A Uniform Grid Structure to Speed Up Example-Based Photometric Stereo

Rafael F. V. Saracchini, Jorge Stolfi, and Helena Cristina da Gama Leitão

***Abstract*—In this paper, we describe a data structure and an algorithm to accelerate the table lookup step in example-based multiimage photometric stereo. In that step, one must find a pixel of a reference object of known shape and color, whose appearance under m different illumination fields is similar to that of a given scene pixel. This search reduces to finding the closest match to a given m-vector in a table with a thousand or more m-vectors. Our method is faster than known previous solutions for this problem but, unlike some of them, is exact, i.e., always yields the best matching entry in the table, and does not assume point-like sources. Our solution exploits the fact that the table is in fact a fairly flat 2-D manifold in m-dimensional space so that the search can be solved efficiently with a uniform 2-D grid structure.**

***Index Terms*—Algorithm, closest match, photometric stereo, uniform grid structure.**

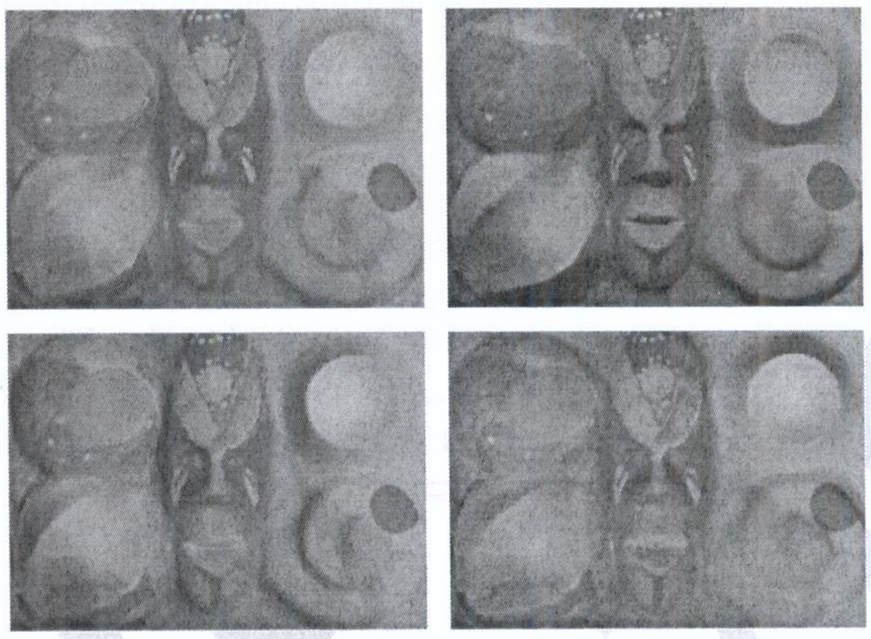


Fig. 1. Images of a painted plaster sculpture under different lightings.

I. Introduction

THE PROBLEM of variable-lighting photometric stereo (VLPS) was considered in [15], [25], and [31]. In this problem, the goal is to determine the 3-D geometry of a scene from a list of $m \geq 3$ monochromatic digital photos $S_1, \ldots S_m$, which are all taken with different lightings but with the same pose and the same viewpoint (see Fig. 1).

Woodham showed that, by analyzing the m pixel intensities $S_i[p]$ at any image point p, one can recover the unit vector $\vec{s}[p]$ that is perpendicular to a surface element, which is visible at p. The third dimension (i.e., depth) at each point can then be recovered by integration of these normals. To perform the above analysis, one must have enough information about the bidirectional radiance distribution function (BRDF) of the surface and about the light field Φ_i in each image S_i. The BRDF of the scene's surface at point p of the image is function $\beta_S[p](\vec{n}, \vec{u}, \vec{v})$ that gives the apparent brightness of a patch of surface with normal $\vec{n}$ which is viewed from direction $\vec{v}$ and illuminated with unidirectional light of unit intensity flowing in direction $\vec{u}$. (We include the geometric light-spread factor $\max\{0, -\vec{u} \cdot \vec{n}\}$ in the BRDF itself).

The problem of VLPS has attracted substantial attention in recent years, e.g., by [5], [6], [24], and [26]. It arises in many diverse applications such as archaeology, art reconstruction [30], face capture [4], dermatology [10], forensics [20], industrial inspection [11], and security [27].

A. Example-Based VLPS

In the example-based variant of VLPS, which is introduced in [33], the BRDF information is given indirectly by m images $G_1, \ldots G_m$ of a reference object, i.e., a sample object of known shape and color, where each image reference object G_i is taken under the same lighting conditions as the corresponding scene image S_i (see Fig. 2). Depending on the application, it may be convenient to include the reference object as part of the scene itself. In that case, each G_i will be just a subimage of S_i.

In this paper, we assume that all images $S_1 \ldots S_m$ have been geometrically corrected, trimmed, and aligned so that each point p on their common domain $\mathcal{S}$ corresponds to the same point on the scene's visible surface. The same condition is assumed for the reference-object images $G_1 \ldots G_m$, whose common domain will be denoted by $\mathcal{G}$.

We also assume linear light sampling so that sample values are directly proportional to the physical light intensity (or radiance) within the pixel. Although the signature-based method we will describe in Section I-C is little affected by the standard power-law ("gamma") intensity encoding, it is sensitive to other kinds of nonlinearity, such as brightness and contrast adjustments, black-level offsets, and sharpening filters that are often enabled by default in certain cameras.

Manuscript received April 19, 2010; revised March 17, 2011; accepted June 01, 2011. This work was supported in part by student grants from CAPES, by CNPq under Grant 304581/2004-6 and Grant 301016/19-5, by FAPESP under Grant 2007/52015-0 and Grant 2007/59509-9, and by FAPERJ under Grant E26/100.532/2007. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. R. P. Millane.

R. F. V. Saracchini and J. Stolfi are with the Institute of Computing, State University of Campinas, 13984-971 Campinas, Brazil (e-mail: ra069320@ic.unicamp.br; stolfi@ic.unicamp.br).

H. C. da Gama Leitão is with the Institute of Computing, Fluminense Federal University, 24210-240 Niterói, Brazil (e-mail: hcgl@ic.uff.br).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2011.2159386

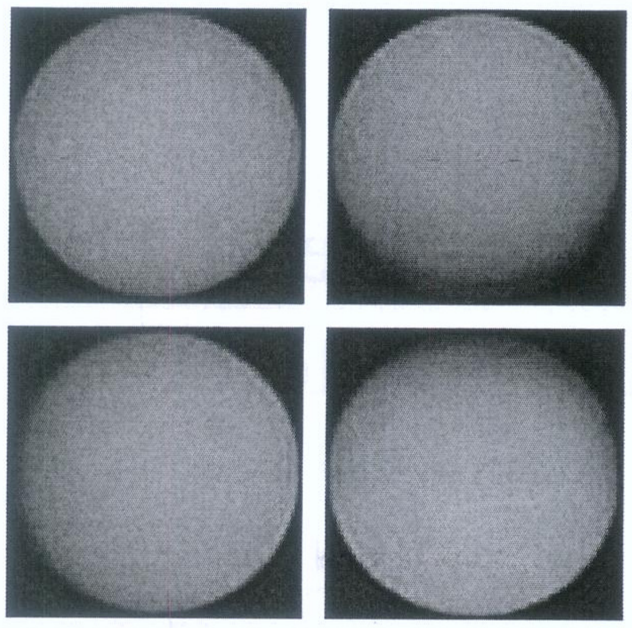


Fig. 2. Images of a spherical reference object under the same illumination conditions as the photos of Fig. 1.

In its basic form, example-based VLPS is viable only if all visible scenes and reference-object surfaces have the same finish everywhere, except for variations in intrinsic color. That is, the BRDF $\beta_S[p]$ of the scene at each image point p and the BRDF $\beta_G[q]$ of the reference object at any point q must be multiples of some fixed BRDF β as follows:

$$\begin{aligned}\beta_S[p](\vec{n},\vec{u},\vec{v}) &= \overset{*}{s}[p]\beta(\vec{n},\vec{u},\vec{v})\\ \beta_G[q](\vec{n},\vec{u},\vec{v}) &= \overset{*}{g}[q]\beta(\vec{n},\vec{u},\vec{v})\end{aligned} \tag{1}$$

for all $\vec{n}$, $\vec{u}$, and $\vec{v}$. The constant factors $\overset{*}{s}[p]$ and $\overset{*}{g}[q]$ in these formulas are the intrinsic lightness or the albedo of the scene and the reference-object surface at those points, respectively. Observe that the albedo of the reference object $\overset{*}{g}[q]$ and the normal direction $\vec{g}[q]$ must be known for all $q \in \mathcal{G}$. Typically, one uses a spherical reference object with uniform albedo, which is preferably white (i.e., $\overset{*}{g}[q] = 1$ everywhere).

Another necessary condition for example-based VLPS is that BRDF β must be dominated by wide-angle scattering, with no mirror-like or glossy scattering. The standard example is the following Lambertian BRDF:

$$\beta(\vec{n},\vec{u},\vec{v}) = \max\{0, -\vec{u}\cdot\vec{n}\}. \tag{2}$$

However, almost any BRDF β will do, as long as it is smooth (without sharp peaks or sharp ridges). For simplicity, we assume a narrow camera field of view and distant light sources so that the viewing direction $\vec{v}$ and the lighting conditions are nearly the same at every point of $\mathcal{S}$ or $\mathcal{G}$.

In the rest of this section, we describe example-based VLPS for monochromatic images. The formulas can be extended easily to multichannel (e.g., RGB) images, yielding the single normal map $\vec{s}\,[p]$ but a different albedo map $\overset{*}{s}_\lambda[p]$ for each spectral band λ. The latter provide the illumination-independent intrinsic color of the scene at each pixel.

B. Fundamental Equations

The key idea of example-based VLPS is that the intensity of each point on the scene photo S_i or the reference-object photo G_i can be analyzed into the product of two factors: the intrinsic albedo $\overset{*}{s}$ or $\overset{*}{g}$ (that depends only on the surface's material and finish) and the lighting factor $L_i(\vec{n})$ (that depends only on the index i and on the surface's normal $\vec{n}$). Specifically,

$$\begin{aligned}S_i[p] &= \overset{*}{s}[p]L_i\left(\vec{s}[p]\right)\\ G_i[q] &= \overset{*}{g}[q]L_i(\vec{g}[q]).\end{aligned} \tag{3}$$

Here, each L_i is the shading function implied by the lighting field Φ_i and BRDF β. It maps each unit vector $\vec{n}$ to the apparent lightness of a white surface perpendicular to $\vec{n}$ and is given by

$$L_i(\vec{n}) = \int_{\mathbb{S}^2} \Phi_i(\vec{u})\beta(\vec{n},\vec{u},\vec{v})d\vec{u}. \tag{4}$$

Factor $\Phi_i(\vec{u})$ is the intensity of the light flow that is incident on the surface from direction $\vec{u}$. This formula assumes that the light flow $\Phi_i(\vec{u})$ is uniform in the sense that it does not depend explicitly on the surface point (p or q) but only on direction $\vec{u}$.

Note that the value of $\Phi_i(\vec{u})$ is irrelevant for directions $\vec{u}$ that point outward from the local surface or for points that are not on the surface; therefore, it is indeed plausible to have the single function Φ_i for the whole scene, which is independent of the local surface orientation. In particular, this lighting model allows attached shadows and is adequate for scenes consisting of a single mostly convex object. On the other hand, this model cannot account for projected shadows, radiosity effects, or sources with uneven light distribution.

