the physical processes as closely as possible. As the obtained results suggest, there is a considerable amount of visual similarity between the real and our model's synthetic images.

**Key-words:** Cathode ray tubes, low resolution graphics, video games, visual experience, rendering.

## Lista de figuras

Figura 1 – Sprites da versão para Family Computer do jogo “Wizardry” (1987). . . . .	14
Figura 2 – Aproximação na tela de um iPhone. . . . .	17
Figura 3 – Representação do tubo de raios catódicos. . . . .	18
Figura 4 – Distribuição de intensidade espacial Gaussiana de luminosidade ao longo de uma tela de CRT. . . . .	19
Figura 5 – Sprites do jogo “The Guardian Legend” para Nintendo Entertainment System. . . . .	19
Figura 6 – Exemplo da utilização de matrizes de convolução. . . . .	20
Figura 7 – Exemplo da aplicação do desfoque gaussiano. . . . .	21
Figura 8 – <i>Pipeline</i> do modelo proposto. . . . .	28
Figura 9 – Demonstração da primeira e segunda etapa . . . . .	29
Figura 10 – Demonstração da terceira etapa . . . . .	29
Figura 11 – Exemplo do formato da máscara gerada pelo modelo. . . . .	29
Figura 12 – Demonstração da saturação do fósforo . . . . .	30
Figura 13 – Demonstração do tratamento logístico . . . . .	31
Figura 14 – Demonstração das <i>scanlines</i> . . . . .	31
Figura 15 – Modelo com ajustes globais na coloração . . . . .	32
Figura 16 – Exemplo de <i>pixels</i> de entrada usados na calibração . . . . .	33
Figura 17 – Exemplo de fotografias dos <i>pixels</i> usados na calibração . . . . .	33
Figura 18 – Exemplo de imagens sintetizadas dos <i>pixels</i> durante a calibração . . . . .	34
Figura 19 – <i>Sprite</i> do game Turrican 2: The Final Fight para Commodore Amiga. . . . .	37
Figura 20 – Comparação entre imagem CRT (a), imagem CRT (b) sintetizada com nosso modelo e (c) sintetizada com <i>shader</i> CRT-Royale, do Turrican 2: The Final Fight para Commodore Amiga. . . . .	37
Figura 21 – <i>Sprite</i> do game Metal Gear Solid 2 para MSX. . . . .	38
Figura 22 – Comparação entre imagem CRT (a), imagem CRT sintetizada (b) sintetizada com nosso modelo e (c) sintetizada com <i>shader</i> CRT-Royale, do Metal Gear Solid 2 de MSX. . . . .	38
Figura 23 – Exemplos de variação no desfoque gaussiano da etapa 2. . . . .	39
Figura 24 – Exemplos de variação na velocidade de saturação do fósforo. . . . .	39

## Lista de tabelas

## **Lista de abreviaturas e siglas**

**AG** Algoritmo Genético

**CRT** Tubo de Raios Catódicos

**KS** Kolmogorov-Smirnov

**LCD** Liquid Crystal Display

**LED** Light Emitting Diode

**NTSC** National Television System Committee

**OLED** Organic Light Emitting Diode

**OpenCV** Open Source Computer Vision

**PAL-M** Phase Alternating Line padrão M

**RF** Rádio Frequência

**RGB** Red Green e Blue

**SDTV** Standart-Definition Television

**SNES** Super Nintendo Entertainment System

**TV** televisão

## Sumário

1	INTRODUÇÃO . . . . .	13
1.1	Organização do documento . . . . .	15
2	FUNDAMENTAÇÃO TEÓRICA . . . . .	16
2.1	Representação digital de imagens em monitores . . . . .	16
2.2	Monitor de tubo de raios catódicos . . . . .	17
2.2.1	Convenções de sinais analógicos . . . . .	17
2.2.2	A tecnologia CRT . . . . .	18
2.2.3	Comportamento da catodoluminescência . . . . .	18
2.2.4	Scanlines . . . . .	18
2.3	Processamento de imagens . . . . .	19
2.3.1	Aplicação de filtros . . . . .	20
2.3.1.1	Desfoque Gaussiano . . . . .	20
2.3.2	<i>Upscaling</i> . . . . .	21
2.4	Testes estatísticos . . . . .	21
2.4.1	Kolmogorov-Smirnov . . . . .	21
2.5	Algoritmos Genéticos . . . . .	22
2.6	Linguagens de programação adotadas . . . . .	22
2.6.1	C++ . . . . .	22
2.6.1.1	OpenCV . . . . .	22
2.6.2	Python . . . . .	23
2.6.2.1	PyGAD . . . . .	23
3	TRABALHOS RELACIONADOS . . . . .	24
3.1	Protocolo de pesquisa . . . . .	24
3.2	Resultados das pesquisas . . . . .	24
3.2.1	Ampliação da resolução de <i>pixel arts</i> . . . . .	24
3.2.2	Simulação dos efeitos fenomenológicos . . . . .	25
4	DESENVOLVIMENTO . . . . .	27
4.1	Ferramentas utilizadas . . . . .	27
4.2	Modelo proposto . . . . .	27
4.2.1	Etapas do modelo . . . . .	28
4.2.2	Método de calibração utilizando algoritmos genéticos . . . . .	32
5	RESULTADOS . . . . .	35
5.1	Evidências empíricas de bons resultados . . . . .	35
5.2	Resultados do método de calibração . . . . .	36

6	CONSIDERAÇÕES FINAIS . . . . .	40
	REFERÊNCIAS . . . . .	41

## 1 Introdução

A televisão (TV) se tornou presente na rotina da humanidade desde que se popularizou após a primeira guerra mundial. O televisor com a tecnologia de CRT deu início a este formato de conteúdo capaz de reproduzir áudio e vídeo, permitindo agora que a informação fosse retransmitida de maneira semelhante aos rádios, agregando a experiência com estímulos visuais. Foram também criados programas de televisão com o intuito de proporcionar entretenimento para as famílias.

As TVs com tecnologia de raios catódicos, apesar de não serem livres de defeitos, eram capazes de reproduzir imagens com resolução em Standart-Definition Television (SDTV) (720x480) que é inferior ao padrão dos tempos modernos mas era suficiente para seus espectadores assistirem o noticiário, séries de TV, desenhos animados, etc. No entanto, o peso e tamanho destes monitores eram bastante inconvenientes para o transporte e até mesmo para sua alocação nas salas de estar.

Nos anos 70 as televisões começaram a ser utilizadas popularmente como um dispositivo de saída para consoles de jogos eletrônicos. Os jogos desta época tiravam vantagem de alguns efeitos das TVs CRT para renderizar os quadros da melhor forma, considerando as limitações de processamento dos dispositivos de reprodução. Efeitos como o desfoque na imagem e alta saturação nas cores foram levados em conta na hora de criar os elementos gráficos destes dispositivos interativos.

Com o surgimento dos monitores de tela plana, as tecnologias de visualização de imagens têm evoluído constantemente, e com isso houve mudanças tanto na apresentação, quanto no caminho do conteúdo até a tela. Essas diversas alterações impactaram muito na forma que o conteúdo é exibido, portanto uma obra que utilizava meios de reprodução da sua época ao ser transposta para as tecnologias atuais não preserva sua exibição exata. Um exemplo disso são consoles como o Super Nintendo Entertainment System (SNES), que foi pensado para ser ligado a televisores CRT via Rádio Frequência (RF) ou via vídeo composto através de um cabo apropriado comprado separadamente, como pode ser observado no manual do Super Nintendo (INSTRUCTION... , 1992). O console gera imagens pouco nítidas, mas quando conectado em monitores modernos apresenta uma nitidez que afeta a qualidade visual dos gráficos.

No século XXI surgiu uma cultura conhecida como *retrogaming*, que segundo Wulf et al. (2018), se refere tanto ao ato de colecionar e jogar consoles e jogos antigos, como também jogos de estilo “*retrô*”, ou seja, que carregam consigo um estilo gráfico antigo e mecânicas de áudio arcaico. Alguns “*retrogamers*” possuem também os televisores e meios de reprodução de áudio e vídeo da época para agregar em sua experiência.

Como qualquer outra forma de artes e mídias, *videogames* refletem o tempo e lugar em que foram criados e podem servir como uma fonte de informação sobre o passado. Existem vários exemplos de museus que se dedicam a jogos, como por exemplo o museu *strong* (STRONG, 2021), que preserva, estuda e interpreta jogos eletrônicos e ma-

teriais relacionados, assim como o museu de artes modernas de Nova Iorque, como visto em MOMA (2020), que possui exemplares de jogos como Tetris<sup>1</sup>, Katamari-Damacy<sup>2</sup>, Magnavox-Odyssey<sup>3</sup>, Minecraft<sup>4</sup> e Myst<sup>5</sup> em sua coleção.

Os monitores CRT estão ficando escassos a ponto de se tornarem itens de colecionadores. Em algum tempo no futuro, os jogos de consoles com o lançamento anterior ao século XXI podem causar estranheza, dado que as imagens de baixa resolução geradas por eles vistas em monitores modernos apresentam grandes quadrados, com cores vibrantes e pouco detalhamento aparente. No entanto, como visto na figura 1, as imagens à esquerda representadas em um monitor CRT, possuem detalhes que não são visíveis nas imagens à direita, que são a representação digital vista em monitores modernos.

Figura 1 – Sprites da versão para Family Computer do jogo “Wizardry” (1987).



Fonte: Auffret (2020)

Existem diversos filtros e técnicas de pós-processamento para reproduzir efeitos que tragam uma exibição aproximada do que se costumava ter ao utilizar as TVs de raios catódicos. No entanto, poucos desses métodos se comprometem com a simulação dos efeitos físicos e buscam cortar caminhos para a redução do custo computacional, a fim de atingir uma parcela maior do público, que possuem usualmente máquinas de baixo desempenho. Além disso, existe pouca documentação a respeito do funcionamento desses métodos, ou sequer explicações de como um determinado efeito está sendo replicado.

<sup>1</sup> [Tetris] (1984), <https://www.igdb.com/games/tetris-4>.

<sup>2</sup> [Katamari-Damacy], <https://www.igdb.com/games/katamari-damacy>.

<sup>3</sup> [Magnavox-Odyssey], <https://olhardigital.com.br/2022/10/04/games-e-consoles/magnavox-odyssey-primeiro-console-da-historia-foi-lancado-ha-50-anos/>.

<sup>4</sup> [Minecraft], <https://www.igdb.com/games/minecraft>.

