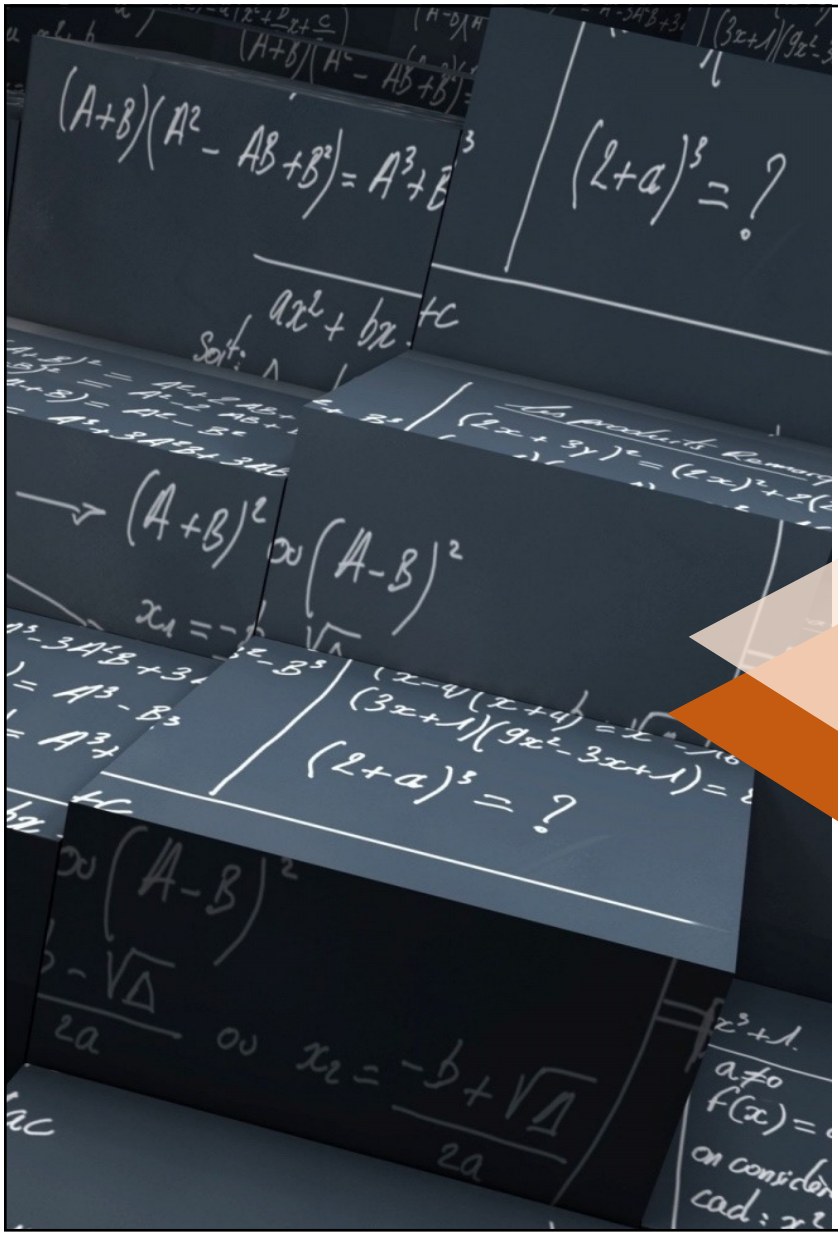
Algorithms and Metrics for

Scalable Multimedia Processing

Prof. Dr.-Ing. Stefan Wagenpfeil, December 2023



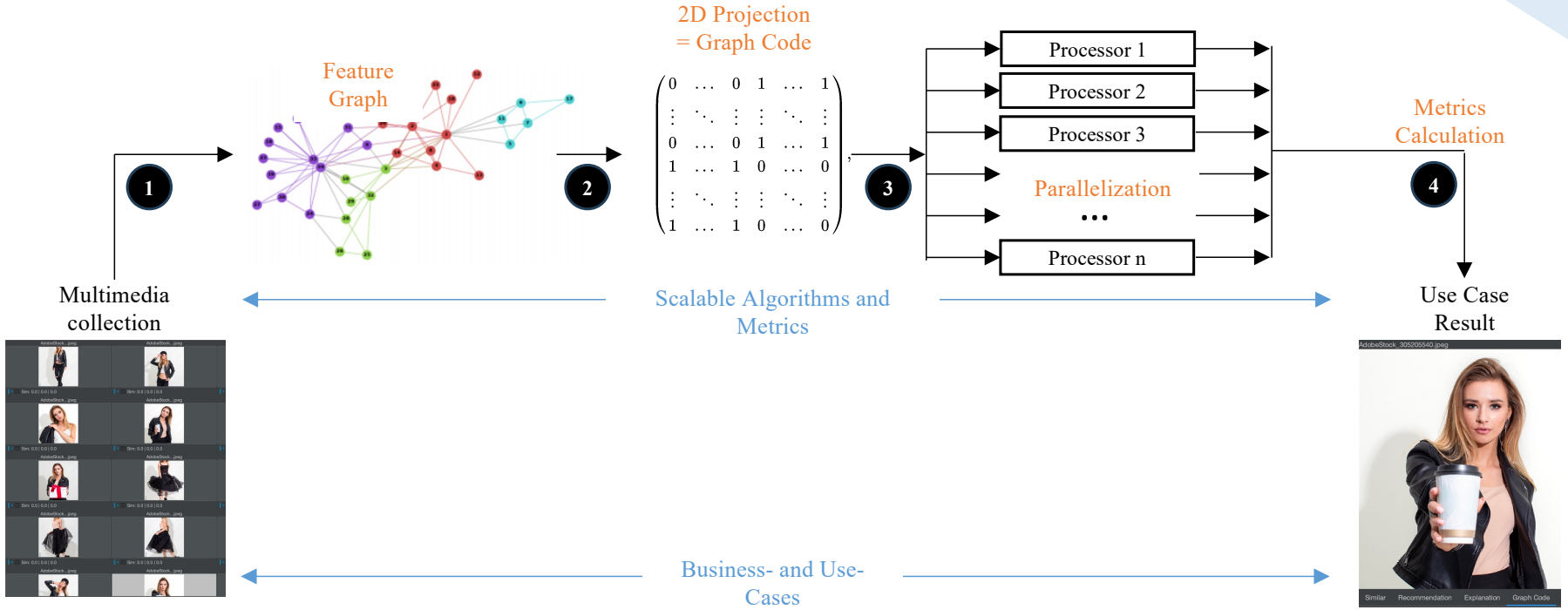
Problem.

Multimedia is everywhere!
… and Multimedia collections grow exponentially. Both the number of assets, and the level-of-detail of each asset increase enormously, which leaves a significant challenge for users and service providers in maintaining, processing, querying, and retrieving relevant assets.

Facts and Figures: every minute, 3.3 million pictures are taken worldwide, 92% taken with Smartphones. 6% of the photos are uploaded and shared, Google Images contains 136 billion images, the digital asset market is about 6.9 billion $ in 2024.

The problem: how can users and/or service providers maintain such huge amounts of high level-of-detail Multimedia assets (particularly, on Smartphones)? How can retrieval processes be made efficient and effective? How can new business cases be derived from such processing?

Solution.



1) Existing feature extraction mechanisms are applied to analyze Multimedia assets. 2) The analyzed data is transformed into a mathematical 2D space for better calculations and processing. 3) Parallelized algorithms employ this 2D space to calculate various metrics. 4) The metrics build the foundation for business- and use-cases.

Metrics I.

Graph Codes can be used to calculate various metrics. In general, the „Graph Code Metric“ is a triple

$$M_{GC} = (M_F, M_{FR}, M_{RT})$$

In this triple, the first element is the „Feature Term Metric“, which basically calculates distances based on the detected feature vocabulary terms.

