OpenSurfaces: A Richly Annotated Catalog of Surface Appearance

Sean Bell Paul Upchurch Noah Snavely Kavita Bala
Cornell University

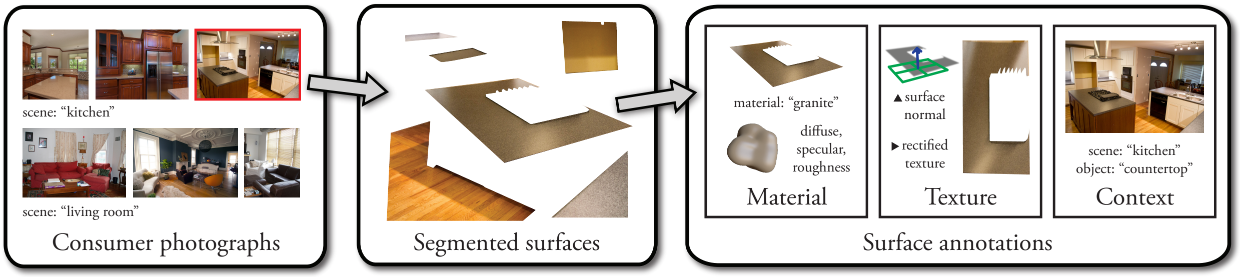


Figure 1: *We present OpenSurfaces, a large database of annotated surfaces created from real-world consumer photographs. Our annotation pipeline draws on crowdsourcing to segment surfaces from photos, and then annotates them with rich surface appearance properties, including material, texture, and contextual information.*

Abstract

The appearance of surfaces in real-world scenes is determined by the materials, textures, and context in which the surfaces appear. However, the datasets we have for visualizing and modeling rich surface appearance in context, in applications such as home remodeling, are quite limited. To help address this need, we present OpenSurfaces, a rich, labeled database consisting of thousands of examples of surfaces segmented from consumer photographs of interiors, and annotated with material parameters (reflectance, material names), texture information (surface normals, rectified textures), and contextual information (scene category, and object names).

Retrieving usable surface information from uncalibrated Internet photo collections is challenging. We use human annotations and present a new methodology for segmenting and annotating materials in Internet photo collections suitable for crowdsourcing (e.g., through Amazon’s Mechanical Turk). Because of the noise and variability inherent in Internet photos and novice annotators, designing this annotation engine was a key challenge; we present a multi-stage set of annotation tasks with quality checks and validation. We demonstrate the use of this database in proof-of-concept applications including surface retexturing and material and image browsing, and discuss future uses. OpenSurfaces is a public resource available at http://opensurfaces.cs.cornell.edu/.

CR Categories: I.4.6 [Image Processing and Computer Vision]: Scene Analysis—Photometry, Shading I.4.6 [Image Processing and Computer Vision]: Feature Measurement—Texture;

Keywords: materials, reflectance, textures, crowdsourcing

Links: DL PDF Web

1 Introduction

The rich appearance of objects and surfaces in real-world scenes is determined by the materials, textures, shape, and context in which the surfaces appear. An everyday room, such as a kitchen, can include a wide range of surfaces, including granite countertops, shiny hardwood floors, brushed metal appliances, and many others. Much of the perceived appeal of such scenes depends on the kinds of materials used, individually and as an ensemble. Thus, many users—ranging from homeowners, to interior designers, to 3D modelers—expend significant effort in the design, visualization, and simulation of realistic materials and textures for real or rendered scenes.

However, the tools and data that we have for exploring and applying materials and textures for everyday problems are currently quite limited. For instance, consider a homeowner planning a kitchen renovation, who would like to create a scrapbook of kitchen photographs from which to draw inspiration for materials, find appliances with a certain look, visualize paint samples, etc. Even simply finding a set of good kitchen photos to look at can be a time-consuming process. Interior design websites, such as Houzz, are starting to provide a forum where people share photos of interior scenes, tag elements such as countertops with brand names, and ask and answer questions about material design. Their popularity indicates the demand and need for better tools. For example, people want to:

- Search for examples of materials or textures that meet certain criteria (e.g. “show me kitchens that use light-colored, shiny wood floors”)
- Find materials that go well with a given material (“what do people with black granite countertops tend to use for their kitchen cabinets?”)
- Retexture a surface in a photo with a new material (“what would my tiled kitchen look like with a hardwood floor?”)
- Edit the material parameters of a surface in a photograph, (“how would my wood table look with fresh varnish?”)
- Automatically recognize materials in a photograph, or find where one could buy the materials online (search-by-texture).

To support these kinds of tasks, we present OpenSurfaces, a large, rich database of annotated surfaces (including material, texture and context information), collected from real-world photographs via crowdsourcing. As shown in Figure 1, each surface is segmented from an input Internet photograph and labeled with material information (a named category, e.g., “wood” or “metal”, and reflectance parameters), texture information (a surface normal and rectified texture), and context information (scene category and object name). Compared to existing material databases, ours takes a "big data" approach, collecting large numbers of example materials captured *in situ* in their surrounding context. Just as massive databases of images and objects have led to new advances in image editing [Hays and Efros 2007; Lalonde et al. 2007] and object recognition [Russell et al. 2008], we believe that a large and comprehensive catalog of *contextual surface appearance properties* is critical for everyday applications involving exploring, editing, and recognizing materials and textures. To our knowledge, ours is the first large-scale database of rich, annotated surface appearance information of its kind.

A central challenge in creating such a catalog is that automatically recovering material properties from images is a notoriously difficult inverse problem that requires careful calibration [Weyrich et al. 2009] or strong assumptions about the image formation model [Romeiro and Zickler 2010]. Our images are scraped from Flickr, so objects appear under a wide range of uncontrolled lighting conditions, with unknown scene geometry. These properties raise the question of whether it is possible to recover *any* usable material information from such images. This drives a key aspect of our system: we ask humans to judge these properties in uncalibrated settings, leveraging the fact that people are good at recognizing and categorizing materials across a range of lighting conditions and image quality. To scale to the large numbers of images, materials, and textures we want, we use crowdsourcing on Amazon's Mechanical Turk (MTurk).

Even with humans in the loop, creating a useful surface catalog is very challenging. Internet photos are noisy, the quality of results from MTurk labelers can vary widely, and interfaces involving material parameters can be difficult for novice users to understand. To get usable results, we designed a multi-stage annotation pipeline, involving multiple types of tasks, to collect and verify surface annotations, including material, texture, and contextual information.

We evaluate the quality of this approach and demonstrate the utility of this database in proof-of-concept applications including surface retexturing and appearance browsing. We believe that the availability of such a database can be helpful to many applications in graphics and vision, beyond the ones we demonstrate.

Our work makes the following contributions:

- A new, large-scale open source database of surface appearance (with thousands of entries and growing) annotated with material, texture, and contextual information available at http://opensurfaces.cs.cornell.edu/.
- A methodology for creating such a database through crowdsourcing annotations of Internet photo collections.
- A publicly available annotation pipeline to spur further exploration and use of such data in graphics and vision applications.
- A demonstration of proof-of-concept uses of such richly annotated surface information.

2 Related work

Image databases. Over the past few years, researchers have shown the utility of "big data"—in the form of large, annotated image databases—for addressing difficult problems in graphics and vision. These databases include 80 Million Tiny Images [Torralba et al. 2008], ImageNet [Deng et al. 2009], the SUN scene database [Xiao et al. 2010], and the LabelMe dataset [Russell et al. 2008]. LabelMe has similar goals to ours and uses a significant amount of user annotation, but their focus is on labeling objects, rather than materials. Other work has extended systems such as LabelMe to material name annotations [Endres et al. 2010].

While the work described above comprises very large databases of images, objects, and scenes, existing databases of natural images of *materials* are relatively small. This category includes the Flickr Materials Database (FMD) [Liu et al. 2010] and the datasets of Hu, et al. [2011]. These datasets are largely made up of close-up photos of objects made of a single substance, such as wood or glass, and have primarily been used for the problem of material categorization. The PSU Near-Regular Texture Database consists of closeups of textured patterns, including material textures [Liu et al. 2004].

In contrast, our aim is to build a database of materials *in context* in photos of everyday scenes, so that we can support applications like interior design that involve whole scenes, rather than single objects or materials. Moreover, prior databases annotate each image with a single category label (e.g. "wood," "glass"), while we collect a much richer set of annotations that include reflectance parameters and surface normals, enabling a wider class of potential applications.

