

Material Memex: Automatic Material Suggestions for 3D Objects

Arjun Jain Thorsten Thormählen Tobias Ritschel Hans-Peter Seidel

MPI Informatik



Figure 1: Given an input query 3D object without materials (left) our approach automatically assigns materials (center) using database information and suggests alternatives to the user (insets) which can be selected interactively to improve the automatic assignment (right).

Abstract

The material found on 3D objects and their parts in our everyday surroundings is highly correlated with the geometric shape of the parts and their relation to other parts of the same object. This work proposes to model this context-dependent correlation by learning it from a database containing several hundreds of objects and their materials. Given a part-based 3D object without materials, the learned model can be used to fully automatically assign plausible material parameters, including diffuse color, specularity, gloss, and transparency. Further, we propose a user interface that provides material suggestions. This user-interface can be used, for example, to refine the automatic suggestion. Once a refinement has been made, the model incorporates this information, and the automatic assignment is incrementally improved. Results are given for objects with different numbers of parts and with different topological complexity. A user study validates that our method significantly simplifies and accelerates the material assignment task compared to other approaches.

Links: [DL](#) [PDF](#) [WEB](#) [VIDEO](#) [DATA](#)

1 Introduction

Assigning materials to parts of a 3D object is a difficult and time consuming task that is performed by specially trained color & lighting artists in movie or game productions. The chosen palette of materials strongly influences the overall appearance of the 3D scene and is essential to allow the object to fit into an environment. There is a wide range of different materials, e. g., for a car there is the specular metallic paint work, the diffuse rubber material on the tire, the aluminium of the rim, the fabrics and leather used in the interior, etc. The compositing of materials is also important, as for example all screws on a tire should not just be metallic, but are likely to be of

the same metallic material. Furthermore, materials are influenced by their context, e. g., for a part of a car’s interior, leather or wood are far more likely materials than for a part of the car’s engine.

Despite these observations, current content creation packages assign materials using a tedious manual process, involving the adjustment of rarely intuitive parameters, or by selecting pre-defined materials from a database using a keyword search. Even an experienced artist requires approximately 45 minutes to assign appropriate materials to all 130 parts of the car shown in Fig. 1.

In this work, we propose an approach to computationally model the relation of shape and material by learning it from a database of hundreds of multi-component 3D objects with materials. This model can then be used to automatically assign materials to 3D objects or can be employed in a user-interface to provide a ranked list of the most likely materials (see Fig. 1).

The paper comprises the following contributions:

- A model of the relation between materials and shape as well as context, called the *material memex*,
- Automatic assignment of materials using this model.
- A novel interface to guide a user when assigning materials by providing ranked material suggestions.
- A user study of task performance when using conventional slider or text interfaces compared to our interface.

The rest of the paper is structured as follows. The next section discusses related work. Section 3 introduces the proposed material memex. Section 4 presents two applications of the material memex: automatic material assignment and ranked material suggestion. Results are given in Section 5, and the paper ends with a discussion and conclusion.

2 Related Work

Material Assignment. Exploiting the relations of materials and shapes so far received only limited attention in the computer vision community, and even less interest from computer graphics. For editing and assigning material, different material design interfaces are employed. In most commercial 3D modelling tools it is still common to directly modify the parameters of the analytic reflectance model such as Phong [1975]. Ngan et al. [2006] propose an interface for BRDF selection that displays material variations with several preview images. There are also special solutions [Kautz et al. 2007;

Pellacini and Lawrence 2007] to edit spatially-varying material representations. Kerr and Pellacini [2010] have performed a user-study to evaluate material design interfaces with either physical sliders, perceptual sliders, or preview image navigation.

Texture Transfer. Closely related to our approach are texture transfer methods [Mertens et al. 2006; Lu et al. 2007]. These approaches model the statistical relationship between local geometric properties, such as curvature and local statistics of reflectance. Using this relation, a texture synthesis on a new object produces shape-dependent textures that capture, e. g., weathering. Our approach is different, as it is neither considering the statistics of local shape descriptors nor the statistics of reflectance. Instead, we work on high-level structure, such as spatial arrangement, shape, and material similarities to capture the global organization instead of local statistics. Chajdas et al. [2010] propose a system that assists a user to assign textures in large virtual environments. A user-provided texture assignment is automatically propagated to similar surfaces in the environment. Textures can also be re-targeted to different surface sizes [Lefebvre et al. 2010]. In contrast, our approach focuses on material properties that are extracted from a database, such as diffuse color, specularity, glossiness, and transparency. We assume that a material is constant for a part of an object and ignore spatially varying properties typically stored in textures.

Data-driven Content Creation. Supporting artists or casual users to effectively create content has recently received much interest. Our approach is inspired by other data-driven 3D content creation tools, e. g., for 3D modeling [Chaudhuri et al. 2011], hand-drawings [Lee et al. 2011], furniture arrangement [Yu et al. 2011], image color themes [Wang et al. 2010], or segmentation and labeling of objects [Kalogerakis et al. 2010]. Fisher et al. [2010; 2011] use a database of objects to search for a suitable object that fits into a given spatial context. They employ the Visual Memex Model from Malisiewicz and Efros [2009] that stores associations between entities instead of categorizing them. The Material Memex model presented in this paper also adopts this methodology but applies it in the application domain of material assignment. Furthermore, because our approach estimates the materials of several parts of an object simultaneously, a more complex probabilistic framework must be employed.

3 Material Memex

Input to our approach is a multi-component 3D object where parts have no assigned material. Output is a suggestion for a suitable material for each part. The part-material relation is learned from a database of multi-component 3D objects.

We follow the *memex model* recently popularized by Malisiewicz and Efros [2009], which is based on Vannevar Bush’s [1945] early concept of a memory extender (memex). The memex does not structure information using categories, but rather stores associations between entities. As most entities have no strong pairwise associations, the memex is typically a sparse representation. Malisiewicz and Efros show that this category-free memex model outperforms category-based approaches on the challenge to detect an object in a 2D image using only contextual cues [Torralba 2003].