$$M_F(GC_i, GC_j) = \frac{|dict_{\cap}|}{|dict_i|}$$

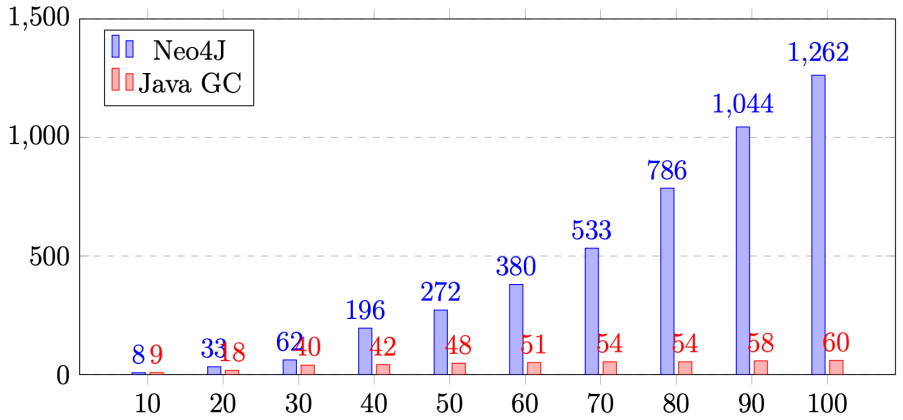
The second element is the „Feature Relationship Metric“, which focuses on the relationships between detected features

$$M_{FR}(GC_i, GC_j) = \frac{\sum AM(M_{\cap i,j}) - n}{|AM(M_{\cap i})| - n}$$

And the third element is the “Relationship Type Metric“, which calculates measures regarding the similarity of relationship types.

$$M_{RT}(GC_i, GC_j) = \frac{\sum_{i,j}^{n, i \neq j}(|M_{\cap i} - M_{\cap j}|)}{|M_{\cap i}| - n}$$

Based on these metrics, the basic calculation of similarity and recommendations can be performed on large Multimedia collections in a parellized and scalable way.



=> Graph Code calculations scale linear instead of exponential. GPU-based scaling has been implemented (CUDA + Apple Metal) and show a significant performance speedup. In experiments, parallel Graph Code processing was up to 16.711 times faster than sequential processing.

Metrics II.

Furthermore, various business- and use-cases can be implemented based on Graph Codes. In general, there are three major unique and innovative cases without any further processing:

The „Feature Discrimination Metric“ describes, how discriminative a certain feature is. If many similar assets are in a collection, this metric helps to find, what distinguishes a certain asset from the rest of the collection

$$M_{DIS}(vt_i, vt_j) = \sum_{k=0}^{n} |m(i,k) \neq 0| - \sum_{k=0}^{n} |m(j,k) \neq 0|$$

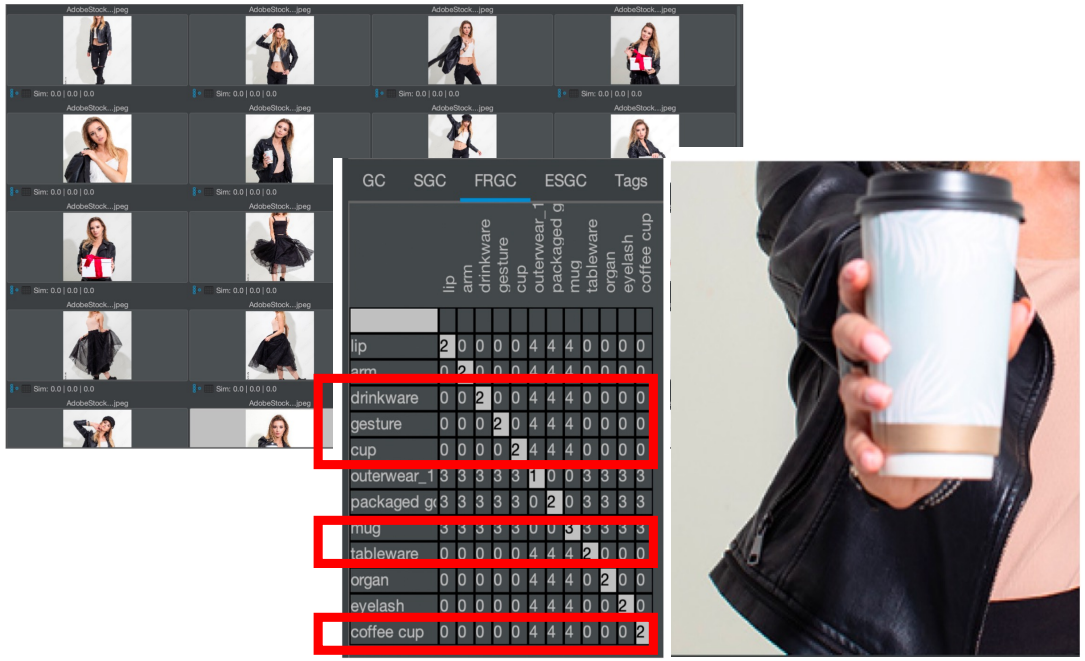
The overall relevance of a certain feature within the whole collection can be measured with the „Feature Relevance Metric“.