<sup>5</sup> [Myst], <https://www.igdb.com/games/myst>

Este trabalho visa desenvolver e implementar um modelo fenomenológico de renderização em dispositivos CRT, preservando assim a estética dos gráficos *retrô* através da simulação dos efeitos do monitor de tubo de raios catódicos nos monitores modernos. O termo fenomenológico refere-se à exploração dos fenômenos daquilo que aparece à consciência, onde neste caso nos referimos à nossa percepção dos efeitos vistos em visores CRT.

No século XXI, as TVs CRT começaram a entrar em desuso e portanto perderam-se muitas características essenciais para imagens em baixa resolução. Em gerações de *videogames* onde a resolução produzida era limitada pelo poder de processamento e pela memória disponível, fazia-se necessário utilizar determinados efeitos colaterais dos CRTs para a produção de alguns aspectos visuais que embelezavam os jogos e conteúdos de baixa resolução. Este projeto, portanto, pode vir a direcionar a qualidade alvo das imagens sintetizadas durante a criação de novos *shaders* ou filtros caso não haja um televisor CRT, nem imagens com qualidade suficiente disponíveis.

## 1.1 Organização do documento

O restante do presente trabalho está organizado da seguinte forma:

- **Capítulo 2 Fundamentação Teórica:** serão abordados os conceitos e técnicas que circundam o projeto;
- **Capítulo 3 Trabalhos relacionados:** serão descritos o processo e os resultados da pesquisa por referências bibliográficas;
- **Capítulo 4 Desenvolvimento:** será dissertado a metodologia desenvolvida, as etapas do modelo e um método heurístico para calibrar o modelo;
- **Capítulo 5 Resultados:** serão apresentados os resultados adquiridos no desenvolvimento do trabalho;
- **Capítulo 6 Considerações:** serão dados os pareceres finais a respeito do projeto.

## 2 Fundamentação teórica

Neste capítulo serão abordados os conceitos, ferramentas e técnicas que são essenciais para o entendimento do trabalho.

### 2.1 Representação digital de imagens em monitores

Os monitores são os dispositivos de saída em grande parte dos equipamentos eletrônicos, como celulares, *tablets*, computadores, entre outros e, ao visualizar imagens nestes dispositivos, é possível enxergar um espectro de cores numeroso, e portanto muitos podem se indagar como essas imagens são representadas. Toda e qualquer representação digital de imagens em monitores é composta por *pixels*, que é uma abreviação de “*picture element*” (do inglês elemento de imagem). Através de um aglomerado destes elementos de imagem, cria-se a ilusão de imagens, ou vídeos ao apresentar diversos quadros em sequência.

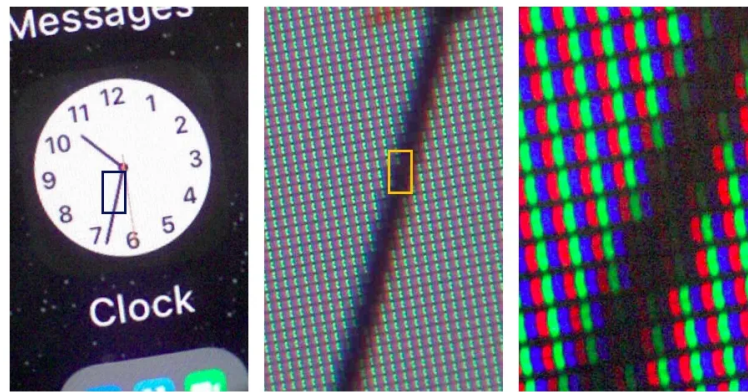
Existem três componentes fundamentais de cor em um *pixel*, denominadas de *subpixels*, os quais possuem as cores primárias de luz (aditivas) vermelho, verde e azul. Essas componentes, que podem ser referenciadas por componentes Red Green e Blue (RGB), ao serem capturadas pelos sensores da retina e processadas conjuntamente pelo cérebro humano se misturam se tornando uma única cor. Através de níveis de intensidade distintos nessas componentes, é possível se criar qualquer cor do espectro de luz visível com diferentes níveis de saturação. No entanto, os níveis de brilho e contraste variam conforme a tecnologia utilizada no monitor.

De acordo com Kandel, Schwartz e Jessell (2000), vemos que a percepção humana das cores são baseadas em três tipos de células, onde cada uma tem maior sensibilidade por faixas específicas do espectro de luz visível. Dentre os três tipos, temos os de tipo “S”, “M” e “L” que são, respectivamente, sensíveis aos comprimentos de onda menores, médios e maiores.

No sistema RGB, cada componente de cor (R, G e B) corresponde aproximadamente à sensibilidade dos diferentes tipos de células do olho humano. O vermelho (R) está mais relacionado à sensibilidade das células do tipo “L” que respondem a comprimentos de onda maiores. O verde (G) é mais perceptível para as células do tipo “M” que respondem a comprimentos de onda médios. Já o azul (B) está associado à sensibilidade das células do tipo “S”, que respondem a comprimentos de onda menores. Essa correlação entre as células do olho humano e as componentes RGB dos monitores é fundamental para a reprodução precisa das cores em dispositivos eletrônicos, permitindo uma representação visual fiel às imagens e conteúdos exibidos.

Ao aproximar-se de um monitor é possível enxergar os *pixels* individuais, como visto na Figura 2. Percebe-se que, conforme analisamos uma porção cada vez menor da tela, a cor branca deixa de existir de forma que enxergamos apenas as cores individuais das componentes.

Figura 2 – Aproximação na tela de um iPhone.



Fonte: Knoepfler (2018)

## 2.2 Monitor de tubo de raios catódicos

Como descrito em (TANNAS JR, 2012), no final do século XIX, o físico alemão chamado Karl Ferdinand Braun construiu um monitor que utilizava um canhão de elétrons, emitindo um feixe de elétrons de alta voltagem que atingia o vidro frontal do dispositivo revestido por uma camada de fósforo, a qual emitia luz onde atingida, produzindo um fenômeno conhecido como catodoluminescência. As TVs de tubo de raios catódicos foram amplamente utilizadas até serem substituídas por tecnologias mais recentes como Liquid Crystal Display (LCD), Light Emitting Diode (LED), Organic Light Emitting Diode (OLED) e por telas de plasma. Ainda assim, as televisões CRT são utilizadas em algumas partes do mundo e são muito valorizadas por colecionadores e entusiastas, pela imagem característica da tecnologia e por sua longa vida útil.

### 2.2.1 Convenções de sinais analógicos

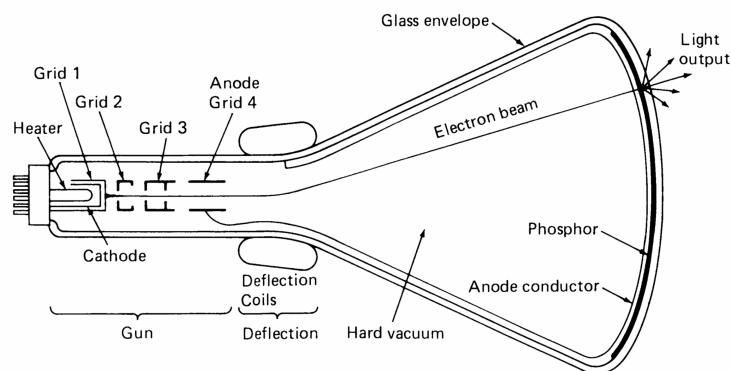
Com o fim de realizar uma padronização, foram criadas algumas convenções que definem como os sinais devem ser transmitidos e como as fabricantes deveriam projetar suas televisões para atender estas convenções. No entanto, foram criadas diversas convenções, mas dentre as principais temos National Television System Committee (NTSC) e Phase Alternating Line padrão M (PAL-M).

Em Bertozzi (2001), é possível perceber que estas convenções ditam detalhes que impactam as codificações dos sinais analógicos e de cores, cada definição tinha seu padrão de resolução SDTV. Um padrão bastante utilizado em regiões que utilizam NTSC é a resolução 480i apresentando 29,97 quadros por segundo. Já o PAL-M, que era utilizado no Brasil, utilizava a mesma resolução e frequência do padrão NTSC porém um sistema de codificação de cores distinto.

### 2.2.2 A tecnologia CRT

Os monitores de raios catódicos utilizam um tubo de vidro a vácuo e, como visto na Figura 3, dentro dele há um dispositivo que esquentava o cátodo emitindo elétrons que são acelerados pelo ânodo, gerando um feixe dessas partículas com alta tensão que denominamos de raios catódicos. Estes raios passam por uma estrutura com furos, que pode ser chamada de *slot mask*, *aperture grille* ou *shadow mask*, e atingem uma camada de fósforo presente no painel frontal de vidro do monitor, emitindo luz através de um fenômeno conhecido como catodoluminescência.

Figura 3 – Representação do tubo de raios catódicos.



Fonte: (TANNAS JR, 2012)

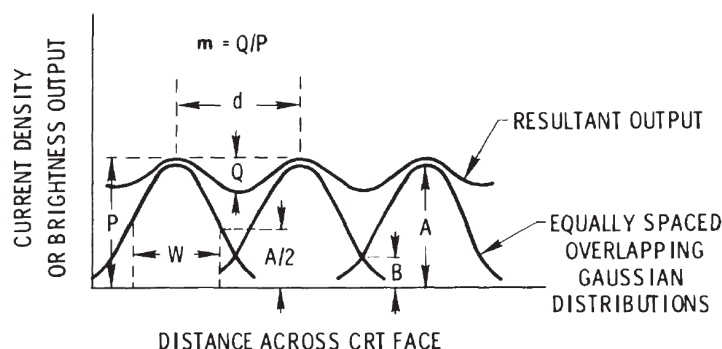
### 2.2.3 Comportamento da catodoluminescência

O fósforo contido nos pixels dos painéis de tubo de raios catódicos irradia a luz de uma forma distinta dos painéis modernos. Como visto na Figura 4, a iluminação dos *subpixels* se propaga com decaimento gaussiano, de modo que a luminosidade acaba se estendendo para os elementos vizinhos da imagem, produzindo o que ficou conhecido como “*color bleeding*”.