Similarly, we propose to store associations between parts and materials of database objects in a *material memex* to generate a suitable material suggestion for a user-provided query object. Using the sparse memex representation, the approach can efficiently compute a likelihood value for each possible material for a part of the input query object. The computed likelihood value can be used, for example, to display a ranking of the best-fitting materials for a part of

the query object to the user. The likelihood value depends on the shape similarity between parts, i. e., the probability for a candidate material is high if there are parts in the database that have a shape similar to the query part and a material similar to the candidate material. Furthermore, the likelihood value depends on the context of the part. In this paper, the context information is captured by the pairwise spatial relation between two parts that are in physical contact. The probability for a candidate material is high if there are two parts in the database which are in physical contact and have similar materials and a similar spatial relationship as the query part and one of its contextual parts in the query object. Additionally, it is very likely that parts with similar shape in the query object have similar materials. For larger databases and query objects with many parts, the sparse memex representation, which can be built once during pre-processing, is the key ingredient making it possible to return material suggestions at interactive speed.

In contrast to previous work, in our application the context is not fully defined, i. e., in our case the probability for a candidate material depends on materials of contextual parts in the query objects, which are also unknown. Consequently, the problem addressed in this paper has many unknown variables and the globally best solution must be estimated by maximizing the joint probability distribution of all variables simultaneously.

Overview In the following, we want to assemble a probabilistic factor graph that represents the joint probability distribution for all the unknown materials of the query object. The individual probability distributions for the material of each part are then inferred by the marginal distributions, which can be calculated with the sum-product algorithm (belief propagation).

The structure of the probabilistic factor graph depends on the query object and the potentials of the individual factors are dependent on the query object and all the objects in the database. In order to determine the structure of the factor graph, we need to determine which parts are in context and are consequently influencing each other. In this paper we assume, for the sake of lower computational complexity, that parts only influence each other if they are in physical contact.

The following paragraphs will introduce the entities required to analyze the input data in order to create the factor graph. First, we describe what our approach expects as input, the decomposition of objects into parts, and the contact analysis. Afterwards, measures for the spatial relation between parts, for shape similarity, and for material similarity are introduced. The defined measures will allow us to create a sparse memex graph for the query and a sparse memex graph for the database objects. The query memex graph will directly lead to the structure of the probabilistic factor graph and the database memex graph is required to efficiently compute the potentials of the individual factors.

Input. Input to the approach is a multi-component 3D object \hat{S} that has no assigned material, which is called the *query* object. Furthermore, there is a database of multi-component objects with assigned materials. This database is defined by the set of objects \mathcal{S} . This set should be large, e. g., in our experiments we used 276 objects. The individual database objects $S_m \in \mathcal{S}$ are given as polygonal meshes where each face is labelled with a material of the Phong [1975] reflection model. We use Phong in our implementation as it is a widespread reflection model for 3D objects from Internet repositories, such as Google 3D Warehouse, Turbosquid, Dosch 3D, etc. Uniform scaling is applied to all 3D objects in the database as well as the query shape. This ensures that all objects have the same size (measured on the dominant eigenvector of the covariance matrix of the mesh vertices). As we are not employing a category-based

learning approach, neither the objects, polygons, or materials of the database require any manual annotations.

Decomposition Into Parts. The material memex stores associations between parts of objects. Before these associations can be computed, all objects must be decomposed into their parts, i. e., it must be defined which faces of an object’s polygonal mesh belong to a particular part. For decomposition, we assume that the designers of the query and database objects have initially created the object from multiple parts. If this part information is still available, it is used for decomposition. However, for approximately 90 percent of our models the part information is lost, as it is typically not stored in the file formats utilized by the Internet repositories. In these cases, we recover the part information by searching for connected components in the polygonal mesh. For the database shapes it is also ensured that all polygons in the connected mesh component have the same material.

Let $\mathcal{P} := \{P_1, \dots, P_k, \dots, P_K\}$ be the set of parts that is created by the decomposition of all objects of the database into parts. As this set comprises all parts of the database, K is the number of all parts in the complete database. Every part P_k in the database has a unique material M_k . Note that in contrast to common practice in computer graphics, in the mathematical description used throughout this section, materials are not shared across different parts even if they are identical, i. e., every part has one material and every material has one part. Consequently, there is a set of known database materials $\mathcal{M} := \{M_1, \dots, M_k, \dots, M_K\}$ which has the same cardinality as the set of parts. The query object \hat{S} is decomposed in exactly the same way, resulting in a set of parts $\hat{\mathcal{P}} := \{\hat{P}_1, \dots, \hat{P}_n, \dots, \hat{P}_N\}$.

Contact Analysis For the query object \hat{S} and each database object $S_m \in \mathcal{S}$ a contact analysis is performed. The contact analysis of the query object will be later used to generate the query memex graph (which in turn captures the structure of the probabilistic factor graph). The contact analysis of the database objects is performed to compare the spatial context of a part in the query object to a spatial context of a part in the database, as will be explained in detail later.

During contact analysis we compute whether the two parts, P_i and P_j , of an object are in physical contact, i. e., are (almost) touching each other. To this end, both parts are resampled to a point cloud. The individual sampling points are placed on the polygonal surface mesh such that their distance is roughly equal (blue noise sampling). We define the minimal spatial distance $d_{i,j}^{\min}$ as the smallest Euclidean distance between any sampled point from part P_i to any sampled point from part P_j . We define two parts P_i and P_j to be in contact if their minimal spatial distance $d_{i,j}^{\min}$ is smaller than 0.01 percent of the total object size.

In the following, we describe three associations between parts that are required to build the material memex: *spatial relation* similarity for part pairings, *shape* similarity, and *material* similarity.

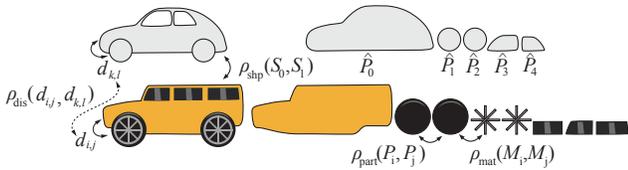


Figure 2: Prerequisites to build a material memex graph. The query object (top) and a database object (bottom) are decomposed into parts. We defined several associations between parts: spatial relation similarity ρ_{dis} , shape similarity ρ_{part} and material similarity ρ_{mat} .

