

# Método de Aprendizado Não Supervisionado Baseado no Produto Cartesiano de Rankings para Busca de Imagens

Lucas Pascotti Valem, Daniel Carlos Guimarães Pedronette

<sup>1</sup> Departamento de Estatística, Matemática Aplicada e Computação (DEMAC)  
Universidade Estadual Paulista (UNESP), Rio Claro, Brasil

lucasvalem@rc.unesp.br, daniel@rc.unesp.br

**Resumo.** Apesar dos avanços significativos em ferramentas de busca de imagens, a definição de uma medida efetiva para a modelagem de similaridade entre imagens continua sendo um desafio em Sistemas de Recuperação de Imagens por Conteúdo (CBIR). Nesse cenário, técnicas de aprendizado não supervisionado de similaridade, capazes de melhorar a eficácia de tarefas de recuperação de imagens sem a intervenção do usuário são indispensáveis. Este trabalho de iniciação científica apresenta o método Cartesian Product of Ranking References (CPRR), o qual foi desenvolvido com esse propósito. Vários experimentos foram conduzidos em quatro coleções de imagens, considerando várias características visuais e diversos aspectos. Além da eficácia, também foram realizados experimentos para avaliações de eficiência, considerando computação paralela e heterogênea em CPU e GPU. O método atingiu ganhos significativos de eficácia que são comparáveis aos resultados de estado da arte mais populares.

**Abstract.** Despite the consistent development of visual features, effectively measuring the similarity among images remains a challenging problem in Content-based Image Retrieval (CBIR) systems. In this scenario, similarity learning approaches capable of improving the effectiveness of retrieval in an unsupervised way are indispensable. This undergraduate research work presents a method called Cartesian Product of Ranking References (CPRR) which was developed with this objective. An extensive experimental evaluation was conducted considering various aspects, four public image datasets, and several features. Besides effectiveness, experiments were also conducted to assess the efficiency, considering parallel and heterogeneous computing on CPU and GPU devices. The method has achieved significant effectiveness gains, including competitive state-of-the-art results on popular benchmarks.

## 1. Introdução

Ao longo dos anos, a evolução das tecnologias de aquisição e compartilhamento de conteúdos visuais provocou um grande crescimento das coleções de imagens (*datasets*). Com a grande quantidade de imagens acumuladas diariamente [Thomee and Lew 2012], os sistemas CBIR (*Content-Based Image Retrieval*) se tornaram a principal solução para a indexação e busca de imagens através de seu conteúdo visual.

Apoiados principalmente pela criação de diversas ferramentas visuais e medidas de distância, os sistemas CBIR se estabeleceram como um recurso essencial em muitos campos. No entanto, apesar do desenvolvimento de métodos de recuperação de imagens,

medir efetivamente a similaridade entre as imagens continua sendo um desafio. Assim, propostas mais recentes têm se direcionado em outras estratégias para os processos de recuperação, não diretamente relacionadas a procedimentos de extração de características de baixo nível [Liu et al. 2007].

Vários métodos de pós-processamento foram propostos para melhorar a eficácia de tarefas de recuperação não supervisionadas [Yang et al. 2009, Yang et al. 2013, Jiang et al. 2011]. No geral, o principal objetivo de tais métodos é substituir distâncias par a par por medidas de afinidade global capazes de considerar a estrutura do *dataset* [Yang et al. 2013]. Apesar de eficazes, métodos baseados em processos de difusão [Yang et al. 2009] e grafos [Wang et al. 2011a] geralmente apresentam altos custos computacionais.

Nesse cenário, métodos que exploram a informação contida em listas ordenadas de resultados (*ranked lists*) têm atraído bastante atenção recentemente, considerando tanto aspectos de eficácia quanto eficiência. Vários métodos recentes demonstram [Pedronette and da S. Torres 2013, Bai et al. 2015, Valem et al. 2015] que uma análise das posições dos elementos nas *ranked lists* fornece uma fonte de informação rica e confiável para obtenção de medidas contextuais de distância ou similaridade. Essas informações podem ser analisadas por diferentes modelos, como a similaridade das *ranked lists* [Pedronette and da S. Torres 2013] ou conjuntos [Bai et al. 2015], recomendações baseadas nas posições dos elementos [Valem et al. 2015], e verificações de consistência das *ranked lists* [Chen et al. 2014]. Em relação à eficiência, uma vez que as informações mais relevantes se encontram nas primeiras posições das *ranked lists*, o esforço computacional necessário é reduzido. Além disso, o modelo de lista ordenada de resultados permite uma representação uniforme, independente de medidas de distância ou similaridade.