### 2.2.4 Scanlines

De forma geral, os monitores CRT utilizavam uma resolução padronizada SDTV, que de acordo com o padrão NTSC era uma resolução entrelaçada de 480i, ou seja, metade das linhas se intercalavam entre si a cada quadro exibido. Jogos antigos da época de consoles como o SNES e Sega Megadrive tiravam proveito destas resoluções intercaladas e utilizavam apenas metade da resolução, neste caso 240 *pixels*. As *scanlines* são então, as linhas de varredura que o canhão de elétrons percorre. No entanto, normalmente são referenciadas pelos entusiastas de videogame como sendo as linhas escuras que aparecem em consoles antigos. O brilho do fósforo das linhas varridas refletiam para as linhas escuras, e alguns artistas que produziam jogos nesta época costumavam construir suas

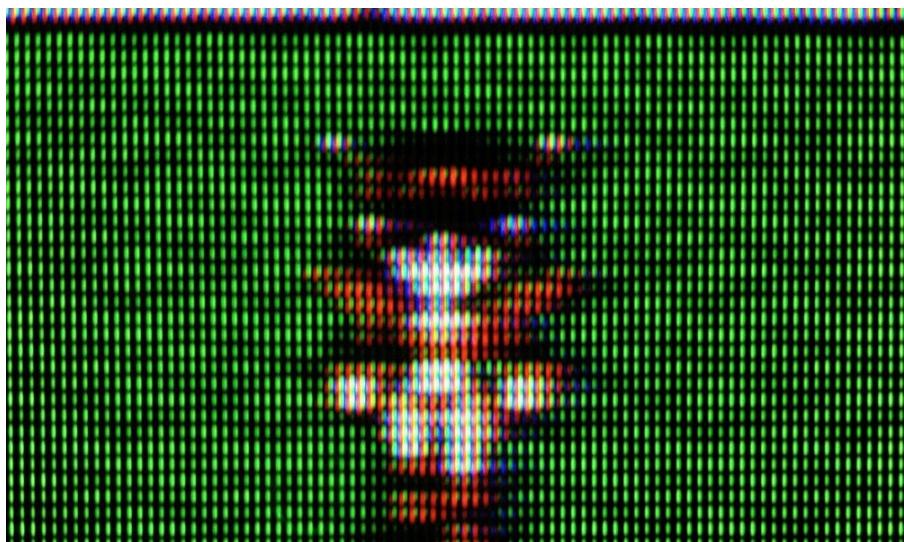
Figura 4 – Distribuição de intensidade espacial Gaussiana de luminosidade ao longo de uma tela de CRT.



Fonte: (TANNAS JR, 2012)

artes tirando proveito deste efeito. Além disso, usando apenas metade das linhas de varredura era possível reproduzir imagens com o dobro da taxa de atualização, dado que os monitores produziam um quadro em metade do tempo. Complementarmente, os consoles precisariam processar uma menor resolução, permitindo que o poder computacional fosse alocado para outros fins, caso desejado. Veja por exemplo na Figura 5, onde podemos perceber as linhas de varredura com lacunas entre si.

Figura 5 – Sprites do jogo “The Guardian Legend” para Nintendo Entertainment System.



Fonte: Adaptado de CRTpixels (2022)

### 2.3 Processamento de imagens

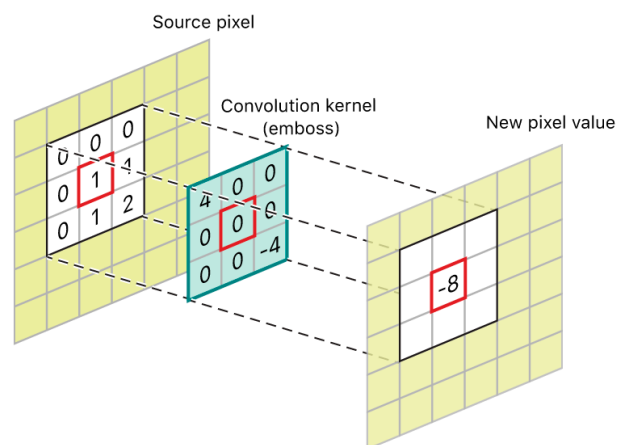
De acordo com Gonzalez e Woods (2007), a área de processamento de imagens digitais se refere a processar imagens digitais utilizando computadores digitais. Portanto refere-se à manipulação de imagens, seja para otimizar, bem como também para extrair

informações. Existem inúmeras aplicações do processamento de imagens na computação, como a restauração de fotografias antigas, ajuste de cores, aplicação de filtros de desfoque, aumento da nitidez, aumento da escala de resolução, etc.

### 2.3.1 Aplicação de filtros

A aplicação de filtros em uma imagem é um dos principais processos do processamento de imagens. Nele produzimos uma imagem de saída com características particulares a partir de uma entrada. Uma forma muito comum de aplicar um filtro é através da criação de uma matriz de convolução, que chamamos de *kernel*, com a qual passamos por cada *pixel* para calcular seu valor de acordo com os valores da matriz, os valores do pixel em destaque e de seus vizinhos, como visto na Figura 6. Existem diversas aplicações de filtros, como para detecção de bordas, desfoque, correção de cores, etc.

Figura 6 – Exemplo da utilização de matrizes de convolução.

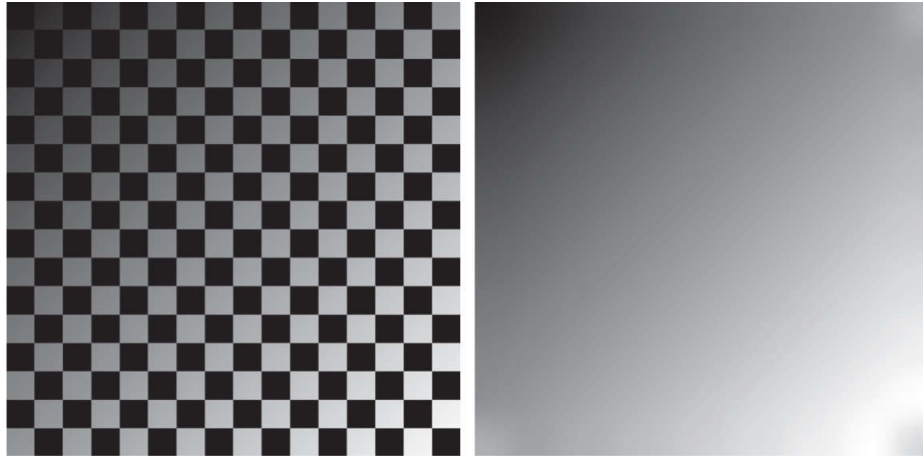


Fonte: Inc. (2023)

#### 2.3.1.1 Desfoque Gaussiano

O desfoque gaussiano é um filtro muito comum no processamento de imagens, obtido através da aplicação de uma matriz de convolução que gera uma aparência suavizada na imagem. Sua aplicabilidade é abrangente e pode ser muito útil, por exemplo, quando se deseja reduzir o ruído em uma imagem, suavizar bordas serrilhadas, ou até mesmo remover rugas e espinhas de um retrato. Seu funcionamento se dá pela utilização de um decaimento gaussiano, processando a imagem de forma a ponderar cada *pixel* de acordo com a sua vizinhança. Um exemplo do efeito deste filtro está representado na Figura 7, onde é possível perceber que o padrão xadrez, visto à esquerda, é suavizado de forma a se misturar formando um degradê, visto à direita.

Figura 7 – Exemplo da aplicação do desfoque gaussiano.



Fonte: Gonzalez e Woods (2007)

### 2.3.2 *Upscaling*

Segundo Bates (2019), o termo *upscaling* se refere ao aumento da definição de uma imagem ou vídeo, de forma que a partir da sua resolução original, seja possível atingir uma imagem com maior contagem de *pixels*. Existem diversos motivos para serem utilizados métodos de *upscaling*, como por exemplo visualizar conteúdos arcaicos com maior nitidez, aumento do desempenho de aplicações em tempo real sendo visualizadas em monitores de alta resolução, aumentar a qualidade de imagens comprimidas, impressão em grandes formatos, etc.

## 2.4 Testes estatísticos

Com base nos conceitos vistos em (WASSERMAN, 2004) é afirmar que testes estatísticos são instrumentos fundamentais no campo da análise de dados, proporcionando a capacidade de fazer grandes inferências a partir dos dados coletados e testar hipóteses formuladas. A escolha do teste estatístico apropriado depende de vários fatores, incluindo o tipo de dados que se está lidando e as perguntas específicas que se deseja responder. Um teste estatístico comumente utilizado para comparar distribuições é o teste Kolmogorov-Smirnov, que será abordado na seção seguinte.

### 2.4.1 Kolmogorov-Smirnov

O teste Kolmogorov-Smirnov (KS) é uma ferramenta estatística poderosa que tem por objetivo comparar duas distribuições de probabilidade. É comumente usado para avaliar a aderência de uma distribuição amostral a uma distribuição teórica, ou para comparar duas distribuições amostrais (JR, 1951).

A premissa básica do teste KS é examinar a diferença máxima entre as funções de distribuição acumulada das duas distribuições que estão sendo comparadas. Para uma

dada distribuição, a diferença é calculada para cada ponto de dados e apenas o valor máximo é considerado como a estatística do teste KS.

## 2.5 Algoritmos Genéticos

Visto em Luke (2013), algoritmos de meta-heurística aplicam um grau de aleatoriedade para a busca de soluções que sejam mais próximas da ótima possível. Os algoritmos genéticos são métodos de busca meta-heurística inspirados no conceito de evolução. Para encontrar uma solução, é gerada uma população de soluções que se cruzam e sofrem mutações na sua essência, sendo pontuadas através de uma função de custo. Então, através da seleção natural, as soluções com maior pontuação são selecionadas.

## 2.6 Linguagens de programação adotadas

As linguagens de programação são amplamente utilizadas como uma forma de expressar em texto as instruções que serão executadas pelos computadores. Esta seção se dedica a descrever as linguagens adotadas no projeto.

### 2.6.1 C++

Dentre as linguagens de programação, C++ é uma das linguagens orientadas a objeto mais famosas no âmbito de programação. Ela foi produzida por Bjarne Stroustrup em 1983 e, como visto em Stroustrup (1985), é uma linguagem compilada, multi-paradigma e de propósito geral. Ela possui muita similaridade com a linguagem C, justamente por ter sido projetada para ser sua sucessora e, assim como sua predecessora, possui gerenciamento de memória manual. Portanto, dada a sua natureza compilada, é uma ótima linguagem para a produção de programas com o foco em desempenho e baixo custo de memória.

#### 2.6.1.1 OpenCV

O Open Source Computer Vision (OpenCV) é uma biblioteca de código livre para visão computacional e processamento de imagens. A biblioteca se encontra disponível para C++, Java e Python, e conta com inúmeras funções otimizadas para manipulação e processamento de imagens, reconhecimento de padrões, rastreamento de objetos, etc. De acordo com (BRADSKI; KAEHLER, 2017), a biblioteca inicialmente foi desenvolvida por um funcionário Intel, mas atualmente esta sendo mantida por uma organização sem fins lucrativos<sup>1</sup>.

