

A Custom AI-Assisted Search Engine for 3D Architectural Assets

Integrating Text and Image-Based Queries

Miro Roman¹, Jorge Orozco²

¹University of Innsbruck, ²ETH Zurich

¹miro.roman@uibk.ac.at, ²jorge@marksearch.online

Abstract. *This paper presents search0more, a custom AI-assisted search engine developed as a specialised instrument for architectural design, deployed online and tested in an academic environment over the past year. The system aggregates 19,123 assets — physically based rendering (PBR) materials, 3D models, and curated raster images — from multiple external repositories including BlenderKit and other online asset libraries. It encodes them as both textual and visual representations to support cross-modal retrieval. Users can query the database with natural language text prompts or reference images, and results can be directly imported into common 3D modelling environments such as Blender. A central contribution lies in demonstrating how domain-specific, intentionally biased databases can be constructed and navigated in the context of contemporary AI-driven design workflows, and subsequently integrated with large language models and text-to-image systems such as ChatGPT and Midjourney. The paper details the complete technical pipeline — data aggregation, encoding, clustering, and retrieval — and discusses the implications of specialised search infrastructures for informed creativity in architectural design. search0more is a research project developed by Jorge Orozco and Miro Roman at the House of Coded Objects, UIBK's Institute für Gestaltung Studio2, in collaboration with Adil Bokhari from the Chair for Digital Architectonics at ETHZ.*

Keywords. *3D search engine, self-organizing maps, image and text queries, architectural databases, AI-driven design tools.*

INTRODUCTION

Architectural design has always been a discipline that works through material: through surfaces, objects, spatial envelopes, and their combinations. The shift toward digital workflows has not diminished this dependency but transformed it — the materials architects now work with are increasingly drawn from online repositories of 3D models, PBR textures, raster images and other

media. They are assembled into virtual environments and rendered for evaluation, communication, and fabrication. The attitude developed in this project is not to build a better general-purpose search engine, but to build a specialised one. search0more is a custom AI-assisted search engine conceived as a design support instrument: purpose-built for architectural assets, intentionally biased toward the vocabularies and material categories relevant to

architectural design practice, and structured for direct integration into design workflows. As AI develops at an accelerating pace, the relevance of specialised and curated databases grows in parallel. The design logic of the system rests on two interconnected arguments. The first concerns cross-modal encoding: the observation that most contemporary online media — 3D assets, images, texts, videos — are described and navigated through combinations of visual representations and textual descriptors (Baltrusaitis, Ahuja and Morency, 2019). Encoding all assets as both image vectors and text vectors enables a unified mode of querying that does not privilege one modality over the other; users can query by description, by example image, or by navigating similarity chains from any retrieved asset. The second argument concerns domain specificity: the claim that 'what is an architectural library' is an open question and can be played with through the design of a library, encoding the conceptual distinctions that matter to specific architectural practice but in relation to those inherited from general-purpose image classification and large-scale language model training. These two arguments together define the scope and ambition of the project. The system integrates AI-based modelling and clustering techniques — Latent Semantic Indexing (Deerwester et al., 1990) over a domain-specific vocabulary for text, and VGG19 feature extraction (Simonyan and Zisserman, 2014) and Self-Organizing Map projection (Kohonen, 1990) for images — to enable seamless retrieval of assets from a unified corpus. Assets retrieved through the system can be directly imported into Blender (Blender Foundation, 2023), and the retrieval API is designed for integration with external AI systems. Rather than operating as an isolated tool, search0more is conceived as one component within a broader ecosystem of domain-specific computational instruments, each addressing different modalities of architectural knowledge while sharing a common logic of curated, biased, and design-oriented informational

instruments. Within this ecosystem, 'ask.alice' provides search over a curated library of more than 1500 books (Roman, 2018); 'Panoramas of Cinema' (Orozco, 2019) applies machine learning to cinematographic image retrieval. Together, these three apps form a set of instruments for context and information driven design. They cover written arguments, moving image archives, and 3D assets, and are deployed in studio teaching at the UIBK Innsbruck and ETH Zurich. Their development reflects a commitment articulated in a broader research programme (Bühlmann, Hovestadt and Moosavi, 2015): that computation and code are a form of literacy and should be treated as such. The eCAADe community has published growing work on AI-assisted design support: Pestriakova, Schubert and Petzold (2025) demonstrate how machine learning can structure design space exploration while preserving creative agency; Kim, Lee and Kim (2025) present a deployed system for structured spatial information extraction from panoramic imagery; Demir, Uçar and Türkmen (2025) build a natural language interface for urban spatial data; Park et al. (2025) develop a graph neural network-based recommendation system for house floor plans, using spatial relationship similarity to retrieve design references. The present work extends this trajectory with emphasis on cross-modal retrieval over 3D assets, (PBR) materials, and images - the layer that conditions which material enters the design process at all. The search0more search engine has been publicly deployed at <https://search.0more.net> and tested in academic teaching environments over the past year, operating as live app rather than a research prototype.

