Geometry Construction from Caustic Images

Manuel Finckh, Holger Dammertz, and Hendrik P.A. Lensch

Institute of Media Informatics, Ulm University
89081 Ulm, Germany

Abstract. In this work we investigate an inverse geometry problem. Given a light source, a diffuse plane and a caustic image, how must a geometric object look like (transmissive or reflective) in oder to project the desired caustic onto the diffuse plane when lit by the light source? In order to construct the geometry we apply an analysis-by-synthesis approach, exploiting the GPU to accelerate caustic rendering based on the current geometry estimate. The optimization is driven by simultaneous perturbation stochastic approximation (SPSA). We confirm that this algorithm converges to the global minimum with high probability even in this ill-posed setting. We demonstrate results for precise geometry reconstruction given a caustic image and for reflector design producing an intended light distribution.

1 Introduction

The automatic construction of geometric objects from a predetermined property is an important engineering task. We address the problem of constructing a transmissive or reflective surface, that, given a predefined light position, creates an a priori defined caustic image. This task of creating a specific caustic occurs in the design of headlights, of parabolic concentrators for solar cells or interior design, see for example Figure 1. The same approach can also be used to reconstruct the surface geometry of a real object given only an image of its caustic. Even though this is an ill-posed problem we show how in many cases reasonable reconstructions can be achieved.

The geometry estimation follows an analysis-by-synthesis approach. The procedure starts with an initial surface whose geometry is subsequently optimized to minimize the *mean squared error* (MSE) between the target caustic and the current caustic image. The MSE is the only measure applied to determine the quality of the constructed geometry. A standard optimization algorithm for such a problem would be the *simulated annealing* (SAN) algorithm [1, 2]. However, our results show that the *simultaneous perturbation stochastic approximation* (SPSA) optimization algorithm [3] is more robust and converges much faster in this setting.

One contribution of our work is the use of SPSA as a *global* optimizer for this kind of problem. For the evaluation of the objective function we present a specialized and optimized implementation that exploits GPUs for a fast evaluation. It efficiently renders single bounce reflections or refractions. The proposed

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part V, LNCS 6315, pp. 464–477, 2010.

Fig. 1. The left image shows four of our optimization targets for which we generated geometry with our proposed optimization approach. The second and third image show a physically correct light transport simulation illustrating the results. For the living room scene we optimized a glass table to cast the predefined caustic. The letters E C C V in the right image are caustics from parabolic reflectors with a small embedded light source. The last image shows one half of such an optimized parabolic reflector.

optimization framework is flexible to work on connected or disconnected triangle meshes and further can operate on C^2 continuous B-spline surfaces. One unique feature of our method compared to other methods like [4, 5] is the degree of freedom we can deal with due to utilizing SPSA for optimization. In the case of the parabolic reflectors in Figure 1 the B-spline patches have about 4000 control points which have to be optimized. We explore the use of our framework for reflector design, for geometry reconstruction of water surfaces and for light concentrating glass objects.

2 Related Work

A recent survey for various ways of reconstructing specular and transparent geometry has been assembled by Ihrke et al. [6]. Direct geometry measurement techniques are based on structured illumination [7, 8] and multiple input images. They apply shape from distortion [9–11], shape from specular highlights [12, 13], optical tomography [14], or inverse ray-tracing [15, 16]. Our reconstruction method falls into the last category but uses a single intensity distribution image as input and therefore requires optimization.

Morris et al. [17] demonstrated the reconstruction of a water surface by utilizing two cameras and a known pattern placed under the water surface. With some restrictions to the setup, for instance secondary refractions or reflections have to be suppressed, they were able to reconstruct the surface correctly. Reconstructing a water surface is not the scope of this work, nevertheless our methods could be applied to this task.

Kutulakos et al. [18] investigate the theoretical background of reconstructing arbitrarily-shaped specular scenes. They reduced the problem of 3D shape reconstruction to reconstruction of light paths that cross the image plane. They showed, that it is impossible to reconstruct a light path when the light is reflected or refracted more than twice. In all other cases, three viewpoints are enough for

successful reconstruction. This insight limits the generality of our approach, as a single caustic image allows us to consider a single surface interaction only. In addition Ramamoorthi et al. [19] provides a theoretical framework for inverse rendering problems.