---

<sup>1</sup> [OpenCV] Open Source Computer Vision Library (OpenCV) (free, BSD license), <http://opencv.org>.

## 2.6.2 Python

Entre as linguagens de programação, Python é uma das linguagens mais versáteis e populares atualmente. Python foi criada por Guido van Rossum e lançado em 1991 e, conforme explicado em Foundation (2023), é uma linguagem interpretada, multi-paradigma e de propósito geral. Python suporta tanto a programação orientada a objetos quanto a programação procedural, além de incluir recursos de linguagens de *script*.

### 2.6.2.1 PyGAD

Vemos em Gad. (2023), uma biblioteca Python flexível e eficiente para a criação de algoritmos genéticos, chamada PyGAD. Ele lida com a técnica heurística, conhecida como Algoritmo Genético (AG), que se inspira no processo de seleção natural. Frequentemente, os algoritmos genéticos são empregados para desenvolver soluções otimizadas para questões de busca e otimização.

### 3 Trabalhos Relacionados

#### 3.1 Protocolo de pesquisa

Neste capítulo será apresentado o caminho utilizado para a pesquisa de literatura e material de referência, junto a seus resultados. O objetivo da pesquisa foi encontrar referências no tópico de simulação dos efeitos dos monitores CRT, bem como formas alternativas de reprodução de imagens de baixa resolução em monitores de alta resolução. Foram utilizados mecanismos de pesquisa como o Google, e ferramentas acadêmicas, como o Google Scholar e IEEE Explorer. Para a seleção dos trabalhos foram utilizadas as seguintes *strings* de busca:

- (pixel art) AND (upscaling OR magnifying)
- (crt) AND (filter OR shader OR simulation OR emulation OR physics)

Baseado nos resultados da pesquisa foram selecionados artigos com enfoque em *pixel art*. Para serem considerados, os trabalhos deveriam tentar reproduzir imagens de baixa resolução em monitores modernos seja através da ampliação de sua resolução ou replicando os efeitos que os monitores de tubo de raios catódicos produziam. Trabalhos que não estavam disponíveis para *download* nem mesmo através do *proxy* da UNIPAMPA foram desconsiderados, assim como aqueles que não possuem versões em português nem inglês.

#### 3.2 Resultados das pesquisas

##### 3.2.1 Ampliação da resolução de *pixel arts*

Em Stepin (2003), vemos que o HQx é um método de aumento de resolução desenvolvido para melhorar a aparência de artes de baixa resolução preservando detalhes da entrada que em algoritmos de aumento de resolução comuns seriam perdidos. Este método possui um algoritmo similar para suas variantes, que produzem imagens com aumento em duas, três e quatro vezes na escala de resolução. O algoritmo realiza uma série de interpolações ao percorrer cada *pixel*, considerando seus valores de cores e os de seus vizinhos, demarcando-as como perto e distante de acordo com uma limiar pré-definida. Em seguida, aplica-se uma tabela de consulta que define como as intensidades de cores de cada pixel serão misturadas de acordo com seus vizinhos baseada na demarcação criada.

Temos em xBR (2012) um método que utiliza-se de técnicas similares ao HQx para atingir uma ampliação de resolução, mas expandindo as regras para melhores resultados. No processo, é utilizado um algoritmo de detecção de bordas pensado pelo próprio autor que indica se é ou não uma borda. Com a informação da borda, é realizado um redimensionamento através do conjunto de duas regras de interpolação. Este procedimento é

reproduzido para cada elemento da imagem e possui um custo computacional maior que o do algoritmo HQx, por exemplo.

O método visto em Mazzoleni (2001) é novamente muito similar ao HQx, e portanto ao xBR, onde para cada elemento da imagem utiliza-se dos *pixels* próximos para redimensionar sua resolução de acordo com regras pré-definidas. No entanto, o autor do site menciona que seu método possui um melhor desempenho do que os métodos citados.

Em Kopf e Lischinski (2011), demonstram um método que realiza a vetorização para ampliar as artes de baixa resolução. O método criado produz representações da entrada que pode ser portada para qualquer resolução com alta nitidez. Para isto, cada pixel da imagem é redimensionado e convertido em pontos (ou vetores) e posteriormente é realizada uma interpolação entre estes pontos utilizando curvas *spline*. Além disso, o algoritmo produz uma otimização destas curvas que produzem resultados muito mais concisos para *pixel arts* em comparação com outros algoritmos de vetorização, dado ser um algoritmo especializado.

### 3.2.2 Simulação dos efeitos fenomenológicos

Em Pittman (2015), Kyle Pittman (desenvolvedor independente) incorporou uma simulação de monitores CRT para monitores modernos. O desenvolvedor comentou que dada a necessidade de desenvolver um jogo que tivesse bom desempenho acabou optando por realizar uma aproximação que fosse agradável em visual, mas deixou de lado a sua aproximação física dada as implicações na carga de processamento gerada pela simulação. Ele utilizou uma série de técnicas para seu filtro, como a utilização de uma máscara para os *pixels*, uma tabela de *look-up* para produzir as cores do padrão NTSC a partir dos valores RGB, uma técnica para realçar as bordas e, além disso, projeta e distorce a imagem em um modelo matemático tridimensional que simula as curvas do vidro presente nos monitores de tubo.

Visto em 4loup (2017), *CRTify* é um pacote de *assets* que simula os efeitos de monitores de tubo, desenvolvido para o motor gráfico da *Epic Games* denominado Unreal Engine (versão 4). Este pacote conta com modelos tridimensionais e materiais para representar o visual de um monitor de tubo de raios catódicos em um ambiente virtual 3D. Estes materiais possuem texturas e *shaders* que, ao serem aplicados a um objeto tridimensional ou bidimensional, tornam possível simular diversos aspectos e distorções do CRT em tempo real. Dentre tais efeitos temos, por exemplo efeitos de distorção da curvatura do vidro e de cores causadas por interferências magnéticas. Os *shaders* deste pacote também permitem alterar a geometria dos *subpixels* do monitor, simulação das linhas de varredura, chiado e etc.

Através de TroogleMonkey (2014) encontramos o *CRT-Royale*, um *shader* de código aberto para o software de emulação *retroarch* com alto poder de personalização que simula vários aspectos dos monitores de tubo de raios catódicos. Através de um arquivo de

configuração, este filtro permite modificar a aparência dos *subpixels*, a geometria da tela, efeito de halo, nível da gaussiana gerada pela iluminação no vidro, etc. Além disso, existem diversas variantes do *CRT-Royale* que entregam certos aspectos prontos para serem utilizados sem ser necessário nenhuma configuração. Um exemplo disso é o *CRT-Royale NTSC* que apresenta efeitos do padrão NTSC.

## 4 Desenvolvimento

Neste capítulo, demonstramos os passos para o desenvolvimento de um modelo fenomenológico para produzir imagens de alta resolução simulando as características de um monitor CRT, muito similar a diversos *shaders* que procuram atingir o mesmo resultado, porém com o foco em reproduzir os efeitos de forma fidedigna e baseada nos impactos físicos da tecnologia de raios catódicos. Ao final desta seção discutimos uma abordagem para calibrar os parâmetros do modelo.

### 4.1 Ferramentas utilizadas

Foi realizada uma análise de ferramentas que iriam compor o projeto sendo levado em conta uma menor complexidade para o desenvolvimento de forma que tenha um desempenho que permita que os testes levem um tempo inferior a uma hora, dado que serão realizados diversos cálculos e a iteração sobre os elementos das imagens de entrada. Após uma breve análise das linguagens de programação para implementação foi escolhido o C++, uma linguagem orientada a objetos compilada que entrega um ótimo desempenho, mas também permite o encapsulamento e, portanto, a abstração de conceitos a fim de permitir que o código fonte seja incrementado com futuras melhorias.

Com o objetivo de facilitar a implementação do modelo, foram aproveitados recursos de processamento e manipulação de imagens da biblioteca OpenCV. Além de facilitar, esta biblioteca tem funções otimizadas para a aplicação de filtros que permitem que os esforços sejam concentrados em aperfeiçoar apenas o nosso modelo. Adicionalmente, a biblioteca também provê métodos de redimensionamento de resolução.

Além disso, usamos a linguagem de programação Python por sua simplicidade e vasta gama de bibliotecas científicas. Usamos a biblioteca PyGAD para a implementação do algoritmo genético voltado para a calibração do modelo, devido à sua flexibilidade e facilidade de uso. E para a métrica KS utilizamos a biblioteca SciPy.

### 4.2 Modelo proposto

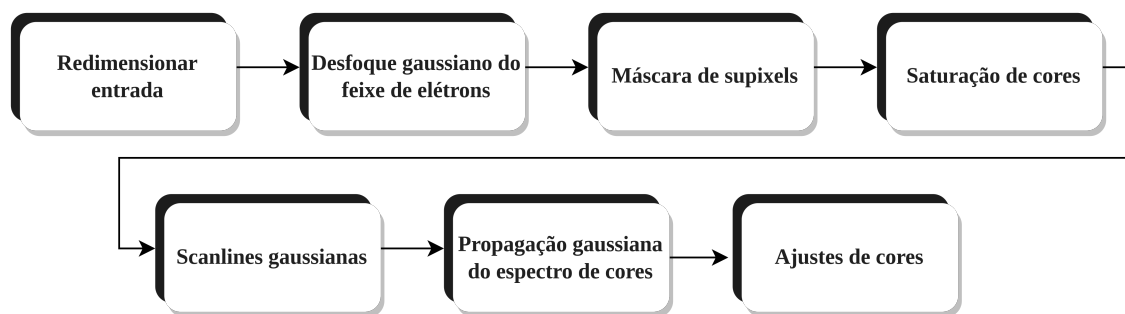
A abordagem ótima para reproduzir a imagem de dispositivos CRT com máxima acurácia física seria simular fisicamente todos os processos de eletroluminescência, ou talvez até realizar uma aproximação utilizando um algoritmo de traçado de raios, no entanto o custo para calibrar os parâmetros e até gerar os quadros não seria aplicável para o uso prático. Portanto, o objetivo principal deste trabalho é produzir um modelo matemático baseado nos efeitos físicos para que, a partir de uma imagem de baixa resolução como entrada, reproduza os efeitos vistos em monitores CRTs. Para isto, iremos modelar o processo envolvido nesses monitores, como o impacto da varredura do feixe de elétrons ao exibir uma imagem em 240p em monitores de resolução interlaçada (480i), e também a interação com a camada de fósforo junto a seus reflexos internos.

