Optimizing Multiple Camera Positions for the Deflectometric Measurement of Multiple Varying Targets

Oleg Lobachev Martin Schmidt Michael Guthe
Universität Bayreuth, Universitätsstraße 30, 95447 Bayreuth, Germany
{oleg.lobachev,martin.schmidt,michael.guthe}@uni-bayreuth.de

ABSTRACT

We present a device for detection of hail dents in passenger cars. For this purpose we have constructed a new multi-camera deflectometric setup for large specular objects. Deflectometric measurements have strict constraints how cameras can be placed – for instance: angular restrictions and distance limitations. An important trait of our system is the static setup – we use a *single* setup for camera configuration for *all* objects to be scanned.
We render the camera images and analyze them for the deflectometric needs to optimize the camera placement w. r. t. multiple parameters. Important ones are the positions of the cameras – reflections of the patterns should be clearly visible. Camera parameters are computed using a global optimization procedure for which we efficiently generate a good starting configuration. We introduce an empiric quality measure of a particular camera configuration and present both visual and quantitative results for the generated camera placement. This configuration was then used to build the actual device.

Keywords

deflectometry, ray tracing, quantum annealing

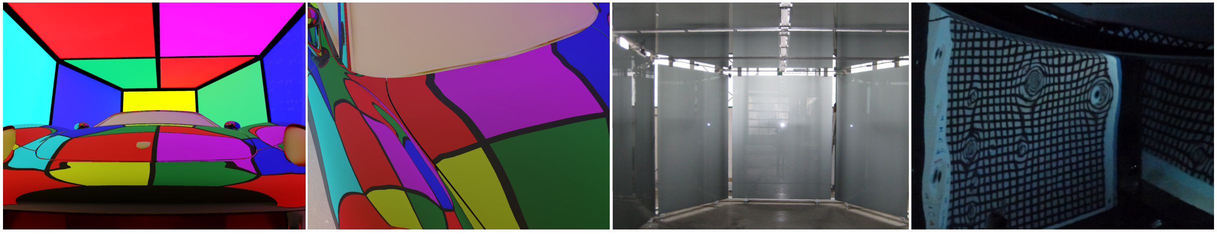

Figure 1: We optimize the camera positions for optimal inspection of specular surfaces using reflections of projected patterns. From left to right: an overview of the setup with color coding to show different projection screens; view through one of the cameras where the borders between screens are clearly visible; a picture of the actual device; hail damage on a (white) car where the dents are clearly visible as distortions of the reflected pattern.

1 INTRODUCTION

One of the most problematic events for car insurance companies are hailstorms as numerous claims need to be processed in the shortest possible time. Currently, each car needs to be assessed and documented by a human expert using a camera and a pattern that is reflected by the damaged surface. As this process is very time consuming, insurance companies are highly interested in automating assessment and documentation. The main challenge is reducing the recording time to process all damaged cars shortly after the event. Analysis and assessment may take longer since the claims settlement – i. e., repair – is the limiting factor anyways.

Traditional methods like laser measurement can only reconstruct an object if it consists of a diffuse material (see, e. g., [19]). For the inspection of specular surfaces, these methods are not applicable. One possible solution to this problem *deflectometry*. Cameras record reflections from a specific pattern on the surface. Once the pattern and the positions of the camera and the pattern generator (e. g., a fixed pattern or a display) are known, the surface normals can be calculated by comparing the distorted pattern reflection with the expected one.

In this joint project with i-Lumica AG, PHIcom GmbH, OGP Messtechnik GmbH, ALTRAN Group, AXA Auto Competence Center and ExactVision Bildverabeitungssysteme GmbH, we developed a specific deflectometric measurement system. The *goal* was to detect hail damage in cars, shown in the rightmost image in Figure 1. The whole measurement procedure, including driving the car in and out should take no more than three minutes. Our setup significantly differs from the traditional approach where a human expert uses a single camera and pattern to document the hail damage:

- We consider complete *large* objects: passenger cars.
- A large *variety* of different objects – any car, not a particular brand and model – should be measured with a completely static setup.
- We do not move a single camera and pattern projector above the surface, but use a multi-camera setup with multiple projectors.

In Figure 1 the third image shows the actual measurement device. It is built using 13 back projection screens, in between cameras can be mounted on aluminum beams.

For objects with a complex form, one camera alone can not measure the complete surface. However, setting up more than one camera can mitigate this problem and lead to an almost complete coverage – we will show this fact practically. Positioning the cameras and maximizing the coverage is performed using global optimization. An important issue in our setting is the optimality of camera positioning for a *broad set* of scanned objects. We do not optimize the setup for a single particular target, but for a set of different targets.