If formulas (3) hold, then we can determine normal $\vec{s}\,[p]$ at point p of the scene by finding a point in the reference object that reacts in the same way as p to changes in lighting directions, except for albedo $\overset{*}{s}[p]$ and albedo $\overset{*}{g}[q]$. More precisely, we must find $q \in \mathcal{G}$ such that the m-vectors given as

$$\begin{aligned}\boldsymbol{S}[p] &= (S_1[p], S_2[p], \ldots, S_m[p])\\ \boldsymbol{G}[q] &= (G_1[p], G_2[p], \ldots, G_m[p])\end{aligned} \tag{5}$$

are multiples of each other. We call $\boldsymbol{S}[p]$ and $\boldsymbol{G}[q]$ as the observation vectors (OVs) of points p and q. Having located the matching point q, we can recover the normal vector $\vec{s}[p]$ and the albedo $\overset{*}{s}[p]$ of the scene at p by the following formulas:

$$\vec{s}[p] = \vec{g}[q] \qquad \overset{*}{s}[p] = |\boldsymbol{S}[p]|\,\frac{\overset{*}{g}[q]}{|\boldsymbol{G}[q]|}. \tag{6}$$

This method will fail if there are two points q' and q'' on the reference-object images, which have different normals (i.e., $\vec{g}[q'] \neq \vec{g}[q'']$) but collinear OVs (i.e., $\boldsymbol{G}[q'] = \alpha\boldsymbol{G}[q'']$ for some scalar α). To avoid this problem, the number of images m must be at least 3, and the light fields $\Phi_1, \ldots \Phi_m$ must be varied sufficiently to break any such ambiguities. We will assume that these conditions are satisfied in the following.

On the other hand, example-based VLPS can work in principle with non-Lambertian BRDFs and arbitrary lighting, as well as with attached shadows and penumbras, since these effects do not destroy the proportionality between vectors $\boldsymbol{S}[p]$ and $\boldsymbol{G}[q]$. In particular, there is no need to identify the images and pixels where attached shadows occur (however, cast shadows and scene-scattered light are still a problem).

C. Table Lookup Step

The most time-consuming part of example-based VLPS is locating point $q \in \mathcal{G}$ such that $\boldsymbol{S}[p]$ is a multiple of $\boldsymbol{G}[q]$. As pointed out in [14], if neither vector is zero, this is equivalent to matching the observation signatures $\boldsymbol{s}[p]$ and $\boldsymbol{g}[q]$, which are defined by

$$\boldsymbol{s}[p] = \frac{\boldsymbol{S}[p]}{|\boldsymbol{S}[p]|} \quad \boldsymbol{g}[q] = \frac{\boldsymbol{G}[q]}{|\boldsymbol{G}[q]|}. \tag{7}$$

Here, $|\cdot|$ is any norm of $\mathbb{R}^m$, e.g., the Euclidean norm $|\boldsymbol{X}| = \sqrt{\sum_{i=1}^{m} X_i^2}$. Note that position q is not meaningful by itself; it is only used to associate the observation signature $\boldsymbol{g}[q]$ to normal $\vec{g}[q]$ and the reference-object albedo factor $\gamma[q] = \overset{*}{g}/|\boldsymbol{G}[q]|$. Therefore, we can replace the reference-object images by a signature table, which is an unordered set of triplets as follows:

$$T = \{(\boldsymbol{g}[q], \vec{g}[q], \gamma[q]) : q \in \mathcal{G}\}. \tag{8}$$

The computation of $\vec{s}\,[p]$ then becomes the closest match table lookup problem, where we look for element $t = (t\boldsymbol{g}, t\vec{g}, t\gamma)$ of table T that minimizes the signature distance $\text{dist}(t.\boldsymbol{g}, \boldsymbol{s}[p])$, in some metric $\text{dist}(\cdot,\cdot)$. We will use here the Euclidean metric of $\mathbb{R}^m$, i.e., $\text{dist}(\boldsymbol{x}, \mathbf{y}) = |\boldsymbol{x} - \mathbf{y}|$.

The brute-force solution to this problem would be to scan table T, computing $\text{dist}(t\boldsymbol{g}, \boldsymbol{s}[p])$ for each entry t in it, while keeping track of the closest-matching entry. However, for accurate results, table T must have tens of thousands of entries. Since the lookup must be repeated for each pixel of the scene domain $\mathcal{S}$, it may take tens of minutes to process a single set of scene images with this method.

D. Previous Work

Several authors have worked on so-called "uncalibrated" photometric stereo methods that do not require a reference object. In general, this approach requires certain assumptions about the BRDF and lighting. For instance, in [21], a linear combination of Lambertian and mirror-like components, with extended light sources of known position, was considered but not glossy materials. In [28], linear combinations of Lambertian, forescatter (glossy), and backscatter BRDFs was allowed but considered only the case of three images, i.e., each taken with a known point-like source. Both papers assumed that the images were free from shadows. In [12] and [29], heuristics was used to exclude images and pixels where there were shadows (cast or attached) but assumed a single point source of known intensity in each image; the former assumed a Lambertian BRDF, whereas the latter allowed also a constant ambient term. Another work further along these lines is [17]. However, unreferenced techniques still seem to have limited applicability because they are unreliable and require too many constraints on the lighting and/or the BRDF. In particular, they may return only an unspecified monotonic function of the slopes rather than the actual slopes and therefore require ad hoc scaling of the height map.

Example-based VLPS seems to be closer to practicality, in spite of the inconvenience of including the reference object in the scene. In this approach, the bottleneck is the signature table lookup step. Several techniques have been proposed in the literature to speed up this step. Woodham [32] used a regular m-dimensional grid spanning hypercube $[0, 1]^m$, with 2^b cells along each axis, for some bit count b. In the preprocessing phase, each reference-object OV $\boldsymbol{G}[q]$ was quantized with b bits per coordinate, yielding the m-tuple of indices of some grid cell where the associated normal vector $\vec{g}[q]$ was stored. (Woodham assumed uniform albedos $\overset{*}{s} = \overset{*}{g}$; therefore, there was no reason to normalize the OVs). In the lookup phase, each OV $\boldsymbol{S}[p]$ of the scene was mapped to a table cell in the same way, and the desired normal $\vec{s}[p]$ was recovered from the grid. One obvious disadvantage of this method is the size of the grid, i.e., 2^{mb} entries, which is about 250 000 for $m = 3$ and $b = 6$.

Later studies have proposed other general m-dimensional nearest point algorithms for this task. For instance, Hertzmann and Seitz [14] use the approximate nearest neighbor (ANN) of [3], whereas Zhong and Little [34] substitute the locally sensitive hashing (LSH) in [16]. One could also consider the k-D tree method in [7]. However, all these methods have a common shortcoming: they consider the set of reference-object signatures $\mathbb{T} = \{\boldsymbol{g}[q] : q \in \mathcal{G}\}$ to be a generic cloud of points scattered in m-dimensional space and therefore use general m-dimensional nearest-neighbor search algorithms, which are inherently expensive in space and/or time. As detailed in Section III, our method is much faster and (unlike ANN and LSH) always gives the exact best match.

II. Fast Table Searching With 2-D Bucketing

We now describe an algorithm to find the best matching entry in the signature table that exploits the special shape of the reference-object signature set $\mathbb{T}$ to achieve a very fast look up at a modest space cost. An earlier version of this algorithm was described in [18].

A. Shape of the Signature Table

The key observation for our improved method is that set $\mathbb{T}$ of all reference-object signatures is essentially a 2-D subset of $\mathbb{R}^m$. Therefore, we can solve the lookup problem very efficiently by a 2-D uniform grid technique, as described in [1].

To understand the key observation above, note that, because of formulas (3) and (7), the observation signatures $\boldsymbol{s}[p]$ and $\boldsymbol{g}[q]$ can be expressed as $\boldsymbol{l}(\vec{s}[p])$ and $\boldsymbol{l}(\vec{g}[q])$, respectively, where $\boldsymbol{l}$ is the lighting signature function as follows:

$$\boldsymbol{l}(\vec{n}) = \frac{\boldsymbol{L}(\vec{n})}{|\boldsymbol{L}(\vec{n})|} \tag{9}$$

and $\boldsymbol{L}(\vec{n}) = (L_1(\vec{n}), \ldots L_m(\vec{n}))$. Note that function $\boldsymbol{l}$ that maps surface normals to lighting signatures is defined only on the hemisphere $\mathbb{H}^2$ of $\mathbb{S}^2$ consisting of the normal directions that deviate less than 90° from the viewing direction $\vec{v}$. On the other hand, a good reference object must provide a fairly dense and uniform sampling of $\mathbb{H}^2$ (which is why spheres are used normally for that purpose). It follows that the set of reference-object signatures $\mathbb{T}$ must be a fairly dense and uniform cover of $\mathbb{K} = \boldsymbol{l}(\mathbb{H}^2)$, i.e., the range of the function $\boldsymbol{l}$.

Now, given our assumption that the reference object's BRDF β lacks the sharp spikes of mirror-like reflection, the shading factors $L_i(\vec{n})$ given by formula (4) are the continuous functions

of the surface normal $\vec{n}$. In fact, L_i is typically fairly smooth, with just a few broad and hardly distinguishable maxima. Furthermore, the example-based VLPS problem is solvable if and only if function $\boldsymbol{l}(\vec{n})$ is invertible, i.e., for every point $\boldsymbol{v}$ of $\mathbb{S}^{m-1}$, there is at most one direction $\vec{n}$ such that $\boldsymbol{l}(\vec{n}) = \boldsymbol{v}$. If this condition holds, the range $\mathbb{K}$ of $\boldsymbol{l}$ is a continuous one-to-one embedding of hemisphere $\mathbb{H}^2$ into $\mathbb{S}^{m-1}$. Finally, since the observation signatures are contained in the positive orthant of $\mathbb{R}^m$, the angular diameter of $\mathbb{K}$, i.e., as seen from the origin of $\mathbb{R}^m$, is at most 90°.

From these considerations, intuition suggests (and experience confirms) that the range $\mathbb{K}$ of $\boldsymbol{l}$ is a relatively flat patch of a 2-D manifold (surface) immersed in $\mathbb{S}^{m-1}$, whose curvature depends on the angular spread of light sources. Moreover, the reference-object signatures $\mathbb{T}$ must be distributed over $\mathbb{K}$ with fairly uniform density.

B. The 2-D Grid Scheme

In our method, the signature table T is preprocessed as follows. We first compute the centroid $\boldsymbol{b}$ of the set $\mathbb{T}$ of all signatures (seen as a set of points of $\mathbb{R}^m$) and two orthogonal unit vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^m$ that define its directions of maximum extent. These vectors are found by computing the $m \times m$-coordinate moment matrix M of the displacements $\boldsymbol{g} - \boldsymbol{b}$, for all $\boldsymbol{g} \in \mathbb{T}$, and by taking the unit-length eigenvectors associated to its two largest eigenvalues. Point $\boldsymbol{b}$ and vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ define the 2-D affine subspace P of $\mathbb{R}^m$, and the signature projection plane, which is roughly coplanar with set $\mathbb{T}$. The orthogonal projection onto P of a given observation signature $\boldsymbol{g}$ will be denoted by $\downarrow \boldsymbol{g}$.

Next, we choose a square uniform grid of $N \times N$ square cells on the projection plane P. This grid is centered on point $\boldsymbol{b}$, has its sides parallel to the vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, and is barely large enough to contain the projection $\downarrow \boldsymbol{g}$ of any signature $\boldsymbol{g}$ in $\mathbb{T}$. More precisely, the grid is a square of size $2R$, where

$$R = \varepsilon + \max\{|(\boldsymbol{g} - \boldsymbol{b}) \cdot \boldsymbol{u}|, |(\boldsymbol{g} - \boldsymbol{b}) \cdot \boldsymbol{v}| : \boldsymbol{g} \in \mathbb{T}\} \quad (10)$$

for some small safety margin ε. Each cell of the grid is then a square with side $\tau = 2R/N$.