$$M_{REL}(vt_i, vt_j) = TFIDF(vt_i, SGC_{Coll}) - TFIDF(vt_j, SGC_{Coll})$$

Finally, the „Aboutness Metric“ can be employed to calculate, what a collection of assets is about.

$$TFIDF(vt_i, SGC) = M_{DIS}(vt_i, vt_j) \cdot log \frac{|SGC_{Coll}|}{M_{DIS}(vt_i, vt_j)}$$

$$M_{ABT} = SGC - SGC_{STOP-ED}$$



These three additional metrics can be used to answer questions like „what's the most important object on this photo“ (see example on the right), „what's the difference between these photos“, „why is this element in my result list“.

Feature Extraction.

A complete Java-based implementation and Multimedia Processing Framework has been developed, which employs various algorithms and mechanisms for Multimedia feature extraction. For each technology, a processing plugin has been implemented, which converts the extracted features into feature graphs and Graph Codes.

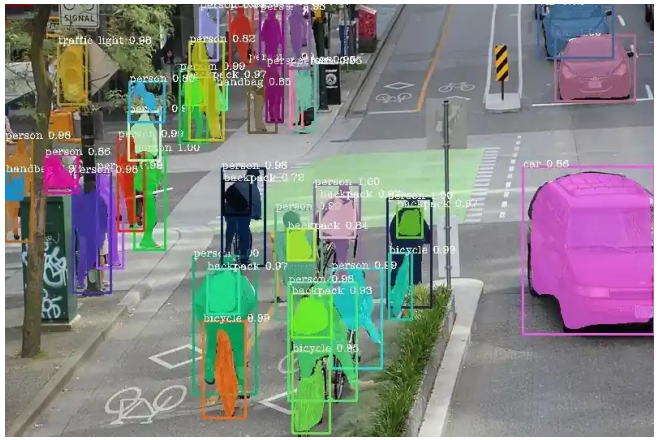


Image
EXIF Data including Camera / Lens model and Location
Object Detection Plugins by
- *Google Vision*
- *Microsoft Vision*
- *Amazon AWS*
- *ClarifAi*
- *Carnet (Car and Car type specialist)*
- *Open CV*
- *Yolo Fruit*
- *Yolo General*

Social Media (Facebook, Instagram, Twitter)
Text Annotations

Audio
Speech to Text (english)
Shazam Integration for Artist, Title, Label
Beats Per Minute Extractor
Pitch Extraction

Other
SQL/JDBC integration, Json, XML

Video
FFMPEG for Video-Processing and Conversion
Scene Detection and Splitting
MPEG7 and MXF Containers
Keyframe Extraction
Audio Track Extraction
Google Vision Video Summary
Live Streaming and Video on Demand
Social Media (Facebook, Instagram, Twitter)

Text
Microsoft Office (Word, Excel, Powerpoint, RTF, Embedded Multimedia Objects)
Bag Of Words Calculations
TFIDF Calculation
Sentence and topic detection
RSS Extractors
XML and XSLT
Social Media (Facebook, Instagram, Twitter, Reddit)

And various AI-models and script integrations

Use Case

Organize Mobile Libraries

of course, also on desktops...

Keywords and Topics

Quickly find fotos by keyword.
Automated topic calculation.