Considerando que existem inúmeros tipos de monitores, cada um com características como cores, nitidez e geometrias dos *pixels* distintas, nosso modelo é parametrizado para que seja possível calibrá-lo através de um conjunto de parâmetros como entrada.

#### 4.2.1 Etapas do modelo

O modelo produzido é capaz de produzir imagens de alta resolução a partir de uma imagem de baixa resolução do tipo *pixel art*. Para isso foi construído um pipeline de processamento que está demonstrado na Figura 8. Em sequência, demonstraremos os processamentos realizados para atingirmos o objetivo deste trabalho.

Figura 8 – *Pipeline* do modelo proposto.



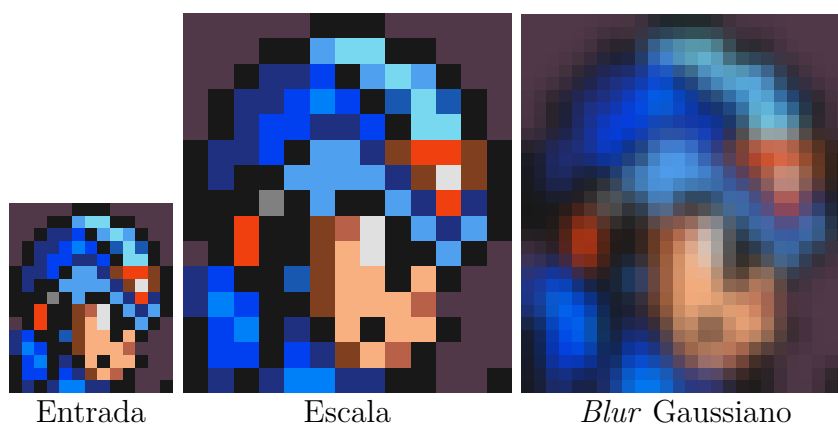
Fonte: O autor

O primeiro passo realizado é redimensionar a escala da imagem de entrada em duas vezes, sem utilizar nenhum método de interpolação. Queremos apenas aumentar a contagem de pixels, para que seja possível aplicar o efeito gerado pela resolução interlaçada, onde as imagens são representadas pela metade da resolução SDTV gerando as “*scanlines*”. Em seguida, no segundo passo, aplicamos um desfoque gaussiano para gerar o efeito causado da falta de nitidez do raio gerado pelo canhão de elétrons. Este passo nos ajuda também a gerar o vazamento das intensidades para a vizinhança, o que acontecia em CRTs ocasionado tanto pelo desfoque do feixe, quanto pelo sinal analógico das conexões de vídeo. Essas etapas estão demonstradas na Figura 9.

No terceiro passo, as intensidades de vermelho, verde e azul de cada *pixel* que forma a imagem são mapeadas para uma textura que geramos baseado na resolução desejada para cada *pixel*, um exemplo está demonstrado na Figura 10. Essa textura, desenvolvemos utilizando 4 pontos de controle para cada *subpixel*, formando uma estrutura retangular que são ligadas através o algoritmo “Catmull-Rom”, que utiliza *B-splines* visto em (CATMULL; ROM, 1974). Devido ao fato dos elementos de uma imagem vista através de monitores CRTs não se disporem em um *grid* quadrado, como em monitores modernos, ao realizar o mapeamento foi necessário dividi-los pela metade nas colunas ímpares, como visto na Figura 11, para a obtenção da geometria hexagonal.

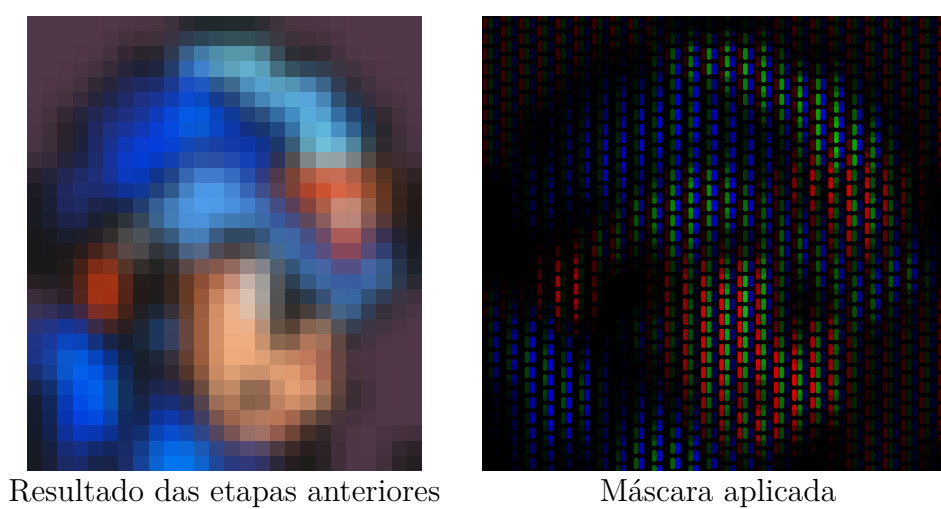
Em seguida, na quarta etapa, nós processamos as intensidades de cada *subpixel*, levando em conta que cada componente de um único *pixel* da imagem de entrada terá

Figura 9 – Demonstração da primeira e segunda etapa



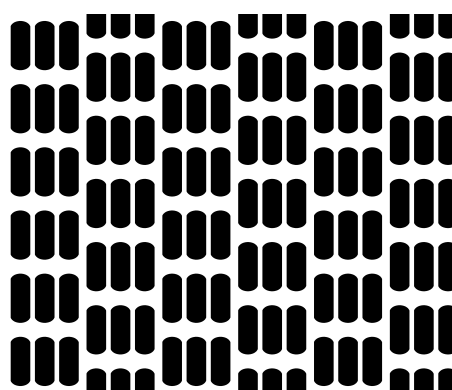
Fonte: O autor

Figura 10 – Demonstração da terceira etapa



Fonte: O autor

Figura 11 – Exemplo do formato da máscara gerada pelo modelo.



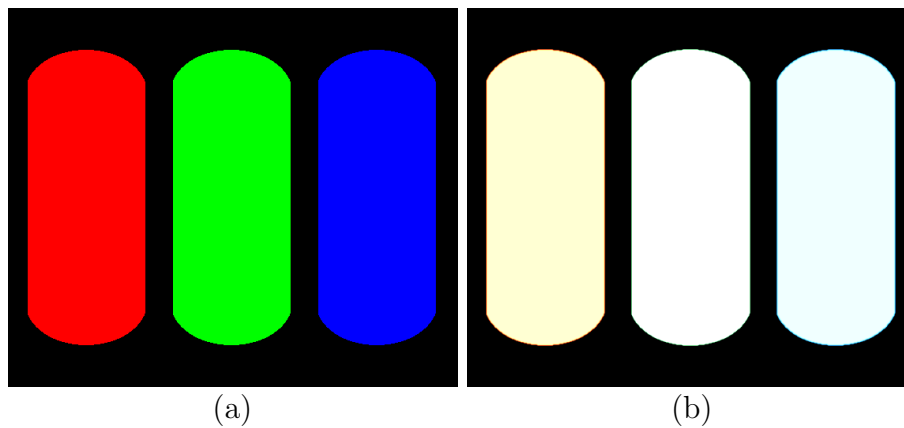
Fonte: O autor

uma representação sintética utilizando as três componentes de cores, permitindo que assim seja gerado o efeito onde o fósforo verde se torna branco. Este procedimento está representado nas equações 4.1 e 4.2, onde  $s$  é o peso logístico resultante da intensidade  $v_c$  do componente do *subpixel* ( $c = \text{vermelho, verde ou azul}$ ),  $v_{c0}$  age como um parâmetro que ajusta a inclinação da curva logística para a esquerda ou direita, enquanto que  $v_{coutput}$  é a intensidade da componente do *subpixel* com a correção. As variáveis  $\sigma_{1c}$  e  $\sigma_{2c}$  estabelecem os estados da distribuição espectral de cada componente do *subpixel*, dado que suas cores não são completamente puras em intensidades elevadas, progredindo para uma coloração quase branca em intensidades máximas.  $\alpha_c$  incorpora a amplificação da transição logística. Um exemplo do resultado desse tratamento está na Figura 12. Além disso, em casos onde as duas metades do mesmo *pixel* possuem cores distintas é realizada também uma interpolação logística entre as cores, um exemplo de comparação pode ser visto na Figura 13.

$$s = \frac{1}{1 + e^{-\alpha_c \times (v_c - v_{c0})/255}} \quad (4.1)$$

$$v_{coutput} = \min(v_{cmin} + (1 - s) \times (\sigma_{1c} \times v_c) + (s \times \sigma_{2c} \times v_c), 255) \quad (4.2)$$

Figura 12 – Demonstração da saturação do fósforo

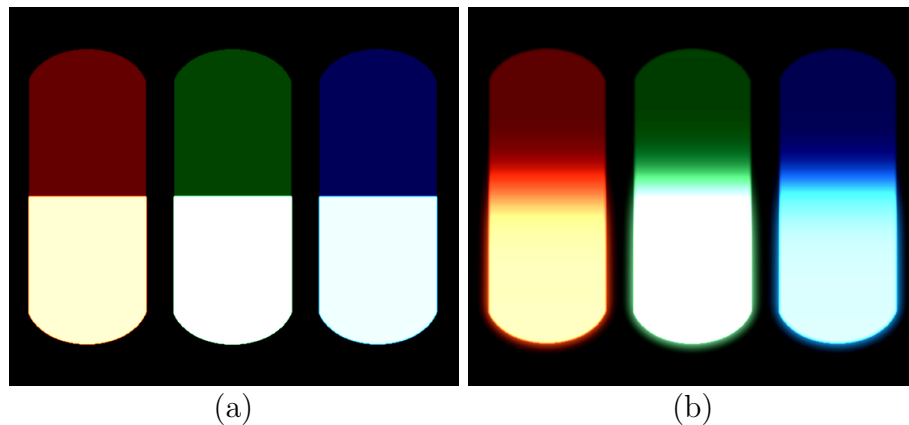


Fonte: O autor

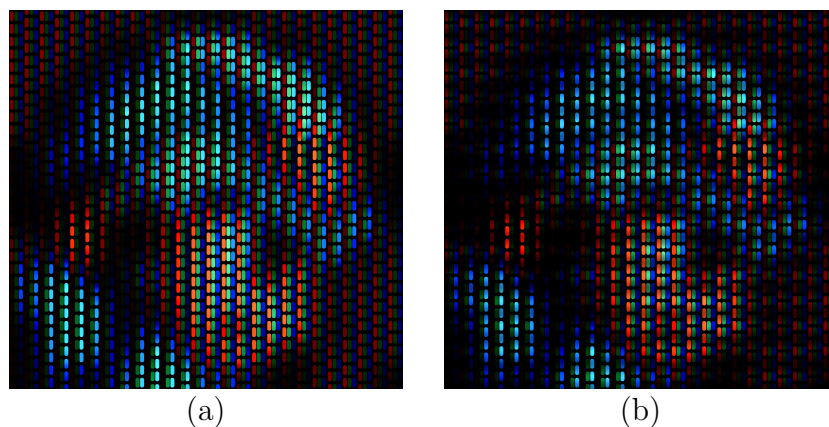
No quinto passo, produzimos o efeito das linhas de varredura, utilizando um decaimento gaussiano. Consideramos as intensidades de cada *pixel* para o escurecimento das linhas. Desse modo, os *pixels* com maior iluminação mantêm sua intensidade, enquanto os componentes mais escuros tendem a se apagar completamente. Um exemplo da aplicação deste efeito está apresentado na Figura 14, onde (a) mostra o *pixel* sem o efeito, e em (b) após o processamento das *scanlines*.