Having chosen the grid, we build, i.e., for each cell $C[i, j]$ of the grid, the linked bucket list $B[i, j]$ of all table entries t in T, whose signatures $t\boldsymbol{g}$ project onto that cell. We also compute the corresponding bucket mean $\mu[i, j]$, which is defined as the barycenter of all signatures $t\boldsymbol{g}$ in list $B[i, j]$, and the bucket radius $\rho[i, j]$, which is defined as the the maximum distance from $\mu[i, j]$ to any signature $t\boldsymbol{g}$ in that list (see Fig. 3).

The 2-D shape of $\mathbb{T}$ means that the entries in $B[i, j]$ are fairly close to each other, even if their mean distance from plane P is large compared with the cell size. This property remains true even when m is greater than 3.

Once the grid structure has been constructed, each scene signature $\boldsymbol{s}[p]$ is looked up with procedure 1 below. Its steps are explained in Sections II-C–II-E.

C. Bucket-Grid Searching

Procedure 1 begins by computing the indices (i, j) of the cell that contains the projection $\downarrow \boldsymbol{s}$ of the given signature $\boldsymbol{s}$ (steps 1–2). It then scans list $B[i, j]$ for the entry that is closest to $\boldsymbol{s}$. If necessary, it continues the search by scanning nearby buckets $B[i', j']$, in some appropriate order (steps 3–4). Note that some buckets may be empty and, moreover, that the best match to query $\boldsymbol{s}$ may not be in bucket $B[i, j]$, i.e., even if that bucket is nonempty.

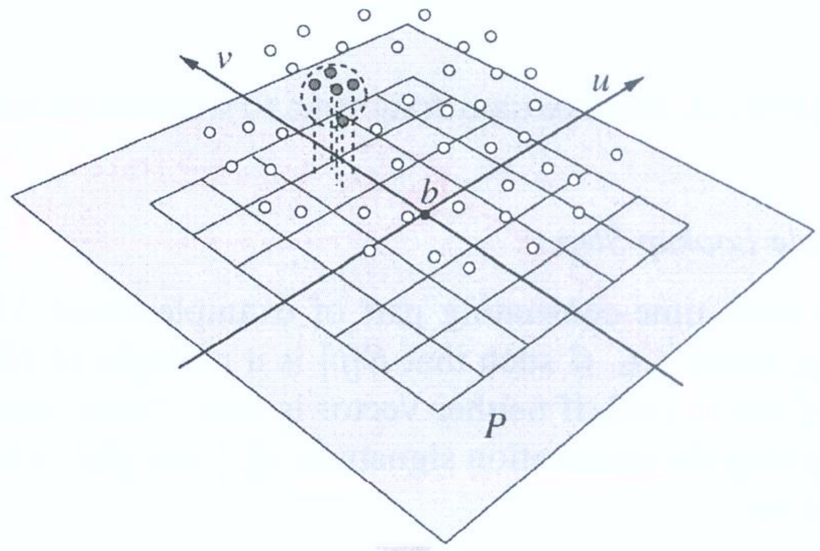


Fig. 3. Illustration of the 2-D bucketing algorithm for $m = 3$, showing some observation signatures in $\mathbb{T}$ (small open circles), the projection plane P (grey quadrilateral), barycenter b and main axes u and v of $\mathbb{T}$, the uniform grid, the bucket list $B[i, j]$ (small filled circles), and its enclosing sphere (large dotted circle) as defined by the centroid $\mu[i, j]$ and radius $\rho[i, j]$.

Procedure 1 (Table lookup): Given an observation signature $\boldsymbol{s}$, finds the entry $t$best in the table T whose signature $t$best$\boldsymbol{g}$ is most similar to $\boldsymbol{s}$.

---

1. $i \leftarrow \lfloor((\boldsymbol{s} - \boldsymbol{b}) \cdot \boldsymbol{u} + R)/\tau\rfloor$;
2. $j \leftarrow \lfloor((\boldsymbol{s} - \boldsymbol{b}) \cdot \boldsymbol{v} + R)/\tau\rfloor$;
3. $d$best $\leftarrow +\infty$; $t$best $\leftarrow$ **null**;
4. For each pair (δ_i, δ_j) in Δ, in order, do
   5. If $d$best $\leq \tau\Lambda(\delta_i, \delta_j)$, return $t$best.
   6. $(i', j') \leftarrow (i, j) + (\delta_i, \delta_j)$;
   7. If $0 \leq i' < N$ and $0 \leq j' < N$, then
   8. If $d$best $>$ dist$(\boldsymbol{s}, \mu[i', j']) - \rho[i', j']$, then
      9. For each t in $B[i', j']$, do
         10. Set $d \leftarrow$ dist$(\boldsymbol{s}, t\boldsymbol{g})$;
         11. If $d < d$best,
             set $d$best $\leftarrow d$ and $t$best $\leftarrow t$.
12. Return $t$best.

The stored bucket attributes $\mu[i, j]$ and $\rho[i, j]$ allow us to skip over quickly buckets that cannot contain possibly a better match to the query signature $\boldsymbol{s}$. More precisely, we should examine bucket $B[i', j']$ only if the query signature $\boldsymbol{s}$ is closer to the bucket's bounding ball than to the best matching signature found so far in the table (step 8), i.e., only if

$$\text{dist}(\boldsymbol{s}, \mu[i', j']) - \rho[i', j'] < \text{dist}(\boldsymbol{s}, t\text{best}\boldsymbol{g}) = d\text{best}. \quad (11)$$

We will call condition (11) the bucket scan condition.

D. Bucket Search Order and Early Return

The bucket scan condition (11) will often allow us to skip bucket $B[i', j']$ without examining its entries. However, if we were to apply this criterion for all buckets, individually, the running time would still be proportional to the number N^2 of buckets in the grid which, as in any hashing scheme, is expected to be proportional to the size of the table. In that case, the grid-based search would improve on the brute-force solution only by a constant factor at best.

To avoid scanning the whole grid, we examine buckets $B[i', j']$ in a specific order, starting with bucket $B[i, j]$ that contains the projected query $\downarrow \boldsymbol{s}$ and moving gradually away from it (step 4). A second criterion (step 5) then allows us to abandon the search as soon as we detect that none of the buckets still to be scanned can contain possibly a better match than the one found so far. Typically, this happens after scanning only a small fraction of the bucket array.

Consider two signatures $\boldsymbol{s}'$ and $\boldsymbol{s}''$ that project into cells $C[i', j']$ and $C[i'', j'']$, respectively. It is easy to see that

$$\text{dist}(\boldsymbol{s}', \boldsymbol{s}'') \geq \text{dist}\,(C[i', j'], C[i'', j'']) \,. \tag{12}$$

In this formula, $\text{dist}(C[i', j'], C[i'', j''])$ is the minimum distance between the two cells, ~~which is~~ seen as the subsets of plane P, i.e.,

$$\text{dist}\,(C[i', j'], C[i'', j'']) = \tau\Lambda(i' - i'', j' - j'') \tag{13}$$

where

$$\Lambda(\delta_i, \delta_j) = \sqrt{\left(|\delta_i| \stackrel{\wedge}{-} 1\right)^2 + \left(|\delta_j| \stackrel{\wedge}{-} 1\right)^2} \tag{14}$$

where

$$x \stackrel{\wedge}{-} y = \max\{0, x - y\}. \tag{15}$$

Note that $\Lambda(\delta_i, \delta_j)$ is a bit smaller than the Euclidean norm $|(\delta_i, \delta_j)|$ of the integer pair.

We conclude that bucket $[i', j']$ can be ignored if the cell distance bound (12) excludes the possibility that a better match can be found within it, i.e., if

$$\tau\Lambda(i' - i, j' - j) \geq \text{dist}(\boldsymbol{s}, t\text{best}\boldsymbol{g}) = d\text{best}. \tag{16}$$

Note that condition (16) is weaker than the bucket scan condition (11). However, condition (16) depends only on the current best match $t$best and the difference $(i' - i, j' - j)$ between the cell indices. Therefore, if we scan buckets (i', j') in such an order that $\Lambda(i' - i, j' - j)$ is increasing, we can stop the search as soon as that condition is satisfied (step 5).

For that purpose, as part of the table preprocessing, we precompute the list Δ of all vectors (δ_i, δ_j) within $|\delta_i|, |\delta_j| < N$, ~~which is~~ sorted by the increasing value of $\Lambda(\delta_i, \delta_j)$ (and breaking ties by $|(\delta_i, \delta_j)|$). For each query signature $\boldsymbol{s}$, we enumerate buckets $B[i', j']$ by taking each displacement (δ_i, δ_j) from the ordered list Δ and computing $(i', j') \leftarrow (i, j) + (\delta_i, \delta_j)$ (step 6), provided that i' and j' lie in $\{0, \ldots N - 1\}$ (step 7). In practice, it is more convenient to work with the squared bound $\Lambda^2(\delta_i, \delta_j)$ instead of $\Lambda(\delta_i, \delta_j)$ (see Fig. 4).

8	5	4	4	4	5	8
5	2	1	1	1	2	5
4	1	0	0	0	1	4
4	1	0	**0**	0	1	4
4	1	0	0	0	1	4
5	2	1	1	1	2	5
8	5	4	4	4	5	8

45	41	33	27	34	42	46
37	21	17	11	18	22	38
29	13	05	03	06	14	30
25	09	01	**00**	02	10	26
31	15	07	04	08	16	32
39	23	19	12	20	24	40
47	43	35	28	36	44	48

Fig. 4. Squared cell distance function $\Lambda^2(\delta_i, \delta_j)$ (left) and the bucket scan order implied by (right) list Δ, for the 7×7 cells surrounding the starting cell.

E. Analysis

The average cost of algorithm 1 for one scene pixel p is roughly $Bb + Dd + O(1)$, where b is the average number of buckets examined (step 5–7), d is the average number of times the signature distance function dist is evaluated (steps 8 and 10), and B and D are the costs associated to those two operations.

In the extreme case when $N = 1$, we will have $b = 1$ and $d = \#T + 1$, which is essentially equivalent to a brute-force search of T. This is also the theoretical worst case, i.e., when almost all the signatures of the table project into the same bucket. However, given the way that the projection is chosen, this only happens when the signatures are very similar all over the reference-object image. This in turn means that all images are taken with nearly isotropic illumination so that the shading is independent of the surface normal. Obviously, such data are inadequate for photometric stereo.

At the other extreme, when N is very large, most buckets are expected to contain at most one entry, and d will tend to a minimum (whose value depends on the curvature of the set $\mathbb{T}$). At the same time, b will increase immediately to about 10 because $\Lambda(\delta_i, \delta_j)$ is 0 for the first nine pairs (δ_i, δ_j) in list Δ. Thereafter, b will grow slowly in proportion to N^2 because the procedure will have to skip increasingly more empty buckets before finding the first nonempty one. The optimal value of N (which minimizes the running time) depends primarily on ratio B/D and, secondarily, on the shape of the signature cloud. It turns out that, in our tests, the effects of increasing b and decreasing d cancel mostly each other so that the running time is nearly independent of the grid size for $\kappa < 0.5$, with a weak minimum somewhere between $\kappa = 0.05$ and $\kappa = 0.25$ (i.e., N between $2\sqrt{\#T}$ and $4\sqrt{\#T}$). (see Table III and Fig. 20).