Crowdsourcing. The use of crowdsourcing to collect data is gaining adoption, and has been used in recent approaches to a range of problems, including understanding shape through gauge figures [Cole et al. 2009], creating a mesh segmentation database [Chen et al. 2009], devising a retargeting evaluation framework [Rubinstein et al. 2010], and for integrating humans into the loop for micro-tasks [Gingold et al. 2012]. The experiences from this body of prior work has informed our design process.

Material acquisition and databases. Material acquisition is an active area of research (for a recent survey, see [Weyrich et al. 2009]). A few public databases exist with carefully calibrated measurements, including the MERL database [Matusik et al. 2003] with 4D BRDF measurements for 105 materials, fit to various BRDF models [Ngan et al. 2005], and CUReT [Dana et al. 1999], with 6D measured BTFs (bidirectional texture functions) for 61 samples with various lighting and illumination conditions. The complexity of acquisition and quality of these databases has typically limited their size. To help address these issues, appearance acquisition research has focused on hardware solutions [Ren et al. 2011], but large databases of measured materials are still difficult to acquire.

Rather than capture detailed, high-quality reflectance information for a small number of materials under controlled conditions, our goal is complementary; we aim to gather large numbers of surface annotations, *in situ*, from photographs taken under a wide variety of uncontrolled settings. Since humans provide annotations, we aim for perceptually plausible appearance data.

3 Overview

We present an overview of OpenSurfaces, and discuss some of our key design decisions. Our annotation pipeline takes as input a set of consumer photos depicting one or more surfaces, in context, in an interior scene, such as a kitchen. Each photo is processed in multiple stages, resulting in segmented surface regions (e.g., countertops, floors, cabinets, drawer handles, etc.), where each segment is annotated with material, texture and contextual information.

Creating this database involves several challenges:

- How can we create a high-quality database from consumer photographs? What kinds of photos should we use?
- How can we help novice labelers annotate surfaces? How can we scale this annotation to build a very large database?
- What information is represented in the database?
- What tasks are needed to build this database?

3.1 Community photo collections

One important motivation of our work is to collect a large range of everyday surfaces *in context* from everyday imagery. By "in context," we mean that we want to capture the settings in which various surfaces appear—for instance, where a given type of material tends to appear in an image, what kinds of objects it belongs to, and what other materials it appears in combination with. We chose to focus on indoor locations such as kitchens, living rooms, and dining rooms, which contain indoor materials of practical use, though it is easy to generalize our approach to broader categories.

We use Creative-Commons-licensed Flickr photos as the main source for our images, as we found that Flickr contains a vast range of real, everyday materials in context in high-quality photos. Images from the SUN database [Xiao et al. 2010] were not usable for our purposes because they are typically not of high enough resolution.

3.2 Human annotation

Online consumer photos are far removed from the carefully calibrated, high-dynamic range images typical of material acquisition. Our photos contain multiple materials on surfaces of unknown geometry, are captured under widely varying and unknown lighting conditions, lack radiometric calibration, and may have been post-processed. Hence, extracting meaningful surface properties from these images is well beyond the state-of-the-art of automatic inverse rendering algorithms. Optimization [Romeiro and Zickler 2010] and machine learning approaches [Dror et al. 2001; Liu et al. 2010] to inferring materials have been studied recently, but do not yet demonstrate the performance necessary to annotate materials in noisy, real-world images. These considerations motivate another major design choice in our approach: using humans to annotate our images via crowdsourcing. Humans are reasonably good at identifying materials and their properties over a range of lighting conditions [Fleming et al. 2003], and the availability of crowdsourcing lets us collect annotations at scale for large image collections.

In this paper we focus on an annotation pipeline we deployed on Amazon's Mechanical Turk (MTurk), as MTurk provides a platform for annotating many images in a short amount of time and at low cost. However, our system can also be run as a stand-alone interface hosted on our servers, so that new photos can continue to be annotated (similar to [Russell et al. 2008] for object labeling).

We faced two main challenges in getting useful annotations from labelers. First, annotating surface properties in a photo is not a familiar task to most people, and even communicating what we mean by a "surface," "material," or "texture" to a novice is difficult. Second, MTurk annotators can be unreliable—users can ignore instructions or intentionally provide bad labels. To deal with these sources of noise, we split our material annotation tasks into several subtasks, with the goal of making each subtask as simple, modular, and intuitive as possible. We also use techniques to account for noise, and to verify the results of each subtask. To improve robustness to noise, we use the CUBAM machine learning algorithm of Welinder, et al. [2010] which uses a model of noisy user behavior for binary tasks (e.g., voting for the quality of a surface) to extract better results. In part, it models the competence and bias of each user based on how often they are in agreement with other users.

3.3 OpenSurfaces data representation

Real-world surfaces can be characterized in many ways, including (in increasing order of complexity): names of material categories (e.g., "wood" vs. "metal" vs. "paper"), image exemplars, simple diffuse reflectance models, parametric BRDF models, 4D BRDF measurements, and 6D BTDF measurements. In choosing a surface representation for our database, we considered several factors. First, different representations are suitable for different tasks. For a material recognition task, one might want a database with segmented materials labeled with a category name ("wood" vs. "plastic"), for use in training classifiers. Other applications, such as interior design ("replace the wood floor in this photo with a shinier one"), might warrant a richer description of materials in terms of their reflectance. Hence, we collect multiple types of annotations for each surface, including material names and reflectance parameters.

Surface normals and rectified textures. While some types of surfaces we consider have a uniform BRDF, many, such as granite or wood, are highly textured. Thus, we chose to store texture information to describe surfaces as well. Because textures in photos can be significantly foreshortened by perspective, we create a subtask where labelers mark regions as planar or non-planar, and indicate the surface normal of planar regions using a 3D perspective grid; this allows us to create and store *rectified* textures.

Reflectance. We ask labelers to annotate material parameters for segmented surfaces. Ideally, we would collect the most detailed BRDF information possible; we especially want to move beyond simple Lambertian models, because specular and glossy materials are extremely common in indoor scenes. On the other hand, there is only so much information that can be recovered from a single, uncalibrated image; moreover, our tasks should not be too difficult for human labelers. Thus, we chose to represent the materials using a simple parametric BRDF model (Section 4.7).

Data representation. Our final surface representation includes the following information for each surface (illustrated in Figure 1): material data (a material name, and reflectance parameters including diffuse albedo, gloss contrast, and gloss roughness), texture data (a surface normal and rectified texture (if planar)), and context data (an object name for the surface, and a scene category in which the surface occurs). Each surface also stores quality information, including a segmentation quality score and a planarity score.

3.4 Annotation stages

To build this representation for surfaces, our labelers perform a series of tasks in an annotation pipeline consisting of the following stages (Stage 0 is automatic, and the rest involve humans in the loop):

0. Download images of various scene categories from Flickr.
1. Filter out images that depict the wrong scene category.
2. Flag images that are improperly white balanced.
3. Segment regions of a single material/texture from each image.
4. Name the material for each region.
5. Name the object associated with each region.
6. Label each region as planar or non-planar.
7. Rectify each planar region by specifying a surface normal.
8. Match reflectance parameters for each white balanced region.

Every stage is carefully validated, with at least five labelers contributing to each decision, with some tasks (such as validating reflectance parameters) shown to up to ten.

4 The OpenSurfaces annotation pipeline

We now describe our annotation pipeline in detail. We first discuss how we obtain an initial set of input images, then describe the pipeline of tasks performed on each image and segmented region. We ran several pilot studies for each of these tasks; we describe how these studies guided the design of our final interfaces and tasks. More details on each task are available in the supplementary document. Figure 2 shows a block diagram of the annotation pipeline.