BACKGROUND

Domain-Specific Retrieval: Exploration Rather Than Specification

What design requires is not faster lookup but a different navigational logic — one that supports exploration rather than specification. A designer

searching for materials rarely begins with a complete description; the process is associative, moving from an approximate intention through visual and conceptual adjacency toward something not yet fully imagined. Two entry points serve this exploration: a text query initiates traversal from a conceptual direction, while an image query initiates it from a perceptual reference. Both map into the same continuous representational space, and from any retrieved asset, similarity chains extend outward through pre-computed neighbourhoods. The corpus index is structured to support this iterative practice: new assets can be encoded and appended, allowing the retrieval space to grow incrementally alongside an expanding collection. Bühlmann, Hovestadt and Moosavi (2015) articulate a broader framework for understanding computing as a medium for architectural thought, one that foregrounds the experimental dimension of computational instruments. In their conception, it is not sufficient to build systems that retrieve correct results; they should encode implicit assumptions about what counts as relevant and make those assumptions operational for users. This orientation directly informs the design of search0more. It can build on a domain-specific corpus and inherit representations from general-purpose models.

Text-Based Retrieval: From TF-IDF to Neural Embeddings

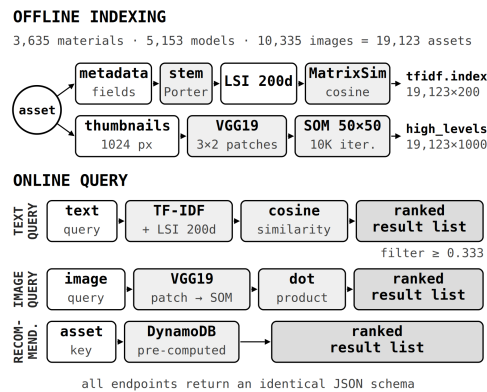
The vocabulary mismatch problem (Furnas et al., 1987; Manning, Raghavan and Schütze, 2008) — the failure of literal keyword search when query vocabulary and indexing vocabulary diverge — is the information retrieval problem this system addresses. Latent Semantic Indexing (Deerwester et al., 1990) resolves it by applying Singular Value Decomposition to the term–document matrix, projecting both documents and queries into a lower-dimensional latent space where semantically related terms cluster regardless of surface form. A query for "rough stone wall" retrieves an asset described as "weathered limestone

surface" when both project to nearby regions of the latent space. Each transformation in the pipeline introduces a lossy statistical compression: stemming collapses morphological variants, TF-IDF discounts corpus-frequent terms, SVD truncates variance below the 200th principal component. The representational space is thus a probabilistic construct shaped by the distributional properties of the corpus — but at query time, retrieval is deterministic: cosine similarity against the full index. The system occupies a characteristic middle ground: its representations are the frozen residue of probabilistic processes, while its inference remains algebraically exact. For a domain-specific corpus of 19,123 assets, 200 latent dimensions capture the dominant conceptual structure without overfitting to surface vocabulary — consistent with the recommendations of Deerwester et al. (1990) for corpora of this scale. Contemporary neural approaches — Word2Vec (Mikolov et al., 2013), Sentence-BERT (Reimers and Gurevych, 2019), CLIP (Radford et al., 2021) — offer richer semantic representations and cross-modal retrieval, but require GPU inference, large model weights, and domain-specific fine-tuning. For a domain-specific system, LSI provides a principled, self-contained alternative: interpretable latent dimensions, fast matrix-multiplication query serving, no external API dependencies, and training entirely from the corpus itself.