F. Extension for Color Images

Our 2-D grid method is adapted easily to color images. If each image has c spectral bands (color channels), the ~~color~~ OVs $\boldsymbol{S}[p]$ and $\boldsymbol{G}[q]$ are the concatenation of c monochromatic OVs with m components each. As before, in order to recover the scene normal $\vec{s}[p]$ at point p, we look for the reference-object point q such that the color signatures $\boldsymbol{s}[p]$ and $\boldsymbol{g}[q]$ match, except that the color signatures are obtained from the color OVs by normalizing the OV in each band separately. The color signatures then become points of space $(\mathbb{S}^m)^c \subseteq \mathbb{R}^{mc}$, but they are still a 2-D manifold in that space and therefore can be organized and searched using a single 2-D grid.

Once the matching table entry is known, formula (6) yields the separate albedo $\overset{*}{s}[p]$ for each channel. These albedos can be

combined into a single color image, i.e., the intrinsic color map of the scene (see Fig. 16).

III. Tests

To evaluate our method, we tested it with a set of synthetic images from a simple geometric scene and a set of digital photos of a real scene. While synthetic images are not acceptable for validating complete VLPS algorithms, they still provide useful information. Synthetic images can provide lookup tables of fairly realistic size and shape and allow us to evaluate the performance of the table lookup procedure in ideal conditions, when the images are free from nonessential defects such as nonlinear camera sensor response, image alignment errors, nonuniform lighting, imperfect reference objects, and pixel noise. On the other hand, tests with photos of a real scene are necessary to estimate the impact of all these factors on the algorithm's accuracy and efficiency.

In both tests, each reference-object image was interpolated at 11 172 sampling points to produce the signature table T. For the tests with real images, some entries along the edges of the reference object had to be discarded because they were too dark in all images (i.e., $|\boldsymbol{G}|$ was very small), leaving 10 967 usable table entries. We used grids with five different sizes $N \times N$, either 422×422, 299×299, 211×211, 149×149, and 105×105, corresponding to approximate average entry-to-bucket ratios $\kappa = \#T/N^2$ 6%, 12%, 25%, 50%, and 100%, respectively. We also processed each image set with the grid size $N = 1$, which is essentially equivalent to the brute-force nearest match algorithm. In all tests, we verified that the observation signatures returned by our table lookup procedure were always identical to the results of the brute-force search. We applied the algorithm to each color channel (R, G, and B) separately, as well as to the computed luminance (grayscale) channel $Y = 0.2989R + 0.5866G + 0.1145B$.

The basic output of any photometric-stereo algorithm is a normal map (or slope map). The computation of a height map from that slope data is a separate problem that does not concern us here. Nevertheless, we performed such conversion for both tests for visualization purposes and as a simple check of the correctness of the output. We used a fast but effective integrator [23]. The data reliability weight map required by that integrator was estimated from the distance between the pixel signature and the best matching reference-object signature. Pixels along height discontinuities (around the tilted cylinder and the double ramp) were marked by hand by setting the corresponding weights to zero. For display, each integrated height map was turned into a fine triangular mesh (with four triangles for each input image pixel) and rendered with POV-Ray, with arbitrary illumination.

All source programs (in the GNU variant of C), test data files, `bash` and `gawk` shell scripts, and usage instructions are available in the Unix `tar` archive format at the site www.liv.ic.unicamp.br/, file saracchini/projects/photo-hash/2011-03-17.tar.gz.

A. Tests With Synthetic Images

The synthetic images were produced with the POV-Ray ray tracer [8], which was from a simple virtual scene consisting of blocks of various shapes and colors lying on a flat checkered surface (see Fig. 5).

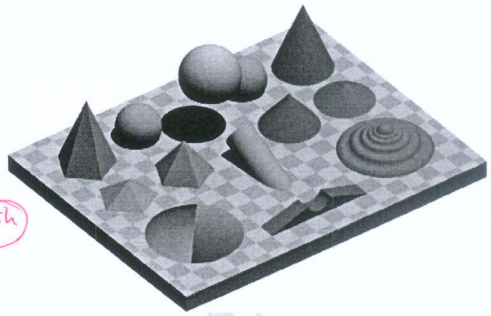


Fig. 5. Perspective image of the synthetic scene.

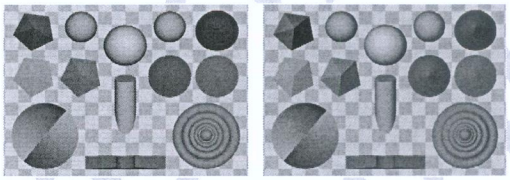


Fig. 6. Two of the 48 synthetic input scene images, with light inclinations (left) $\theta = 5°$ and (right) $\theta = 14°$.

All objects had a Lambertian finish with various colors. The object at bottom left, in particular, was a simple spherical cap painted with gradually varying shades of blue gray to give a false impression of relief.

We generated 48 images of the scene; each image being produced with a single pointlight source (see Fig. 6). The source-to-scene distance was set to about 50 000 times the width H of the visible part of the scene, which was large enough to ensure practically uniform illumination over the entire scene, and the `shadowless` feature of POV-Ray was used to eliminate cast shadows while retaining the attached shadows. The directions of the light sources were distributed evenly by hand with the maximum inclination $\theta_{\max} = 25°$ from the vertical. The distance from the virtual camera to the scene was set to 500 H so as to achieve nearly parallel orthographic projection. The POV-Ray scene images were generated as 16-bit PNG files [22], which were reduced from 630×450 to 420×300 pixels with Lanczos filtering [9] and converted to 16-bit PPM format [13]. For each scene image, we also computed a synthetic image of a white spherical reference object using the same lighting and the same Lambertian BRDF. Fig. 7 shows the normal map $\vec{s}$ computed by our program for $m = 24$ input images.

Fig. 8 shows the computed albedo map $\overset{*}{s}$. Fig. 9 shows the normal error map, i.e., the absolute difference between the normal vectors $\vec{s}[p]$ computed by our program and the true normal vectors (known from the POV-Ray model). We attribute those errors to various sources: color mixing at pixels that straddle sharp edges and silhouettes, inadequate sampling of the reference object where its surface was nearly vertical, and quantization noise in the input images. Note that the errors are significant only at the edge pixels and where the albedo is

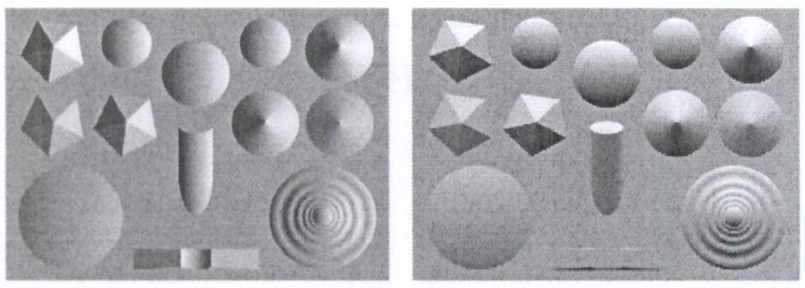


Fig. 7. The X and Y components of the computed normal map, using $m = 24$ images.

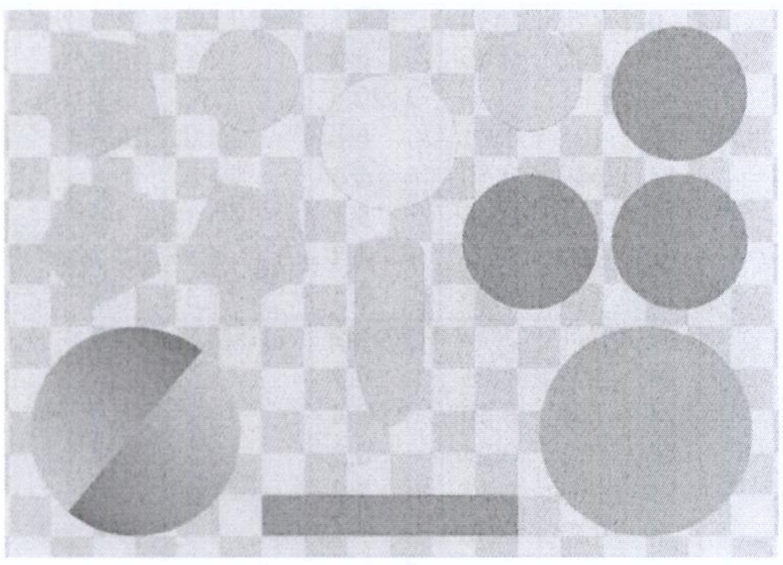


Fig. 8. Computed intrinsic color map of the scene, which is a composite of the R, G, and B albedo maps.

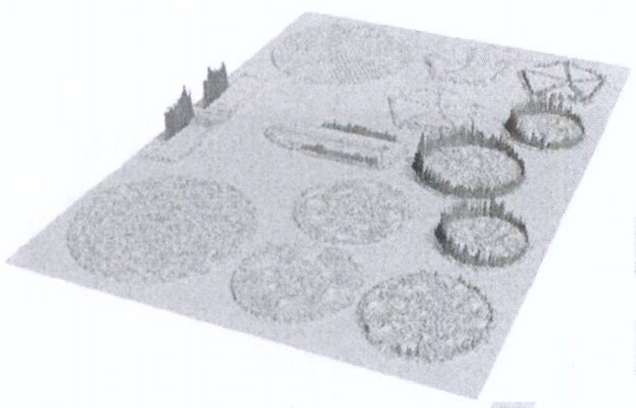


Fig. 9. Plot of the errors in the computed normals of Fig. 7. The maximum error corresponds to an angle of 0.16 rad (about 9°) and the root-mean-square error to 0.009 rad (0.52°).

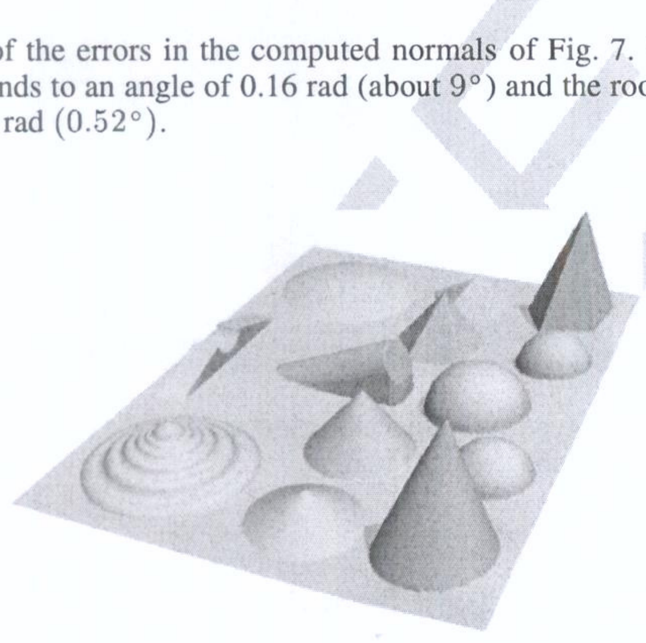
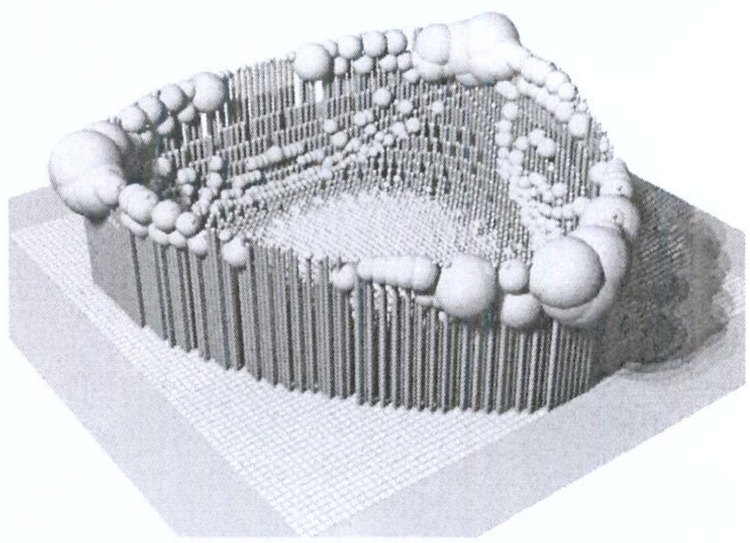


Fig. 10. A 3-D perspective view of the height map obtained by integrating the normal map of Fig. 7.

lowest. Fig. 10 shows a 3-D view of the height map obtained by integrating the computed normal map.