Spatial Relationship of Parts Besides the structural information produced by the contact analysis, spatial relation similarity for part pairings is another important cue to determine whether a context in the query object and in the database object is similar. The Euclidean distances $d_{i,j}$ between the centers of mass of parts P_i and P_j is used to capture their spatial relationship.

In our approach, we have made the design decision to rely solely on Euclidean distances between parts to capture their spatial relationship.

The advantage of using the Euclidean distances between parts instead of a more complex spatial relation measure is that the Euclidean distance is invariant to joint rotational and translational transformation of the two involved parts. Consequently, the measure is not dependent on the choice of the local coordinate system, which is typically not consistent for the parts of the 3D objects. However, the measure is not scale-invariant, which is why we initially scale all objects to have the same size.

To define a similarity measure of two spatial relations $d_{i,j}$ and $d_{k,l}$, it is assumed that due to noise the difference between two spatial relations (that should be considered similar) obeys a Gaussian distribution with zero mean. This is motivated by the central limit theorem. We define the similarity measure of two spatial relations $d_{i,j}$ and $d_{k,l}$ as

$$\rho_{\text{dis}}(d_{i,j}, d_{k,l}) = \exp\left(\frac{-(d_{i,j} - d_{k,l})^2}{2\sigma_d^2}\right) \quad , \quad (1)$$

where σ_d denotes the standard deviation of spatial relations difference ($d_{i,j} - d_{k,l}$) that should be considered similar. We chose σ_d such that the resulting zero-mean Gaussian distribution captures the smallest $\tau_{\sigma_d} = 3.0$ percent of all spatial relation differences in the complete database of parts. This spatial similarity measure is close to 1.0 if the two spatial relations are very similar and close to 0.0 if they are very different.

Shape Similarity. The similarity $\rho_{\text{shp}}(P_i, P_j)$ between the parts P_i and P_j is calculated using the approach by Chen et al. [2003]. In our application, the scale invariance of their descriptor is undesirable. We extend it by adding three entries to their descriptor vector that represent each shape’s size in the x -, y - and z -directions measured in the space spanned by the three eigenvectors of the covariance matrix of the part’s sampling points.

We measure the dissimilarity between part P_i and part P_j using the L1-distance $|\mathbf{s}(P_i) - \mathbf{s}(P_j)|$ of the extended descriptor vectors. Assuming a Gaussian distribution of the L1-distances that should be considered similar, we define the shape similarity of parts to be

$$\rho_{\text{part}}(P_i, P_j) = \exp\left(\frac{-|\mathbf{s}(P_i) - \mathbf{s}(P_j)|^2}{\sigma_p(P_i, P_j)^2}\right) \quad . \quad (2)$$

We chose the standard deviation $\sigma_p(P_i, P_j)$ such that the resulting zero-mean Gaussian distribution captures the smallest $\tau_{\sigma_p} = 1.0$ percent of all L1-distances that involve either P_i or P_j in the complete database. Due to this definition, the standard deviation $\sigma_p(P_i, P_j)$ is a function of P_i or P_j . This is required, as we have observed in our experiments that the absolute value of the L1 distance depends strongly on the involved parts P_i or P_j . This shape similarity measure is close to 1.0 if the two parts are very similar and close to 0.0 if they are very different.

Besides geometric shape similarity between parts, shape similarity can also be computed between whole objects. In particular, we will

require a measure for the shape similarity $\rho_{\text{shp}}(\hat{S}, S_m)$ between the query object \hat{S} and an object in the database S_m , when computing the potentials of the factor graph. When computing these potentials we can weight the influence of a part higher if it belongs to an object that is similar to the query object.

To compute $\rho_{\text{shp}}(\hat{S}, S_m)$, we again employ the approach of Chen et al. but now rendering and comparing whole objects and not parts. The only difference is that we extend the aligned descriptor created by Chen et al.’s approach differently. When comparing objects we only add a single entry, which is the number of parts of the object. This is motivated by the fact that our approach works best if the query and database objects have compatible complexity because the part segmentation is more likely to be similar in this case. Adding the number of parts in the shape similarity object has the result that parts from objects with similar complexity are preferred when computing the potentials of the factor graph.

Formally, we define the shape similarity of objects to be

$$\rho_{\text{shp}}(\hat{S}, S_m) = \exp\left(\frac{-|\mathbf{s}(\hat{S}) - \mathbf{s}(S_m)|^2}{\sigma_s(\hat{S}, S_m)^2}\right) . \quad (3)$$

We choose the standard deviation $\sigma_s(\hat{S}, S_m)$ such that the resulting zero-mean Gaussian distribution would capture the smallest 15 L1-distances between \hat{S} and all database shapes $S_m \in \mathcal{S}$.

Material Similarity. The material similarity function between material M_i and M_j is denoted by $\rho_{\text{mat}}(M_i, M_j)$. We employ a custom difference of Phong reflection parameters inspired by the perceptual re-parametrization by Pellacini et al. [2000]. In particular, the specular color is parametrized by the third root of RGB color values instead of using the RGB values directly. Further, we have a parameter for the glossiness of the material (where we define glossiness to be exponent of the cosine function typically used to compute the specular component). We divide Phong glossiness by an assumed maximum of 300 and afterwards take the fourth root to linearize its perceptual influence. Consequently, our Phong material descriptor vector contains eight elements: 3 for the diffuse color values in RGB color space, 3 for the third root of specular color values in RGB color space, 1 for the fourth root of the glossiness, and 1 for transparency. All descriptor elements can take values in the range of 0.0 to 1.0.