Patow [20] and Patow et al. [21, 22] investigated the specific problem of reflector design to obtain surfaces that produce an intended light distribution for a given light source position. They applied an analysis-by-synthesis approach using a brute force search and SAN in order to optimize the reflector for the distribution. Early work on the reflector design problem was done by Neubauer [23], Caffarelli et al. [24], and Wang [25, 26].

Recent approaches directly operating on NURBS-surfaces and utilizing an analysis-by-synthesis approach are presented by Anson et al. [4] and Mas et al. [5]. However, the search space was restricted to only two dimensions and rotationally symmetric reflectors, and four dimensions respectively.

Recently, Weyrich et al. [27] presented a method of fabricating micro geometry with custom reflectance probability. They first computed a set of micro facets which produces the selected reflectance distribution. However, the resulting micro facets are not connected, so they utilized a SAN optimization process to arrange them in a (nearly) tileable way. Our approach generates comparable results on disconnected meshes but is flexible enough to optimize watertight surfaces as well. Beyond that, our SPSA-based system works efficiently for both reflective and refractive surfaces.

3 Optimization Framework

Caustics occur when a specular or refractive object focuses light onto a diffuse surface. Caustics are often caused by water surfaces, glass objects, such as lenses, or concave mirrors as one finds them in headlights. In photo-realistic image synthesis a common task is to simulate these caustics [28]. In this work, we ask for the inverse problem: Given the caustic image, what is the geometry of the caustic generating surface? During optimization we apply a simplified rendering system (Section 5) which ignores multiple scattering inside the reflector or refractor.

We can not use methods as in [17, 18], because the directional information of the incident light onto the diffuse surface is not available in a single caustic image. Hence, we are not able to directly reconstruct the light paths which would allow us to reconstruct a single reflective or refractive surface directly. Furthermore, as the incident direction to the diffusor is unknown the problem is not well-defined, even the simple case where all the light is focused on a single point is not uniquely solvable. The solution could be a small lens directly in front of the light source, or a huge lens further away from the light source.

Therefore, we use an analysis-by-synthesis approach to find an appropriate solution. The reflecting or transmitting surface is represented as a triangle mesh or in most examples as C^2 continuous B-spline patch which is either initialized as a planar surface, or a parabolic surface (mainly used for reflector optimization).

The number of control points is arbitrary, for the results presented here we use 12^2 up to 64^2 control points. The control points are fixed in their xy-position, only the z-coordinate is modified during optimization. Allowing for varying xy-coordinates would not only increase the dimensionality of the problem, it would also complicate the optimization process, i.e., preventing self occlusion would not be trivial in such a setting. The B-spline patch is described by one state vector θ, with dimensionality p equal to the number of control points. Hence, we search for an optimal solution vector θ^* in the problem space $\Theta \subseteq \mathbb{R}^p$. The optimal solution is defined by the global minimum of the objective function $L(\cdot)$, which in our case is the MSE between the current caustic image and the target caustic image.

The missing pieces to fully describe our framework are the employed optimization algorithm and the evaluation of the objective function which includes the costly computation of the caustic. These points are discussed in the next two sections.

4 Optimization Using SPSA

The optimization is carried out by performing a random walk on the problem space Θ. In each iteration a new candidate solution θ_{k+1} is computed by adding a specific step vector to the current state vector θ_k. In contrast to simulated annealing which takes a completely random approach, the SPSA algorithm computes an approximate gradient to determine the best search direction in each iteration.

The SPSA algorithm belongs to the family of *stochastic approximation* (SA) algorithms [29]. The basic form of the SA algorithm when there is no analytic gradient available is the Kiefer-Wolfowitz finite-difference SA (FDSA) algorithm [30, 31]. The disadvantage of this algorithm is that it needs $2p$ objective function evaluations in order to approximate a gradient. Introduced by Spall [3], the SPSA algorithm overcomes this disadvantage. It consumes only two objective function evaluations in each iteration in order to approximate a gradient regardless of the dimensionality of the problem.