As expected, the main height errors occur near discontinuities in the height map, where the normal map gives no information about the relative height of adjacent regions. The perceptible tilt of the pyramids and cones away from the vertical is an unavoidable parallax effect due to the finite camera-to-scene distance.

Fig. 11 shows the shape of the signature set $\mathbb{T}$.



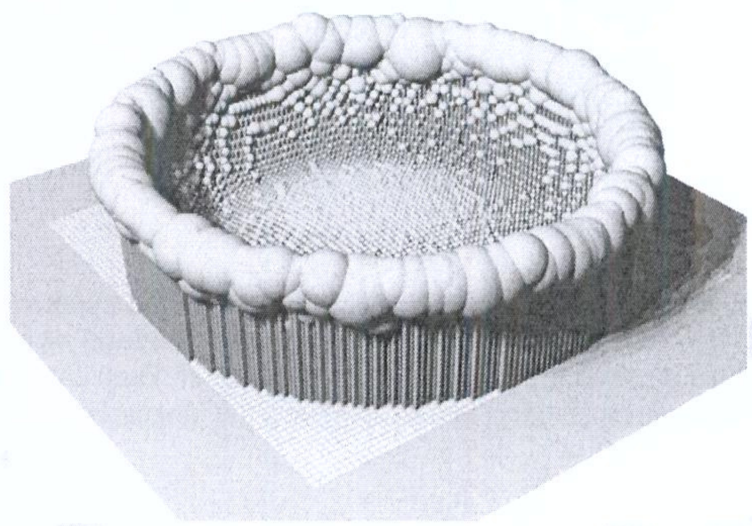


Fig. 11. Shape of the signature set $\mathbb{T}$ from the synthetic data sets, for maximum light inclination $\theta_{\max} = 25°$, grid size $N \times N = 75 \times 75$, and (top) $m = 6$ images or (bottom) $m = 48$ images. The darker quadrangle represents square $[-1, +1] \times [1, +1]$ on the projection plane P; the lighter quadrangle is the part of P covered by the cell grid. The height of each stick is the distance from the centroid $\mu[i, j]$ of the entries in each bucket to the center of the grid cell $c[i, j]$ on plane P. The radius of the sphere is the bucket radius $\rho[i, j]$. The same scale is used for all quantities.

Fig. 11 is not a simple projection of $\mathbb{T}$ because the displacement vectors $\mu[i, j] - c[i, j]$ are not necessarily parallel in $\mathbb{R}^m$ as they appear to be in the figure. Therefore, $\mathbb{T}$ may be twisted in complicated ways. Still, Fig. 11 depicts accurately the deviation of $\mathbb{T}$ from plane P. The figure thus confirms our claim that set $\mathbb{T}$ is fairly flat over most of its extent.

Fig. 12 shows the number of entries in each bucket list $B[i, j]$ observed in those tests. The table grid size was $N \times N = 211 \times 211$, corresponding to a mean entry-to-bucket ratio $\kappa = 25\%$. Note that the observation signatures are distributed fairly evenly over a substantial fraction of the grid.

Tables I and II shows various cost metrics for a single call of the table lookup operation. The columns are the number m of input images, the grid size N and the corresponding entry-to-bucket ratio κ, the lookup time t in microseconds, the number b of buckets $B[i', j']$ that were examined, and the number d of table entries that are tested (i.e., the number of times $\text{dist}(\boldsymbol{s}, \boldsymbol{g})$ was computed). All values are averaged the over all pixels of $\mathcal{S}$. The tests were run on a machine with an Intel Xeon 2.5 GHz clock and 8 GB random-access memory device; the absolute time t will depend obviously on the platform and the implementation.

B. Tests With Photos of a Real Scene

For the tests with real data, we used a scene consisting of a plaster sculpture hand-painted with tempera colors (see

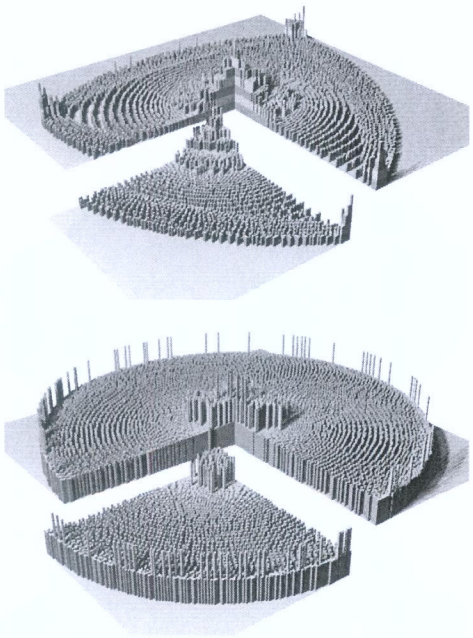


Fig. 12. Number of entries in each grid bucket for the synthetic scene, for $\theta_{\max} = 25°$, $N = 211$, and (top) $m = 6$ images or (bottom) $m = 48$ images. The quadrilateral shows the grid extent on the plane P. Each color stripe on the bars represents one table entry. In the top image, the longest bucket list has eight entries.

TABLE I
MAIN SYMBOLS USED

Symb.	Definition	Section
m	Number of input images	I
S_i, G_i	Scene and reference object images	I, I-A
Φ_i	Lighting field for images S_i, G_i	I
L_i	Shading function for images S_i, G_i	I
$\mathscr{S}$, $\mathscr{G}$	Domains of scene and ref. object images	I, I-A
p, q	Points in $\mathscr{S}$ and $\mathscr{G}$	I, I-A
β_S, β_G	BRDFs of scene and ref. object	I-A
β	Common unscaled BRDF	I-A
λ	Spectral band index	I-A
$\dot{s}$, $\dot{g}$	Albedos of scene and ref. object	I-A
$\boldsymbol{S}$,$\boldsymbol{G}$	Observation vectors of scene and ref. object	I-B
$\boldsymbol{s}$, $\boldsymbol{g}$	Signatures of scene and ref. object	I-C
γ	Ref. Object albedo factor $\dot{g}/\lvert\boldsymbol{G}\rvert$	I-C
$\vec{s}$, $\vec{g}$	Normal directions of scene and ref. object	I, I-A
T	Signature table	I-C
$\mathbb{T}$	Set of all ref. object signatures $\boldsymbol{g}$ in T	I-D
$\boldsymbol{b}, \boldsymbol{u}, \boldsymbol{v}$	Centroid and principal axes of $\mathbb{T}$	II-B
P	Plane of the grid in $\mathbb{R}^m$	II-B
R	Half-side of grid	II-B
N	Number of rows and columns in grid	II-B
τ	Size of grid cells	II-B
$C[i,j]$	Grid cell in column i, row j	II-B
$B[i,j]$	List of signatures in $C[i,j]$	II-B
$\mu[i,j]$	Barycenter of $B[i,j]$	II-B
$\rho[i,j]$	Radius of $B[i,j]$ measured from $\mu[i,j]$	II-B
Λ	Min cell-to-cell distance	II-B
Δ	Precomputed search order list	II-B
θ	Inclination of light from vertical	III

Fig. 13). The reference objects were four table tennis balls spray-painted with matte white enamel paint. The reference objects were nested into shallow conical cups made of black

TABLE II
AVERAGE COSTS AND OPERATION COUNTS OF THE TABLE LOOKUP PROCEDURE FOR SYNTHETIC IMAGES WITH VARIOUS VALUES OF m AND N. THE ENTRIES WITH $N = 1$ CORRESPOND TO THE BRUTE-FORCE TABLE SEARCH (WITHOUT THE GRID-BASED SPEED-UP)

m	N	κ	t	d	b
3	1	—	1159.4	11172.0	1.0
3	105	1.01	3.9	21.1	10.0
3	149	0.50	3.1	14.4	10.0
3	211	0.25	2.9	12.3	10.1
3	299	0.12	2.5	9.2	10.4
3	422	0.06	2.5	7.7	11.2
6	1	—	1499.7	11172.0	1.0
6	105	1.01	5.9	27.9	10.2
6	149	0.50	4.7	18.4	10.3
6	211	0.25	4.0	14.2	10.6
6	299	0.12	3.7	11.3	11.3
6	422	0.06	3.7	10.1	12.6
12	1	—	2120.5	11172.0	1.0
12	105	1.01	6.6	18.0	10.4
12	149	0.50	6.0	14.0	10.8
12	211	0.25	5.2	10.3	11.5
12	299	0.12	4.8	8.5	13.1
12	422	0.06	4.5	5.3	16.4
24	1	—	3745.2	11172.0	1.0
24	105	1.01	9.9	15.4	10.4
24	149	0.50	9.4	13.7	10.9
24	211	0.25	8.3	10.0	11.7
24	299	0.12	7.8	7.6	13.5
24	422	0.06	7.1	5.0	17.0
48	1	—	7110.6	11172.0	1.0
48	105	1.01	17.1	15.0	10.5
48	149	0.50	15.8	12.5	11.0
48	211	0.25	14.2	9.8	11.9
48	299	0.12	12.9	6.8	14.0
48	422	0.06	11.7	4.4	18.0

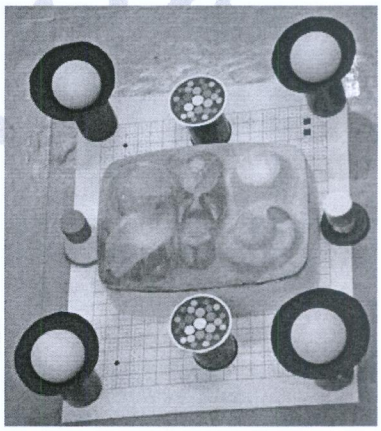


Fig. 13. Perspective photo of the test object used in the real-data test. Note the reference objects inside the light baffles.

paper to block stray light from the scene. We also included in the scene two grayscale calibration charts.

We took 48 digital pictures of this scene with a 3-Mpixel consumer camera (Sony DSC-W50), which is mounted on a photographer's stand 119 cm above the scene's background paper sheet (see Fig. 1). The camera's flash and sharpening filter were turned off, and time-lapse triggering was used to reduce vibrations. Lighting was provided by a 500-W cylindrical halogen lamp in a 8 cm × 16 cm reflector, held about 1 m from the scene.