Stage 0: Collecting images. First, we needed to gather a set of high-quality images of indoor scenes. We obtain our images from Flickr, using search terms for each room type, such as "kitchen" and "living room" (see supplementary for full list). Since our goal

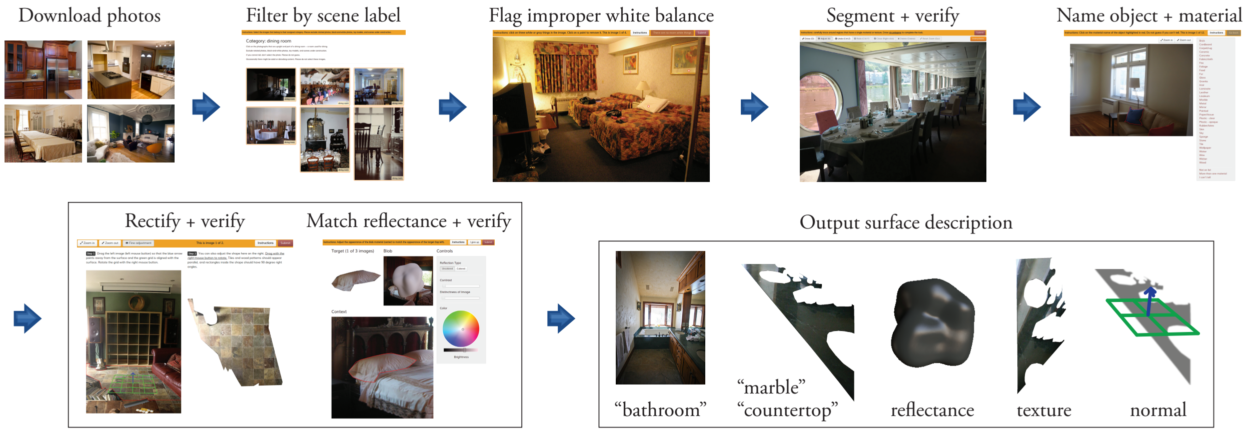


Figure 2: *OpenSurfaces annotation pipeline. Each stage contains a typical example.*

is to recover realistic parameters, we exclude images with the tag "hdr," which are typically highly stylized. We also limit our search to Creative Commons photos that allow "sharing" and "remixing," to ensure that our database can be used in a variety of applications.

We then group the remaining images by scene type. In total, we downloaded 1,099,277 images. We further pruned the list keeping only those photos that are: (a) color JPEG high-resolution (≥ 6 megapixels), (b) at most 32MB in size (to control our disk footprint), and (c) have focal length information in their Exif headers (which we use for the rectification in Section 4.6). After this filtering step, we were left with a final set of approximately 207K images; we picked a few scene categories to focus on, giving us about 92K images.

4.1 Stage 1: Filtering images by scene category

Though we use the text tags associated with Flickr photos to download an initial set of photos, in practice these tags are quite noisy. For example, an image tagged "kitchen" may depict a bar named "The Kitchen," or something else entirely. As in previous work on obtaining images of categories [Deng et al. 2009], we must curate the images of each scene category to find the relevant ones. In this task a labeler is shown a grid of 50 images drawn from a given category (e.g., "dining room"), and is asked to select all the images that belong to that category.

Each image was shown to five labelers. We found that labelers are very fast and reliable at this task, and pruned the image list down to about 25K images. Our database mostly contains scenes from the following categories: kitchen, living room, bedroom, bathroom, staircase, dining room, hallway, family room, and foyer.

4.2 Stage 2: Flag images with improper white balance

While human beings are able to judge material properties in a range of lighting conditions due to color constancy, there are limits to this ability [Brainard et al. 1997], and the lighting conditions in online consumer photographs can be poor and highly variable from photo to photo. So that our labelers can reason about the true color of a surface as accurately and reliably as possible, we filter images that are significantly distorted in color space. To do so, we designed a task where labelers click on objects that they believe are white, and use this feedback to reject images that appear to be improperly white balanced. Users are prevented from clicking on pixels that are close to saturated ($R, G, B \geq 253$), or within 70 pixels of another point.

Each selected pixel is converted to $\mathrm{L^*a^*b^*}$ color space. If the median value of $||(\mathrm{a}^*, \mathrm{b}^*)||$ is ≤ 15, then that user's submission counts as one vote towards the photo being white balanced. Each photo is seen by five labelers; we run CUBAM on the full set of resulting votes, which yields a score (positive or negative) for each photo. If a photo receives a positive CUBAM score, then we consider it white balanced. We opted for this stringent rule because of the large size of our image collection; we can eliminate many images, and still have a large pool of remaining images to label. Compared to majority (3 out of 5) voting, CUBAM changed 14% of labels from good (white balanced) to bad (not white balanced), and 0.8% from bad to good.

Since material recognition is still possible in distorted color spaces [Sharan et al. 2009], we only use white balancing to filter inputs for appearance matching (Stage 8).

4.3 Stage 3: Material segmentation

The next task is to segment regions of constant material or texture from each image. These regions will become the surfaces that are annotated in later tasks.

Pilot study. This task is related to the object segmentation task in LabelMe, but when we tried adapting LabelMe for our task, the interface proved to be cumbersome. We created a new interface with features (smooth zoom, undo/redo, automatic pan) designed to encourage better material segmentations; user feedback was very positive. The smooth zoom and automatic pan features were especially important, yielding segmentations with greater accuracy and more vertices. Without automatic pan (scrolling the view when users click near the edge), users often submitted clipped polygons.

For this task, a labeler is presented with an image and instructed to segment six regions based on material and texture, and *not* object boundaries. The user is shown several examples of good and bad segmentations. An ideal segmentation contains a single material or texture, and tightly hugs the boundary of that material region. The user creates polygons by clicking in the image, or entering a mode where they can adjust an existing polygon (the interface zooms in for fine-scale adjustments; the user can also zoom in or out). The interface saves a full undo/redo stack, and logs all actions with a timestamp for later replay. The interface disallows self-intersecting polygons, but separate polygons can be nested or overlapping.

Once a labeler has segmented regions from the image, we post-process the polygons to create a set of disjoint shapes. This step is to address common cases that arise in these kinds of tasks where a large region of a single material contains a smaller region of a second material (e.g., a door with a handle, or a shower stall with a drain). The easiest way to label these kinds of surfaces is to provide the boundary of the outer shape, and separately the boundary of the inner shapes. Our post-processing stage detects such intersections and yields new shapes as follows: if one shape contains another shape, the inner shape is unchanged, and the outer shape has a hole corresponding to the inner shape; if two shapes partially intersect, three regions are generated to capture cases where the foreground partially intersects the background. The output is stored as a 2D mesh triangulated by [CGAL]. The supplementary material describes this process in more detail. While this technique can occasionally over-segment overlapping regions, we found it to be very useful in addressing common configurations.

4.3.1 Voting for material segmentation quality

The quality of segmentations from this task varies widely; while many regions were surprisingly well-segmented, some were too small, or had sloppy boundaries, and others were good object (but not material) segmentations. We created an additional task where users vote on the quality of each segmentation; these votes are used to determine a quality score for each segmented surface region.

This voting task is somewhat subjective, since it is not always clear what constitutes a "single material" or how labelers interpret the word "texture". Thus, we accept shapes as high quality only if there is a certain amount of consensus. We asked five voters to vote on the quality of each segmentation, and ran CUBAM on the resulting votes. Compared to majority (3 out of 5) voting, CUBAM changed about 7.0% of the bad examples to good, and about 8.38% of the good examples to bad. By default, we discard shapes with a CUBAM-computed quality score below a threshold, but this threshold can be adjusted for applications that need higher-quality segmentations, or which can tolerate lower-quality segmentations.

As we ran the material segmentation task, we noticed that a few users produced exceptionally detailed segmentations, with an accuracy higher than the output of the above voting step. After collecting about 30,000 segmentations, we restricted the task to the best 26 workers (out of 530, using MTurk qualifications) and removed the voting step. This doubled the average detail from 11.6 to 20.3 vertices while reducing our total effective cost from $0.035 to $0.025/shape (including bonuses), since we were no longer paying for voting or for shapes that we later rejected. Even with the smaller set of workers, submission rates remained above 4,000 shapes/day.

4.4 Stages 4 and 5: Naming materials and objects

Finally, we want semantic information for each material segmentation: a **material** name, such as granite or wood, and an **object** name, such as wall, floor, or countertop. These kinds of labels can enable better searching of the database, interesting analytics ("what materials do countertops tend to be made of?"), and category labels for recognition and search.