Na sexta etapa, emulamos a propagação das cores do fósforo, fenômeno que ocorre em consequência da reflexão interna no vidro. Fazendo uso da máscara exibida na Figura

Figura 13 – Demonstração do tratamento logístico



Fonte: O autor

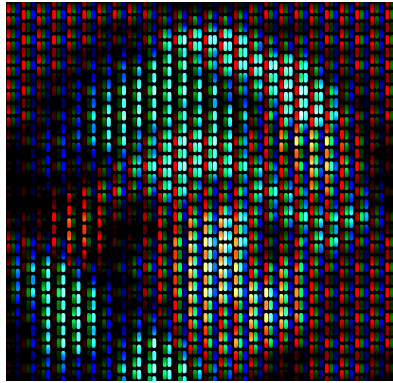
Figura 14 – Demonstração das *scanlines*

Fonte: O autor

11, que demarca as áreas desprovidas de fósforo, aplicamos um desfoque gaussiano à imagem produzida na etapa anterior. Então, selecionamos apenas a região correspondente à máscara, preservando a área dos fósforos da imagem original.

No sétimo e último passo, aplicamos alguns ajustes globais de cor à imagem usando funcionalidades da biblioteca OpenCV. Este procedimento representado na Figura 15, tem como objetivo otimizar os resultados dos processamentos anteriores, permitindo-nos modificar especificamente o contraste e a saturação da imagem. Quanto ao brilho, adicionamos uma constante a todas as componentes de cor da imagem, garantindo que nenhuma intensidade exceda o valor máximo permitido. Dessa maneira, aprimoramos a qualidade geral da imagem, garantindo um resultado final mais realista e visualmente agradável.

Figura 15 – Modelo com ajustes globais na coloração



Fonte: O autor

#### 4.2.2 Método de calibração utilizando algoritmos genéticos

Na abordagem que adotamos para calibrar os parâmetros do modelo, consideramos unicamente as distribuições cumulativas de cores das imagens. Utilizamos essa distribuição para executar a métrica do teste de KS, utilizando-a como meio de comparar as imagens, independente de formato e ângulo, o que é um fator relevante principalmente considerando que as imagens de referência são fotografias retiradas de monitores CRT.

Contudo, basear-se exclusivamente no espectro de cores poderia conduzir a resultados paradoxalmente opostos aos esperados, ainda que com níveis de cores corretos e, portanto, decidimos implementar o uso de  $n$  imagens. Estas são selecionadas por sua relevância para partes específicas do processo de calibração. Inicialmente, focamos nos *pixels* individuais R, G e B, assim como em uma combinação dos três em uma única imagem, além de mantê-los separadamente, conforme ilustrado na Figura 16.

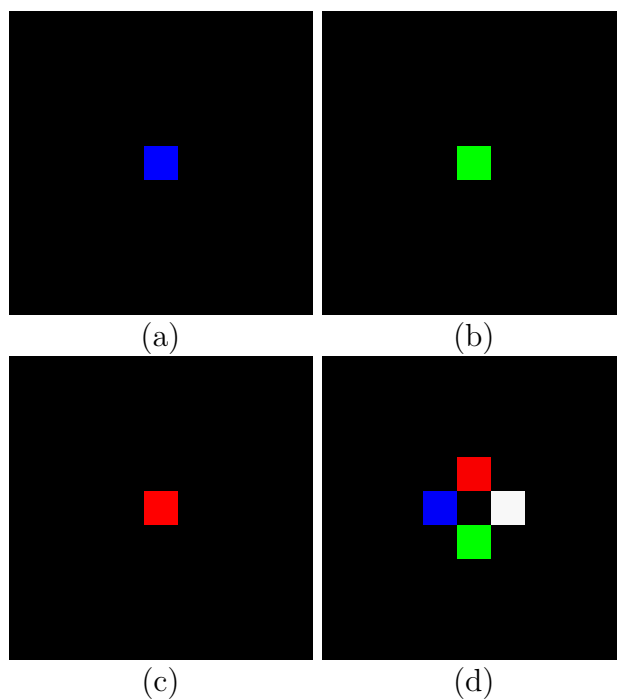
Para a função de pontuação, somamos os resultados individuais das comparações entre a imagem fotografada (conforme Figura 17) e as imagens sintetizadas (conforme Figura 18).

A soma dos “KS” é calculada de acordo com a seguinte equação:

$$\Theta = \sum_{i=1}^n KS(\Phi_i, \Psi_i) \quad (4.3)$$

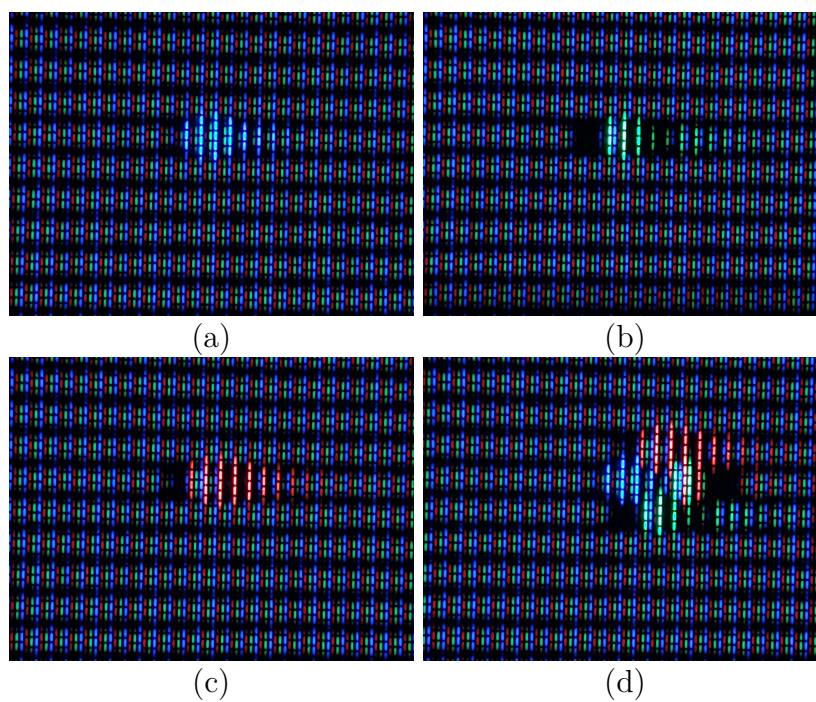
Nessa equação,  $\Theta$  é a pontuação total,  $KS$  é a função que calcula a estatística de Kolmogorov-Smirnov,  $\Phi_i$  é a  $i$ -ésima distribuição cumulativa de cores da imagem fotografada e  $\Psi_i$  é a  $i$ -ésima distribuição cumulativa de cores da imagem sintetizada.

Figura 16 – Exemplo de *pixels* de entrada usados na calibração

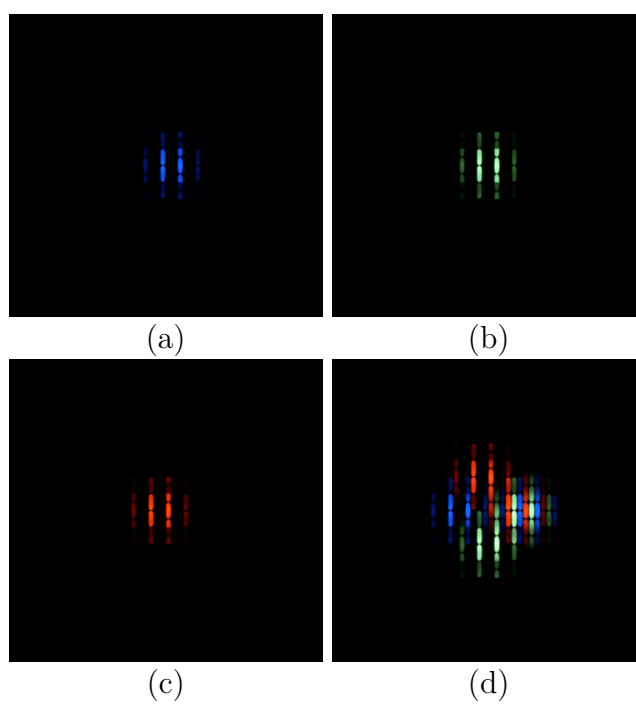


Fonte: O autor

Figura 17 – Exemplo de fotografias dos *pixels* usados na calibração



Fonte: O autor

Figura 18 – Exemplo de imagens sintetizadas dos *pixels* durante a calibração

Fonte: O autor

## 5 Resultados

Neste capítulo, demonstramos os resultados obtidos no desenvolvimento deste projeto. Foi possível obter resultados que cumprem o principal objetivo deste trabalho, e de forma empírica, evidenciamos a qualidade das imagens sintéticas utilizando imagens de um televisor real.