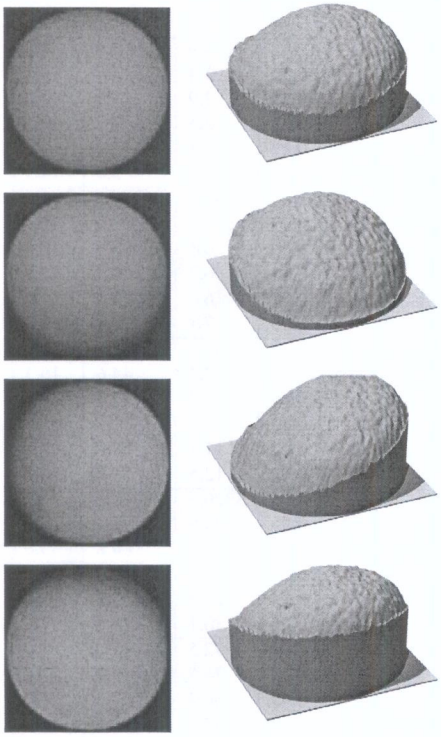


Fig. 14. Images of the bottom right reference object of Fig. 13, under the same lighting as the images of Fig. 1 (left column) and 3-D plots of their intensities at the lookup table sampling points (right column).

The lamp was moved around so that the inclination θ varied between $0°$ and $25°$ in various directions. The pictures were shot at night with the room lights turned off, but we made no special attempt to suppress secondary illumination by light scattered from the lamp by nearby walls and furniture.

The photos were saved in "fine" quality JPEG at 2816×2122 pixels. They were corrected in the computer for radial distortion, translation-aligned to ± 0.25 pixel, cropped to 834×1158 pixels (the sculpture proper), and finally scaled down to 278×386 pixels in order to reduce the camera and JPEG compression noise.

The relatively large field of view (about 30 cm at 120 cm) resulted in noticeable parallax distortion near the edges of the image. In particular, the reference-object outlines were distinctly elliptical, and the reference-object point with normal (0, 0, 1) was offset perceptibly from the ellipse's center. These distortions were taken into account when building the lookup tables by assuming that each reference object was imaged by a parallel but oblique projection that locally matched the camera's perspective projection. We found that the reference-object images were affected by camera noise and small defects in the spray-on paint (see Fig. 14). Surprisingly, these high-frequency defects had relatively little impact on the computed normal and height maps. For the tests reported here, we used the reference object at the bottom right of Fig. 13.

Fig. 15 shows the scene's normal map computed by our method from the luminance channel of $m = 24$ input images (the maps obtained in separate tests with $m = 6$ to $m = 48$ images were barely distinguishable from this one).

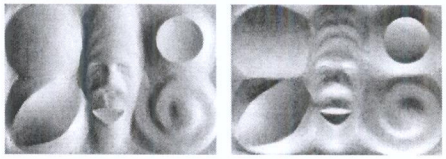


Fig. 15. The X and Y components of the normal map computed from $m = 24$ luminance images of the real scene.

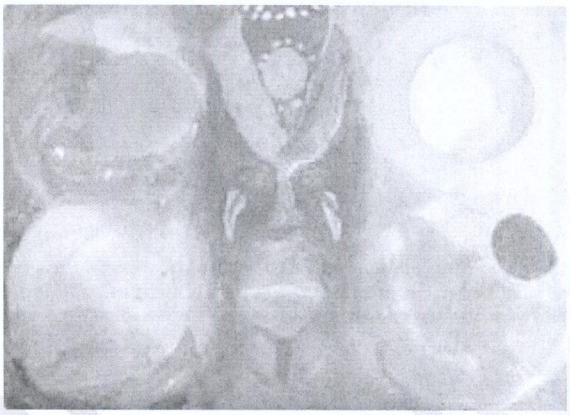


Fig. 16. Computed intrinsic color map of the real scene, which is a composite of the R, G, and B albedo maps.

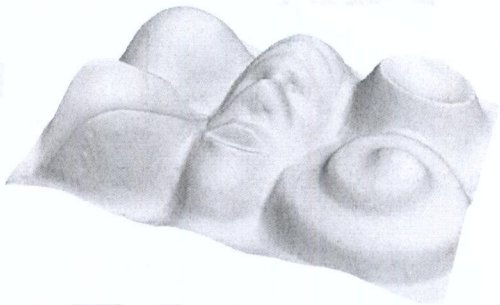


Fig. 17. A 3-D perspective view of the height map obtained by integrating the normal map of Fig. 15.

Fig. 16 shows the scene's intrinsic color map, which is obtained by composing the albedo maps computed separately for the three color channels. Fig. 17 is a 3-D rendering of the height map obtained by integrating the normal map, as in Fig. 10. Fig. 18 shows the shape of the signature set $\mathbb{T}$ extracted from the reference-object images. As in the tests with synthetic images, we observe that the set stays fairly close to the the grid's projection plane and is nearly 2-D. The first observation is attested by the lengths of the sticks and the second one by the smooth variation in the lengths of the sticks and by the relatively small radii of the bucket-enclosing balls.

Fig. 19 shows the lengths of bucket lists $B[i, j]$ in selected tests with a grid of $N \times N = 211 \times 211$ buckets, i.e., the mean entry-to-bucket ratio $\kappa = 25\%$. Table III and Fig. 20 show the performance metrics for our lookup procedure, running on a monochromatic (luminance channel) version of this test data, for various numbers of lights m and grid size N. The columns are the same as in Table II. Notice that b decreases and

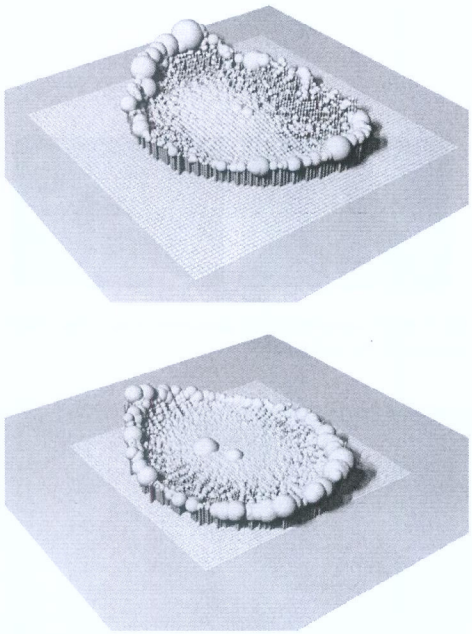


Fig. 18. Shape of the signature set $\mathbb{T}$ obtained from photos of the real reference object, with the same conventions as Fig. 11, for (top) $m = 6$ images and (bottom) $m = 48$ images. The grid size was set to $N \times N = 75 \times 75$.

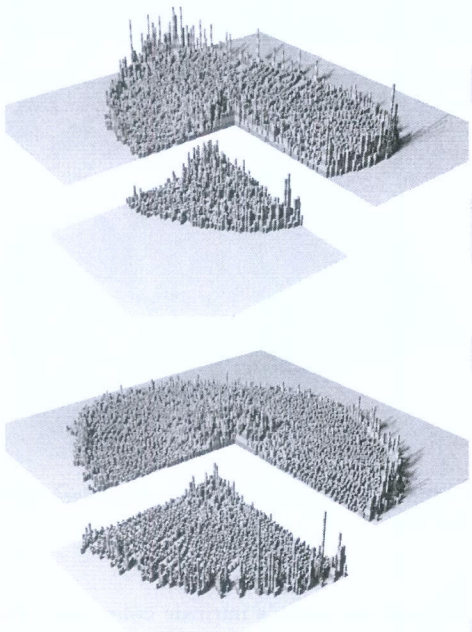


Fig. 19. Number of entries in each bucket list for photos of the real reference object, for $N = 211$ and (top) $m = 6$ images or (bottom) $m = 48$ images. The conventions are those of Fig. 12. The longest bucket list for $m = 6$ had 13 entries.

TABLE III
AVERAGE COSTS AND OPERATION COUNTS OF OUR TABLE LOOKUP PROCEDURE FOR VARIOUS VALUES OF m AND N. THE ENTRIES WITH $N = 1$ CORRESPOND TO BRUTE-FORCE TABLE SEARCH (WITHOUT THE GRID-BASED SPEED-UP)

m	N	κ	t	d	b
3	1	—	1134.9	10967.0	1.0
3	105	0.99	3.4	15.8	10.0
3	149	0.49	2.9	11.3	10.0
3	211	0.25	2.5	7.9	10.3
3	299	0.12	2.3	5.4	11.5
3	422	0.06	2.1	3.9	14.3
6	1	—	1458.2	10967.0	1.0
6	105	0.99	7.5	37.5	12.1
6	149	0.49	6.5	28.6	15.8
6	211	0.25	6.3	24.0	22.9
6	299	0.12	6.6	20.7	35.0
6	422	0.06	7.1	18.2	56.1
12	1	—	2137.1	10967.0	1.0
12	105	0.99	12.1	44.6	16.5
12	149	0.49	11.5	37.3	24.6
12	211	0.25	11.0	31.9	37.6
12	299	0.12	11.2	27.2	60.5
12	422	0.06	12.5	23.8	102.1
24	1	—	3920.5	10967.0	1.0
24	105	0.99	28.9	73.8	25.7
24	149	0.49	27.1	64.8	39.6
24	211	0.25	26.5	58.4	63.8
24	299	0.12	27.3	53.2	108.7
24	422	0.06	30.1	48.5	191.4
48	1	—	6972.5	10967.0	1.0
48	105	0.99	74.5	117.7	28.4
48	149	0.49	66.4	102.3	44.0
48	211	0.25	62.2	90.4	71.8
48	299	0.12	59.8	80.6	123.2
48	422	0.06	60.6	72.6	218.5

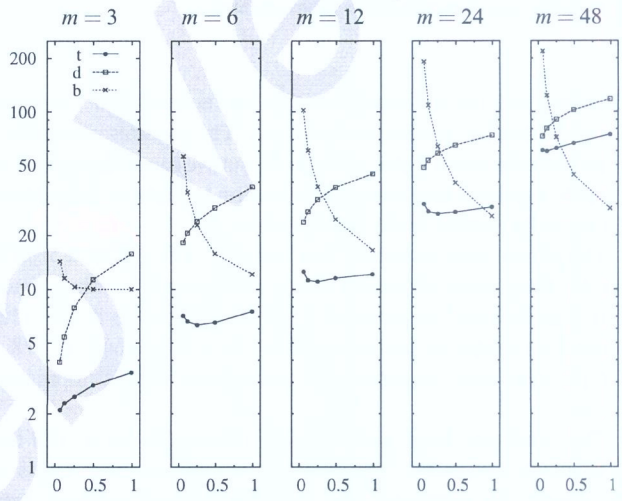


Fig. 20. Average times (t) and operation counts (b, d) of a table look-up for various values of m, as a function of the grid occupancy κ (horizontal axis).

d increases as κ increases and that the choice $\kappa = 0.25$ gives near-optimal times for all m.

C. Comparison With k-D Tree Nearest Neighbor

For each test, we compared our results and times with a brute-force search and with our implementation of the k-D tree nearest neighbor algorithm in [7]. We used the same images above, the same signature table with 10 967 entries, and a 2-D grid with $N \times N = 211 \times 211$ buckets ($\kappa \approx 0.25$). Both methods are equally accurate as they return the same result as a brute-force search. However, our 2-D grid-based method was always much faster than both, although the speed-up depended on the quality of the input images, i.e., from a factor of 5 or more with real

TABLE IV
COMPARISON OF OUR GRID-BASED METHOD WITH THE k-D TREE NEAREST NEIGHBOR ALGORITHM AND THE BRUTE-FORCE SEARCH, ON SYNTHETIC IMAGES, SHOWING THE AVERAGE LOOKUP TIME t (MICROSECONDS PER PIXEL) AND THE AVERAGE NUMBER d OF DISTANCE COMPUTATIONS PER LOOKUP

	t			d		
m	Grid	kd-Tree	Brute	Grid	kd-Tree	Brute
3	2.9	33.8	1159.4	12.3	143.2	11172.0
6	4.0	59.8	1499.7	14.2	179.5	11172.0
12	5.2	77.6	2120.5	10.3	160.9	11172.0
24	8.3	114.8	3745.2	10.0	155.5	11172.0
48	14.2	208.0	7110.6	9.8	169.0	11172.0

TABLE V
COMPARISON OF OUR GRID-BASED METHOD WITH THE k-D TREE ALGORITHM AND THE BRUTE-FORCE SEARCH FOR PHOTOS OF A REAL SCENE

	t			d		
m	Grid	kd-Tree	Brute	Grid	kd-Tree	Brute
3	2.5	28.1	1134.9	7.9	118.3	10967.0
6	6.3	66.3	1458.2	24.0	201.9	10967.0
12	11.0	101.5	2137.1	31.9	216.8	10967.0
24	26.5	217.6	3920.5	58.4	300.9	10967.0
48	62.2	468.3	6972.5	90.4	387.3	10967.0

photos to a factor of 100 or more with synthetic images (see Tables IV and V).