In this paper we address the *optimization* of the camera placement. We use optimization metaheuristics to maximize the *coverage*, i.e., the amount of the surface that the cameras can inspect. Car insurance companies require at most 3% tolerance in the number of hail dents. If we reach a sufficient coverage, we can presume that the remaining area has the same density of hail dents or less. This assumption is feasible as the uncovered area is both visually and physically less accessible and thus a lower density of hail dents can be assumed. If we obtain a coverage of 94% and estimate $\sim 3.2\%$ more hail dents than found, we have 3% too many if there are none and 3% too few if the density is the same as on the covered area.

The main contributions of this paper are:

- We present and discuss our novel setup and its optimization.
- Using ray tracing we obtain qualitative measures for the *coverage* of a given camera configuration.
- We propose an algorithm for the initial camera placement that produces good starting values, essential for the subsequent optimization.
- Using quantum annealing for further optimization, we have achieved almost complete coverage ($\geq 94\%$ of the visible surface).
- We verified our method on a broad set of car models, including a car that was not used during optimization.
- We use a *static* setup, i.e. a single configuration is used for all models.

2 RELATED WORK

There are many applications for which it is necessary to reconstruct the surface of an arbitrary object. Often, only contactless measurement methods are possible [2]. Existing approaches that measure the quality and characteristics of a surface mainly depend on diffuse materials. Highly specular materials and objects with complex form are harder to measure. Two different methods form the so called *traditional techniques*. These include, but are not limited to, stereoscopic vision and triangulation.

Stereoscopic vision uses two images that are recorded from different positions to create a 3D representation of a given object [9]. The two images can be matched by a set of feature points or by dense matching. The distance to the camera can then be estimated for each pixel based on the parallax. This is also used by almost all primates for spatial vision [26]. Stereoscopic vision fails in case of specular surfaces because the points can not be matched in the exact surface plane, but only in the mirror plane, which lies behind the actual surface.

The second approach is the triangulation of incident light on a surface [19]. A laser scanline is projected onto the surface and recorded by a camera above the laser. Then the distance to the surface can be calculated from the vertical displacement using triangulation. This process depends on diffuse or partially diffuse surfaces and does not work for highly specular objects. If the diffuse reflection is too dim, the surface needs to be coated, which is not possible for all types of objects. These techniques can be used as a fallback for surfaces that are not specular enough and therefore not suitable for deflectometry.

2.1 Deflectometry

The deflectometric inspection of specular surfaces [15, 27] simulates the way how humans recognize the shape of such surfaces: we look at the distortion of the reflected patterns and infer the shape of the surface from the reflection and the original form of the pattern. A stereo setup makes it possible to infer the point cloud of the surface from a set of specular reflections [2, 20]. The reflected patterns are projected on a back light screen using a pattern generator. A camera records the reflected image of the pattern on the surface. Because the pattern has a known structure and can additionally be modified when taking several consecutive images, it clearly shows the distortions induced by the shape of the surface.

Although deflectometry has been known for 25 years, practical methods became feasible only recently [7, 14, 16, 18]. Our method differs from these in several aspects, although others also compare with a reference [2, 16].

multiple cameras Most approaches use a single camera and move it along the object. We use many fixed cameras.

static setup Our setup is not repeatedly optimized for each particular model, but the camera positions and orientations remain fixed in productive use.

multiple objects Our system scans multiple different cars without modifications of the setup in-between.

optimization We optimize the camera positions, which is not considered in most papers on deflectometry. We discuss related work for optimal camera placement below.

stereo We could also use deflectometric stereo, i. e., the reconstruction of 3D image with multiple sensors [2, 3, 18, 28]. In an interesting alternative approach, Zheng et al. [31] rotate their objects in order to spare the number of cameras and projection screens.

Directly related to our work are the papers by Balzer et al. [2] and Hong et al. [14]. The first uses a deflectometric setup to find a dent in a part of a passenger car. In contrast to our work, a single sensor and a single projection screen are moved on a robot arm. The second utilize two cameras and five screens for deflectometry of solder joints. They were not concerned with the optimization of camera positions; further we use much more cameras.