### 5.1 Evidências empíricas de bons resultados

Com o filtro criado, a partir de uma imagem de entrada em baixa resolução são aplicados os passos de processamento, resultando em uma imagem sintética que simula os efeitos das TVs CRT. A fins de comparação, selecionamos duas imagens que capturamos de um televisor e, com o filtro criado, foram sintetizadas imagens a partir de capturas de tela de emuladores sem nenhum processamento dos jogos Turrigan 2: The Final Fight para Commodore Amiga e Metal Gear Solid 2 de MSX.

Utilizando a Figura 19 como entrada foi possível gerar as imagens da Figura 20 (b) e Figura 20 (c). Nesta Figura, temos na Figura 20 (a) uma fotografia de um monitor CRT, sendo a Figura 20 (b) uma imagem sintetizada através de nosso modelo, e na Figura 20 (c) uma utilizando o *shader* CRT-Royale. Nessa Figura vemos a comparação de capturas de tela levemente diferentes, mas que baseando-se no personagem é possível analisar a semelhança entre elas.

Ao analisarmos a Figura 20 (c) em comparação com a Figura 19, sem processamento algum, notamos uma melhoria significativa em termos visuais, porém quando comparamos a Figura 20 (a), vemos que faltam determinados detalhes. Observando detalhes na armadura e arma do personagem na imagem usada como entrada, percebemos que não há tanto detalhamento. Já na imagem sintetizada na Figura 20 (b), é possível notar os braços do personagem segurando uma espécie de “arma alienígena”, assim como na imagem fotografada visto na Figura 20 (a).

Através da Figura 21, geramos novamente duas imagens em suas representações CRT, como também fotografamos a reprodução através de um monitor CRT. Vemos estas imagens na Figura 22, onde na Figura 22 (a) temos a fotografia do tubo de raios catódicos, na Figura 22 (b) uma imagem sintetizada utilizando nosso filtro e na Figura 22 (c), o *shader* “CRT-Royale”.

Temos uma comparação similar à anterior, onde novamente é visível a semelhança entre as três imagens. Se voltarmos nossa atenção aos olhos do personagem, percebemos que são representados por apenas um *pixel* branco e traços do seu rosto e testa não possuem detalhes discerníveis na imagem de entrada. Na Figura 22 (c) vemos uma melhoria significativa mas, novamente, existe a falta de maior detalhamento nos efeitos, como também na resolução. Em contrapartida, na Figura 22 (b) é possível perceber que apresenta bastante semelhança com Figura 22 (a), com olhos arredondados e maior detalhamento gráfico, percebendo-se regiões elevadas em sua testa.

No entanto, vale notar que a calibração ideal do brilho e do contraste pode ser aprimorada nas imagens geradas por nosso modelo.

Foram criadas algumas animações utilizando nosso modelo que demonstram como se comportaria em tempo real. Para gerar as animações aplicamos o nosso modelo em cada *frame*, juntando as imagens geradas em uma animação final. A coleção dessas animações podem ser conferidas no repositório científico *figshare*, através de Tamiosso (2023).

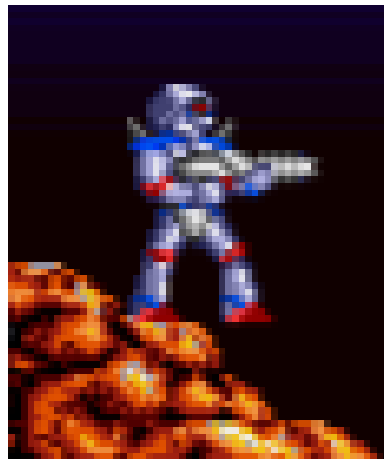
A seguir demonstramos mais alguns resultados variando alguns parâmetros do nosso filtro. Na Figura 23 temos o desfoque gaussiano com parâmetros distintos. Já na Figura 24 temos uma aceleração na função de saturação do fósforo, onde é possível perceber que em (b) a parte azul do mar já sofreu a saturação, enquanto que em (a) a cor ainda se mostra pura no *background*.

## 5.2 Resultados do método de calibração

Durante o processo de calibração, nosso método demonstrou um progresso notável nos estágios iniciais, convergindo para um conjunto de parâmetros que oferecia uma representação colorimétrica substancialmente melhor. As cores de cada componente tornaram-se um pouco mais precisas em comparação com os resultados iniciais onde, por exemplo, a cor verde estava sendo representada como roxa e a cor vermelha como rosa.

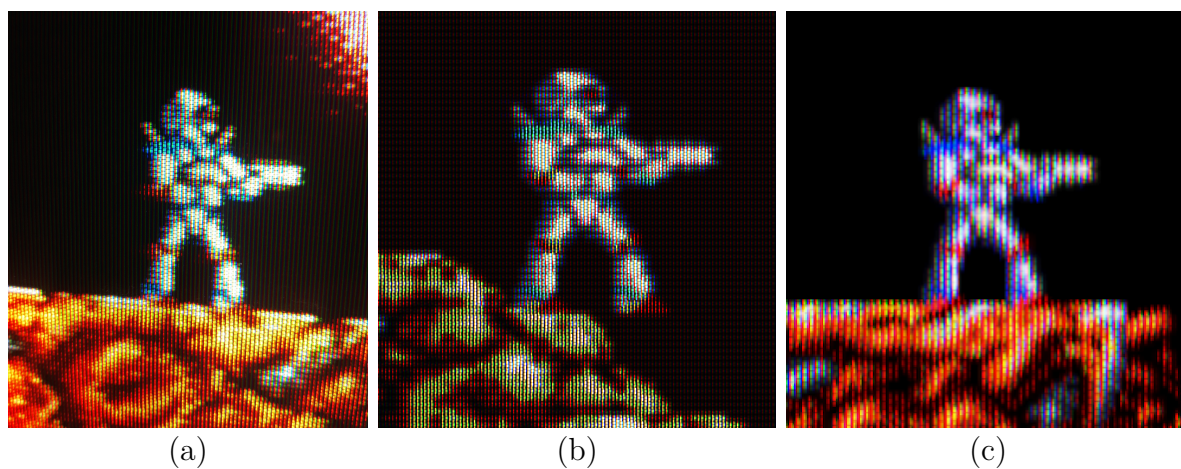
No entanto, ao atingir um certo ponto nessa progressão, o algoritmo estagnou. Apesar da cor azul estar sendo reproduzida corretamente desde o início e das outras cores terem se ajustado significativamente, os avanços subsequentes foram mínimos. Essa estagnação, um fenômeno comum em algoritmos genéticos quando chegam a um ótimo local, indicou que havíamos alcançado o limite da capacidade do nosso algoritmo de calibrar o modelo de forma eficaz. Isso sugere que talvez seja necessário reavaliar a estrutura do nosso algoritmo genético e as imagens utilizadas para fins da calibração.

Figura 19 – *Sprite* do game *Turrican 2: The Final Fight* para Commodore Amiga.



Fonte: LemonAmiga (2004)

Figura 20 – Comparação entre imagem CRT (a), imagem CRT (b) sintetizada com nosso modelo e (c) sintetizada com *shader* CRT-Royale, do *Turrican 2: The Final Fight* para Commodore Amiga.



(a)

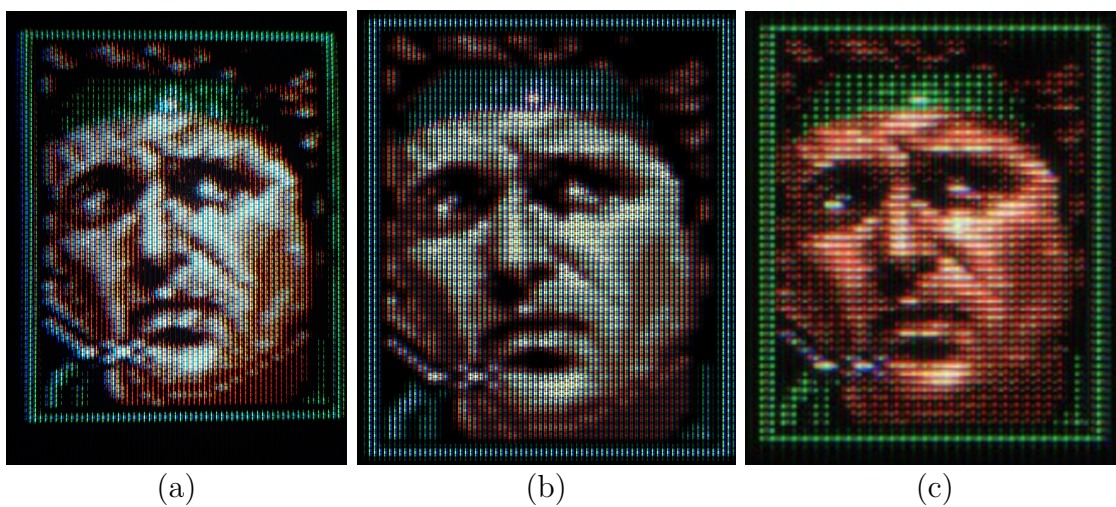
(b)

(c)

Fonte: O autor

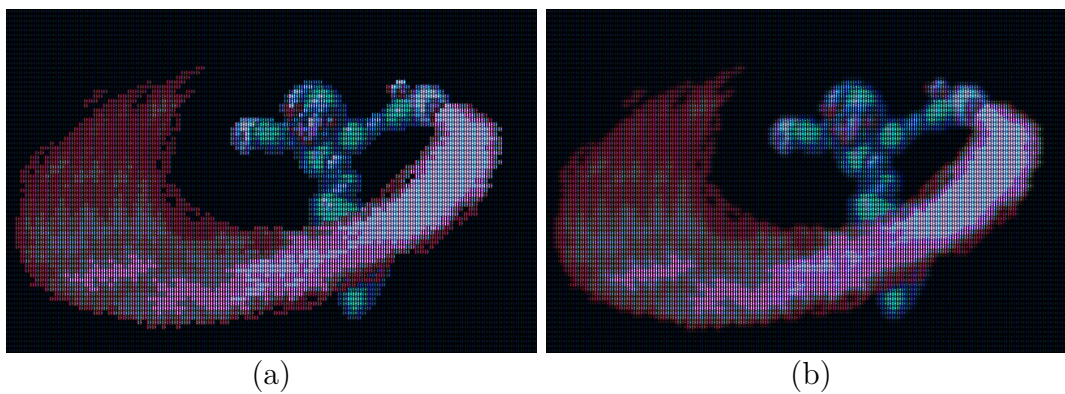
Figura 21 – *Sprite* do game Metal Gear Solid 2 para MSX.

Fonte: MyAbandonware (2012)

Figura 22 – Comparação entre imagem CRT (a), imagem CRT sintetizada (b) sintetizada com nosso modelo e (c) sintetizada com *shader* CRT-Royale, do Metal Gear Solid 2 de MSX.