The increased search cost observed with real photos is due mainly to perturbations of the pixel signatures, i.e., both in the scene and in the reference-object images. These perturbations are due to various reasons, including camera and JPEG compression noise, cast shadows, highlights, dust specks, and nonuniform illumination. These perturbations displace usually the signature away from the 2-D manifold. As a consequence, the starting bucket obtained by projection is removed further from the bucket containing the best match, and the search termination criteria become less effective.

D. Comparison With ANN and LSH

Another method whose aims are comparable to ours is the LSH, as used in [34]. According to Zhong and Little, it is an improvement on the ANN method used in [14].

The LSH method projects the data points (i.e., the reference-object signatures from the table) onto a number of unidimensional uniform grids, which are defined by suitably chosen vectors in m-space. The query point is projected onto those same grids, the selected bucket lists from these grids are combined, and the nearest match is sought in the combined list. The optimum number of grids depends on m, and the result is not guaranteed to be the best match.

We were unable to obtain the LSH implementation used by Zhong and Little. We tried using the the LSH method of [16], as implemented by [2], which is available online. However, the method is designed to return a list of all points contained approximately in a given radius r. Selecting this parameter appears to be a nontrivial problem: if r is too small, the algorithm often returns an empty list, and if it is too large, the list is often quite long.

Although running times on different machines are difficult to compare, Zhong and Little report that their program was about 50 times faster than a brute-force search, for a specific example (their "bottle" scene, with 10 000 table entries, $m = 24$ images). For comparison, in our real-photo test (see Section III-B), we obtained a speed-up of about 130 over brute force, when running with similar parameters. Thus, we conclude tentatively that our acceleration method is at least twice as effective as their LSH-based method. In any case, it should be noted that their method returns an approximate best match (according to the article, with normal errors of up to 8°), whereas ours always returns the exact best match.

E. Effect of Image Noise

The presence of noise in the input scene images S_i can affect: (1) the search time; (2) the accuracy of the recovered normals; and (3) the accuracy of the integrated height field. To show these effects, we processed the synthetic images (420 × 300 pixels) and real-scene photographs (278 × 386 pixels) pertubed by varying amounts of artificial noise. Specifically, we used the same 24 images used in the main tests, where each pixel was mixed with 2%, 5%, 10%, and 20% of random noise uniformly distributed in [0, 1].

(1) Search cost: Image noise increases the search cost by displacing the query signature vector away from the 2-D manifold of table entries, which reduces the effectiveness of the search cutoff criteria. Even so, we found that image had a little effect on the cost of our algorithm, whereas it significantly increased the cost of the k-D tree algorithm. The running times are shown in Table VI. In these tests, we used the grid size of 211 × 211 buckets, which gave best performance for the noise-free images.

(2) Indeed, in these tests with noisy images, we found that a large fraction of the search time (both for our algorithm and for the k-D tree nearest neighbor) is due to a very small number of "outlier" pixels that have totally inconsistent signatures and require thousands of distance computations. If those outliers are excluded, the average search cost per pixel is fairly independent of the image resolution (see Fig. 21), since the best matching entry for those outlier pixels is useless, one could improve the performance of the algorithm in practice by terminating the search after a reasonable number of distance evaluations. For this paper, we choose instead to return the best match in every case even for the outliers.

(3) Normal error: With 2%, 5%, 10%, and 20% added noise, the root-mean-square error in the computed normal was 0.014, 0.03, 0.08, and 0.17 rad (0.8°, 2.1°, 5.1°, and 10.2°), respectively.

(4) Observe that the accuracy of the recovered normals is not really a property of our grid-based algorithm but of the matching criterion chosen (minimum Euclidean distance between normalized OVs) and of the density of table T. Thus, any other exact algorithm will have the same sensitivity to noise as ours, and any approximate algorithm can only be less accurate than ours.

(5) Height field error: To conclude, we show in Fig. 22 the result of integrating the normal map obtained from 24 images with 10% added pixel noise. As in Fig. 17, we used the integrator of [23]. (However, recall that the accuracy of the integrated height field is not relevant to this paper.)

TABLE VI
AVERAGE NUMBER OF SIGNATURE DISTANCES PER PIXEL COMPUTED BY THE GRID AND k-D TREE SEARCH ALGORITHMS FOR VARIOUS PERCENTAGES η OF MIXED NOISE

η	d					
	Synthetic			Real		
	Grid	k-d T.	Brute	Grid	k-d T.	Brute
00%	10.0	155.5	11171.0	58.4	300.9	10966.0
02%	22.9	203.4	11171.0	82.1	351.3	10966.0
05%	63.6	279.0	11171.0	190.7	521.1	10966.0
10%	177.0	390.0	11171.0	517.5	847.3	10966.0
20%	558.9	612.3	11171.0	1467.9	1446.2	10966.0

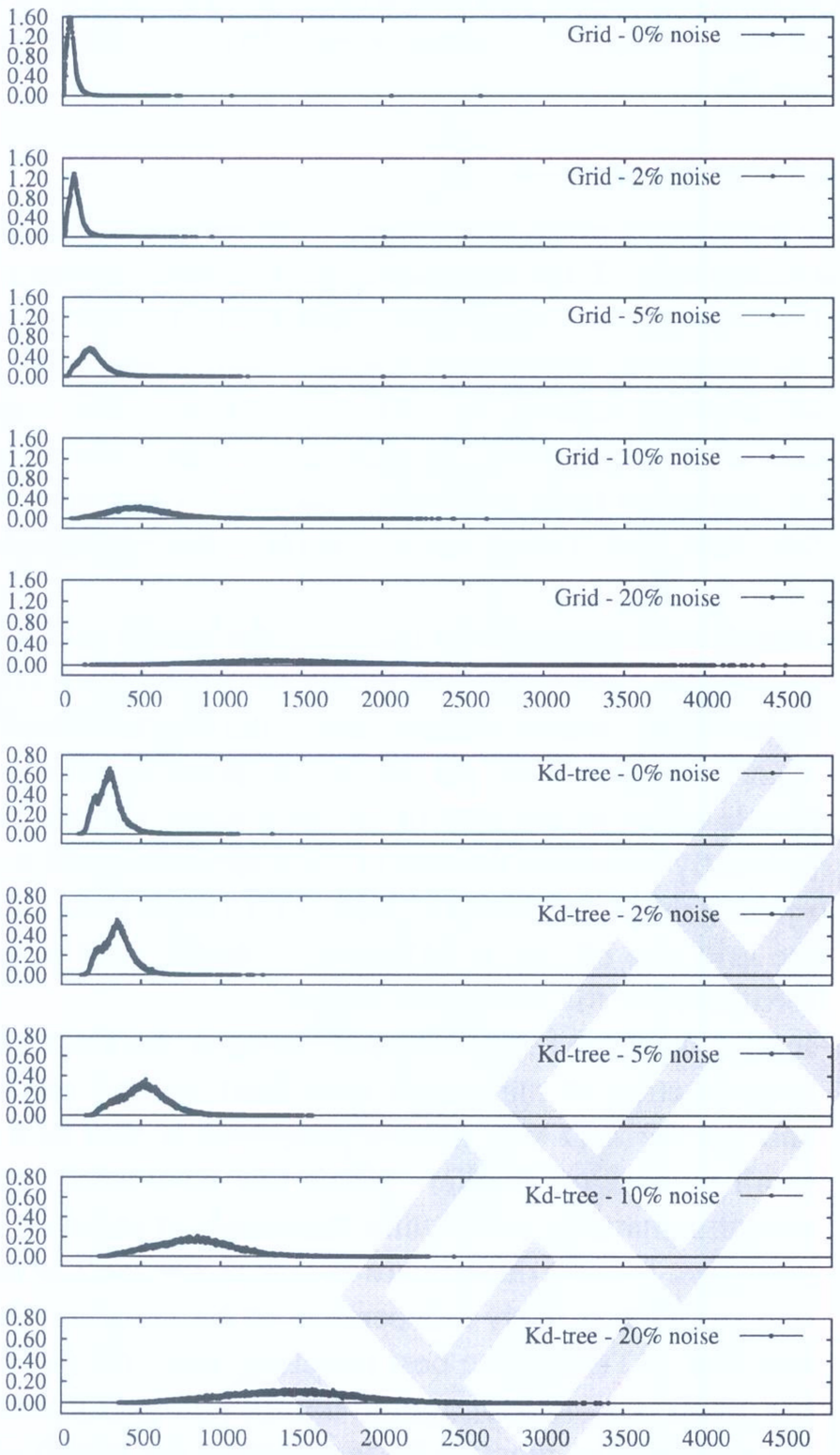


Fig. 21. Histograms of the percentage of image pixels (vertical axis) that require a given number of distance comparisons (horizontal axis) with the grid-based and k-D tree search algorithms for the real-scene images with various error percentages η.

IV. CONCLUSIONS AND FUTURE WORK

We have described a technique for VLPS, which is based on a 2-D uniform grid that provides significant speed-up for the table lookup step over other methods for an arbitrary number of input images.

Unlike uncalibrated VLPS methods such as in [12] and [29], our method works for a broad class of non-Lambertian BRDFs (which need not be explicitly known) and for arbitrary light

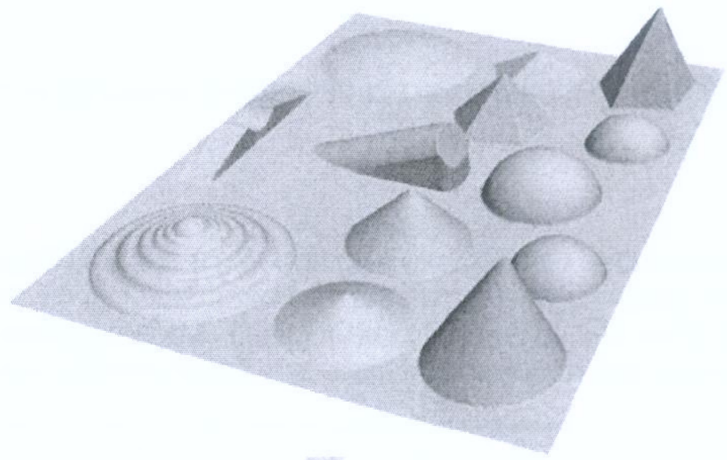


Fig. 22. Height map obtained by integrating the normal map derived from $m = 24$ scene images with $\eta = 10\%$ noise.

fields (not just point-like sources). Our method can can also handle attached shadows directly without the heuristic image and pixel pruning required by those methods. Unlike some other acceleration schemes for example-based VLPS, our method always yields the best matching entry in the table and not just a close approximation thereof.