Material names. The material name is meant to indicate the "stuff" [Adelson 2001] that gives the surface its appearance. In a pilot study, we designed an interface, inspired by LabelMe [Russell et al. 2008], where a user is presented with a material segment and asked to enter a freeform text label, with an "auto complete" feature to suggest material names from a database. However, we found a huge amount of noise in the labels that users entered—multiple different words for the same substance, and misspelled or non-English terms were common. Based on this study, we moved to an interface where a user chooses a material name from a discrete set of choices. We selected the potential names from the results of our pilot study, and taxonomies in interior design [Juracek 1996]. In total, we selected 34 possible material names. After observing that labelers struggled with painted surfaces (walls and ceilings), we introduced the category "painted" to include all surfaces with an outer layer of opaque paint. Without this label, users would guess the underlying surface, causing a split between "wallboard", "wood", "plaster", and "concrete". Users also struggled with laminate surfaces designed to resemble wood or granite. This was resolved by instructing workers to place fake granite and real granite into the same "granite" category (similarly for wood). Users were also given the option of selecting "not on list," "more than one material," and "I can't tell." As the task progressed, we added 6 items to the list by observing items consistently labeled "not on list". The full list of 34 material names is provided in the supplementary material.

Object names. We also create a separate task in which we collect object names, such as "floor" and "countertop," for each segmented surface region. Because we are most interested in categories of objects involved in material design, such as structural elements or worktops, we also limit users to a discrete set of object labels, where the list of labels depends on the material (e.g., "clothing" for fabric, "handles" for metal). We otherwise use the same interface as in the material naming task above. The full list of object names is provided in the supplementary material.

To handle noise, we only keep a semantic label if 3 out of 5 labelers agree, which we found to be reliable. For both types of labels, labelers agreed on a label for about 80% of the segmentations.

4.5 Stage 6: Planarity voting

To rectify each surface region, or transfer that region to a new photo, we need to know the geometry of that surface. For many regions (e.g., a chandelier), the geometry can be quite complex; hence, we chose to identify regions with simple, planar geometry and treat these specially in our database. Planar regions are very common and account for an interesting class of objects that one might want to transfer between images (such as floors and countertops). Thus, we created a task in which workers vote for whether a given segment lies on a single plane. For this task, we use an interface similar to that of the quality voting task, presenting workers with a grid of images zoomed into segments and instructing them to click on all segments that are planar. Each segment is shown to five users, and the results are aggregated using CUBAM.

4.6 Stage 7: Rectified textures

For each planar surface, we seek to create a rectified texture that appears to look "head on" at that surface. For this task, labelers are shown a photo with a planar region annotated in red, with a small grid and surface normal inside the region. They are instructed to "drag the blue arrow to point away from the surface." As they drag the grid, the surface is transformed in real time, via a homography computed from the normal and the image's Exif focal length, and shown to the right of the photo (see Figure 3(a)). The user adjusts the grid until the normal looks correct and the texture looks rectified. Users can also adjust the rectified result directly by dragging on it. We instruct users "tiles and wood patterns should appear parallel, and rectangles inside the shape should have 90 degree right angles." We also provide many examples of good and bad rectifications, and found that users produced higher quality results when shown negative examples featuring rectifications that are only slightly incorrect.

The grid and surface normal are rendered in perspective according to the image focal length. We set the 3D depth so that the grid projects to a constant width when facing forward. The live rectification is performed by constructing a WebGL scene and encoding the desired homography into the camera projection matrix. Since the scaling within the texture plane is arbitrary, we apply a scaling transform to the homography so that the bounding box of the rectified shape fits exactly within the view.

While prior work typically uses a gauge figure to represent a surface normal figure [Koenderink et al. 1992; Cole et al. 2009], we found a 3D perspective grid to be much more effective. We collected 10,000 normals of each type, and found that the RMS error between the submitted normals and final selected normals was 32.1 degrees for gauge figures and 14.9 degrees for planar grids.

When given a continuous space to explore, we found that users did not provide surface normals of sufficient accuracy to rectify surfaces, and so there was often some skew left in the rectified texture. We address this by detecting vanishing points (VPs) and snapping normals to the closest VP. We obtain VPs by finding line segments with LSD [von Gioi et al. 2010], then clustering the segments with J-Linkage [Toldo and Fusiello 2008], and solving for the optimal VP for each cluster [Tardif 2009; Feng et al. 2010]. We then have five users vote on both the original and snapped normals. If both are voted "correct", we use the snapped normal.

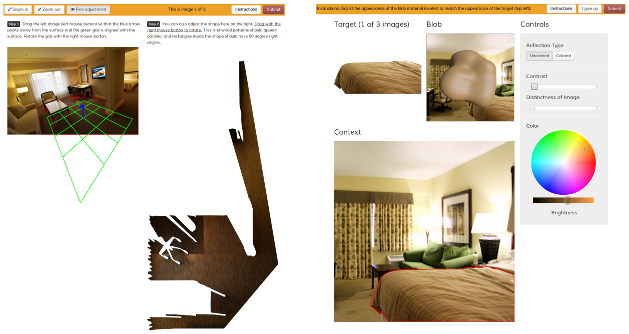
4.7 Stage 8: Appearance matching

Our final task is to find reflectance parameters that match the appearance of each segmented surface (Figure 3(b)). In this task, a synthetic object is rendered alongside the surface to be matched (and with the photo as a backdrop). The design of this task involved several considerations, especially (1) the choice of synthetic scene to render (so as to give effective, accurate visual feedback to labelers with the goal of good appearance matching) and (2) the generality of the material representation and ease of use of the interface.

Choice of synthetic scene. To create a synthetic scene for effective appearance matching, we had to choose a shape, material, and lighting to be rendered with the current user-selected parameters to give the labeler feedback. We ran several pilot studies before we settled on a set of design choices for our scene and interface.

Shape. Vangorp et al. [2007] recommend the use of a blob shape for improved perception of material reflectances. We ran a study with an interface that included both a sphere and a blob, but found that the blob shape was preferred; our final interface uses just the blob (see Figure 3(b)).

Lighting. Initially, we hypothesized that matching the lighting of the input photo in our synthetic scene would provide the most effective cues to the user in this task. Because automatically inferring lighting is difficult, we ran a pilot study where we asked labelers to adjust spherical harmonic coefficients of an environment map to roughly match the lighting of the input photo—for instance, making the left or right side brighter based on windows present in the real scene. However, we found that users were not skilled at recreating lighting [Ramanarayanan et al. 2007], especially in the low-dynamic range input images we provide. Fleming et al. [2004] recommend the use of environment maps with natural lighting statistics to improve material perception. We selected the high dynamic range environment map of the Ennis-Brown House [Debevec 1998] (a high-quality environment map of an indoor scene). Other techniques for recreating lighting with user annotation (e.g. [Karsch et al. 2011]) would be worth considering in the future.



Step 1 Drag the left image (left mouse button) so that the blue arrow points away from the surface and the green grid is aligned with the surface. Rotate the grid with the right mouse button.

Step 2 You can also adjust the shape here on the right. Drag with the right mouse button to rotate. Tiles and wood patterns should appear parallel, and rectangles inside the shape should have 90 degree right angles.

Instructions: Adjust the appearance of the blob material (center) to match the appearance of the target (top left).

(a) Texture rectification (b) Appearance matching

Figure 3: *(a) Stage 7: Rectifying planar textures interface. This figure shows a successfully completed task, where the perspective grid on the left appears to lie flat against the surface, and the texture, shown on the right, is correctly rectified. (b) Stage 8: Interface for appearance matching to recover material parameters.*

Material choice. The choice of material representation was a tradeoff between accuracy and ease of labeling. Common choices in graphics are Ward-based models and microfacet-based models (e.g., Cook-Torrance and variants). We made our choice based on pilot studies as well as the material design study of Kerr and Pellacini [2010], which found no preference between three broad classes of models: Ward [1992], perceptual Ward [Pellacini et al. 2000], and a microfacet-based model. In our pilot studies with these models, we found that the intuitive explanation of the perceptual Ward parameters (contrast gloss (c), and distinctness of image (d) respectively) worked well in the MTurk setting, so we selected the perceptual Ward with a balanced optimization for grazing angles [Geisler-Moroder and Dür 2010] as our model of choice. Note that because the intensity of the input illumination is unknown for our photos, the user-specified diffuse albedo is only correct up to a scale factor; however, the roughness values are absolute.