The L1-distance $|\mathbf{m}(M_i) - \mathbf{m}(M_j)|$ between the material descriptor vectors $\mathbf{m}(M_i)$ and $\mathbf{m}(M_j)$ is used to measure the dissimilarity between material M_i and part M_j . Assuming again a Gaussian distribution of the L1-distances, the material similarity function is

$$\rho_{\text{mat}}(P_i, P_j) = \exp\left(\frac{-|\mathbf{m}(M_i) - \mathbf{m}(M_j)|^2}{\sigma_m^2}\right) , \quad (4)$$

where σ_m denotes the standard deviation over all L1-distances of materials in the complete database that should be considered similar. We chose σ_m such that it captures the smallest $\tau_{\sigma_m} = 1.0$ percent of all L1-distances in the complete database.

Database Memex Graph. For all parts in the database we extract a Memex graph $\hat{G} = (\mathcal{V}, \mathcal{E}_d \cup \mathcal{E}_s \cup \mathcal{E}_m)$ consisting of a set of nodes \mathcal{V} and the union of the three sets of edges \mathcal{E}_d , \mathcal{E}_s , and \mathcal{E}_m . The set of nodes \mathcal{V} is equivalent to the set of parts \mathcal{P} , i. e., each node represents a part in the database (Fig. 3). The set of spatial context edges \mathcal{E}_d is determined by finding parts that are in contact (as defined during the contact analysis). If two parts P_i and P_j , are in contact, their index pair (i, j) is added to the set of edges \mathcal{E}_d . Note, that only parts from the same object in the database can be in contact. This results in very sparse connectivity of nodes. In fact, the cardinality

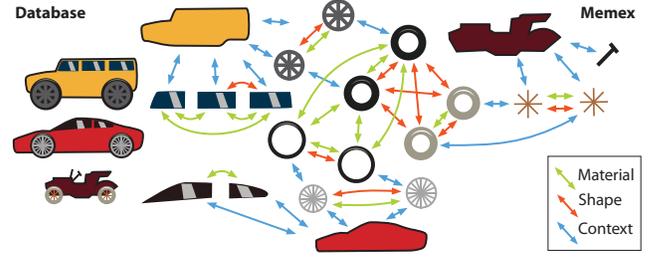


Figure 3: A database of objects (top) and the database memex graph that we extract (bottom) with spatial context, shape similarity, and material similarity edges.

of \mathcal{E}_d grows only linearly with the number of objects in the database. Additionally, the graph contains a set of shape similarity edges \mathcal{E}_s . If two parts P_i and P_j , are of similar shape, i.e., their shape similarity function $\rho_{\text{part}}(P_i, P_j)$ is above a user defined threshold, their index pair (i, j) is added to the set of edges \mathcal{E}_s . The third type of edges are material similarity edges that are added to the set \mathcal{E}_m if the material similarity $\rho_{\text{mat}}(M_i, M_j)$ is above a user-defined threshold.

Query Memex Graph. Similarly, a graph $\hat{G} = (\hat{\mathcal{V}}, \hat{\mathcal{E}}_d \cup \hat{\mathcal{E}}_s)$ for the query object \hat{S} is generated. The set of nodes $\hat{\mathcal{V}}$ is defined by the set of parts $\hat{\mathcal{P}}$ in the query object. The set of spatial context edges $\hat{\mathcal{E}}_d$ and the set of part similarity edges $\hat{\mathcal{E}}_s$ is generated in exactly the same way as for the database objects. Obviously, in contrast to the database memex graph, the query memex graph has no material similarity edges because the materials of the query object are unknown. In contrast to the database memex graph, the query memex graph cannot be computed in a pre-processing step, but rather only once the query object is defined.

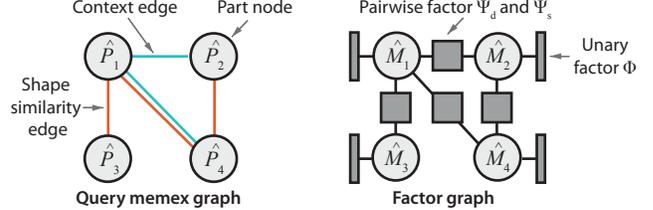


Figure 4: Converting the memex graph of the query object into a factor graph for probabilistic inference.

Probabilistic Factor Graph. Using the information stored in the query and database memex graphs, we can generate a factor graph for probabilistic inference to find an optimal material assignment for each part of the query object. The structure of the factor graph is directly given by the query memex graph $\hat{G} = (\hat{\mathcal{V}}, \hat{\mathcal{E}}_d \cup \hat{\mathcal{E}}_s)$ as described in the following. For each node $\hat{\mathcal{V}}$ in the query memex graph a random element vertex in the factor graph is created. This random element represents the unknown material \hat{M}_n of part \hat{P}_n (see Fig. 4). In total, N random element vertices are created. Furthermore, the factor graph contains unary and pairwise factors.

For each unknown material \hat{M}_n , a unary factor is created and linked to the corresponding random element vertex in the factor graph. We define the unary potential of these factors to be

$$\phi(\hat{M}_n) = \sum_{k=1}^K \rho_{\text{mat}}(\hat{M}_n, M_k) \rho_{\text{part}}(\hat{P}_n, P_k) \rho_{\text{shp}}(\hat{S}, S(P_k)) , \quad (5)$$

which depends on all parts P_k and their materials M_k in the database. Our choice for the unary potential in Eq. 5 is motivated by the

intuitive assumption that the potential should be large if there is a part P_k in the database that has a similar material as \hat{M}_n and similar shape as \hat{P}_n . Furthermore, the contribution of a part P_k is weighted by the shape similarity between the query object and the object $S(P_k)$ containing part P_k .

For each graph edge of the query memex graph in both sets $\hat{\mathcal{E}}_d$ and $\hat{\mathcal{E}}_s$ a pairwise factor is created. This pairwise factor is linked to the corresponding vertices in the factor graph, as illustrated in Fig. 4.