Gradient Approximation. The idea is to randomly perturb all elements of θ_k to obtain two (probably noisy) measurements of the objective function. More formally, let $y(\cdot)$ denote a noisy measurement of $L(\cdot)$, i.e., $y(\cdot) = L(\cdot) + noise$. Each component i of the k-th approximate gradient $g_k(\theta_k)$ is now determined by,

$$g_{ki}(\theta_k) = \frac{y(\theta_k + c_k\Delta_k) - y(\theta_k - c_k\Delta_k)}{2c_k\Delta_{ki}} \,. \tag{1}$$

A simple choice for the Δ_k random vector is to use a Bernoulli$\pm$1 distribution for each component Δ_{ki} of the vector (with probability 1/2 for each $\pm$1), in general each Δ_{ki} has to be independent and symmetrically distributed about 0 with finite inverse moments $E(|\Delta_{ki}|^{-1}) < \infty$ for all k, i [32]. The so-called gain

Table 1. Choice of the gain sequences

a_k	$a/(A+k+1)^\alpha$
α	0.602 (practically effective), 1.0 (asymptotically optimal)
A	$< 10\%$ of maximum (expected) iterations
a	$a/(A+1)^\alpha \cdot g_0(\theta) \approx$ smallest desired change among elements in θ
c_k	$c/(k+1)^\gamma$
γ	0.101 (practically effective), 1/6 (asymptotically optimal)
c	small positive number $\approx$ standard deviation of the measurement
Δ_k	Bernoulli$\pm$1 distribution

sequence c_k controls the distance between the sample points. It is monotonically decreasing in each iteration to ensure high quality gradients when approaching the optimum.

The optimization is carried out by moving along this approximated gradient g_k, formally,

$$\theta_{k+1} = \theta_k - a_k g_k(\theta_k) \ , \tag{2}$$

where the a_k is another gain sequence generating monotonically decreasing step lengths.

The Gain Sequences. Unfortunately, there is no generally optimal choice for the gain sequences a_k and c_k and the random vector Δ_k in practice, only theoretically optimal choices are at hand. In [33], some suggestions are given how to tune these parameters in order to improve convergence, for a summary see Table 1. The specific choice of SPSA parameters for our purpose is discussed in Section 6.

Convergence. In general, the order of the error is $\epsilon = \mathcal{O}(k^{-1/3})$ [3, 34, 35]. This is only the local convergence rate of the algorithm. But as pointed out in [36] SPSA may work as global optimizer without adding an extra random vector to the SA-recursion (Eq. 2) [37].

5 Caustic Rendering on the GPU

As the objective function has to be evaluated several thousand or even million times, a fast implementation of the caustic rendering step is crucial. The main part of the evaluation of the objective function is the synthesis of the caustic (about 99% of the computation time is spent there).

The caustic is synthesized by Monte Carlo light transport simulation [28] ignoring multiple interactions. We can optimize this simulation specifically for our task. The simple scene geometry allows for omitting self occlusion, and further allows direct handling of the ground plane as accumulation buffer. For efficient simulation we split the B-spline into multiple cubic Beziér patches [38] (see Figure 2b). The splitting computation is done on the CPU, the resulting Beziér

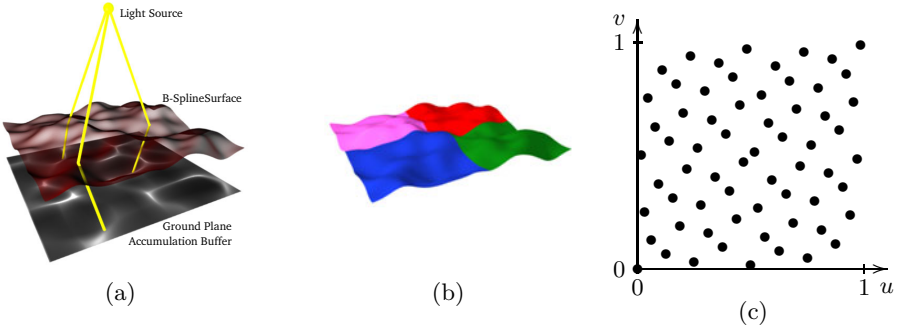


Fig. 2. Illustration of the rendering process. In (a) the basic process is illustrated. Starting from a light source, light rays are connected to the B-spline surface, refracted (or reflected) and intersected with the ground plane accumulation buffer. (b) shows how, for sampling purposes, the B-spline surface is split into multiple Beziér patches and (c) shows 64 Hammersley points used to generate sample points on each Beziér patch.

patches are than transfered to to GPU memory and the remaining computation of the caustic is done on the GPU. The basic steps performed on the GPU are the following (illustrated in Figure 2a):