Image-Based Retrieval and Visual Encoding

Content-based image retrieval through deep convolutional networks produces semantically meaningful feature representations that can be compared by distance. VGG19 (Simonyan and Zisserman, 2014), pretrained on ImageNet (Deng et al., 2009), extracts activation vectors capturing features from low-level texture and edge responses to higher-level semantic categories. Self-Organizing Maps (Kohonen, 1990) provide unsupervised topology-preserving dimensionality

reduction: similar feature vectors map to adjacent positions on a 2D grid, organising the visual feature space without labelled data. In the image pipeline, asset thumbnails are subdivided into spatial patches and passed through VGG19, yielding per-patch probability distributions over 1,000 visual categories. These are projected onto a Self-Organizing Map trained through unsupervised learning and mean-pooled into a single per-asset vector. The resulting feature space inherits the statistical structure of ImageNet pretraining while serving as a fixed index at query time. The subdivision of each thumbnail into a 3 ×



2 spatial grid before extraction captures the whole dataset as an 'image based dictionary' (Kohonen, 1990) — a topology-preserving map in which proximity encodes visual similarity.

SYSTEM ARCHITECTURE

The system is structured around two clearly separated phases (Figure 1): an offline corpus construction and encoding phase, computationally intensive but infrequent; and an online query-serving phase, lightweight and handling all live traffic. This separation is the central architectural

decision — it enables real-time semantic retrieval on commodity infrastructure without GPU resources.

Corpus Construction and Data Aggregation

The corpus aggregates assets from three source domains, totalling 19,123 entries (Figure 2):

- BlenderKit PBR materials (3,635 assets): surface shaders with metadata including category, verbal description, name, and tag arrays. Materials span architectural surface vocabularies from concrete and stone to fabric and glass, with rich textual descriptions authored by their creators.
- BlenderKit 3D models (5,153 assets): geometric objects with metadata including category, object label, verbal description, name, and tags, spanning furniture, structural elements, vegetation, and fixtures — all importable directly into Blender (Blender Foundation, 2023).
- Curated PNG image collection (10,335 assets): a raster image dataset with metadata including category and label, supplemented by object-detection labels generated by a YOLO-based classifier (Redmon et al., 2016) during preprocessing.

All binary assets — the actual image files — are stored in an AWS S3 bucket; structured metadata — names, descriptions, categories, tags, and source links — in an AWS DynamoDB table, keyed by a composite source–identifier tuple. The encoding

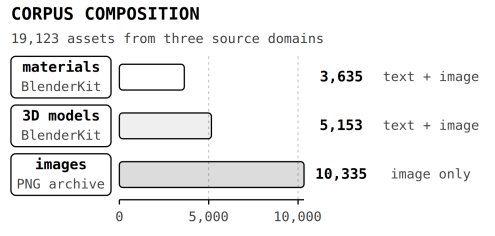


Figure 1
Two-phase system architecture: offline corpus encoding producing serialised index files; online query serving for text, image, and recommendation queries.

Figure 2
Corpus composition: asset counts per source domain and metadata schema by source type.

pipeline runs entirely against local serialised index files, with no database calls during offline processing. At query time, DynamoDB is consulted only at the final step, to enrich ranked results with metadata before the JSON response is returned. Heavy computation stays offline and self-contained; the live database handles only lightweight lookups.

Figure 3

Data encoding pipelines: the text path (top) runs from raw metadata through stemming, TF-IDF weighting, and LSI projection, while the image path (bottom) runs from thumbnail patches through VGG19 feature extraction, SOM mapping, and dot-product similarity.

Text Encoding Pipeline

Each asset's metadata fields are concatenated into a single document string. This passes through a ten-step normalisation and encoding pipeline, illustrated in Figure 3 with concrete examples:

- Concatenation of category, description, label, name, and tag fields.
- Lowercasing and alphanumeric filtering, removing punctuation and apostrophes.
- Stop-word removal using a standard English stop-word list.
- Porter stemming (Porter, 1980), reducing each token to its morphological root — applied identically at corpus build time and query time, ensuring vocabulary alignment.
- Dictionary construction, producing 11,602 unique stems across all 19,123 documents, stored as a Gensim object (Řehůřek and Sojka, 2010).
- Bag-of-words encoding to sparse (token_id, count) vectors.
- TF-IDF reweighting, amplifying distinctive terms and suppressing corpus-wide common terms.
- LSI projection via truncated SVD with 200 retained dimensions.
- MatrixSimilarity index (tfidf.index): the 19,123 × 200 projection matrix serialised for fast cosine similarity lookups at query time.
- Asset pointer index (Mi.pickle): (source, category, asset_id, thumbnail_path) tuples indexed to match the MatrixSimilarity row ordering.