If $(i, j) \in \hat{\mathcal{E}}_d$, which means that it is a spatial context edge, the pairwise potential of the factor is defined by

$$\psi_d(\hat{M}_i, \hat{M}_j) = \frac{1}{|\mathcal{E}_d|} \sum_{(u,v) \in \mathcal{E}_d} [\rho_{\text{mat}}(\hat{M}_i, M_u) \rho_{\text{mat}}(\hat{M}_j, M_v) \rho_{\text{part}}(\hat{P}_i, P_u) \rho_{\text{part}}(\hat{P}_j, P_v) \rho_{\text{dis}}(d_{i,j}, d_{u,v}) \rho_{\text{shp}}(\hat{S}, S(P_u))] \quad (6)$$

Otherwise, if $(i, j) \in \hat{\mathcal{E}}_s$, which means that it is a shape similarity edge, the pairwise potential is simply

$$\psi_s(\hat{M}_i, \hat{M}_j) = (1 - \rho_{\text{mat}}(\hat{M}_i, \hat{M}_j)) \quad (7)$$

The pairwise potential in Eq. 6 for spatial context edges is high if the two parts of the query object \hat{P}_i and \hat{P}_j have similar materials, similar shape, and a similar spatial relation as two parts P_u and P_v from the database. The potential is weighted by the shape similarity between the query object and the object $S(P_u)$ containing part P_u and P_v .

The pairwise potential in Eq. 7 for shape similarity edges is high if parts with similar shape in the query object have similar material.

As the number of parts and the cardinality of \mathcal{E}_d grows linearly with the number of objects in the database and only the database memex can be precomputed, the computation time for the factors at runtime also grows linearly. However, by defining a threshold on shape similarity $\rho_{\text{shp}}(\hat{S}, S(P_u))$ between the query and the database objects in Eqs. 5 and 6 a constant computation time can be achieved once the shape similarity is known. By applying the threshold, the summation includes only those parts and edges that belong to similar database objects, which is an approximately constant quantity (due to our choice of the shape similarity of objects in Eq. 3).

Inference The factor graph defined in the previous paragraph represents the joint probability distribution of all the unknown materials $\hat{M}_n \in \hat{\mathcal{M}}$ of the query object, which is given by

$$P(\hat{\mathcal{M}}) = \frac{1}{Z} \prod_{\hat{M}_n \in \hat{\mathcal{M}}} \phi(\hat{M}_n) \prod_{(i,j) \in \hat{\mathcal{E}}_d} \psi_d(\hat{M}_i, \hat{M}_j)^\alpha \prod_{(i,j) \in \hat{\mathcal{E}}_s} \psi_s(\hat{M}_i, \hat{M}_j)^\beta, \quad (8)$$

where Z is a normalizing constant known as the *partition function*. The exponents $\alpha = 1.0$ and $\beta = 20.0$ are weighting parameters. Note, that these exponents correspond to linear weighting multipliers of the pairwise terms if the joint probability distribution is transferred in the log-domain as is typically done during inference.

In theory, the unknown materials \hat{M}_n are continuous entities. However, for efficient inference using the factor graph, we define a discrete set of candidate materials \mathcal{C} . This set is derived from the set \mathcal{M} of all the materials in the database. We cluster the database materials using a maximum of $D = 100$ cluster centers. Afterwards, for each cluster center, the database material $M_n \in \mathcal{M}$ with the smallest distance ρ_{mat} to the cluster center is added to the set \mathcal{C} of candidate materials. Consequently, the pairwise factors of the factor graph can

be represented as $|\mathcal{C}| \times |\mathcal{C}|$ matrices and the unary factors are $|\mathcal{C}| \times 1$ vectors (see Fig 4).

The task is now to assign a candidate material from set \mathcal{C} to each random variable of the factor graph, i. e., to each material $\hat{M}_n \in \hat{\mathcal{M}}$, so that the marginals $P(\hat{M}_n)$ of the joint probability $P(\hat{\mathcal{M}})$ are maximized:

$$\arg \max_{\hat{M}_n} P(\hat{M}_n) = \arg \max_{\hat{M}_n} \left(\sum_{\hat{\mathcal{M}} \setminus \hat{M}_n} P(\hat{\mathcal{M}}) \right) \quad \forall \hat{M}_n \in \hat{\mathcal{M}} \quad (9)$$

These maxima can be approximated efficiently using the sum-product algorithm (loopy belief propagation) on the factor graph [Kschischang et al. 1998]. The required computation time can be further reduced by using a GPU shader implementation with parallel message updating. Our CPU implementation requires 4897 ms for 50 iterations on a factor graph with 211 unary factors and 623 pairwise factors for 50 candidate materials. The parallel GPU implementation requires 158 ms for the same task on an NVIDIA Geforce GTX 275.

4 Applications

In the following, we describe different applications that become possible with the material memex model, such as automatic assignment of materials to a 3D object or providing user assistance when assigning materials.

4.1 Automatic Material Assignment

To automatically assign materials to a query shape without materials, Eq. 9, i. e., the maximum marginal for each random variables \hat{M}_n , needs to be solved, resulting in a set of materials that can be interactively previewed. The entire material memex is pre-computed and serialized to disk (10 hours for a database of 276 shapes). The most time-consuming steps at run-time are computing the unary (Eq. 5) and the pairwise factors (Eqs. 6 and 7) as well as executing the loopy belief propagation. Fortunately, both steps can be performed using a GPU: dependent on the size of the query object, computing factors takes 500 to 3000 ms, and resp. 20 ms to 200 ms for the loopy belief propagation. Please note, that pre-computation of the factors is not possible as they depend on the choice of the query object.

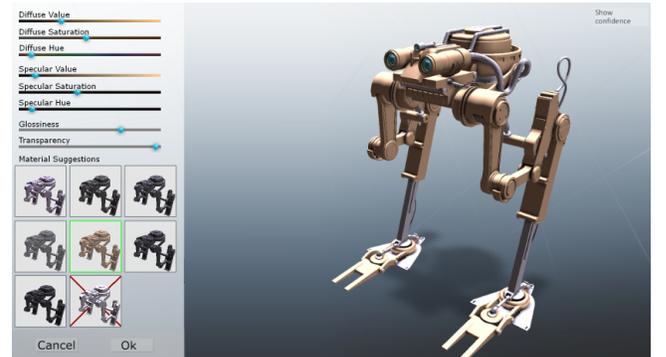


Figure 5: The user interface for ranked material suggestion.