Search

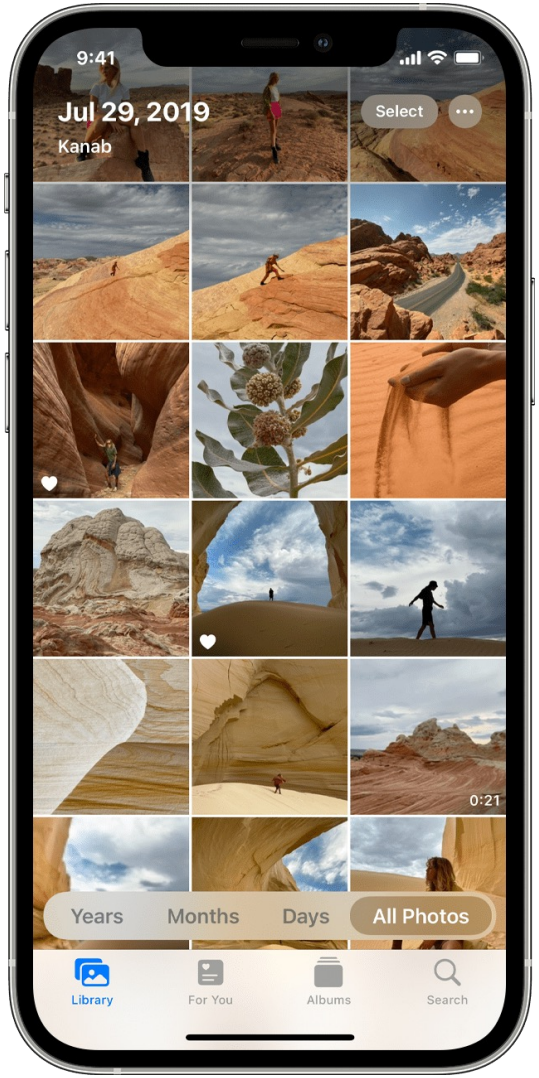
Find and organize similar fotos
by highlighting the differences
and discriminative features

Group and Create

Automated grouping in
Albums or Slideshows.

Scaling

Fast and easy maintenance
of large mobile libraries.



Use Case

Local Libraries.

Many creatives manage their libraries locally. This means, that e.g. from a photo-shooting, hundreds of (almost) similar pictures are stored in a local library.

The „Aboutness“ metric can automatically organize libraries, albums, and similar assets. Preview images representing the most important features of a library can be calculated, and differences between images be highlighted.
Furthermore, relevant aspects of Multimedia assets can be detected according to the overall library. Common features of a collection will be eliminated as irrelevant (e.g., when a model wears the same clothes during the photo-shooting), while discriminating features will be highlighted (e.g., when a model shows a coffee mug into the camera).
Features can be automatically highlighted, visualized, used for querying, filtering, or result presentation. As all these calculations can be performed by using the computer‘s GPU (and thus in parallel) and are extremely fast – independent from the collection‘s size.

Graph Codes can be calculated for any Multimedia asset, e.g. photos, videos, text-documents (including MS Word), audio files, etc. Therefore, Graph Code enabled applications can benefit from cross-media indexes and retrieval. Also Desktop-search benefits significantly from Graph Codes.

Use Case

Stock Material.