A iniciação científica teve como propósito o desenvolvimento do algoritmo de aprendizado não supervisionado chamado de *Cartesian Product of Ranking References* (CPRR). O principal objetivo do CPRR é maximizar a informação de similaridade disponível nos *rankings* através de operações de produto cartesiano. Enquanto o algoritmo CPRR considera apenas um subconjunto das *ranked lists* para reduzir os custos computacionais, o produto cartesiano é usado para expandir as relações de similaridade. A ideia central consiste no uso de consultas de vizinhos mais próximos e mais próximos reversos (*k*NN e *reverse k*NN) para calcular os conjuntos de imagens, os quais são usados pelas operações de produto cartesiano. Baseado no nosso conhecimento atual da literatura, o CPRR é o primeiro método de aprendizado não supervisionado que explora o produto cartesiano de conjuntos de vizinhança e vizinhança reversa para tarefas de recuperação de imagens. Além disso, o método desenvolvido pode ser usado em operações de agregação de listas e pode ser executado em paralelo, apresentando tempos de execução bastante reduzidos.

Para uma avaliação consistente do método, diversos experimentos foram conduzidos, considerando diferentes aspectos. Quatro coleções de imagens públicas e diversos descritores de imagens distintos foram considerados. Os resultados experimentais confirmam os ganhos significativos de eficácia do método desenvolvido, atingindo ganhos relativos de até +32.57%. Experimentos realizados para avaliar a eficiência do método em computação paralela e heterogênea (CPU e GPU) demons-

traram que o algoritmo apresenta tempos de execução muito baixos para coleções de imagens de diferentes tamanhos. Dentre os *baselines* considerados, está o algoritmo *RL-Recommendation* [Valem et al. 2015], que possui ótimos resultados de eficácia e eficiência, embora os resultados do CPRR tenham se revelado superiores. O algoritmo CPRR também possui resultados comparáveis a métodos recentes de recuperação considerados estado da arte. Destaca-se, por exemplo, que o algoritmo alcançou um *N-S score* de 3.93 no *dataset* UKBench, frequentemente utilizado como *benchmark*, resultado superior aos mais recentes na literatura. Esse trabalho deu origem a publicações e prêmios, dentre eles uma publicação em conferência relevante da área premiada como *Best Full Paper* [Valem and Pedronette 2016].

O restante do texto é organizado como se segue: a Seção 2 apresenta uma definição formal do problema abordado. A Seção 3 apresenta o algoritmo CPRR, desenvolvido pelo aluno. A Seção 4 descreve a avaliação experimental. A Seção 5 lista as premiações e submissões efetuadas. Por fim, a Seção 6 discute as conclusões.

## 2. Definição Formal do Problema

Seja  $\mathcal{C} = \{img_1, img_2, \dots, img_n\}$  uma coleção de imagens, onde  $n$  é o tamanho da coleção. Seja  $\rho: \mathcal{C} \times \mathcal{C} \rightarrow \mathbb{R}$  uma função de similaridade, tal que  $\rho(img_i, img_j)$  denota a similaridade entre duas imagens  $img_i, img_j \in \mathcal{C}$ . Para simplificar a leitura, a notação  $\rho(i, j)$  é utilizada ao longo do texto.

A similaridade entre todas as imagens  $img_i, img_j \in \mathcal{C}$  definida pela função  $\rho(i, j)$  pode ser aplicada para calcular uma matriz de afinidade  $W$ . A matriz  $W$ , por sua vez, é comumente utilizada como uma matriz de adjacência por vários métodos baseados em grafos e em processos de difusão. Contudo, esse método geralmente resulta em um armazenamento e complexidade temporal de pelo menos  $O(n^2)$ , impedindo resultados eficientes e escaláveis em grandes coleções de imagens.

O modelo de lista ordenada de resultados representa uma solução efetiva e eficiente nesse cenário. Diferente das funções de similaridade, as quais estabelecem relações apenas entre pares de imagens, as *ranked lists* armazenam informação de similaridade entre a imagem de consulta e toda a coleção de imagens. Além disso, embora uma *ranked list* possa conter informações de toda uma coleção, espera-se encontrar a maior parte das informações relevantes nas suas primeiras posições. Portanto, uma constante  $L \ll n$  pode ser usada tal que apenas um subconjunto das primeiras  $L$  posições da *ranked list* são consideradas, reduzindo o custo computacional requerido pelo algoritmo.

Formalmente, uma *ranked list*  $\tau_q = (img_1, img_2, \dots, img_L)$  pode ser definida como uma permutação de uma coleção de imagens  $\mathcal{C}_L \subset \mathcal{C}$ , a qual contém as imagens mais similares à imagem de consulta  $img_q$ , tal que  $|\mathcal{C}_L| = L$ . Uma permutação  $\tau_q$  é uma bijeção do conjunto  $\mathcal{C}_L$  para o conjunto  $[L] = \{1, 2, \dots, L\}$ . A notação  $\tau_q(i)$  pode ser interpretada como a posição (classificação ou *rank*) da  $img_i$  na *ranked list*  $\tau_q$ . Portanto, se a  $img_i$  está classificada antes da  $img_j$  na *ranked list* da  $img_q$ , isto é,  $\tau_q(i) < \tau_q(j)$ , então  $\rho(q, i) \geq \rho(q, j)$ .