4.2 Ranked Material Suggestion

Besides assigning the top material to each part of a query shape, our approach can also be used to present an ordered list of alternative material assignments. In our prototype (see Fig. 5), after a fully-automatic initial assignment, a user can freely select parts, and



Figure 6: Automatic material assignment: database of cars where the materials are assigned by a human designer (1st row), results of automatic material assignment (2nd row). Results are generated by leaving out the particular query car from the database.

is presented a list of alternative material suggestions, ranked by likelihood (Eq. 9). After selecting a suggestion as a part’s material, this material becomes *fixated*, i. e., its random variable is removed from the factor graph, and all neighbouring pairwise factors become unary factors by keeping only the row (or column) of the pairwise factor matrix that represented the fixated material. After fixation, the inference is restarted, possibly changing other materials in the query, e. g., after assigning a black rubber material to a tire of a car, all other tires are very likely to be assigned the same material if they are linked with a part similarity edge. To remove a material fixation, the random variable is re-included in the factor graph and the unary and pairwise factors are restored.

Furthermore, the user has the option to create a new material for a selected part using a slider interface to control the parameters of the perceptually parametrized Phong [Kerr and Pellacini 2010] reflectance model (see Fig. 5). New materials are added to the set C of candidate materials. Consequently, all unary factor vectors get an additional element, and all pairwise factor matrices get an additional row and column. The user can seamlessly repeat this procedure as many times as required to achieve the desired goal.

5 Results

In this section, several results that are generated with the proposed approach are presented. Similar results as well as a demonstration of our interactive user-interface for material suggestion, are shown in the supplemental video. All results shown in this section are generated based on the same database that contains 276 objects with known material that we collected from internet repositories. The full database is shown in the supplemental video.

Figure 6 shows results of the fully automatic assignment of materials. Six different car models are employed as query objects and the automatic material assignment is run six times producing the six results shown in the 2nd row of the figure. The car models have a complexity of 120 to 250 parts. Fig. 6 also shows the manual material assignment created by a designer. Creating such a manual assignment for a single car takes approximately 45 minutes.

It can be observed that the difference in quality between the designer and automatic material assignment is small. Almost all of the resulting automatic material assignments look very plausible, e.g., all cars have specular body colors and diffuse dark rubber material at the tires; the windows are transparent; also details, such as the disk brake calipers, indicators, or head lights, have suitable material assignments. However, there are also several wrong material assignments, e.g., the hood cover of the convertible is transparent. Such estimation errors are more likely to occur if parts with similar shapes and spatial configurations are not observed in the database. This is also illustrated in Fig. 7, where the query object is an unusual aircraft and the most similar objects in the database are of quite

different objects. As there are no suitable pairwise associations in the database, the material assignment does not look realistic.

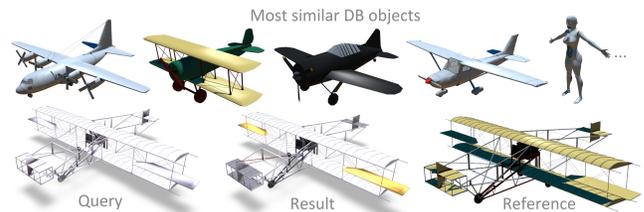


Figure 7: Failure case for automatic material assignment: database (top), query object (left), our result (middle), reference (right).



Figure 8: Comparison of using only the unary factors during inference (left) vs. unary and pairwise factors (right).

Figure 8 is a comparison of a result where only the unary factors (see Eq. 5) are employed for an automatic material assignment, versus our approach of using unary factors and pairwise factors in a probabilistic factor graph. In the result that uses only the unary term, several wrong assignments are visible. Using context information that is stored in the pairwise factors allows removing these artefacts.

Figure 9 shows several examples where the proposed user-interface for ranked material suggestion is employed to refine an automatic assignment. All shown refinements required a maximum of two manual material selections from the ranked list to achieve the shown results. It is evident that the approach works for a large range of query objects with different complexity and topology. The supplemental video shows how a user employs our interface to generate similar results. It also visualizes those database objects, which are found to be most similar to the individual query objects.

Furthermore, the supplemental video demonstrates that shape similarity edges and spatial context edges can cause other parts to change their automatic material assignment when a user manually assigns a material to a part. The spatial context edges typically influence only other parts in the vicinity of the part. It can be observed that for query objects with a smaller factor graph, material changes occur more often. For very complex objects, where most random elements of the factor graph have many connected edges, manual assignments of a material typically do not influence the margins of the other parts strongly enough to provoke a material change.