2.2 Optimization

For deflectometric measurements, the camera must be placed in such a way that the whole surface of the object is visible and entirely covered by the reflected pattern. For objects with increased complexity, such as car bodies, this is not feasible using a single camera and pattern only. Multiple cameras must record the surface. The necessary condition is that every point on the surface can be seen by at least one camera. This is known as the *art gallery problem* and has received profound research attention. While this is necessary for any measurement, it is not sufficient. The cameras must view the surfaces in such an angle that every point on the surface is also covered by a reflection of the pattern [8, 11, 23].

In mechanical measurement apparatuses, the placement of cameras can not be arbitrary. Certain conditions must be met, such as traverses and beams that can be fitted with cameras and there must be a minimum distance between two cameras. As the installation of cameras is tedious and time-consuming, a preceding optimization of camera positioning is crucial to build the device. Camera position optimization [13] was most often researched for a single camera. The random walk method [29] is often used for camera positioning, while some other approaches, like genetic algorithms [24], are viable. See also the references in Olague and Mohr [24].

Random walk. This optimization method [6, 29], abbreviated here with RW, is a typical approach to camera placement [30]. It is often used in computer vision [5, 12, 22]. The camera positions and target directions are "jittered" a bit with random values, then the new position is evaluated. If it is better than the previous one it is kept and otherwise discarded. This method often gets stuck in a local optimum if the step size is too small.

Adaptive simulated annealing. The adaptive simulated annealing method, in short ASA, is a global optimization metaheuristic [17]; it is adaptive because the step size is adjusted to the current state of the problem. ASA has an important property of being a *global* optimization method. "Usual" optimization algorithms are local, i. e., as soon as the local optimum is found, no better solution is reached. In contrast, the ASA can "move away" from the local optimum in a search for a global optimum.

Quantum annealing. This method [1, 10] (abbreviated QA) slightly differs from ASA. To achieve global maximum, ASA allows certain "setbacks": the result might be worse temporarily. In contrast, QA "tunnels" through regions between local optima, which also allows QA to find a global optimum. More formally, in QA the tunneling distance for the current step determines possible candidates for the next state; in ASA this depends on the notion of temperature in the annealing analogy. QA can be seen as a quantum Monte-Carlo method. QA is also similar to RW, the key difference lies in how the "jitter" radius is defined, depending on the step count. In RW this radius is constant. In QA the radius is chosen based on a normal distribution. The variance of this distribution is steadily decreasing, but very large, "tunneling" jumps are still possible, although less and less probable with increasing number of steps.

While ASA and QA could be used in our case, the latter has several advantages. The possible camera positions do not form a connected manifold and thus jumps are inevitable. While this is natural in QA, it needs to be specially handled in ASA. In addition, QA has been evidenced to need relatively few iterations compared to ASA in order to find a good solution [21]; in our case iterations are quite costly for setups with many cameras and objects.

3 MEASUREMENT SYSTEM

Our setup includes 13 projection screens placed in a "greenhouse" structure as shown in Figure 1. Note the spaces between the screens. First of all, this is a construction-induced limitation; however, we also use these spaces to place cameras without obstructing the projection screens. The placement of the cameras on the supporting beams induces some limitations to the camera placement, as discussed below in Section 3.1.

Each camera records several images using different patterns on the projection screens. Based on that data, the hail dents are detected as shown in Figure 2. To register the dents on the car and remove duplicate ones, we need to determine the exact position of the car. For this purpose, we mount four wide-angle cameras: to the left, to the right, behind and above the car. Then we use background subtraction [25] to detect the car. We compare the captured images to the views of 3D models, minimizing the difference between the covered pixels. This gives us the position and actual reference model of the car. Note that the reference car models are segmented, as we need to assign the hail damage to construction parts. Projecting the images back onto the reference model gives us the exact positions of the hail dents. Subsequently, duplicates are removed. The final list per part is the result of the deflectometric measurement.

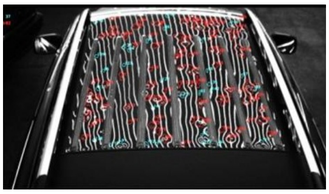

Figure 2: Camera image showing detected hail dents. In addition, the dents are classified into those that can be repaired without much effort (cyan circles) and those that cannot (red circles).

3.1 Constraints

Our approach considers a number of constraints regarding the installation and then optimizes the complete camera set. All parameters are subject to a global optimization procedure that will generate the parameter values with the highest coverage possible with respect to the specified constraints.