We start a fixed number of threads (i.e. 64) on the GPU. The number of samples we take for each Beziér patch is also fixed and a multiple of the number of threads (so each thread has to take $\#work = \#samples/\#threads$ samples). For the generation of the sample points we use the Hammersley quasi-Monte Carlo point set [39] (see Figure 2c). They are well distributed and very simple to compute,

$$(x, y) = (i/n, (reverse\ bits)(i)/0x100000000LL)\ , \tag{3}$$

where $n = \#samples$, and $i = sample\_index$.
Each thread computes now:

```
I.  initialize ground plane accumulation buffer with zeros
II. for each Bezier patch:
    1. sample_index = k + thread_num * work,
       where k in range (0,work)
    2. compute point (x,y)
    3. transform into 3D point on Bezier patch
    4. connect point to light source and compute refraction ray
    5. intersect with ground plane
    6. accumulate contribution with correct weighting
```

Essential for the correctness of the simulation is the weighting of the samples. As we directly sample points on the surface patches we have to weight them according to the projected differential surface area and the squared distance r.

The differential surface area is given by the length of the cross product of both directional derivatives (length of the surface normal n), and the projection results in an additional cosine factor (between normalized surface normal $\hat{n}$ and direction $\hat{d}$ to the light source),

$$w = \frac{1}{r^2}||n|| \cdot | < \hat{n} \cdot \hat{d} > | \ . \tag{4}$$

Additional Optimization. The accumulation of the contribution in step 6 results in random memory read-write access. Further the different threads may write to the same part of the accumulation buffer, which either leads to the need of atomic writes or a falsified computation of the caustic. So we simply save the contribution along with the pixel index in a consecutive array. The array is transfered to the main memory and the final caustic image is then assembled by the CPU. By asynchronously calling the CUDA kernel this computation along with the additional memory transfer can be done mostly synchronous to the GPU computation.

Efficiency of the method. For example the water surface (see Section 6) the B-Spline is split into 81 Beziér patches, each patch is sampled with 2048 samples which results in 165888 samples per objective function evaluation. 1000 objective functions are evaluated in less than 4 seconds, which results in 30 minutes overall run time for convergence (System: Intel Core 2 Duo E6850 with 3 GHz and NVIDIA GeForce GTX 285).

6 Results

We apply our framework to a set of different scenarios.

Designing Reflective and Refractive Concentrators. In Figure 1, we design refractive and reflective surfaces to generate sharp, high contrast patterns at a specific focus plane. The optimization clearly renders the letters for the glass table and the parabolic reflector, but can only approximate the sharp transition. Due to the C^2 continuity, the optimized caustic cannot perfectly match the original which leads to some background noise in those areas which are intended to be black. The height variation necessary to produce the output is surprisingly small compared to the object size.

Headlight Design. A real-world application is shown in Figure 3. Here, a parabolic head light is optimized to cast a non-blinding, almost homogeneous spot of predefined shape onto the street. Note the even intensity distribution in the illuminated region which avoids the hot spot close to the light source typically generated by headlights. Additional results can be seen in Figure 4.

Optimizing Reflectance Distributions. Inspired by Weyrich et al. [27], we further investigate generating a specific radiance distribution of an almost planar, reflective surface for directional illumination. We demonstrate the flexibility of

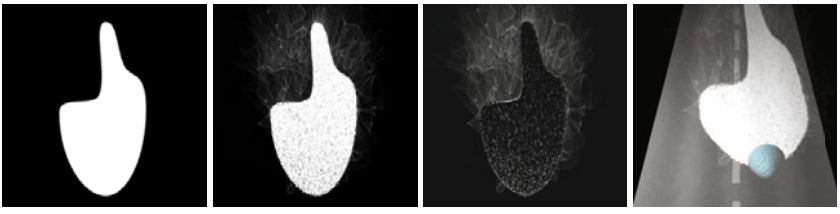
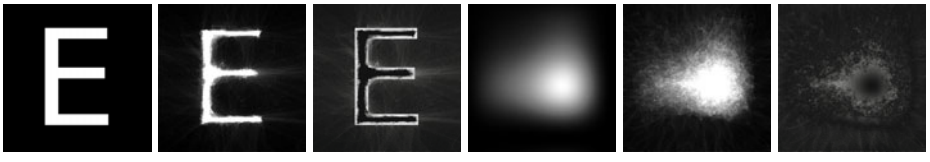


Fig. 3. Headlamp Design – from left to right: Predefined shape of the light cone when projected onto the street. Our resulting distribution after optimization. The difference to the original distribution. Illustration of the result.