Toda imagem em uma coleção pode ser tomada como uma imagem de consulta  $img_q$  e sua respectiva *ranked list* pode ser calculada. Sendo assim, o conjunto de *ranked lists*  $\{\tau_1, \tau_2, \dots, \tau_n\}$  fornece um modelo compacto e efetivo de lista ordenada de resultados.

### 3. Algoritmo CPRR

O principal objetivo do *Cartesian Product of Ranking References* (CPRR) é maximizar a informação de similaridade codificada nas *ranked lists* utilizando operações de produto cartesiano.

O método desenvolvido pode ser descrito em duas etapas principais:

- **Normalização das Posições:** as referências recíprocas das posições são analisadas melhorando a simetria da vizinhança e, conseqüentemente, a eficácia das *ranked lists*;
- **Produto Cartesiano das Referências:** o produto cartesiano é calculado considerando as primeiras  $k$  posições dos conjuntos de vizinhança e dos conjuntos de vizinhança reversa. Os resultados obtidos são usados para definir uma medida iterativa de similaridade que é utilizada para reordenar as *ranked lists*.

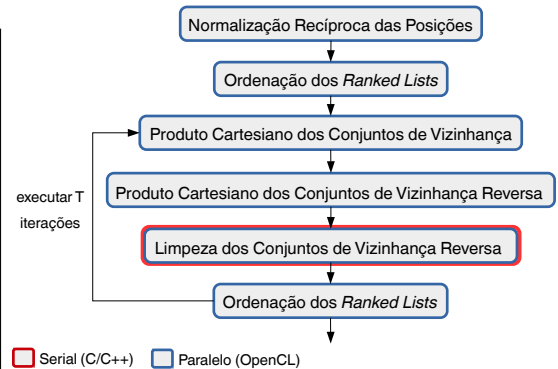
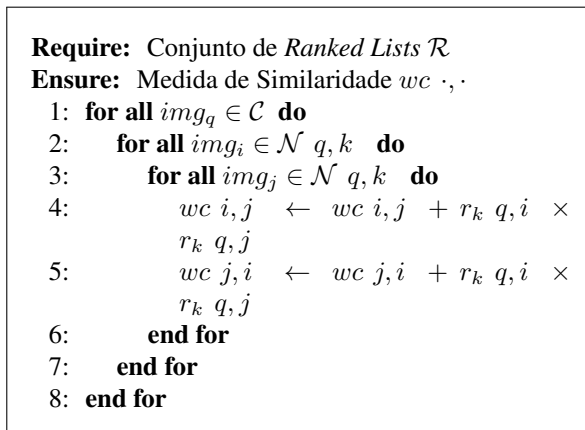
#### 3.1. Produto Cartesiano dos Conjuntos de Vizinhança

Enquanto o algoritmo considera apenas um subconjunto das *ranked lists* para reduzir os custos computacionais, o produto cartesiano é usado para expandir as relações de similaridade. Sejam  $\mathcal{N}(i, k)$  e  $\mathcal{N}(j, k)$  os conjuntos de vizinhança de tamanho  $k$  das imagens  $img_i, img_j$ . Seja  $\mathcal{C}$  o conjunto que representa a coleção de imagens. A medida de similaridade  $wc(i, j)$  é calculada considerando toda imagem de consulta  $img_q \in \mathcal{C}$  que apresenta  $img_i$  e  $img_j$  como vizinhos. Além disso, a medida leva em consideração a posição da imagem, ou seja  $r_k(q, i) \times r_k(q, j)$ , onde  $r_k(q, i) = k - \tau_q(i) + 1$ . Formalmente, a medida  $wc(i, j)$  pode ser definida como:

$$wc(i, j) = \sum_{q \in \mathcal{C}} \sum_{i \in \mathcal{N}(q, k)^2} \sum_{j \in \mathcal{N}(q, k)^2} r_k(q, i) \times r_k(q, j). \quad (1)$$

Em termos de complexidade assintótica, a medida de similaridade apresenta complexidade de apenas  $O(n)$ , uma vez que  $k$  é uma constante. A Figura 1 apresenta o algoritmo para a realização dessa etapa. A principal ideia consiste em realizar a análise apenas das primeiras  $k$  posições para o cálculo do produto cartesiano de cada conjunto de vizinhança. O produto cartesiano da vizinhança reversa é realizado de maneira análoga, mas considerando as referências realizadas a  $img_q$  nas demais *ranked lists* do *dataset*.