In another preliminary study we found that matching the color to the real object was the hardest part of the task (as also reported in [Kerr and Pellacini 2010]). To assist labelers with color matching, we modified our interface in three ways. First, we simplified our model to avoid having two separate color wheels for diffuse and specular color. Instead, we let the user select either: (a) a diffuse color, with uncolored specularity and specular roughness (shown in Figure 3(b)); or (b) a colored specular material with roughness, but no diffuse component for colored gloss. Second, to start our users off on the right track, we initialize the diffuse color based on an analysis of the photo. We perform a k-means clustering (with 4 clusters), on the pixels of the segmented surface, and use the mean of the largest cluster to initialize the color widget. Third, users may click on pixels in the segmented surface to set the color. These three modifications significantly improved the quality of our appearance matches, as users spent less time hunting for matching colors; users stopped reporting that they had trouble finding colors, and 85% of submissions were voted to have a perceptual match for color.

Figure 3(b) shows our interface, including a target image region to be matched cropped from the photo, the surrounding context with the region highlighted in red, the rendered 3D blob, and a series of perceptual sliders to edit blob appearance. To specify color, we use a HSV (hue, saturation, value) widget, as in [Kerr and Pellacini 2010].

Rendering and precomputation. Since the task runs in the browser, we made minimal assumptions about users' graphics cards and bandwidth capabilities. This precluded the use of sophisticated precomputation-based methods [Ben-Artzi et al. 2006] with high compute and memory requirements. To achieve interactive performance, we ignore inter-reflections in the blob. Instead, we prefilter environment maps to obtain a diffuse map and 16 gloss maps at different roughnesses sampled according to the d axis from the perceptual Ward model, with $\alpha \in [0, 0.2]$. The HDR prefiltered maps are packed into RGBE textures and encoded as PNG files (1.6MB). At render time, the blob normal is rasterized, and two texture lookups are performed for the diffuse and gloss components from the prefiltered maps; the result is tone-mapped using Reinhard [2002]. We assume that the log-average luminance (required by [Reinhard et al. 2002]) is approximately constant, so it can be stored offline.

Quality. To filter out low quality submissions, we show every match to ten users: five evaluate the color, and five evaluate the gloss

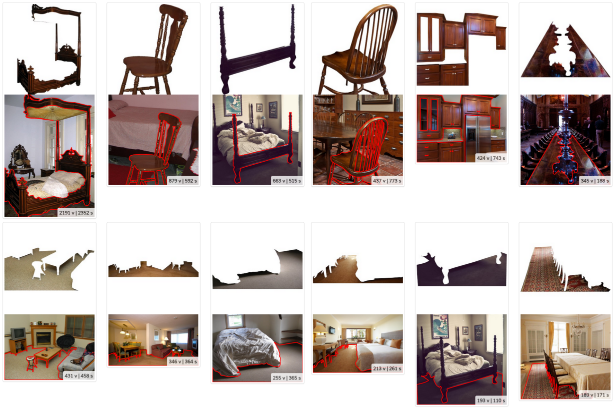


Figure 4: *Example segmented surfaces labeled "wood" (top row) and "carpet" (bottom row). Each item shows the number of vertices and the time spent.*

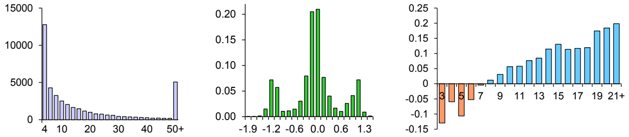


Figure 5: *Statistics over shapes in our database. Left: Histogram of vertex counts, Center: Histogram of CUBAM scores from voting on shape quality, Right: Average CUBAM score for each vertex count.*

if the color matches. We found that evaluating both properties at once causes users to ignore gloss and focus only on color. As with the other voting tasks, we aggregate the results with CUBAM.

5 Results

In this section, we analyze the OpenSurfaces data collected to date (these statistics reflect a snapshot in time, and will change as the database continues to grow). We also discuss the cost and effectiveness of each task (see Table 1).

5.1 OpenSurfaces statistics

We present statistics on the surface data (segmentations, textures, reflectances, and contextual data) collected using our methodology.

Images. Our database currently contains 25,357 curated images, from an initial set of 91,868 sent through the curation task. From our large set of downloaded photos, we prioritized room categories with the largest number of images: kitchens, bathrooms, and living rooms; these have the greatest representation in our database.

Material segmentations. Our database consists of 70,005 segmentations deemed to be good by CUBAM out of 103,513 user-submitted polygons. Figure 5 summarizes the shapes in our database, as well as their quality, as determined by user voting and the CUBAM outputs. Figure 4 shows examples of our segmented surfaces (see the website for many more). Figure 5 (left) shows a histogram of vertex counts over all submitted polygons. Users were required to create polygons with at least four sides, though there are a considerable number of polygons (over 12,000) with more than 30 vertices (the most complex polygon to date has 2,191 vertices). Figure 5 (center) shows a histogram of CUBAM-computed quality scores for all shapes after quality voting. The peak near 0 corresponds to shapes with high disagreement among labelers. This accords well with user feedback about ambiguous cases.

We observed that the amount of disagreement between labelers varies considerably depending on the task. Image curation (Stage 1) had the most agreement, with the fewest number of items with small CUBAM scores. Finally, Figure 5 (right) shows the average CUBAM quality score for shapes as a function of their vertex count. Not surprisingly, more detailed shapes also tend to be of higher quality (as observed anecdotally in other work, including LabelMe).

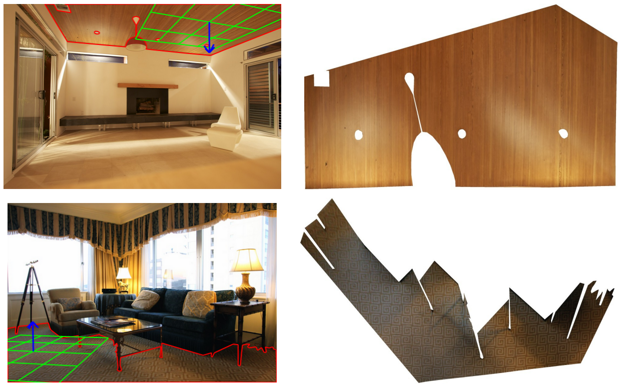


Figure 6: Example rectified textures. *Left: original photographs with segmented surface highlighted and user-specified surface normal. Right: rectified texture with provided surface normal.*

Material and object names. In total, we have 58,928 surfaces annotated with a material name, and 33,378 annotated with object names. These labels allow for interesting analytics, such as: What distributions of materials tend to appear in each type of room? What kinds of objects tend to appear in each material category? Our website includes up-to-date statistics for typical material distributions. Because our dataset has a few forms of bias (see Section 5), these numbers are not necessarily representative of rooms or objects in general, but still reveal interesting trends in our data.

Planarity. Surprisingly, many users struggled with the idea of planarity. For validation, we manually reviewed 35,000 planar outputs and observed a precision of 96.5%. Almost all of the mistakes were from users conflating piecewise-planar with planar, selecting surfaces such as two walls, despite specific instructions and examples to avoid this case. While it only took a few hours to find and remove the 3.5% of the outputs, we could add a follow-up stage that specifically filters out piecewise-planar regions.

Rectified textures. Compared to the earlier tasks, rectifying a planar surface (by specifying a surface normal in the image) is evidently much more difficult. Out of 21,808 shapes that were input to our rectification task, about 16,882 (77.4%) shapes had a surface normal (or snapped surface normal) that was voted as "correct". While individual users were generally poor at the rectification task, even lazy labelers were accurate enough that snapping to the nearest vanishing point often produced the correct normal.

When voting on submissions, labelers were reasonably capable of judging "correct" normals, but struggled to interpret the resulting rectified texture. On average, incorrect normals that were judged to be "correct" were 12 degrees away from the correct normal. For some reason, we found a particularly high number of malicious users for this task; it was necessary to separately detect and block users who selected "yes" for every proposed match. Examples of the final rectifications are in Figures 6 and 7 (see website for more).

Reflectance parameters. Appearance matching (by specifying reflectance parameters) was also a challenging task; out of 40,148 surfaces annotated with reflectance parameters, about 27,648 (69%) were accepted as having sufficient quality. Figures 8 and 9 show examples of high-quality appearance matches (see website for more).