(a) (b) (c) (d) (e) (f)

Fig. 4. Results for parabolic reflector design, (a,d) show the target distribution, (b,d) the distribution of the resulting reflector, and (c,f) the difference to the target distribution.

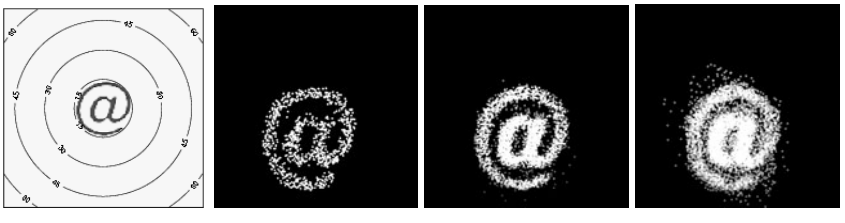


Fig. 5. Results for the micro facet optimization. The first image shows the target distribution, the second image the result by Weyrich et al. [27] with single disconnected quads. The third image shows the result of applying our optimization approach to disconnected quads and the last image on a connected triangle mesh. Note that Weyrich et al. directly samples normals from the target distribution and generates corresponding quads, hence there is no sample outside the target distribution. With our method it is possible to directly optimize a closed triangle mesh. We improved our results by adding a minimal Gauss-blur to the target distribution as it introduces a well-defined gradient to the error function.

our framework and compare the performance for connected and unconnected triangle meshes. As shown in Figure 5, optimizing unconnected quads yields results slightly inferior to Weyrich et al., because in this case the normals can be directly sampled from the target distribution and need not to be optimized. However, with our approach it is possible to optimize a closed triangle mesh which is not possible with their method.

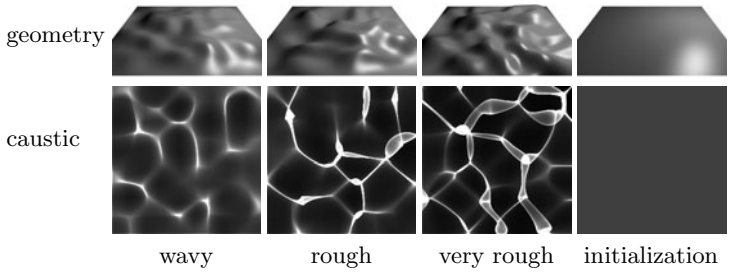


Fig. 6. Top: Visualization of the ground truth geometry for the global convergence test. Bottom: Resulting caustics from the above geometry that where fed into the optimization process. For these tests the optimization is alway initialized with a flat surface, which results in an average grey caustic.

Geometry Reconstruction for Water Surfaces. In Figure 6, we show the caustics generated by water surfaces of varying roughness. Our system generates geometry that reproduces the caustics up to a small error. The Figures 8, 9, and 10 demonstrate the relationship between the MSE of the caustic images and the error of the geometry. The geometry error was computed by sampling the surface at 10000 fixed locations and summing up the squared differences to the original surface. The size of the surface is 10 by 10cm. The results show that the optimized geometry most often corresponds well to the original surface.

7 Discussion

To generate the results we used SPSA with the standard gain sequences a_k and c_k as given in Table 1 with $\alpha = 0.602$ and $\gamma = 0.101$. The asymptotically optimal values of $\alpha = 1.0$ and $\gamma = 1/6$ would lead to faster local convergence, however, at the cost of greatly reducing the probability of convergence to a global optimum. The parameters a and c are adapted to each specific problem and are experimentally chosen observing the beginning of the optimization. The choice of c influences the gradient approximation and depends on the noise of the objective function (Eq. 1). One has to trade off a noisy vs. artificially smoothed gradient estimation. The value of a controls the step size and thus the convergence behavior. Often, if the initial curve of the MSE is too smooth, as it should be at the end of an optimization, the algorithm directly converges to a nearby local minimum. a needs to be increased to allow for a more random exploration of the error landscape. Choosing a too large though will slow down convergence.

Global Convergence. The SPSA algorithm only guarantees a probabilistic convergence to a global optimum [36]. For decent choices of parameters a and c, most often, the global optimum in the water surface test cases was found with the first sequence (see Figure 6, 8, 9, and 10). Whenever the final MSE was still too large, as in the failure case in Figure 10, we restarted the optimization with a different random seed, which was never required more than twice in all

presented cases and could be automatized. Our experimental results verify that SPSA is suitable for global optimization in our setting.