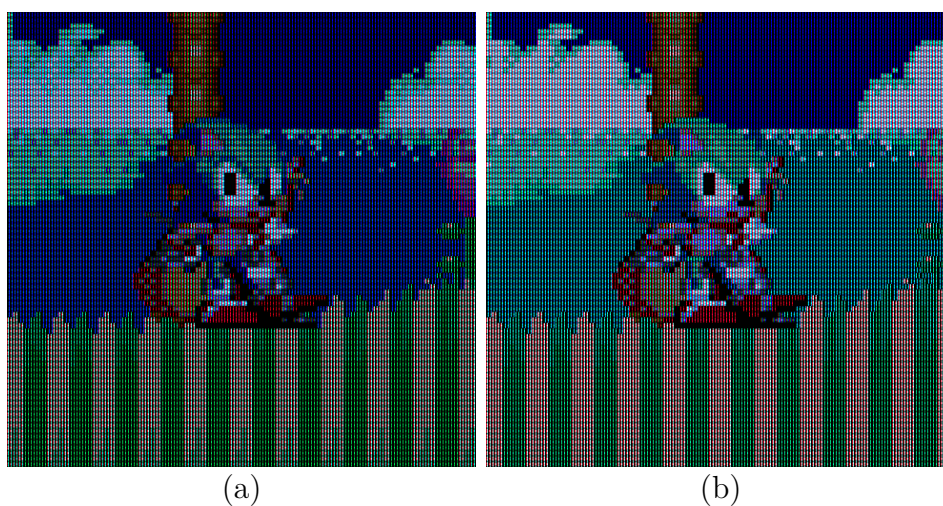
Fonte: O autor

Figura 23 – Exemplos de variação no desfoque gaussiano da etapa 2.



Fonte: O autor

Figura 24 – Exemplos de variação na velocidade de saturação do fósforo.



Fonte: O autor

## 6 Considerações Finais

O projeto já alcançou significativo êxito na produção de resultados para a representação de imagens de baixa resolução em monitores modernos. As imagens sintetizadas apresentam uma semelhança razoável com as imagens reproduzidas em TVs CRT, quando exibidas em monitores contemporâneos. Por meio de um modelo parametrizável, fomos capazes de calibrar os resultados manualmente, permitindo obter representações bem próximas à realidade.

Além disso, conseguimos otimizar significativamente o tempo de processamento. Com o código atualizado, a produção da saída para imagens com baixa contagem de *pixels* é de apenas 12 segundos, que é uma melhoria significativa em relação aos 30 minutos necessários nos estágios iniciais do projeto.

A saturação de cores, antes realizada com valores experimentais e não parametrizados, agora é manipulada de forma parametrizada, proporcionando maior controle e consistência sobre os resultados.

Para futuros aprimoramentos, pretendemos investigar métodos de processamento de imagens executados em placas de vídeo, utilizando bibliotecas que implementem funções em CUDA. Temos como objetivo conduzir testes com o usuário para que seja possível validar de maneira mais efetiva os resultados do nosso modelo. Além disso, daremos maior enfoque na calibração automatizada para gerar resultados ainda melhores, de modo que seja possível calibrar o modelo com base em uma fotografia de um monitor CRT.

## Referências

- 4LOUP. *CRTify*. 2017. Disponível em: <<https://4loup.itch.io/crtify>>. Acesso em: 18 de Out. 2022. Citado na página 25.
- AUFFRET, D. **Designing 2D graphics in the Japanese industry**. 2020. Disponível em: <<https://vgdensetsu.tumblr.com/post/179656817318/designing-2d-graphics-in-the-japanese-industry>>. Acesso em: 2 de Jan. 2023. Citado na página 14.
- BATES, P. **What Is Upscaling? How Does It Work? Is It Worth It?** 2019. Disponível em: <<https://www.makeuseof.com/tag/upscaling-how-does-it-work-and-is-it-worth-it/>>. Acesso em: 20 de Dez. 2022. Citado na página 21.
- BERTOZZI, A. D. **TV PAL-M, princípios básicos de Funcionamento**. [S.l.]: ABERT, 2001. Disponível em: <[http://www.bertozi.com/adb/Artigos\\_Publicados/tvpalm.html](http://www.bertozi.com/adb/Artigos_Publicados/tvpalm.html)>. Acesso em: 9 de Jan. 2023. Citado na página 17.
- BRADSKI, G. R.; KAEHLER, A. **Learning OpenCV 3**. Sebastopol, CA: O'Reilly Media, 2017. Citado na página 22.
- CATMULL, E.; ROM, R. A class of local interpolating splines. In: BARNHILL, R. E.; RIESENFELD, R. F. (Ed.). **Computer Aided Geometric Design**. Academic Press, 1974. p. 317–326. ISBN 978-0-12-079050-0. Disponível em: <<https://www.sciencedirect.com/science/article/pii/B97801207905000500205>>. Citado na página 28.
- CRTPIXELS. 2022. Disponível em: <<https://twitter.com/CRTpixels/status/1497359045203738628?cxt=HHwWiICy6ciH2McpAAAA>>. Acesso em: 29 de Out. 2022. Citado na página 19.
- FOUNDATION, P. S. **Python Documentation**. 2023. Disponível em: <<https://www.python.org/doc/>>. Acesso em: 6 de Jun. 2023. Citado na página 23.
- GAD., A. F. **PyGAD - Python Genetic Algorithm!** 2023. Disponível em: <<https://pygad.readthedocs.io/en/latest/>>. Acesso em: 28 de Mai. 2023. Citado na página 23.
- GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. 3. ed. Upper Saddle River, NJ: Pearson, 2007. Citado 2 vezes nas páginas 19 e 21.
- INC., A. **Apple Developer**. 2023. Disponível em: <[https://developer.apple.com/documentation/accelerate/blurring\\_an\\_image](https://developer.apple.com/documentation/accelerate/blurring_an_image)>. Acesso em: 2 de Jan. 2023. Citado na página 20.
- INSTRUCTION Manual - Super Nintendo Entertainment System (HW(A)-SNS-USA-2). [S.l.]: Nintendo of America Inc, 1992. Citado na página 13.
- JR, F. J. M. The kolmogorov-smirnov test for goodness of fit. **Journal of the American Statistical Association**, Taylor & Francis, v. 46, n. 253, p. 68–78, 1951. Citado na página 21.
- KANDEL, E. R.; SCHWARTZ, J. H.; JESSELL, T. M. **Principles of neural science**. [S.l.]: McGraw-Hill, 2000. Citado na página 16.

- KNOEPFLER, P. P. **Putting your iPhone under the microscope: big surprise about screen**. 2018. Disponível em: <<https://ipscell.com/2018/03/putting-iphone-microscope-big-surprise-screen/>>. Acesso em: 5 de Jan. 2023. Citado na página 17.
- KOPF, J.; LISCHINSKI, D. Depixelizing pixel art. **ACM Transactions on Graphics (Proceedings of SIGGRAPH 2011)**, v. 30, n. 4, p. 99:1 – 99:8, 2011. Citado na página 25.
- LEMONAMIGA. 2004. Disponível em: <<https://www.lemonamiga.com/games/details.php?id=1081>>. Acesso em: 30 de Nov. 2022. Citado na página 37.
- LUKE, S. **Essentials of Metaheuristics**. second. [S.l.]: Lulu, 2013. Available for free at <http://cs.gmu.edu/~sean/book/metaheuristics/>. Citado na página 22.
- MAZZOLENI, A. **Scale2x**. 2001. Disponível em: <<https://scale2x.sourceforge.io/>>. Acesso em: 18 de Set. 2022. Citado na página 25.
- MOMA. **The Museum of Modern Art**. 2020. Disponível em: <<https://www.moma.org/>>. Acesso em: 12 de Nov. 2022. Citado na página 14.
- MYABANDONWARE. **METAL GEAR 2: SOLID SNAKE MSX - 1990**. 2012. Acesso em: 21 de Nov. 2022. Citado na página 38.
- PITTMAN, K. **CRT simulation in Super win the game**. 2015. Disponível em: <<https://www.gamedeveloper.com/programming/crt-simulation-in-super-win-the-game>>. Acesso em: 12 de Set. 2022. Citado na página 25.
- STEPIN, M. **HQ2X, HQ3X e HQ4X**. 2003. Disponível em: <<https://web.archive.org/web/20070930190232/http://www.hiend3d.com/>>. Acesso em: 12 de Out. 2022. Citado na página 24.
- STRONG, M. W. **Strong National Museum of Play**. 2021. Disponível em: <<https://www.museumofplay.org/about/>>. Acesso em: 8 de Dez. 2022. Citado na página 13.
- STROUSTRUP, B. **The C++ programming language**. Reading, Massachusetts: Addison-Wesley, 1985. Citado na página 22.
- TAMIOSSO, G. L. **Implementação de um modelo fenomenológico para geração de imagens CRT**. figshare, 2023. Disponível em: <[https://figshare.com/collections/Implementa\\_o\\_de\\_um\\_modelo\\_fenomenol\\_gico\\_para\\_gera\\_o\\_de\\_imagens\\_CRT/6741594/1](https://figshare.com/collections/Implementa_o_de_um_modelo_fenomenol_gico_para_gera_o_de_imagens_CRT/6741594/1)>. Citado na página 36.
- TANNAS JR, L. E. **Flat-panel displays and CRTs**. Dordrecht, Netherlands: Springer, 2012. (Van Nostrand Reinhold Electrical/Computer Science and Engineering Series). Citado 3 vezes nas páginas 17, 18 e 19.
- TROGGLEMONKEY. **CRT-Royale**. 2014. Disponível em: <<https://emulation.gametechniki.com/index.php/CRT-Royale>>. Acesso em: 8 de Out. 2022. Citado na página 25.

WASSERMAN, L. **All of statistics**. 1. ed. New York, NY: Springer, 2004. (Springer Texts in Statistics). Citado na página 21.

WULF, T. et al. Video games as time machines: Video game nostalgia and the success of retro gaming. **Media and Communication**, Cogitatio Press, n. 2, p. 60–68, 2018. Disponível em: <<http://nbn-resolving.de/urn/resolver.pl?urn=nbn:de:bvb:19-epub-68743-7>>. Citado na página 13.

XBR. 2012. Disponível em: <<https://forums.libretro.com/t/xbr-algorithm-tutorial/123>>. Acesso em: 28 de Set. 2022. Citado na página 24.