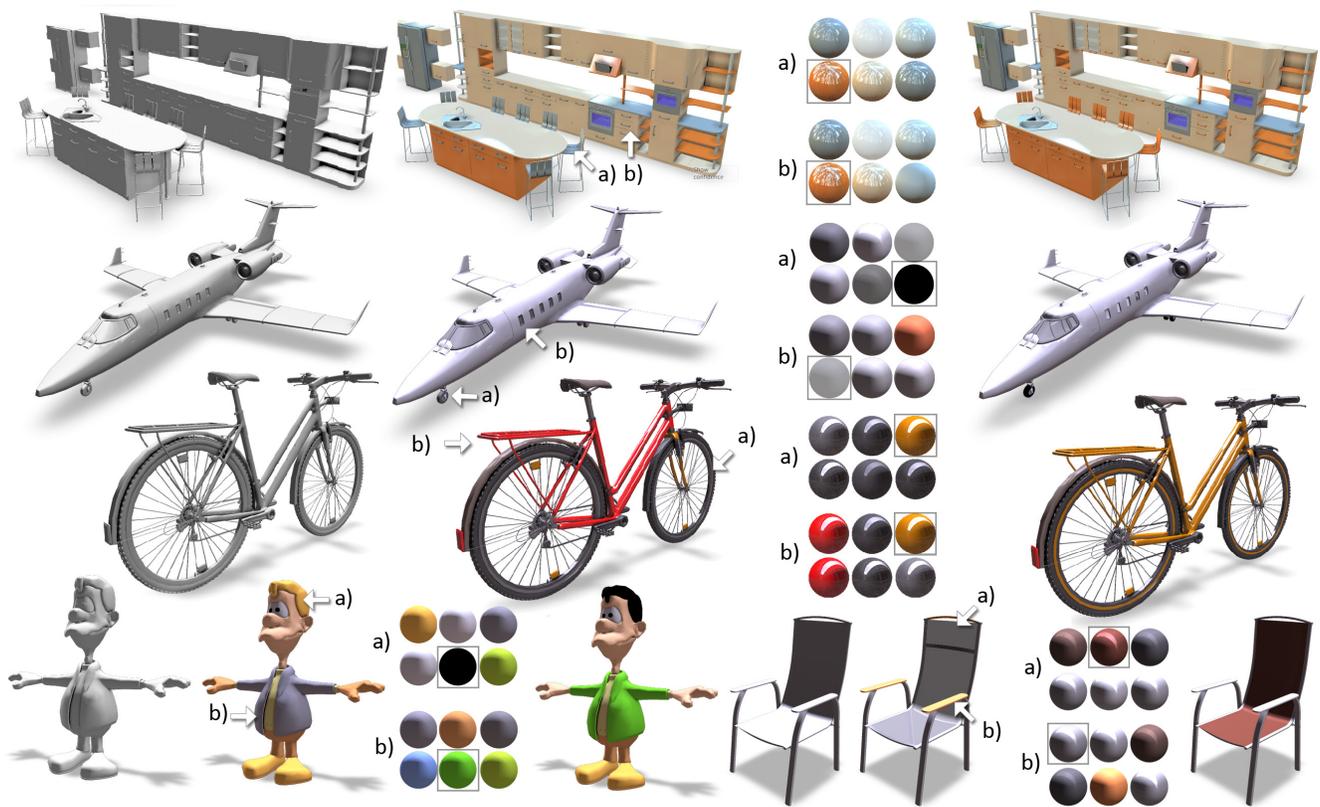


Figure 9: Material suggestion (left to right): query object without material, results of automatic material assignment, user interaction (arrows indicate the clicked part, colored spheres show suggested materials ranked by likelihood), refined results after the user interaction.

Another application of our approach is shown in Figure 10. If a single object is added to the database and the approach is run on several query objects, all query objects will be given similar material assignments. This can be employed, e. g., to indicate that the resulting group belongs to the same team in a computer game.

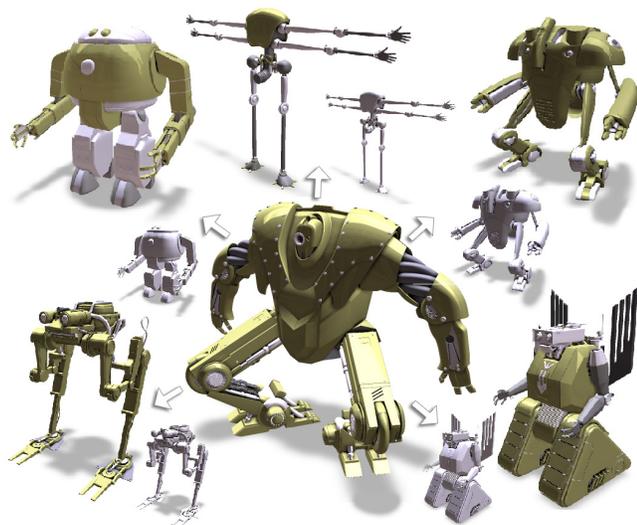


Figure 10: Material transfer from a single database object to multiple other objects: database object (center), query objects (inner circle), query object with transferred materials (outer circle).

User study. The effectiveness of our approach is evaluated in a user study. The first study shows that our interface significantly improves human task performance in a material-assignment session relative to other common user interfaces. The second study shows, that the suggested approach produces better material assignments than common user interfaces. Third, we show how simpler methods for material assignment will likely fail to achieve the same quality as our approach. Finally, a study of material assignment preference over database size indicates that our database sizes are sufficient.

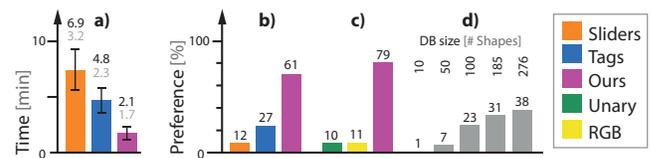


Figure 11: Results of our study (Please see text).

First, we compare task performance in terms of completion time achieved by users of our interface to two other common user interfaces. The first interface uses perceptual *sliders* and color pickers to select the parameters of the Phong reflection model. In the second interface, a user specifies one or multiple *keywords* by typing, and the interface presents preview icons of all materials that partially match one or multiple given keyword. The user selects the final material by clicking an icon. The database of 86 materials was labelled manually with keywords. Note that such keywords are typically not available. In total, 24 subjects participated in the first study. After a short tutorial, each subject performed 6 trial experiments. In each, the participant was asked to assign materials to two objects using three interfaces (sliders, keywords, or ours) in a random order. The maximum time allowed to complete the task was 10 minutes.

A significant difference was found between all interfaces and an improvement of 227 s ($p < 10^{-15}$) comparing ours to the slider and 105 s ($p < 10^{-8}$) when comparing ours to the keyword interface (Fig. 11a). This suggests that our interface can save considerable time and effort.

In a second study, 57 subjects were presented 10 random pairs of images produced with the different approaches in the previous study and had to choose the one with the better material assignment in a two-answer forced choice (2AFC) task. Results from our method were chosen to be preferred in 61 percent of all answers. Our method is significantly better when comparing its score to sliders ($p < 10^{-15}$) and keyword tags ($p < 10^{-9}$), see Fig. 11b.