SPSA vs. Simulated Annealing. The good convergence probability of SPSA is in contrast to SAN which rarely converged to a decent optimum in bearable time. The drawback of SAN is that it does not explore the error landscape in a controlled way. It randomly samples the neighboring space and can not walk into a specific direction as it does not exploit any gradient information, not even an approximation. The best results with SAN were achieved for the smooth test case of Figure 6 where the initial solutions is already near the global optimum. With SAN, we obtained decent results on the unconnected triangle mesh in Figure 5, but it completely failed at optimizing the parabolic reflectors, probably due to the complexity of the problem.

Manufacturing. We did not manufacture the reflectors and refractors we optimized, but we verified our results by the means of physical light transport simulation (see Figure 1). We also simulated the precision of a typical milling machine and randomly perturbed the estimated geometry (see Figure 7).

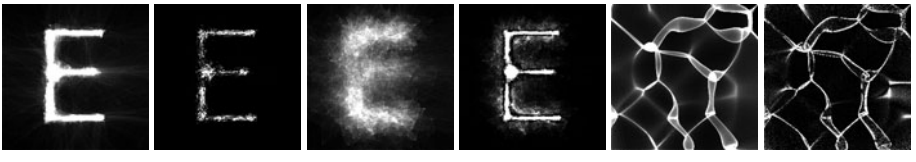


Fig. 7. Results for the precision simulation, from left to right: E-reflector with randomly perturbed control points ($+-0.02$mm) and the difference to the result with no induced error. Again the E-reflector now with a random perturbation of $+-0.2$mm, the "E" nearly vanishes in the noise. The last two images show the SPSA optimization of the very rough test case after 250000 iterations, the geometry error corresponds to an average error of about 0.1mm.

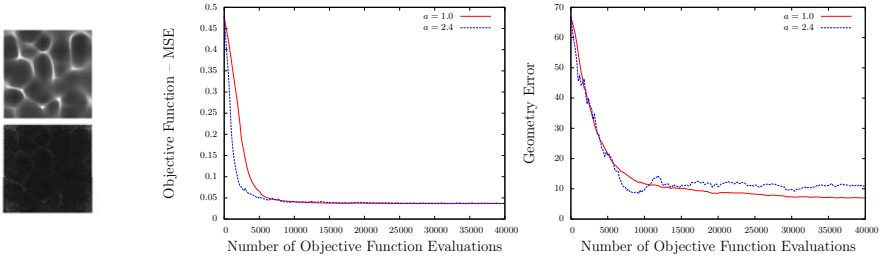


Fig. 8. On this input data the a parameter can be chosen arbitrarily out of a large interval. The reason for this is, that the next nearby local optimum is identical to the global optimum. However, with even smaller values for a the convergence rate of the algorithm would be greatly reduced and with larger ones the algorithm might jump over the global optimum. Note that the remaining difference in the geometric error is extremely small and corresponds to an average error of less than 0.01mm.

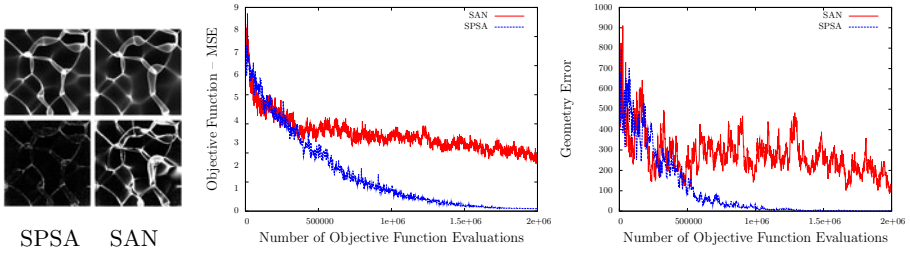


Fig. 9. SPSA and SAN applied to the very rough test case given in Figure 6. SPSA converges to the correct result, SAN converges much slower. The left images show the result after the optimization run (top) and the difference to the target caustic (bottom). We can also see by comparing the objective function graph with the associated geometry error graph that a smaller MSE of the objective function does not always results in a smaller geometry error. This indicates local optima. Also note the smooth MSE graph at the end of the optimization process, indicating local (and in this case global) convergence.