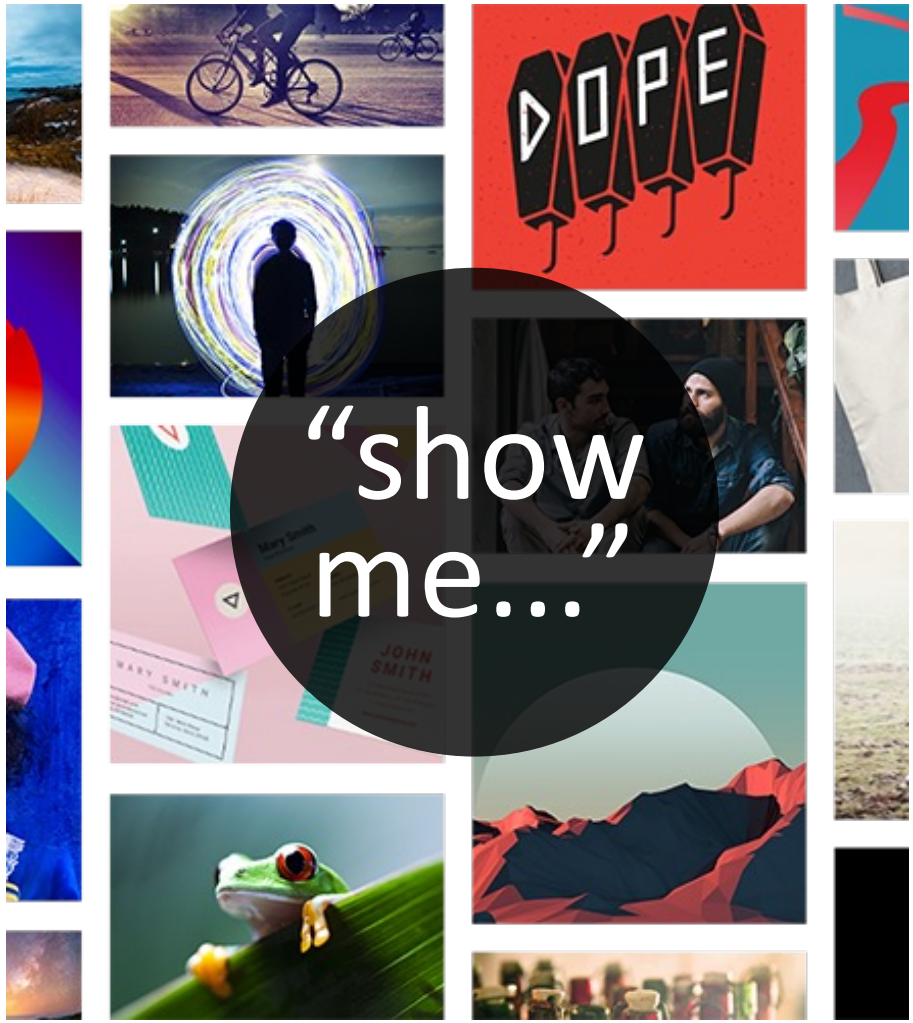
LARGE SERVER-SIDE COLLECTIONS

Semantic Keyword-based Retrieval

Particularly, for server-side collections, fast and effective retrieval is essential. With Graph Codes, entered keywords can be automatically transformed to semantic concepts and matched against the library.

New content can be annotated automatically to support effective querying. Furthermore, recommendations, similar, or discriminative-similar assets can be found fast and effective.

If users allow access to their local assets, Graph Codes can calculate requested topics, missing assets (or parts of assets), and with AI-support provide automated suggestions to creative users.



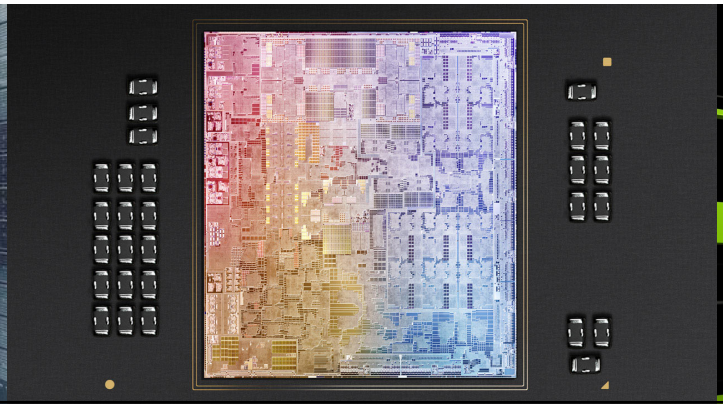
Use Case

Large Multimedia databases on internet or intranet servers, cloud applications, or at service providers benefit from these algorithms and metrics in two respects: 1) a higher level-of-detail in feature processing can be achieved when Graph Codes are used for indexing, and 2) the scalability of Multimedia databases can be improved.

Graph Codes can not just be parallelized ”in“ a machine (vertical scaling), the processing can also be distributed „across“ machines (horizontal scaing), providing even more flexibility and computing power.

Large scale GPU server farms can be employed to scale both vertically and horizontally. Scaling without limits and utilizing available hardware allows unseen performance in the area of Multimedia processing.

Server-Side Processing.



Use Case

Social Communities.

1 **Level-Of-Detail**

The Graph Code algorithms and metrics provide access to a higher level-of-detail in text, images, videos, audio, and any combination of these.

2 **Large Collections**

The number of Multimedia assets maintained by Social Community providers is enormous. Graph Codes can speedup processing and retrieval times.