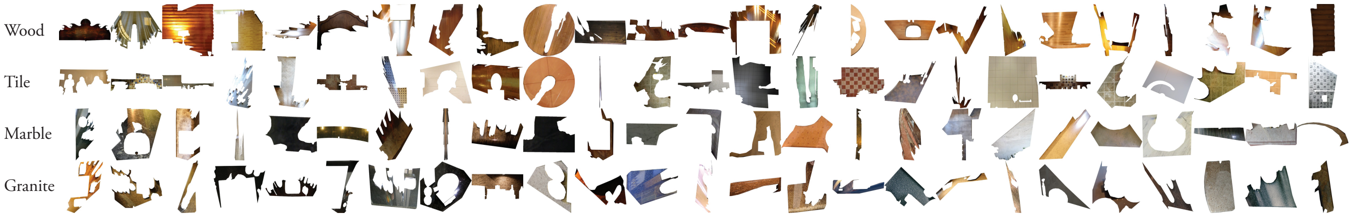


Figure 7: Example rectified textures. *Each row: textures rectified using normals from our database, sorted by decreasing vertex count. Each row contains a single type of material: wood, tile, marble, and granite.*

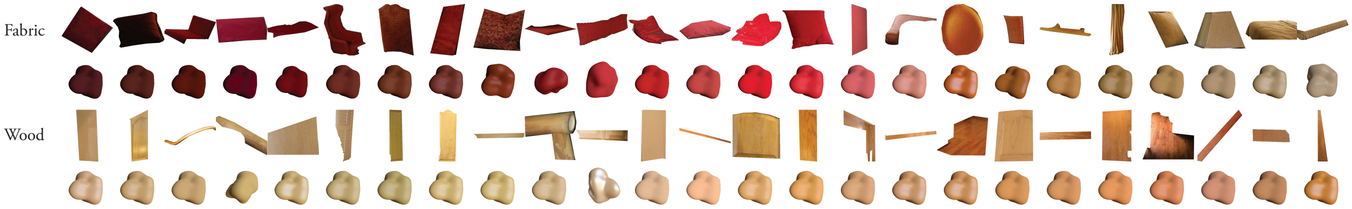


Figure 8: Example reflectance parameters. *First and third rows: material segmentations from our database. Second and fourth rows: blobs rendered with user-specified perceptual Ward parameters.*

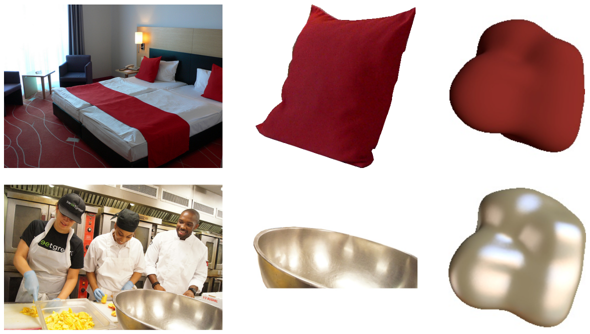


Figure 9: Example reflectance parameters. *Left: original photograph. Center: region to be annotated. Right: blob rendered with the user-specified perceptual Ward parameters.*

As one would expect, users were much more skilled at judging the correctness of matches compared to providing new matches. For each shape, the variance of the perceptual Ward parameters (c and d), decreased from 0.00714 to 0.00452 (34%) and from 0.00297 to 0.00287 (3%) respectively as a result of the two quality voting stages. To further explore the effect of d, and further validate the quality of this pipeline, we added a synthetically rendered image rendered with a state-of-art global illumination algorithm [Walter et al. 2012]. Compared to ground truth, the recovered roughness parameter (d) had an RMS error of 0.057 (28% of the range). Fleming et al. [2003] found an RMS error of 16% for roughness (d) when matching identical spheres across varying illumination. When comparing materials across different geometries (same illumination), Vangorp et al. [2007] found that users can correctly decide if two different objects have the same material 62% of the time. Our images with varying shape and lighting are more challenging, and our matches appear in line with these perceptual studies.

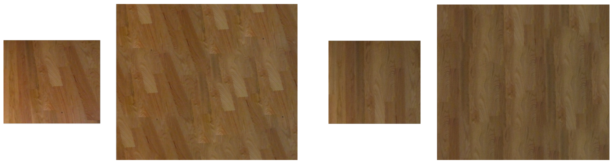
5.2 Task analytics

We now present statistics about our set of tasks, as well as additional observations about the results. Table 1 shows summary statistics for each type of task, including (1) the average payment for each item per task, (2) the average amount of time spent per item, (3) the number of input items processed, and (4) the fraction of output items that were rated as "good". A total of 1,770 MTurk labelers contributed to the database. As shown in the table, we observe a large disparity in the time and cost required across the set of tasks (by up to two orders of magnitude, in terms of cost). The fastest and least expensive task was curating the initial photo set, where each processed photo cost a fraction of a cent. The other voting tasks were also relatively inexpensive and fast, each taking less than 4 seconds. Interestingly, it required more pay to get labelers to quickly name materials compared to objects; labelers seemed to prefer thinking about objects instead of materials. The task which took the most time per item was rectification, which took more than 30 seconds per shape, on average. In our experience with this task, it often takes a significant amount of fine adjustment of the surface normal to achieve a good rectification. Perhaps unsurprisingly, rectification and appearance matching were the two most expensive tasks.

Task	Pay/item	Time	N in	N out	% out
Scene curation	$0.0004	1.8 s	91,868	25,357	27.6%
White balance	$0.0036	16.4 s	24,771	17,839	72.0%
Segmentation	$0.0160	25.8 s	16,282	103,513	N/A
Seg. Quality	$0.0009	3.1 s	54,963	29,467	53.6%
Material name	$0.0033	7.6 s	70,376	58,928	83.7%
Object name	$0.0028	7.2 s	43,058	33,378	77.5%
Planarity	$0.0010	2.7 s	60,800	38,446	63.2%
Rectification	$0.0200	35.6 s	21,808	21,808	N/A
Rect. Quality	$0.0020	3.6 s	21,808	16,882	77.4%
Reflectance	$0.0138	20.0 s	40,148	40,148	N/A
Color Quality	$0.0015	2.9 s	40,148	33,935	84.5%
Gloss Quality	$0.0014	2.9 s	33,791	27,648	81.8%

Table 1: Summary statistics. *For each task we present: the average cost per item in USD (includes MTurk commission), the average time, the number of items assigned to labelers (**N in**). For naming tasks, **N out** is the number of shapes that had at least 3/5 agreement. For voting and quality tasks, **N out** is the number of items classified as "good" as a result of performing that task. As discussed in Section 4.3.1, we stopped measuring segmentation quality after limiting to the 26 best workers.*

An important metric for evaluating MTurk use in collecting a large database is the rate of data collection at a particular price point. We found that submission rate was strongly correlated with price, the speed at which we approve tasks, and the reputation we built with workers. Our submission rate continued to improve as we built trust and communicated with workers. With a cost of about 13 cents per surface (10 cents if non-planar), our initial submission rates improved from a few hundred per day to rates of 4,200 segmentations, 4,500 reflectances, and 15,000 semantic labels per day.



(a) Unrectified synthesis (b) Rectified synthesis

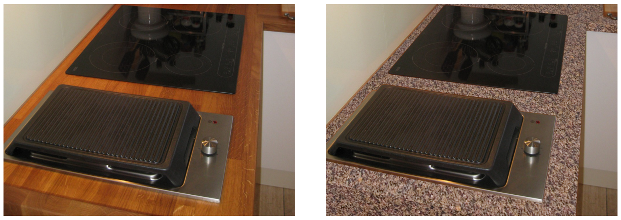
Figure 10: Better exemplars for texture synthesis. *(a) Synthesis using an unrectified exemplar, showing artifacts of foreshortening. (b) A texture synthesized from a rectified exemplar from our database.*

Observations. In developing our tasks and working with labelers on MTurk, we made a number of interesting observations. Many users told us they really enjoyed the segmentation task, and produced beautifully detailed segmented surfaces. We found that our best segmentations were produced by a handful of people, some of whom created thousands of surfaces. On the other hand, some people created good *object* segmentations. In the future, these shapes could be sent back into the pipeline for further segmentation.

The two most difficult tasks were rectification and appearance matching. For appearance matching, users especially struggled with matching near-mirror materials. While they were skilled at selecting color using our interface, many labelers seemed hesitant to select low roughness (high d). This could be due to the fact that at low roughness, users can see the difference between the reflected environment map and the true scene. In the future, we plan to test specialized reflectance models for objects labeled as "mirror" or "glass". On the other hand, we observed that users often correctly used contextual information, such as the appearance of a different object in the image, to aid in appearance matching. This was especially helpful for matching gloss, where highlights suggesting appearance may only appear in other parts of the scene. For example, labelers might use the gloss on nearby cabinet doors to attribute roughness parameters to the similar surface being annotated, despite no discernible gloss being visible on it. Finally, some surfaces in our database are multicolored; we found that in such cases, users consistently matched the dominant color as initialized by k-means clustering.