**Figura 1. Produto Cartesiano dos Conjuntos de Vizinhança.**



**Figura 2. Design do CPRR Paralelo.**

### 3.2. Design Paralelo

A Figura 2 apresenta todas as etapas do algoritmo e seu *design* paralelo usando o padrão OpenCL. O algoritmo foi paralelizado em seis *kernels*. Um *kernel* pode ser entendido como uma etapa da paralelização, cada instância de um *kernel* é executada em uma unidade chamada *work-item*. O mesmo código é executado em paralelo por diferentes *work-items* com dados distintos. Dois modelos de transferência de memória foram usados: *Writer Buffer*, o qual requer a transferência dos dados para a memória do dispositivo; e o *Map Buffer*, o qual requer apenas a transferência dos ponteiros dos dados.

Como as operações de produto cartesiano são processadas em paralelo e as medidas de similaridade são armazenadas na memória global, acessos concorrentes podem provocar perdas de atualizações. No entanto, o *overhead* associado a operações atômicas no OpenCL é alto. Portanto, atualizações diretas de medidas de similaridade são permitidas devido ao impacto pouco significativo na eficácia do algoritmo (como apresentado na seção experimental).

Ao término de cada iteração, um procedimento de ordenação é executado para atualizar as primeiras posições das *ranked lists* de acordo com uma nova medida de similaridade. O algoritmo *insertion sort* é usado, uma vez que tende a ser linear quando a entrada está quase ordenada.

## 4. Avaliação Experimental

Esta seção apresenta um resumo dos experimentos realizados para avaliar o método proposto. Uma descrição completa dos experimentos, *datasets*, descritores e parâmetros utilizados pode ser encontrada em [Valem and Pedronette 2016].

### 4.1. Protocolo Experimental

Nos experimentos, são considerados quatro *datasets* públicos diferentes, com tamanhos indo de 280 a 10.200 imagens. Com a finalidade de explorar as características distintas dos *datasets*, diversos descritores globais foram usados, considerando propriedades de cor, textura e contorno. A extração de características a partir de redes neurais convolucionais foram realizadas utilizando o *Caffe framework* [Jia et al. 2014]. Em relação aos descritores locais, o descritor SIFT [Lowe 1999] é considerado em conjunto a um variante do *vocabulary tree based retrieval* (VOC) [Wang et al. 2011b]. Os *datasets* e descritores são brevemente descritos na Tabela 1.

Diferentes métodos recentes foram considerados como *baselines*, dentre eles os principais: *Pairwise Recommendation* [Pedronette and da S. Torres 2012], *RL-Sim* [Pedronette and da S. Torres 2013] e *RL-Recommendation* [Valem et al. 2015].

A avaliação de eficácia considera todas as imagens de cada *dataset* como imagens de consulta. Como uma medida de eficácia, o *Mean Average Precision* (MAP) é usado na maior parte dos *datasets*, exceto para o *dataset* UKBench que usa o *N-S Score*. Para o *dataset* MPEG-7, o *Recall@40* também é usado em conjunto com o MAP. A maior parte dos experimentos também reportam o ganho relativo de eficácia do método CPRR em relação ao resultado fornecido pelo descritor.

O tempo médio das execuções é calculado a partir da média de 10 execuções utilizando intervalo de confiança de 95%. O *hardware* utilizado nos experimentos é composto

**Tabela 1. Datasets e descritores usados na avaliação experimental.**

<b>Soccer [van de Weijer and Schmid ]:</b> 280 cenas, compostas por imagens de 7 times de futebol com 40 imagens por classe. Avaliação de eficácia utilizando MAP (%).
<b>Descritores:</b> <i>Auto Color Correlograms (ACC)</i> [Huang et al. 1997], <i>Border/Interior Pixel Classification (BIC)</i> [Stehling et al. 2002] e <i>Global Color Histogram (GCH)</i> [Swain and Ballard 1991].
<b>MPEG-7 [Latecki et al. 2000]:</b> 1.400 imagens com diferentes formas divididas em 70 classes. Frequentemente utilizada para avaliação de métodos de aprendizado não supervisionado. Avaliação de eficácia utilizando MAP (%) e Recall@40.
<b>Descritores:</b> <i>Articulation-Invariant Representation (AIR)</i> [Gopalan et al. 2010], <i>Aspect Shape Context (ASC)</i> [Ling et al. 2010], <i>Beam Angle Statistics (BAS)</i> [Arica and Vural 2003], <i>Contour Features Descriptor (CFD)</i> [Pedronette and da S. Torres 2010], <i>Inner Distance Shape Context (IDSC)</i> [Ling and Jacobs 2007] e <i>Segment Saliences (SS)</i> [da S. Torres and Falcão 2007].
<b>Brodatz [Brodatz 1966]:</b> 1.776 imagens, composta por 111 texturas diferentes divididas em 16 classes. Avaliação de eficácia utilizando MAP (%).
<b>Descritores:</b> <i>Color Co-Occurrence Matrix (CCOM)</i> [Kovalev and Volmer 1998], <i>Local Activity Spectrum (LAS)</i> [Tao and Dickinson 2000] e <i>Local Binary Patterns (LBP)</i> [Ojala et al. 2002].
<b>UKBench [Nistér and Stewenius 2006]:</b> 10.200 imagens, composta de 2.550 objetos capturados de 4 diferentes maneiras (ponto de vista, distância e condições de iluminação). Avaliação de eficácia utilizando N-S score.
<b>Descritores:</b> <i>ACC</i> [Huang et al. 1997], <i>BIC</i> [Stehling et al. 2002], <i>Convolutional Neural Network by Caffe</i> [Jia et al. 2014] <i>framework (CNN-Caffe)</i> , <i>Fuzzy Color and Texture Histogram (FCTH)</i> [Chatzichristofis and Boutalis 2008], <i>Scale-Invariant Feature Transform (SIFT)</i> [Lowe 1999] e <i>Vocabulary Tree (VOC)</i> [Wang et al. 2011b]