3 **Feature based search**

Independent from the Multimedia type, features are harmonized and integratable. Querying and retrieval is much more effective.

Use Case

“Multi”-Media.

As Multimedia is everywhere, Graph Codes and the corresponding algorithms and metrics do not only apply for consumer products.

Also many applications in other business domains (e.g., VR/AR, medical applications, automotive, finance) can benefit a lot from employing fast and effective algorithms on a feature level. Common in such business domains is, that typically the level-of-detail and the amount of assets is even higher than in the consumer areas.

Particularly, VR/AR devices have limited computing resources but huge data to process. Therefore, fast and efffective algorithms are crucial to many applications.

Approved.

Various peer-reviewed publications

Fast and Effective Retrieval for Large Multiemdia Collections (2021, MDPI): https://www.mdpi.com/2504-2289/5/3/33

Automated Semantic Explainability of Multimedia Feature Graphs (2021, MDPI): https://www.mdpi.com/2078-2489/12/12/502

Graph Codes – 2D Projections of Multimedia Feature Graphs for Fast and Effective Retrieval (2021, Waset): https://publications.waset.org/vol/180

Parallelization strategies for Graph-Code-Bases Similarity Search (2023, MDPI): https://www.mdpi.com/2504-2289/7/2/70

Smart Multimedia Information Retrieval (2022, MDPI): https://www.mdpi.com/2813-2203/2/1/11

Semantic Query Construction and Result Representation (2021, BIRDS, 52-66): http://ceur-ws.org/Vol-2863/#paper-06

AI-based Semantic Multimedia Indexing and Retrieval for Social Media on Smartphones (2021, MDPI): https://www.mdpi.com/2078-2489/12/1/43

Integration.

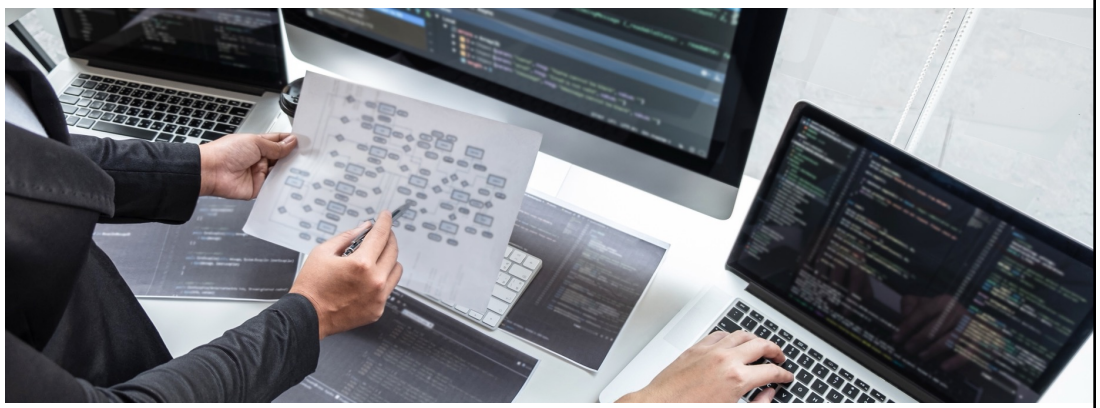
HOW TO INTEGRATE

You know your products the best!

Therefore, the integration of Graph Code algorithms and metrics should be done in the core of your products, by your developers, according to your development process and your quality criteria. Demo-Implementations in Java, CUDA, Objective-C are available. Additionally, a complete standalone Multimedia Information Retrieval Framework can be used in terms of feature extraction, different Multimedia asset types, or integration with existing standards.

To maximize the benefit for your application and business-case, a know-how-transfer and consulting phase is required, where your developers deeply understand the technology behind the algorithms and metrics and where adjustments are made according to your requirements.

The final solution is completely your's. No licenses, no copyright topics, no dependencies.



Contact.

Prof. Dr.-Ing. Stefan Wagenpfeil
E-Mail: sw@stefan-wagenpfeil.de
Web: http://www.stefan-wagenpfeil.de
Skype: stefan_wagenpfeil
Mobile: +49/160/4071976