In a third experiment, we compare our approach against two simpler alternatives: a hypothetical method without a graphical model that only uses unary potentials but no pairwise potentials and a method that does not account for materials and uses only the diffuse RGB color during inference. In a 2AFC task (20 subjects, 10 trials each), we find a significant difference between our approach and simpler alternatives (Fig. 11c), suggesting that material assignment is context-dependent ($p < 10^{-7}$), and a suitable representation of materials is crucial for plausible material assignment ($p < 10^{-8}$).

The last experiment investigates the dependency of quality on database size (Fig. 11d). Subjects were presented images of objects with our material assignment, but produced from databases of increasing size. Again, subjects were asked to select the preferred material composition when presented 320 random pairs from all four database sizes in a 2AFC (details are given as supplemental material). While there is a significant difference between databases with 10 ($p < 10^{-15}$), 50 ($p < 10^{-9}$), 100 ($p = 0.0063$) shapes, no significant difference is found between 185 and 276 shapes ($p = 0.042$). This indicates, but does not prove, that there is only a small effect when increasing the database size beyond 185 exemplars, i. e., that the database size used gives a good indication of the achievable quality.

6 Discussion and Conclusion

This paper proposed an approach to model the relation of shape and material computationally by learning it from a database of 3D objects with materials. It represents associations between database parts, i. e., if two parts have similar shape, similar material, or if they are in context. This information allows automatically assigning materials to other 3D objects and assists users when designing materials by providing ranked material suggestions. Our current implementation employs the Phong reflection model which is widespread in Internet 3D object repositories but is supposed to generalize to general BRDFs using metric such as the one of Ngan et al. [2006]. Our approach requires a decomposition of objects into parts. To this end, we assume that the designers of the 3D objects have created them from multiple parts. This assumption is often not true for natural objects, e. g., animals or human bodies. Automatic segmentation algorithms (e. g., [Kalogerakis et al. 2010]) would be required to make such 3D models accessible for our framework. In future work, our approach could extend to spatially varying materials (textures), but would require cross-parametrization of query and database shapes.

References

- BUSH, V. 1945. As we may think. *Atlantic Monthly* 176, 101–108.
- CHAJDAS, M. G., LEFEBVRE, S., AND STAMMINGER, M. 2010. Assisted texture assignment. In *Proc. I3D*, 173–179.
- CHAUDHURI, S., KALOGERAKIS, E., GUIBAS, L., AND KOLTUN, V. 2011. Probabilistic reasoning for assembly-based 3D modeling. *ACM Trans. Graph. (Proc. SIGGRAPH)* 30.
- CHEN, D.-Y., TIAN, X.-P., SHEN, Y.-T., AND OUHYOUNG, M. 2003. On visual similarity based 3D model retrieval. *Comput. Graph. Forum (Proc. Eurographics)* 22, 223–232.
- FISHER, M., AND HANRAHAN, P. 2010. Context-based search for 3D models. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 29, 182:1–182:10.
- FISHER, M., SAVVA, M., AND HANRAHAN, P. 2011. Characterizing structural relationships in scenes using graph kernels. *ACM Trans. Graph. (Proc. SIGGRAPH)* 30, 34:1–34:12.
- KALOGERAKIS, E., HERTZMANN, A., AND SINGH, K. 2010. Learning 3D mesh segmentation and labeling. *ACM Trans. Graph. (Proc. SIGGRAPH)* 29, 102:1–102:12.
- KAUTZ, J., BOULOS, S., AND DURAND, F. 2007. Interactive editing and modeling of bidirectional texture functions. *ACM Trans. Graph. (Proc. SIGGRAPH)* 26.
- KERR, W. B., AND PELLACINI, F. 2010. Toward evaluating material design interface paradigms for novice users. *ACM Trans. Graph.* 29, 35:1–35:10.
- KSCHISCHANG, F. R., FREY, B. J., AND LOELIGER, H.-A. 1998. Factor graphs and the sum-product algorithm. *IEEE Trans. Information Theory* 47, 498–519.
- LEE, Y. J., ZITNICK, C. L., AND COHEN, M. F. 2011. Shadow-draw: real-time user guidance for freehand drawing. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 27:1–27:10.
- LEFEBVRE, S., HORNUS, S., AND LASRAM, A. 2010. By-example synthesis of architectural textures. *ACM Trans. Graph.* 29, 84:1–84:8.
- LU, J., GEORGHIADES, A. S., GLASER, A., WU, H., WEI, L.-Y., GUO, B., DORSEY, J., AND RUSHMEIER, H. 2007. Context-aware textures. *ACM Trans. Graph.* 26.
- MALISIEWICZ, T., AND EFROS, A. A. 2009. Beyond categories: The visual memex model for reasoning about object relationships. In *Proc. Neural Information Processing Systems*.
- MERTENS, T., KAUTZ, J., CHEN, J., BEKAERT, P., AND DURAND, F. 2006. Texture transfer using geometry correlation. In *Proc. EGSR*, 273–284.
- NGAN, A., DURAND, F., AND MATUSIK, W. 2006. Image-driven navigation of analytical BRDF models. In *Proc. EGSR*, 399–407.
- PELLACINI, F., AND LAWRENCE, J. 2007. AppWand: Editing measured materials using appearance-driven optimization. *ACM Trans. Graph. (Proc. SIGGRAPH)* 26.
- PELLACINI, F., FERWERDA, J. A., AND GREENBERG, D. P. 2000. Toward a psychophysically-based light reflection model for image synthesis. In *Proc. SIGGRAPH*, 55–64.
- PHONG, B.-T. 1975. Illumination for computer generated pictures. *Communications of the ACM* 18, 6, 311–317.
- TORRALBA, A. 2003. Contextual priming for object detection. *Int. J. Comput. Vision* 53, 169–191.
- WANG, B., YU, Y., WONG, T.-T., CHEN, C., AND XU, Y.-Q. 2010. Data-driven image color theme enhancement. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 29, 146:1–146:10.
- YU, L.-F., YEUNG, S. K., TANG, C.-K., TERZOPOULOS, D., CHAN, T. F., AND OSHER, S. 2011. Make it home: Automatic optimization of furniture arrangement. *ACM Trans. Graph. (Proc. SIGGRAPH)* 30, 86.