de uma CPU Intel Xeon CPU E3-1240 e de uma GPU AMD Radeon HD 7900. O ambiente de software utiliza o sistema operacional Linux 3.11.0-15 - Ubuntu 12.04 e OpenCL 1.2 AMD-APP. O código foi compilado usando g++ 4.6.3 com a *flag* de otimização -O3.

## 4.2. Avaliação de Eficácia

Primeiramente, o algoritmo proposto é avaliado em tarefas de recuperação de imagem, considerando três *datasets* e diferentes características globais (contorno, cor e textura). Os resultados de MAP são mostrados na Tabela 2, considerando as implementações serial e paralela do algoritmo CPRR. Como pode ser observado, os resultados de eficácia para as implementações serial e paralela são muito parecidos. Os ganhos relativos de eficácia, calculados levando em consideração as execuções seriais, atingiram ganhos de até +32.57%. Resultados de métodos recentes de aprendizado não supervisionado são usados como *baselines*. Nós podemos observar que o método desenvolvido, o CPRR, alcança os melhores resultados para a maior parte dos descritores.

**Tabela 2. Avaliação de eficácia do algoritmo CPRR considerando vários datasets e descritores (MAP como medida de eficácia).**

Descritor	Dataset	MAP Original	Pairwise Rec.	RL-Rec.	CPRR Serial	CPRR Paralelo	Ganho Relativo
SS	MPEG-7	37.67%	39.90%	48.68%	49.94%	49.947 % ± 0.009	+32.57%
BAS	MPEG-7	71.52%	77.65%	79.58%	80.60%	80.611 % ± 0.006	+12.70%
IDS	MPEG-7	81.70%	86.83%	88.80%	89.42%	89.432 % ± 0.004	+9.45%
CFD	MPEG-7	80.71%	91.38%	91.39%	92.15%	92.157 % ± 0.005	+14.17%
ASC	MPEG-7	85.28%	91.80%	91.34%	92.32%	92.323 % ± 0.005	+8.26%
AIR	MPEG-7	89.39%	95.50%	96.12%	97.80%	97.796 % ± 0.007	+9.41%
GCH	Soccer	32.24%	32.35%	34.38%	35.47%	35.307 % ± 0.054	+10.02%
ACC	Soccer	37.23%	40.31%	41.23%	47.14%	46.965 % ± 0.072	+26.62%
BIC	Soccer	39.26%	42.64%	45.15%	47.29%	47.172 % ± 0.095	+20.45%
LBP	Brodatz	48.40%	51.92%	51.26%	49.07%	49.073 % ± 0.006	+1.38%
CCOM	Brodatz	57.57%	66.46%	64.34%	64.81%	64.816 % ± 0.007	+12.58%
LAS	Brodatz	75.15%	80.73%	79.71%	79.34%	79.346 % ± 0.004	+5.58%

**Tabela 3. Avaliação de eficácia no dataset Ukbench.**

Descritor	Tipo	Score Original	RL-Rec.	CPRR	Ganho Relativo
SIFT	Local	2.54	2.88	2.99	+17.72%
FCTH	Cor/Text.	2.73	2.80	2.90	+6.23%
BIC	Cor	3.04	3.15	3.28	+7.89%
Caffe	CNN	3.31	3.46	3.61	+9.06%
ACC	Cor	3.36	3.53	3.62	+7.74%
VOC	BoW	3.54	3.65	3.72	+5.08%

**Tabela 4. Eficácia para agregação de listas.**

Dataset	Descritores	Score	Métrica
Soccer	ACC+BIC	48.25%	MAP
MPEG-7	CFD+AIR	99.95%	MAP
MPEG-7	CFD+ASC	98.83%	MAP
Brodatz	CCOM+LAS	83.26%	MAP
UKBench	VOC+Caffe	3.88	<i>N-S score</i>
UKBench	ACC+Caffe	3.88	<i>N-S score</i>
UKBench	ACC+Caffe+VOC	3.93	<i>N-S score</i>