The 2-D grid that we use is considerably more efficient (by a factor of 5–10) than the k-D tree nearest neighbor algorithm and appears to be at least twice as fast as Zhong and Little's LSH-based method. As shown in Figs. 11 and 18, radius $\rho[i, j]$ of each bucket is much smaller than distance $|\mu[i, j] - c[i, j]|$, meaning that spherical enclosure tests for each bucket often allow us to eliminate an entire bucket with a single distance function evaluation. Moreover, Tables II and III show that we need to scan only a few buckets surrounding the initial cell, and compute only a few signature distances inside those cells.

1) Multiple materials: Our method can be used with scenes containing two or more materials with distinct BRDFs as long as a separate reference object is provided for each material. In that case, we would build a separate lookup table and a grid structure for each reference object. Then, we would find the best match to each scene signature $\boldsymbol{s}$ in each table and select the best among those matches.
2) Spatially variable materials: The 2-D grid technique is not appropriate for situations where the scene BRDF has additional position-dependent parameters (besides the albedo factor) since the manifold $\mathbb{T}$ of observation signatures is no longer bidimensional. This applies, for example, to anisotropic materials with variable orientation, such as wood, velvet, or brushed steel, and to materials whose BRDF is a linear combination of two or more BRDFs, such as a Lambertian body with a variable amount of Phong highlighting.
3) If the BRDF has only one extra parameter, the signature manifold is 3-D. One could consider adapting our method to use a uniform 3-D grid, but it space cost would be much higher. On the other hand, if the BRDF has one extra parameter but the normal directions are restricted to a single plane (as considered in [14]), manifold $\mathbb{T}$ is still bidimensional; therefore, the 2-D grid method should be just as effective.
4) Nonuniform lighting: A significant obstacle to practical application of example-based VLPS schemes is the common occurrence of nonuniform lighting, as produced, e.g., by

light sources (or large lightly colored objects) located near the scene. Lighting variations cause a gradual distortion of the computed normal $\vec{s}[p]$ as p is further removed from the reference object. This distortion will cause a planar surface to become curved in the reconstructed height field. We are experimenting with reference-object interpolation methods that combine three or more reference objects into a single position-dependent "virtual reference object" that approximates the lighting at each scene point p. We are also extending the uniform 2-D grid method described here for position-dependent virtual reference objects.

5) Projected shadows: If some of the images contain projected shadows, highlights, or secondary lighting, the corresponding OV components $S_i[p]$ can be anomalously low or high. These errors will affect all components of the normalized signature $\boldsymbol{s}$, so that the best matching entry in the table will not be the correct one. In a separate paper [19], we describe a distance function for OVs that is less sensitive to those disturbances. Although the bucket-grid scheme described here does not work for that distance function, we can use multiple bucket grids, in a RANSAC-like approach, to achieve speedups comparable to those reported here.

6) Alternative grid projections: The large remaining source of inefficiency in our method appears to be the function that we use to map the signatures to the grid cells (steps 1 and 2 of algorithm 1). Since we project perpendicularly to the plane, rather than to the manifold $\mathbb{T}$, the nearest table entry is more likely to be in a different bucket than the one we start searching, particularly if the query signature is somewhat displaced from the $\mathbb{T}$ manifold, due to camera noise or other perturbing factors.

7) One could easily remove that problem by using a fancier projection function and/or some other surface instead of the plane P. However, any change in these features of our algorithm would invalidate the logic behind the precomputed bucket search order Δ and the stopping criterion (16), which appears to be essential to the performance of our algorithm.

REFERENCES

[1] V. Akman, W. R. Franklin, M. Kankanhalli, and C. Narayanaswmi, "Geometric computing and uniform grid technique," *Comput. Aided Des.*, vol. 21, no. 7, pp. 410–420, Sep. 1989.

[2] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for near neighbor problem in high dimensions," in *Proc. 47th IEEE Symp. FOCS*, 2006, pp. 459–468.

[3] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, "An optimal algorithm for approximate neareast neighbor searching in fixed dimensions," *J. Assoc. Comput. Mach.*, vol. 45, no. 6, pp. 891–923, Nov. 1998.

[4] G. A. Atkinson, A. R. Farooq, M. L. Smith, and L. N. Smith, "Facial reconstruction and alignment using photometric stereo and surface fitting," in *Proc. Iberian Conf. Patt. Recog. Image Anal.*, 2009, pp. 88–95.

[5] S. Barsky and M. Petrou, "The 4-source photometric stereo technique for three-dimensional surfaces in presence of highlights and shadows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1239–1252, Oct. 2003.

[6] R. Basri, D. Jacobs, and I. Kemelmacher, "Photometric stereo with general, unknown lighting," *Int. J. Comput. Vis.*, vol. 72, no. 3, pp. 239–257, May 2007.

[7] J. L. Bentley, J. H. Friedman, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Trans. Math. Softw.*, vol. 3, no. 3, pp. 209–226, Sep. 1977.

[8] C. Cason, POV-Team, Persistence of Vision Raytracer 2008 [Online]. Available: http://www.povray.org/

[9] J. Cristy, ImageMagick: Convert, Edit and Compose Images 2008 [Online]. Available: http://www.imagemagick.org

[10] Y. Ding, L. N. Smith, M. L. Smith, R. Warr, and J. Sun, "Obtaining 3D malignant melanoma indicators through statistical analysis of 3D skin surface disruptions," *Skin Res. Technol.*, vol. 15, no. 3, pp. 262–270, Aug. 2009.

[11] A. R. Farooq, M. L. Smith, L. N. Smith, and P. S. Midha, "Dynamic photometric stereo for on line quality control of ceramic tiles," *J. Comput. Ind.—Spec. Edition Mach. Vis.*, vol. 56, no. 8, pp. 918–934, Dec. 2005.

[12] H. Hayakawa, "Photometric stereo under a light source with arbitrary motion," *J. Opt. Soc. Amer. A*, vol. 11, no. 11, pp. 3079–3089, Nov. 1994.

[13] B. Henderson, Netpbm Website 2008 [Online]. Available: http://netpbm.source forge.net/

[14] A. Hertzmann and S. M. Seitz, "Example-based photometric stereo: Shape reconstruction with general, varying BRDFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1254–1264, Aug. 2005.

[15] B. K. P. Horn, R. J. Woodham, and W. M. Silver, "Determining shape and reflectance using multiple images," in *A. I. Memo*. Cambridge, MA: MIT, 1978, pp. 1–9.

[16] P. Indyk and R. Motwani, "Approximate nearest neighbour: Towards removing the curse of dimensionality," in *Proc. 13th Annu. ACM STOC*, 1998, pp. 604–613.

[17] G. Kay and T. Caelli, "Estimating the parameters of an illumination model using photometric stereo," *Graph. Models Image Process.*, vol. 57, no. 5, pp. 365–388, Sep. 1995.

[18] H. C. G Leitão, R. F. V Saracchini, and J. Stolfi, "A bucket grid structure to speed up table lookup in example-based photometric stereo," in *Proc. 20th SIBGRAPI*, 2007, pp. 221–227.

[19] H. C. G. Leitão, R. F. V. Saracchini, and J. Stolfi, "Matching photometric observation vectors with shadows and variable albedo," in *Proc. 21st SIBGRAPI*, 2008, pp. 179–186.

[20] T. Malzbender, B. Wilburn, D. Gelb, and W. Ambrisco, "Surface Enhancement using real-time photometric stereo and reflectance transformation," in *Proc. Eur. Symp. Rendering*, 2006, pp. 245–250.

[21] S. Nayar, K. Ikeuchi, and T. Kanade, "Determining shape and reflectance of a hybrid surface by photometric sampling," *IEEE Trans. Robot. Autom.*, vol. 6, no. 4, pp. 418–431, Aug. 1990.

[22] G. Roelofs, Portable Network Graphics: An open, extensible image format with lossless compression 2008 [Online]. Available: http://libpng.org/pub/png/

[23] R. F. V. Saracchini, J. Stolfi, H. C. G. Leitao, G. A. Atkinson, and M. L. Smith, "Multi-scale depth from slope with weights," in *Proc. 21st BMVC*, 2010, pp. 40.1–40.12.

[24] L. Shen, T. Machida, and H. Takemura, "Efficient photometric stereo for three-dimensional surfaces with unknown BRDF," in *Proc. 5th Int. Conf. 3DIM*, 2005, pp. 326–333.

[25] W. M. Silver, "Determining shape and reflectance using multiple images," M.S. thesis, EECS Dept., MIT, Cambridge, MA, 1980.

[26] J. Sun, M. L. Smith, L. N. Smith, S. P. Midha, and J. Bamber, "Object surface recovery using a multi-light photometric stereo technique for non-Lambertian surfaces subject to shadows and specularities," *Image Vis. Comput.*, vol. 25, no. 7, pp. 1050–1057, Jul. 2007.

[27] J. Sun, M. L. Smith, A. R. Farooq, and L. N. Smith, "Concealed object perception and recognition using a photometric stereo strategy," in *Proc. 11th Int. Conf. Adv. Concepts Intell. Vis. Syst.*, 2008, pp. 445–455.

[28] H. D. Tagare and R. J. P. Figueiredo, "A theory of photometric stereo for a class of diffuse non-Lambertian surfaces," *IEEE Trans. Pattern Analy. Mach. Intell.*, vol. 13, no. 2, pp. 133–152, Feb. 1991.

[29] A. Yuille and D. Snow, "Shape and albedo from multiple images using integrability," in *Proc. IEEE Conf. CVPR*, 1997, pp. 158–164.

[30] A. R. Willis and D. B. Cooper, "Computational reconstruction of ancient artifacts: From ruins to relics," *IEEE Signal Process. Mag.*, vol. 25, no. 4, pp. 65–83, Jul. 2008.

[31] R. J. Woodham, "Photometric method for determining surface orientation from multiple images," *Opt. Eng.*, vol. 19, no. 1, pp. 139–144, Jan/Feb. 1980.

[32] R. J. Woodham, "Determining surface curvature with photometric stereo," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1989, vol. 1, pp. 36–42.

[33] R. J. Woodham, "Gradient and curvature from the photometric stereo method, including local confidence estimation," *J. Opt. Soc. Amer. A*, vol. 11, no. 11, pp. 3050–3068, Nov. 1994.

[34] L. Zhong and J. J. Little, "Photometric stereo via locality sensitive high-dimension hashing," in *Proc. 2nd Can. Conf. CRV*, 2005, pp. 104–111.

Rafael F. V. Saracchini received the B.Sc. degree from Fluminense Federal University, Niterói, Brazil, in 2007. He is currently working toward the Ph.D. degree from the State University of Campinas, Campinas, Brazil.

His research interest include 3-D data recovery through digital images using geometric and photometric Stereo from non-Lambertian and glossy/shiny surfaces and its applications in medicine, industry, and security.

Jorge Stolfi received the B.E. degree in electrical engineering and the M.Sc. degrees in computer science from University of Sao Paulo, Brazil, in 1973 and 1979, respectively, and the Ph.D. degree in computer science from Stanford University, Stanford, CA, in 1989.

He is currently a Full Professor with the Institute of Computing, University of Campinas, Campinas, Brazil. His research interests include natural language processing, computational geometry, computer graphics, numerical analysis, and image processing and applications.

Helena Cristina da Gama Leitão received the B.Sc. and M.Sc. degrees in computer science from the Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, in 1988 and 1993, respectively, and the Ph.D. degree from the State University of Campinas, Campinas, Brazil, in 1999.

She is currently an associate Professor with the Institute of Computing, Fluminense Federal University, Rio de Janeiro. Her research interests include pattern recognition for archaeological, medical, and molecular biology applications.