Image Encoding Pipeline

Each thumbnail is subdivided into a 3 × 2 spatial grid of patches. VGG19 (pretrained on ImageNet (Deng et al., 2009), weights frozen) extracts the 1,000-dimensional softmax vector from each patch independently. Patch feature vectors from the full

TEXT ENCODING PIPELINE

example: "concrete wall material with rough texture"

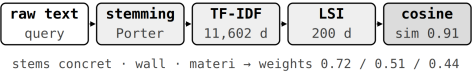
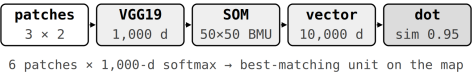


IMAGE ENCODING PIPELINE

example: one reference photograph, 1024 px thumbnail



corpus are collected and a 50 × 50 Self-Organizing Map is trained for 10,000 iterations, producing a topology-preserving arrangement of the visual feature space. Per-asset mean activation vectors are stored as high_levels.pickle (approximately 172 MB). At query time, a submitted image's VGG19 patch features are compared against the stored corpus vectors via dot-product similarity, returning ranked results by visual similarity. Figure 3 shows both encoding paths with step-by-step data examples.

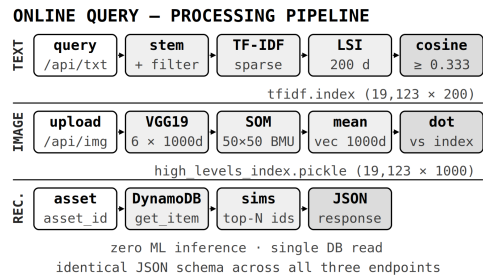
Online Query Serving

The deployed application is a Flask web server running on AWS Elastic Beanstalk (eu-central-1) with three API endpoints, each loading a subset of the serialised index files at startup. Figure 4 illustrates the query processing flow for all three modes. The system exposes its functionality through a simple HTTP interface, enabling both direct user interaction and integration into external computational workflows. For example, a text query can be issued via a request such as:

<https://search.0more.net/api/txt?q=concrete+wa>

||

Text query (/api/txt): The query string undergoes identical preprocessing to the offline pipeline — stemming, dictionary filtering, Bag-of-Words encoding, TF-IDF transformation, LSI projection — and the resulting 200-dimensional vector is compared against the MatrixSimilarity index by cosine similarity. Results below the confidence threshold (0.333) are suppressed. Image query (/api/img): The uploaded image is subdivided into the same 3 × 2 patch grid, processed through VGG19



(weights loaded from file, no GPU required), and its mean patch activation vector is compared against the stored corpus vectors by dot product. Matching image results are combined with semantically similar text-retrieved results for the same image content. Recommendation query (/api/rec): Per-asset similarity scores are pre-computed offline and stored in DynamoDB, indexed by asset identifier. At query time, a simple DynamoDB lookup returns the pre-computed nearest neighbours for any asset, with zero ML inference.

USER INTERFACE AND INTERACTION DESIGN

The user-facing interface is designed around three primary interaction modes: text search, image upload search, and asset-driven recommendation (Figure 5). The homepage presents a single search bar and upload interface against a minimal background of floating asset thumbnails. The visual

communication of the system is the corpus itself: rotating asset previews render the available material directly visible before any query is issued. Search results are presented as a grid of sphere-rendered PBR material previews, model thumbnails, or image thumbnails depending on asset type, with filtering tabs to restrict results by category. Each result carries a small confidence indicator and an expandable panel with full metadata, download options, and direct Blender import instructions. Clicking any result triggers the recommendation endpoint, surfacing visually or semantically similar assets and enabling associative browsing through the corpus rather than strict query-response cycles. The interface supports two distinct search entry points: the typed query (useful when the designer can articulate a description) and the image upload (useful when the designer has a visual reference but not a vocabulary). Both converge on the same ranked result presentation. The separation of text and image retrieval at the backend is invisible to the user — from the interface, the system behaves as a unified visual-linguistic search environment.