Um experimento considerando tarefas de recuperação de imagens e um conjunto bastante distinto de descritores foi conduzido no *dataset* UKBench. A Tabela 3 apresenta os resultados de eficácia dados pelo *N-S score*. O *N-S score* é definido no intervalo [1,4] (o maior valor possível é 4) e corresponde ao número de imagens relevantes entre as primeiras quatro imagens retornadas. O pequeno número de imagens por classe (só 4) torna esse *dataset* um grande desafio para algoritmos de aprendizado não supervisionado. Apesar desse fato, o CPRR alcançou altos ganhos indo de +5.08% a +17.72% e superior ao *baseline*.

O algoritmo desenvolvido foi também avaliado em tarefas de agregação de listas, considerando os melhores descritores de cada *dataset*. A Tabela 4 apresenta os resultados de eficácia para vários *datasets*. Nós podemos observar que todos os resultados de agregação são superiores aos dos descritores isolados, alcançando valores muito significativos para todos os *datasets*.

Um experimento analisando conjuntamente aspectos de eficácia e eficiência foram realizados no *dataset* MPEG-7. A Figura 3 apresenta os resultados do CPRR e de *baselines* recentes. Os resultados de MAP e os tempos de execução determinam a posição do algoritmo no gráfico. Portanto, um algoritmo ideal, com alta eficácia e baixo tempo de execução, é posicionado no canto superior esquerdo do gráfico. Perceba que as melhores posições são ocupadas pelo algoritmo CPRR (serial e paralelo).

### 4.3. Avaliação de Eficiência

Experimentos foram conduzidos para avaliar a eficiência do método desenvolvido, considerando diferentes aspectos: vários *datasets*, implementações serial e paralela, dispositivos distintos (CPU, GPU) e diferentes modelos de transferência de memória. O tempo de *OpenCL build* e *environment time* não são considerados nos resultados, já que podem ser executados uma vez *offline* e o *environment time* é constante independente do *dataset*.

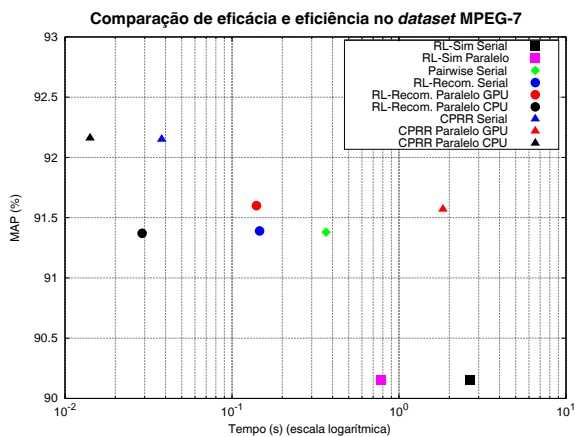
A Tabela 5 apresenta a média dos tempos de execução e intervalos de confiança para o algoritmo CPRR. Para propósito de comparação, o tempo de execução de dois *baselines* recentes são reportados. O melhor resultado de performance para cada *dataset* está destacado em negrito. Como nós podemos observar, o algoritmo CPRR requer tempos de execução muito baixos para todos os *datasets*, menores que os dos *baselines* até mesmo para as implementações seriais.

Uma comparação mais geral do CPRR (tanto serial quanto paralelo) é apresentada na Figura 4. O tempo de execução do *RL-Sim* (serial e paralelo), *RL-Recommendation* (serial e paralelo) e *Pairwise Recommendation* (serial) são reportados como *baselines*. O *dataset* MPEG-7 é considerado para o experimento. Perceba que, mesmo usando uma

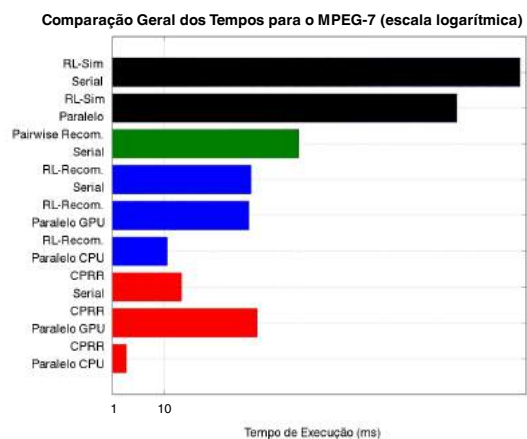
**Tabela 5. Avaliação de Eficiência: tempo de execução (em segundos) do CPRR para diferentes dispositivos e datasets.**