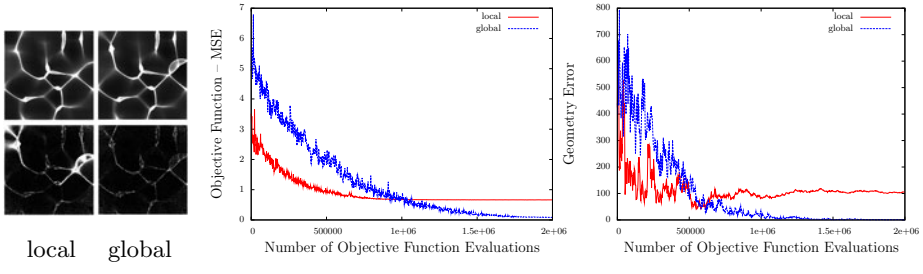


Fig. 10. Illustration of a failure case, where SPSA did not find the global optimum at the first but the second run. In the left columns the top images show the optimization result and bottom images the difference to the original caustic (compare to Figure 6). Note the local minimum of the geometry error of the failure case at about 500000 and the monotonic decrease of the corresponding objective function. The algorithm is clearly converging to a local minimum.

8 Conclusion

The proposed system successfully produces surface geometry that matches a specific reflection or refraction distribution pattern. It can be used to reconstruct geometry from a given caustic image or to shape special purpose reflectors. Key aspects of our efficient optimization are the use of the SPSA algorithm which ensures global convergence with high probability as well as the GPU accelerated caustic rendering approach. Our framework can cope with various surface representations, allowing for the flexibility of unconnected triangles or for the slightly more restrictive but easy to manufacture continuous B-spline surfaces.

Currently, our framework is restricted to single reflective or refractive surfaces. Adding solid objects with multiple scattering events results in a much more complex evaluation of the objective function. It will be a challenge to deal with the additional ambiguities which result in much more distinct local optima of the objective function [18]. However, the optimization framework based on SPSA is independent of the rendering technique and can be exchanged with other methods.

Acknowledgments

This work has been partially funded by the DFG Emmy Noether fellowship (Le 1341/1-1) and by an NVIDIA Professor Partnership Award.

References

1. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computer machines. Journal of Chemical Physics 21, 1087–1091 (1953)
2. Kirkpatrick Jr., S., Gellatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science 220(4598), 671–680 (1983)
3. Spall, J.C.: A Stochastic Approximation Algorithm for Large-Dimensional System in the Kiefer-Wolfowitz Setting. In: IEEE Conf. Decision Contr., pp. 1544–1548 (1988)
4. Anson, O., Seron, F.J., Gutierrez, D.: NURBS-based inverse reflector design. In: Proceedings of CEIG 2008, pp. 65–74 (2008)
5. Mas, A., Martín, I., Patow, G.: Fast inverse reflector design (FIRD). Comput. Graph. Forum 28, 2046–2056 (2009)
6. Ihrke, I., Kutulakos, K.N., Lensch, H.P.A., Magnor, M., Heidrich, W.: State of the art in transparent and specular object reconstruction. In: STAR Proceedings of Eurographics, pp. 87–108 (2008)
7. Morris, N.J.W., Kutulakos, K.N.: Reconstructing the surface of inhomogeneous transparent scenes by scatter-trace photography. In: Proceedings of IEEE International Conference on Computer Vision (ICCV) (2007)
8. Hullin, M.B., Fuchs, M., Ihrke, I., Seidel, H.P., Lensch, H.P.A.: Fluorescent Immersion Range Scanning. ACM Trans. on Graphics (SIGGRAPH 2008) 27, 87:1–87:10 (2008)
9. Oren, M., Nayar, S.K.: A theory of specular surface geometry. International Journal of Computer Vision (IJCV) 24, 105–124 (1996)
10. Tarini, M., Lensch, H.P.A., Goesele, M., Seidel, H.P.: 3d acquisition of mirroring objects. Graphical Models 67, 233–259 (2005)
11. Efros, A.A., Isler, V., Shi, J., Visontai, M.: Seeing through water. In: Advances in Neural Information Processing Systems 17, pp. 393–400. MIT Press, Cambridge (2004)
12. Zisserman, A., Giblin, P., Blake, A.: The information available to a moving observer from specularities. Image and Vision Computing 7, 38–42 (1989)
13. Wang, J., Dana, K.J.: Relief texture from specularities. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 28, 446–457 (2006)