DEPLOYMENT AND EVALUATION

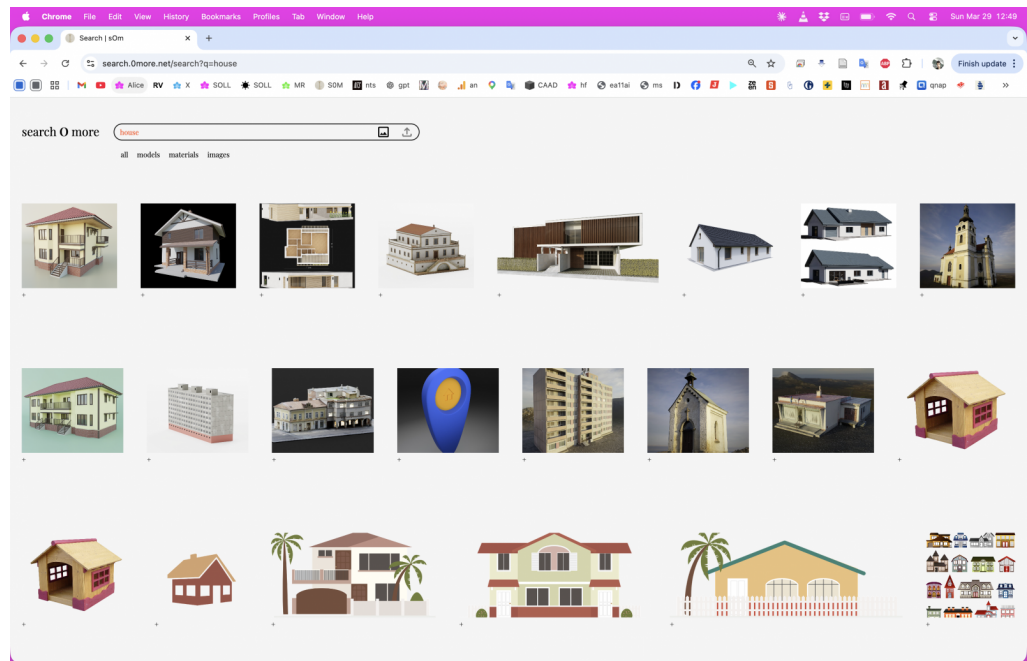
search0more has been publicly accessible at <https://search.0more.net> for over a year, operating as live infrastructure for studio teaching at UIBK Innsbruck and ETH Zurich. Students and researchers have used the system as a standard tool for asset discovery in design projects involving rendering, modelling, and computational workflows. The goal of the evaluation presented here is not to establish performance superiority over existing platforms, but to demonstrate that the system behaves coherently with its intended role as a multimodal, domain-specific retrieval instrument for architectural design. To assess retrieval consistency across modalities, a set of 15 representative queries was constructed spanning three categories: material-oriented text queries (e.g. "weathered concrete", "translucent glass panel", "rough stone wall"), object-oriented text queries (e.g. "wooden

Figure 4
Online query processing flow: text query (top), image query (middle), and recommendation query (bottom), all serving from pre-computed indexes without runtime GPU inference.

chair", "steel railing", "concrete column"), and image-based queries using reference photographs of architectural surfaces and objects. A result is considered relevant if it aligns with the intended material, object category, or visual characteristics implied by the query. Across this set, text-based queries consistently return visually and semantically coherent results, while image-based queries of similar subjects retrieve assets with comparable texture and material characteristics. This cross-modal alignment — where a photograph of a brick surface and the text query "brick wall" converge on overlapping result sets — indicates that the shared embedding space supports retrieval across both modalities without privileging either one. The system demonstrates particular strength in responding to architectural vocabulary. Queries using descriptive, material-oriented language

("load-bearing column", "perforated facade panel", "brushed aluminium") consistently return assets that reflect materiality and construction logic rather than generic object labels. This domain alignment is a direct consequence of building the vocabulary, TF-IDF weights, and latent dimensions from the corpus itself: the semantic space encodes distinctions that matter in architectural practice because it was trained on mostly architectural metadata. In contrast to general-purpose search systems, which tend to prioritise object identity or tag popularity, the system responds to the kind of imprecise, descriptive, and material-oriented queries that characterise early-stage design search. A key limitation in this regard is coverage asymmetry between asset categories: BlenderKit materials and models carry rich creator-provided metadata, whereas the PNG image collection

Figure 5
The search0more interface: search results for "concrete wall" showing sphere-rendered PBR material previews with filter tabs.