Algoritmo	Exec.	Dispositivo	Soccer	MPEG-7	Brodatz	UKBench
<i>Pairwise Rec.</i>	Serial	CPU	0.1149 ± 0.00018	0.3663 ± 0.00094	0.6672 ± 0.00140	14.802 ± 0.11059
<i>RL-Rec.</i>	Serial	CPU	0.0607 ± 0.00000	0.1462 ± 0.00021	0.1108 ± 0.00102	0.1868 ± 0.00018
<b>CPRR</b>	Serial	CPU	0.0058 ± 0.00021	0.0381 ± 0.00041	0.0501 ± 0.00077	0.1767 ± 0.00102
<b>CPRR</b>	Paralelo	GPU <sup>1</sup>	0.0711 ± 0.00463	0.1640 ± 0.00781	0.1560 ± 0.00570	0.2038 ± 0.00874
<b>CPRR</b>	Paralelo	CPU <sup>1</sup>	0.0032 ± 0.00015	0.0164 ± 0.00031	0.0214 ± 0.00054	0.1834 ± 0.00052
<b>CPRR</b>	Paralelo	CPU <sup>2</sup>	<b>0.0027 ± 0.00000</b>	<b>0.0131 ± 0.00018</b>	<b>0.0143 ± 0.00075</b>	<b>0.1082 ± 0.00051</b>

Modelo de Transferência de Memória: <sup>1</sup>Write Buffer; <sup>2</sup>Map Buffer.



**Figura 3. Análise de eficiência e escalabilidade no dataset MPEG-7.**

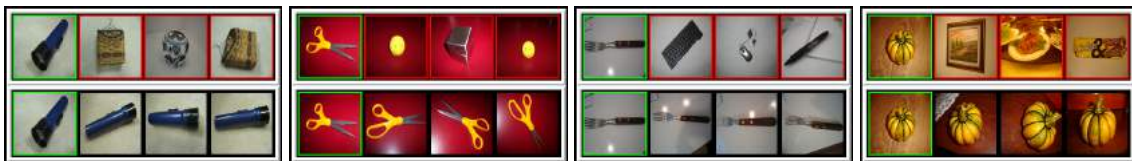


**Figura 4. Tempos de execução no dataset MPEG-7.**

escala logarítmica, o tempo de execução do algoritmo CPRR (em vermelho) é significativamente menor que outros métodos considerados.

#### 4.4. Resultados Visuais

Quatro exemplos visuais de resultados são apresentados na Figura 5. A imagem apresentada em quadro verde é a imagem de consulta. As *ranked lists* obtidas são apresentadas à direita, com imagens incorretas em quadros vermelhos. São apresentados quatro exemplos de consultas, com os respectivos resultados antes e após a aplicação do algoritmo, considerando o dataset UKBench. É notável a melhoria de eficácia obtida pelo algoritmo para as diferentes consultas analisadas.



**Figura 5. Exemplos de resultados visuais no dataset UKBench.**

### 5. Prêmios e Publicações

O trabalho de iniciação científica originou um artigo [Valem and Pedronette 2016] na categoria *Full Paper* na *Conference on Graphics, Patterns and Images - SIBGRAPI 2016*



(Qualis B1). O SIBGRAPI é uma conceituada conferência na área de recuperação de imagens, visão computacional, processamento de imagens e computação gráfica. O artigo foi premiado como *Best Full Paper of SIBGRAPI 2016*.

O trabalho também foi reconhecido com o prêmio de **Melhor Trabalho de Iniciação Científica da Área de Ciências Exatas no XXVIII CIC** - Congresso de Iniciação Científica da UNESP.

O algoritmo está disponível em uma **plataforma open-source** de métodos de aprendizado não supervisionado desenvolvida pelo aluno e que está em constante aprimoramento. O código fonte pode ser acessado em `github.com/lucasPV/UDLF`.

## 6. Conclusões

Nesse trabalho de iniciação científica, foi apresentado um novo método de aprendizado não supervisionado para tarefas de recuperação de imagens. O método inova usando uma abordagem baseada em operações de produto cartesiano de conjuntos tomados a partir das *ranked lists*. Vários experimentos foram conduzidos em quatro coleções de imagens diferentes e diversos descritores. Os resultados experimentais revelam a eficácia, a eficiência e a escalabilidade do método mesmo quando comparado com métodos de estado da arte. Como trabalho futuro, pretende-se avaliar o método em outros cenários multimídia (descritores de vídeo, som, texto e outros). Também deseja-se avaliar o método em casos onde a imagem de busca não faz parte da coleção de imagens.