14. Trifonov, B., Bradley, D., Heidrich, W.: Tomographic reconstruction of transparent objects. In: Proceedings of Eurographics Symposium on Rendering 2006, pp. 51–60 (2006)
15. Ihrke, I., Goldluecke, B., Magnor, M.: Reconstructing the geometry of flowing water. In: Proceedings of IEEE International Conference on Computer Vision (ICCV), pp. 1055–1060 (2005)
16. Miyazaki, D., Ikeuchi, K.: Inverse polarization raytracing: Estimating surface shape of transparent objects. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 910–917 (2005)
17. Morris, N.J.W., Kutulakos, K.N.: Dynamic refraction stereo. In: Tenth IEEE International Conference on Computer Vision (ICCV), pp. 1573–1580 (2005)
18. Kutulakos, K.N., Steger, E.: A theory of refractive and specular 3d shape by light-path triangulation. In: Tenth IEEE International Conference on Computer Vision (ICCV), pp. 1448–1455 (2005)
19. Ramamoorthi, R., Hanrahan, P.: A signal-processing framework for inverse rendering. In: SIGGRAPH 2001: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pp. 117–128. ACM, New York (2001)
20. Patow, G.: Reflector Shape Design From Radiance Distributions. CAD for luminaries. PhD thesis, Universitat Politècnica de Catalunya (2005)
21. Patow, G., Pueyo, X., Vinacua, A.: User-guided inverse reflector design. Computers & Graphics 31, 501–515 (2007)
22. Patow, G., Pueyo, X.: A survey of inverse surface design from light transport behavior specification. Computer Graphics Forum 24, 773–789 (2005)
23. Neubauer, A.: The iterative solution of a nonlinear inverse problem from industry: design of reflectors. In: Proceedings of the international conference on Curves and surfaces in geometric design, pp. 335–342. A. K. Peters, Ltd., Natick (1994)
24. Caffarelli, L.A., Kochengin, S.A., Oliker, V.I.: On the numerical solution of the problem of reflector design with given far-field scattering data. Contemporary Mathematics 226, 13–32 (1999)
25. Wang, X.J.: On the design of a reflector antenna. Inverse Problems 12, 351–375 (1996)
26. Wang, X.J.: On the design of a reflector antenna II. Calc. Var. PDE 20, 329–341 (2004)
27. Weyrich, T., Peers, P., Matusik, W., Rusinkiewicz, S.: Fabricating microgeometry for custom surface reflectance. ACM Transactions on Graphics (Proc. SIGGRAPH) 28 (2009)
28. Veach, E.: Robust Monte Carlo Methods for Light Transport Simulation. PhD thesis, Stanford University (1997)
29. Robbins, H., Monro, S.: A stochastic approximation method. Annals of Mathematical Statistics 29, 400–407 (1951)
30. Kiefer, J., Wolfowitz, J.: Stochastic estimation of a regression function. Annals of Mathematical Statistics 23, 462–466 (1952)
31. Blum, J.R.: Multidimensional stochastic approximation methods. Annals of Mathematical Statistics 25, 737–744 (1954)
32. Spall, J.C.: An overview of the simultaneous perturbation method for efficient optimization. Johns Hopkins Apl Tech. Digest. 19(4), 482–492 (1998)
33. Spall, J.C.: Implementation of the simultaneous perturbation algorithm for stochastic optimization. IEEE Trans. Aerosp. Electron. Syst. 34(3), 817–823 (1998)
34. Spall, J.C.: Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE Trans. Autom. Control. 37, 332–341 (1992)

35. Yin, G.: Rates of convergence for a class of global stochastic optimization algorithms. SIAM J. on Optimization 10, 99–120 (1999)
36. Maryak, J.L., Chin, D.C.: Global random optimization by simultaneous perturbation stochastic approximation. In: WSC 2001: Proceedings of the 33nd conference on Winter simulation, pp. 307–312. IEEE Computer Society, Washington (2001)
37. Jones, M.H.: Jr., White, K.P.: Stochastic approximation with simulated annealing as an approach to global discrete-event simulation optimization. In: Winter Simulation Conference, pp. 500–507 (2004)
38. Piegl, L., Tiller, W.: The NURBS book, 2nd edn. Springer, New York (1997)
39. Lemieux, C.: Monte Carlo and Quasi-Monte Carlo Sampling. Springer, Heidelberg (2009)