Feedback and quality incentives. Rather than raising the pay for all tasks uniformly, which did not result in higher quality (as has been observed by others [Marge et al. 2010]), we targeted good users for bonuses and feedback on the quality of their work. For segmentation, we paid up to 10 times the base rate depending on the complexity of the submission. On average, we paid an extra 25% in bonuses. Since users are paid a fixed rate by default, there is otherwise no incentive to spend extra time producing detailed submissions. One user replied "It's difficult but I appreciate your positive feedback when you approve/reject the HITs, so I'm motivated to please you!" In addition to providing positive feedback, we found it was necessary to prevent 159 users from doing our tasks because they were either malicious or had an accuracy below 50%. The use of sentinels is recommended by [Gingold et al. 2012] to check labeler quality. However, we found that our tasks are not amenable to this approach since our surfaces occur in distinctive photographs, and repetition tips off the labeler. We would like to revisit this in future work.

Sources of bias. In terms of representing places and surfaces, our dataset is biased in a few ways. First, our photos are from Flickr, which is biased towards higher-quality imagery (compared to other sites, such as Facebook), and geographically (Flickr is more widely used in the U.S. and Europe). Second, the keywords we use when searching for photos (intentionally) bias our data towards clean, uncluttered rooms. Third, our users are likely biased towards segmenting certain types of surfaces (as we require that they segment a fixed number of surfaces per image, not the entire photo). Factors such as saliency or ease of labeling likely play a role in a user's decision about what surfaces to label. Finally, since our environment map is fixed, differences in lighting between the true scene and our proxy environment map could affect the reflectance judgement of users and introduce small color shifts. Our pipeline reduces this effect by rejecting improperly white balanced photos. However, further study is needed to understand the extent of these errors.



(a) Target photo (b) Retextured

Figure 11: Retexturing example. *The input target is a segmented photo. A rectified granite countertop surface from OpenSurfaces is synthesized and applied to the input using the correct perspective.*

6 Proof-of-Concept Applications

Our surface catalog can both assist and enhance existing and new applications in material search, browsing, editing, and classification. Here we describe a few such supporting proof-of-concept uses of the database, and discuss future applications.

6.1 Texturing

Richer texture exemplars. OpenSurfaces provides a large set of rectified real-world textures for use as texture exemplars. These improve over the corresponding unrectified and foreshortened textures, letting users of the catalog to expand their access to interesting real-world textures. Figure 10 illustrates this advantage.

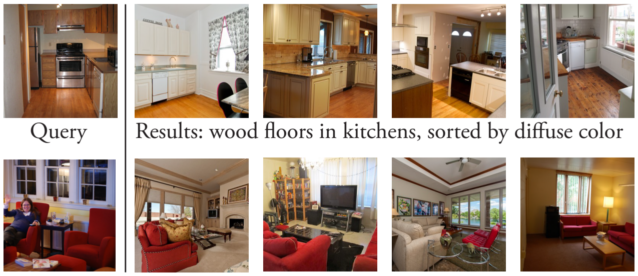
Retexturing. Consider a user with a photograph containing a segmented surface (e.g., a countertop) who wants to visualize a home remodeling change to that surface. The user can browse OpenSurfaces for a suitable replacement using the rich annotated data to search on color, material properties, object name, or substance type. The rectified textures found in our catalog can be used as the input to a texture synthesis algorithm (see Figure 11 for a synthesized granite texture drawn from our database to retexture a kitchen counter, with minor touch-ups applied as a post-process). While texture synthesis requires a clean example, and most of our shapes contain shadows or lighting, we did not find it difficult to select suitable shapes, since many shapes with undesired effects contain a clean inner region.

6.2 Informed scene similarity

Our database can be used to enhance image and material search (e.g., to generate ideas for interior design projects), by allowing for more targeted queries. With our rich annotations, we can explore interior scenes in ways previously not possible. For example, Figure 12 shows a search for wood flooring in kitchens and a search for fabric sofas in living rooms. Using our user-annotated reflectance parameters, we can further refine and sort the results by diffuse color similarity to the input query. While the input query is a segmented material from our database, the query can also be constructed by searching for a phrase (e.g. "kitchen wood floor") and selecting photos that have the desired color.

6.3 Future applications

There are many potential applications of a database of materials and surfaces beyond the proofs-of-concept we describe.



Query

Results: wood floors in kitchens, sorted by diffuse color

Query

Results: fabric sofas in living rooms, sorted by diffuse color similarity

Figure 12: Informed scene similarity. *Left: query object. Right: photos containing objects of the same name and material, sorted by diffuse color similarity to the query object.*

Search. Searching for photos that contain shiny or rough materials, or particular colors, can aid people interested in material design (e.g., interior decorating or home remodeling). Another search problem, search-by-material, could identify which paint or fabric can be bought from a store to best match an object in a photo. Further, our large database of community-gathered photos can be analyzed to discover common co-occurrences or relationships between certain materials. For example, we could tell what materials, objects or colors are often found above, below or nearby a given object. Similarly, we could offer suggestions to homeowners, answering such questions as: "for kitchens with black granite countertops, what kinds of materials are often used for cabinets?" We could also use existing materials to automatically recognize that the query contained black granite countertops, thus requiring the user to only submit an input photo. As our database grows, we could discover geographic or temporal trends in material design, given suitably tagged imagery.

Editing. Aside from search, material editing and transfer is also useful. Synthetic object insertion into photographs [Debevec 1998; Karsch et al. 2011] could use our material parameters for accurate compositing. Alternatively, one can imagine extending Photo Clip Art [Lalonde et al. 2007] to the domain of material compositing, where rather than attempting to accurately simulate lighting and shadows, one could search for a surface that contains the desired effects. The presence of surface normals can also aid reasoning about perspective or scene geometry. Finally, surfaces such as wood may appear very different depending on how the surfaces are treated (e.g., painted, or unfinished). Our database can potentially be combined with image editing software to visualize such effects.

Classification. We believe that one of the strongest applications of our database is enabling the automatic classification of materials and material properties. Our materials can form useful training data and priors for estimating the category, reflectance, and roughness of materials. Our database and photos are completely open, and we hope will serve as a useful resource for these and other applications.

7 Conclusions and future work

This paper takes the first steps towards building a "big data" catalog of surface properties of everyday materials in our world, with reflectance, texture and contextual information for these surfaces. While the current data collected, with thousands of surfaces (and growing), can immediately be useful to many graphics and vision applications, many promising research directions remain.

Labelers often are faced with difficult questions; what is the reflectance of a mirror reflecting a pink wall, pink or glass? Developing better guidelines and interfaces for difficult surfaces like transparent and translucent materials is an interesting avenue of future work.

To further increase scalability, we plan to explore more automation to help users, especially for difficult tasks like rectification and appearance matching—it is often easier to improve on a good answer, or accept or reject an automatically generated result, than to create an answer from scratch. For example, our use of clustering of surface colors significantly improved labeler speed and accuracy in appearance matching. Similarly, detecting vanishing points assisted in the rectification task. Indeed, we hope that our database can effectively bootstrap itself, by creating large amounts of training data useful for improving algorithms for tasks like material recognition [Liu et al. 2010] and reflectance estimation [Dror et al. 2001].

Further, we would like to address the sources of bias in our input collections, by expanding to include more categories, and develop incentive systems to get a more complete annotation of images.

While we include quality controls at every stage of the pipeline, we are interested in further evaluating the quality of our data in comparison to controlled, in-lab methodologies. For reflectance, we plan to physically insert known objects into scenes and compare the reflectance returned by the pipeline to the measured material properties for those known objects. In addition, we plan to render and annotate a larger collection of synthetic scenes to validate surface normals, reflectances, and segmentations.

8 Acknowledgements

This work was supported by the National Science Foundation (grants IIS-1161645, NSF-1011919, and NSF-1149393) and the Intel Science and Technology Center for Visual Computing. We are grateful to Albert Liu and Keven Matzen, and to Kevin for the paper video. We also thank our MTurk workers, and Patti Steinbacher in particular. In the supplementary material, we acknowledge the Flickr users who released their images under Creative Commons licenses. Finally, we thank the reviewers for their valuable comments.