## Referências

- Arica, N. and Vural, F. T. Y. (2003). BAS: a perceptual shape descriptor based on the beam angle statistics. *Pattern Recognition Letters*, 24(9-10):1627–1639.
- Bai, X., Bai, S., and Wang, X. (2015). Beyond diffusion process: Neighbor set similarity for fast re-ranking. *Information Sciences*, 325:342 – 354.
- Brodatz, P. (1966). *Textures: A Photographic Album for Artists and Designers*. Dover.
- Chatzichristofis, S. A. and Boutalis, Y. S. (2008). Fcth: Fuzzy color and texture histogram - a low level feature for accurate image retrieval. In *WIAMIS*, pages 191–196.
- Chen, Y., Li, X., Dick, A., and Hill, R. (2014). Ranking consistency for image matching and object retrieval. *Pattern Recognition*, 47(3):1349 – 1360.
- da S. Torres, R. and Falcão, A. X. (2007). Contour Saliency Descriptors for Effective Image Retrieval and Analysis. *Image and Vision Computing*, 25(1):3–13.
- Gopalan, R., Turaga, P., and Chellappa, R. (2010). Articulation-invariant representation of non-planar shapes. In *ECCV*, volume 3, pages 286–299.
- Huang, J., Kumar, S. R., Mitra, M., Zhu, W.-J., and Zabih, R. (1997). Image indexing using color correlograms. In *CVPR*, pages 762–768.
- Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*.
- Jiang, J., Wang, B., and Tu, Z. (2011). Unsupervised metric learning by self-smoothing operator. In *ICCV*, pages 794–801.
- Kovalev, V. and Volmer, S. (1998). Color co-occurrence descriptors for querying-by-example. In *ICMM*, page 32.

- Latecki, L. J., Lakmper, R., and Eckhardt, U. (2000). Shape descriptors for non-rigid shapes with a single closed contour. In *CVPR*, pages 424–429.
- Ling, H. and Jacobs, D. W. (2007). Shape classification using the inner-distance. *IEEE TPAMI*, 29(2):286–299.
- Ling, H., Yang, X., and Latecki, L. J. (2010). Balancing deformability and discriminability for shape matching. In *ECCV*, volume 3, pages 411–424.
- Liu, Y., Zhang, D., Lu, G., and Ma, W.-Y. (2007). A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1):262 – 282.
- Lowe, D. (1999). Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157.
- Nistér, D. and Stewénius, H. (2006). Scalable recognition with a vocabulary tree. In *CVPR*, volume 2, pages 2161–2168.
- Ojala, T., Pietikäinen, M., and Mäenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *PAMI*, 24(7):971–987.
- Pedronette, D. C. G. and da S. Torres, R. (2010). Shape retrieval using contour features and distance optimization. In *VISAPP*, volume 1, pages 197 – 202.
- Pedronette, D. C. G. and da S. Torres, R. (2012). Exploiting pairwise recommendation and clustering strategies for image re-ranking. *Information Sciences*, 207:19–34.
- Pedronette, D. C. G. and da S. Torres, R. (2013). Image re-ranking and rank aggregation based on similarity of ranked lists. *Pattern Recognition*, 46(8):2350–2360.
- Stehling, R. O., Nascimento, M. A., and Falcão, A. X. (2002). A compact and efficient image retrieval approach based on border/interior pixel classification. In *CIKM*, pages 102–109.
- Swain, M. J. and Ballard, D. H. (1991). Color indexing. *International Journal on Computer Vision*, 7(1):11–32.
- Tao, B. and Dickinson, B. W. (2000). Texture recognition and image retrieval using gradient indexing. *JVCIR*, 11(3):327–342.
- Thomee, B. and Lew, M. (2012). Interactive search in image retrieval: a survey. *International Journal of Multimedia Information Retrieval*, 1(2):71–86.
- Valem, L. P. and Pedronette, D. C. G. (2016). Unsupervised similarity learning through cartesian product of ranking references for image retrieval tasks. In *Conference on Graphics, Patterns and Images (SIBGRAPI'2016)*, pages 249–256.
- Valem, L. P., Pedronette, D. C. G., Torres, R. d. S., Borin, E., and Almeida, J. (2015). Effective, efficient, and scalable unsupervised distance learning in image retrieval tasks. *ICMR*.
- van de Weijer, J. and Schmid, C. Coloring local feature extraction. In *ECCV*.
- Wang, J., Li, Y., Bai, X., Zhang, Y., Wang, C., and Tang, N. (2011a). Learning context-sensitive similarity by shortest path propagation. *Pattern Recognition*, 44(10-11):2367–2374.
- Wang, X., Yang, M., Cour, T., Zhu, S., Yu, K., and Han, T. (2011b). Contextual weighting for vocabulary tree based image retrieval. In *ICCV'2011*, pages 209–216.
- Yang, X., Koknar-Tezel, S., and Latecki, L. J. (2009). Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. In *CVPR*, pages 357–364.
- Yang, X., Prasad, L., and Latecki, L. (2013). Affinity learning with diffusion on tensor product graph. *IEEE TPAMI*, 35(1):28–38.