contains comparatively sparse textual annotations. YOLO-generated object-detection labels partially compensate for this imbalance; however, image assets remain systematically disadvantaged in text-based retrieval. Interaction patterns observed over the deployment period indicate that users frequently engage in non-linear exploration through the recommendation system, typically navigating through 4–6 associative steps before identifying a target asset. Rather than converging immediately on a single result, users treat the system as a space to traverse — refining, expanding, or redirecting their search trajectory through similarity chains. This behaviour suggests that the system operates not only as a retrieval tool but as a navigable design space, supporting the kind of associative, open-ended exploration that characterises design work. The system is not optimised for ranking performance in a competitive benchmark setting; it is designed to consistently return meaningful results across a wide range of inputs, including underspecified or ambiguous queries where conventional keyword search returns nothing. The emphasis on robustness, interpretability, and modularity — with encoding, indexing, and retrieval components fully decoupled — positions the system as an adaptable infrastructure capable of integrating future advances in multimodal representation learning without restructuring the overall pipeline.

DISCUSSION

Cross-Modal Encoding as Navigation Strategy

Most contemporary digital media carries both a visual form and a textual description — complementary facets of the same object. Encoding both and making both available for query is a structural response to how digital media is found online. This dual encoding enables navigation that neither pure text search nor pure image search supports alone: a designer can upload a reference photograph when keywords fail, follow

visual similarity chains from a retrieved asset, or pass LLM-generated descriptions directly to the text API. The modalities are interchangeable entry points to the same representational space.

The Case for Intentional Domain Bias

A recurring design decision has been to deliberately constrain scope: specific source repositories, a vocabulary calibrated to architectural practice, clustering models trained on domain-specific data. This intentional bias is sometimes framed as a limitation. The argument of this paper is that the bias is the contribution. A retrieval system covering all domains is calibrated to no particular domain's actual needs. The architectural asset search problem is about finding objects whose character, scale, material behaviour, and formal vocabulary are relevant to architectural design — distinctions encoded in the metadata conventions and quality standards of architectural repositories. A car enthusiast domain has a different probabilistic space or 'flavour' than an architecture enthusiast. The parallel with companion systems is instructive. 'ask.alice' and 'Panoramas of Cinema' operate on similarly domain-specific corpora, each structured around particular knowledge forms of its field. Their value lies in the depth of coverage, consistency of metadata, and alignment between retrieval vocabulary and the vocabulary of their communities.

Integration with Broader AI Workflows

The longer-term significance of domain-specific retrieval libraries lies in their role as structured knowledge sources for larger AI workflows. The search0more API returns machine-readable JSON — asset identifiers, metadata, similarity scores, download URLs — consumable directly by large language models or automated systems (e.g. <https://search.0more.net/api/txt?q=concrete+wall>). The most immediate integration is through the Blender Python API and LLMs such as Claude or ChatGPT: retrieved assets can be downloaded and imported into Blender scenes programmatically

via bpy, transforming the search system from a browser-based tool into a pipeline component. Extending further, image-generation tools like Midjourney accept reference images — assets from search0more can serve as such references, and conversely, generated images can be uploaded to locate physically based assets matching the generated visual language. The retrieval system bridges AI-generated imagery and the deployable, physically accurate asset library. Domain-specific retrieval libraries are increasingly the missing layer between general-purpose AI systems and design practice. As architectural workflows incorporate more AI-generated material, the value of this infrastructure grows correspondingly.

CONCLUSION

This paper presented search0more, a domain-specific, cross-modal search system for architectural assets, and articulated three main contributions through its implementation. First, it demonstrates the integration of text- and image-based retrieval within a single system, enabling queries that reflect the dual nature of contemporary design media as both visual and textual. Second, it establishes domain specificity as a deliberate design strategy: rather than pursuing general-purpose coverage, the system operates within a curated and biased dataset, allowing retrieval to align with the vocabularies, materials, and associations of a particular design community. Third, it positions the system as part of a broader AI workflow, exposing its functionality through an API that enables integration with generative models and other computational design tools. Together, these elements suggest a shift from universal search toward situated retrieval infrastructures, where value emerges through alignment, coherence, and usability within a specific domain. In this context, search0more is not conceived as a tool for optimal ranking, but as an instrument for navigating a probabilistic design space, supporting exploration, association, and iterative design

processes. Future work will extend the modular architecture with additional modalities and evolving models, while maintaining domain specificity as a central principle for structuring interaction between data, tools, and design practices.