References

ADELSON, E. H. 2001. On seeing stuff: the perception of materials by humans and machines. *Proc. SPIE Human Vision and Electronic Imaging 4299.*

BEN-ARTZI, A., OVERBECK, R., AND RAMAMOORTHI, R. 2006. Real-time BRDF editing in complex lighting. In *SIGGRAPH Conf. Proc.*

BRAINARD, D. H., BRUNT, W., AND SPEIGLE, J. 1997. Color constancy in the nearly natural image. *J. of the Optical Society of America 14*, 9.

CGAL, Computational Geometry Algorithms Library. http://www.cgal.org/.

CHEN, X., GOLOVINSKIY, A., AND FUNKHOUSER, T. 2009. A benchmark for 3D mesh segmentation. In *SIGGRAPH Conf. Proc.*

COLE, F., SANIK, K., DECARLO, D., FINKELSTEIN, A., FUNKHOUSER, T., RUSINKIEWICZ, S., AND SINGH, M. 2009. How well do line drawings depict shape? In *SIGGRAPH Conf. Proc.*

DANA, K., VAN-GINNEKEN, B., NAYAR, S., AND KOENDERINK, J. 1999. Reflectance and texture of real world surfaces. *ACM Transactions on Graphics 18*, 1.

DEBEVEC, P. 1998. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *SIGGRAPH Conf. Proc.*

DENG, J., DONG, W., SOCHER, R., LI, L.-J., LI, K., AND FEI-FEI, L. 2009. ImageNet: A large-scale hierarchical image database. In *Proc. Comp. Vision and Pattern Recognition*.

DROR, R., ADELSON, E. H., AND WILLSKY, A. 2001. Estimating surface reflectance properties from images under unknown illumination. In *Proc. SPIE Human Vision and Electronic Imaging*.

ENDRES, I., FARHADI, A., HOIEM, D., AND FORSYTH, D. 2010. The benefits and challenges of collecting richer object annotations. In *Workshop on Advancing Computer Vision with Humans in the Loop*.

FENG, C., DENG, F., AND KAMAT, V. R. 2010. Semi-automatic 3D reconstruction of piecewise planar building models from single image. In *Int. Conf. on Construction Appl. of Virtual Reality*.

FLEMING, R. W., DROR, R. O., AND ADELSON, E. H. 2003. Real-world illumination and the perception of surface reflectance properties. *J. of Vision 3*, 5.

FLEMING, R. W., TORRALBA, A., AND ADELSON, E. H. 2004. Specular reflections and the perception of shape. *J. of Vision 4*, 9.

GEISLER-MORODER, D., AND DÜR, A. 2010. A new Ward BRDF model with bounded albedo. In *Proc. Eurographics Symp. on Rendering*.

GINGOLD, Y., SHAMIR, A., AND COHEN-OR, D. 2012. Micro perceptual human computation. *ACM Transactions on Graphics 31*, 5.

HAYS, J., AND EFROS, A. A. 2007. Scene completion using millions of photographs. In *SIGGRAPH Conf. Proc.*, 4:1–4:7.

HU, D., BO, L., AND REN, X. 2011. Toward robust material recognition for everyday objects. In *Proc. British Machine Vision Conf.*

JURACEK, J. 1996. *Surfaces: Visual Research for Artists and Designers*. Norton.

KARSCH, K., HEDAU, V., FORSYTH, D., AND HOIEM, D. 2011. Rendering synthetic objects into legacy photographs. In *SIGGRAPH Asia Conf. Proc.*

KERR, W. B., AND PELLACINI, F. 2010. Toward evaluating material design interface paradigms for novice users. *ACM Transactions on Graphics 29*, 4.

KOENDERINK, J. J., DOORN, A. J. V., AND KAPPERS, A. M. L. 1992. Surface perception in pictures. *Perception & Psychophysics*.

LALONDE, J.-F., HOIEM, D., EFROS, A. A., ROTHER, C., WINN, J., AND CRIMINISI, A. 2007. Photo clip art. In *SIGGRAPH Conf. Proc.*

LIU, Y., LIN, W.-C., AND HAYS, J. 2004. Near regular texture analysis and manipulation. In *SIGGRAPH Conf. Proc.*

LIU, C., SHARAN, L., ADELSON, E., AND ROSENHOLTZ, R. 2010. Exploring features in a Bayesian framework for material recognition. In *Proc. Comp. Vision and Pattern Recognition*.

MARGE, M., BANERJEE, S., AND RUDNICKY, A. I. 2010. Using the Amazon Mechanical Turk for transcription of spoken language. In *Int. Conf. on Acoustics, Speech, and Signal Processing*.

MATUSIK, W., PFISTER, H., BRAND, M., AND MCMILLAN, L. 2003. A data-driven reflectance model. *ACM Transactions on Graphics 22*, 3.

NGAN, A., DURAND, F., AND MATUSIK, W. 2005. Experimental analysis of BRDF models. In *Proc. Eurographics Symp. on Rendering*.

PELLACINI, F., FERWERDA, J. A., AND GREENBERG, D. P. 2000. Toward a psychophysically-based light reflection model for image synthesis. In *SIGGRAPH Conf. Proc.*

RAMANARAYANAN, G., FERWERDA, J., WALTER, B., AND BALA, K. 2007. Visual equivalence: Towards a new standard for image fidelity. In *SIGGRAPH Conf. Proc.*

REINHARD, E., STARK, M., SHIRLEY, P., AND FERWERDA, J. 2002. Photographic tone reproduction for digital images. In *SIGGRAPH Conf. Proc.*

REN, P., WANG, J., SNYDER, J., TONG, X., AND GUO, B. 2011. Pocket reflectometry. In *SIGGRAPH Conf. Proc.*

ROMEIRO, F., AND ZICKLER, T. 2010. Blind reflectometry. In *Proc. European Conf. on Comp. Vision*.

RUBINSTEIN, M., GUTIERREZ, D., SORKINE, O., AND SHAMIR, A. 2010. A comparative study of image retargeting. In *SIGGRAPH Asia Conf. Proc.*

RUSSELL, B. C., TORRALBA, A., MURPHY, K. P., AND FREEMAN, W. T. 2008. LabelMe: A database and web-based tool for image annotation. *Int. J. of Computer Vision 77*, 1-3.

SHARAN, L., ROSENHOLTZ, R., AND ADELSON, E. H. 2009. Material perception: What can you see in a brief glance? *J. of Vision 9*, 8.

TARDIF, J.-P. 2009. Non-iterative approach for fast and accurate vanishing point detection. In *Proc. Int. Conf. on Comp. Vision*.

TOLDO, R., AND FUSIELLO, A. 2008. Robust multiple structures estimation with J-linkage. In *Proc. European Conf. on Comp. Vision*.

TORRALBA, A., FERGUS, R., AND FREEMAN, W. T. 2008. 80 million tiny images: A large data set for nonparametric object and scene recognition. *Trans. on Pattern Analysis and Machine Intelligence 30*, 11.

VANGORP, P., LAURIJSSEN, J., AND DUTRÉ, P. 2007. The influence of shape on the perception of material reflectance. *ACM Transactions on Graphics 26*, 3.

VON GIOI, R. G., JAKUBOWICZ, J., MOREL, J.-M., AND RANDALL, G. 2010. LSD: A fast line segment detector with a false detection control. *Trans. on Pattern Analysis and Machine Intelligence 32*, 4.

WALTER, B., KHUNGURN, P., AND BALA, K. 2012. Bidirectional lightcuts. In *SIGGRAPH Conf. Proc.*

WARD, G. 1992. Measuring and modeling anisotropic reflection. In *SIGGRAPH Conf. Proc.*

WELINDER, P., BRANSON, S., BELONGIE, S., AND PERONA, P. 2010. The multidimensional wisdom of crowds. In *Proc. Neural Information Processing Systems*.

WEYRICH, T., LAWRENCE, J., LENSCH, H. P. A., RUSINKIEWICZ, S., AND ZICKLER, T. 2009. Principles of appearance acquisition and representation. *Foundations and Trends in Computer Graphics and Vision 4*, 2.

XIAO, J., HAYS, J., EHINGER, K. A., OLIVA, A., AND TORRALBA, A. 2010. SUN database: Large-scale scene recognition from abbey to zoo. In *Proc. Comp. Vision and Pattern Recognition*.