REFERENCES

- Baltrusaitis, T., Ahuja, C., & Morency, L.-P. (2019). Multimodal Machine Learning: A Survey and Taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2), 423–443. <https://doi.org/10.1109/TPAMI.2018.2798607>
- Blender Foundation. (2023). *Blender*. <https://www.blender.org/>
- Bühlmann, V., Hovestadt, L., & Moosavi, V. (2015). *Coding as Literacy: Metalithikum IV* (V. Bühlmann, L. Hovestadt, & V. Moosavi, Eds.; Vol. 4). Birkhäuser.
- Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6), 391–407. [https://doi.org/10.1002/\(SICI\)1097-4571\(199009\)41:6<391::AID-AS11>3.0.CO;2-9](https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-AS11>3.0.CO;2-9)
- Demir, B. K., Uçar, B., & Türkmen, İ. U. (2025). A Data-Driven Interface for Understanding Urban Interactions in Public Green Spaces. In A. Gönenç Sorguç & others (Eds.), *Confluence, Proceedings of the 43rd eCAADe Conference* (Vol. 1, pp. 317–326). METU. <https://doi.org/10.52842/conf.ecaade.2025.1.317>
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 248–255. <https://doi.org/10.1109/CVPR.2009.5206848>
- Furnas, G. W., Landauer, T. K., Gomez, L. M., & Dumais, S. T. (1987). The Vocabulary Problem in Human-System Communication. *Communications of the ACM*, 30(11), 964–971.

- <https://doi.org/10.1145/32206.32212>
- Kim, Y., Lee, J.-K., & Kim, H. (2025). Development of 360SpatialAI: Indoor Spatial Information Extraction and Management Web App Based on 360 Image and AI Technology. In A. Gönenç Sorguç & others (Eds.), *Confluence, Proceedings of the 43rd eCAADe Conference* (Vol. 1, pp. 269–276). METU.
<https://doi.org/10.52842/conf.ecaade.2025.1.269>
- Kohonen, T. (1990). The Self-Organizing Map. *Proceedings of the IEEE*, 78(9), 1464–1480.
<https://doi.org/10.1109/5.58325>
- Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *Proceedings of the International Conference on Learning Representations (ICLR) Workshop*.
<https://doi.org/10.48550/arXiv.1301.3781>
- Orozco, J. (2020). *Panoramas of Cinema*.
<https://panoramasofcinema.ch>
- Park, H., Heo, M., Seo, Y., Panya, D. S., Gu, H., & Choo, S. (2025). Expert Evaluation of a House Floor Plan Recommendation System Using Graph Neural Networks. In A. Gönenç Sorguç & others (Eds.), *Confluence, Proceedings of the 43rd eCAADe Conference* (Vol. 1, pp. 183–189). METU.
<https://doi.org/10.52842/conf.ecaade.2025.1.183>
- Pestriakova, E., Schubert, G., & Petzold, F. (2025). Navigating Design Space in the Creative Process: Integrating Machine Learning for Structured Exploration Avoiding Design Fixation. In A. Gönenç Sorguç & others (Eds.), *Confluence, Proceedings of the 43rd eCAADe Conference* (Vol. 1, pp. 713–722). METU.
<https://doi.org/10.52842/conf.ecaade.2025.1.713>
- Porter, M. F. (1980). An Algorithm for Suffix Stripping. *Program*, 14(3), 130–137.
<https://doi.org/10.1108/eb046814>
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning Transferable Visual Models From Natural Language Supervision. *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 8748–8763.
<https://proceedings.mlr.press/v139/radford21a.html>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788.
<https://doi.org/10.1109/CVPR.2016.91>
- Řehůřek, R., & Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 45–50.
<https://is.muni.cz/publication/884893/en>
- Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 3982–3992.
- Roman, M. (2020). *Ask Alice*.
<https://ask.alice-ch3n81.net>
- Roman, M. & Alice_ch3n81. (2022). *Play Among Books: A Symposium on Architecture and Information — Spelled in Atom-Letters*. Applied Virtuality Book Series, Vol. 17., Birkhauser.
<https://www.degruyterbrill.com/document/doi/10.1515/9783035624052/pdf>
- Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv Preprint arXiv:1409.1556*.
<https://doi.org/10.48550/arXiv.1409.1556>