We change both *position* and *orientation* of the camera, as well as the focal length of the lens. Hence, each camera theoretically possesses 7 degrees of freedom: 6 DOFs for the camera position and orientation and one for the focal length, which we choose from a fixed set. The position is limited by the construction constraints: the cameras are placed on beams between the projection screens (see Figure 3) and the orientation is also limited to 2 DOFs. So in total there are only $3+1$ DOFs per camera. A further constraint is the *distance* between cameras. Due to construction limitations two cameras need to be almost 10 cm apart. We set this distance to exactly 10 cm in our optimization.

Summarizing, some constraints are implied by the physical properties of the cameras: focal distance, depth of field, resolution of the sensor. Other constraints emerge from the physical construction: we place cameras not everywhere, but only where it is mechanically feasible.

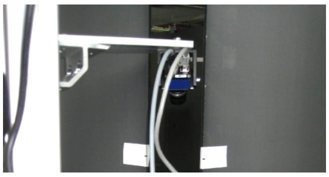

Figure 3: Camera mounted between two projections screens. Note that while the view direction can be modified, the up vector of the camera is fixed.

We define several criteria that need to be fulfilled for each point on the surface of the car body in order to be classified as covered. These include basic visibility (visible by at least one camera), extended visibility (visible by at least two cameras if 3D reconstruction is desired), visible reflection of pattern display, and reflection adequate for deflectometry. In the following, we will discuss these criteria in detail.

Notably, surface parts with interreflections (e. g., concave parts of the car surface) are marked as not covered.

3.2 Visibility

The most basic requirement for coverage is the visibility of each point. This is easily determined: If a point can be seen by at least one camera, it is visible. To compute the visibility of a certain point, several steps are performed. The images resulting from all of the cameras in the scene are processed. Covered pixels are transformed into covered patches on the surface of the object. Any area not covered by a camera is not visible.

3.3 Reflections of pattern display

For each point, the eye vector describes the direction from which the camera sees this point. A reflection can occur when the reflection vector $\vec{R}$ belonging to the eye vector $\vec{V}$ hits one of the pattern displays. Figure 4 illustrates this concept.

One problem during the optimization with respect to the visibility of reflected patterns are the gaps between the display segments. They result from technical and mechanical constraints during construction of the system.

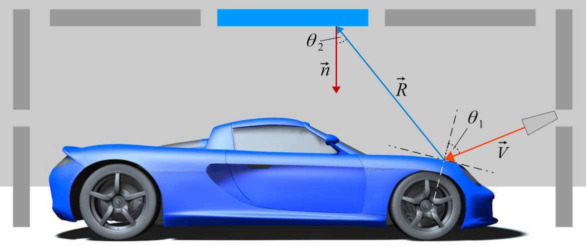

Figure 4: Fresnel reflection angles on the surface and display.

Solving this problem is easy, as the affected points have to be covered by a different camera.

The reflection quality is the most interesting and difficult requirement. An adequate reflection is one that fulfills the deflectometric requirements [16]. For our system, we can reduce these requirements to the following:

- Possible reflection
- Actually visible reflection
- Reflected image must not hide details.

The possible reflection is already guaranteed by the preceding paragraph. It is necessary for any further calculations. Whether the reflection is actually visible depends on several factors. The viewing angle influences the strength of reflections on the surface. This is known as the Fresnel effect. The angle between surface normal and viewing direction θ_1 differs between points on the surface. The glass panels of the projection screens induce another limitation of the viewing angles. The angle θ_2 between the normal of the screen surface and the view vector $\vec{R}$ must not be too large, otherwise the Fresnel reflection creates an overlay on the pattern. In our experiments, we found that up to 60° the reflection does not cause measurement artifacts.

Furthermore, we want to maintain a good quality of the reflected image. Shrinking[1] the image in one direction leads to severe loss of resolution. We do not want the shrinking in lateral direction to exceed 1 : 2. This leads to the second limitation: the viewing angle θ_1 between surface normal and view vector $\vec{V}$ must be less than $\theta_{\max}$, the angle where the lateral shrinking is 0.5. Any angles larger than $\theta_{\max} = 60^\circ$ are still visible, but lead to distortions due to the extreme shrinking of the reflected pattern. Since we want to avoid such errors, $\theta_{\max}$ is the upper bound for the acceptable reflection angle.

The final restriction is the minimum resolution on the surface. The projected pixel should be smaller than a given threshold in both directions. Note that the projected size is not only distance and focus dependent, but also changes with θ_1.

4 OPTIMIZATION

The goal of our simulation approach is to optimize the camera placement in a scene of multiple cameras and an object. The more parts of the object's surface are visible, the better the placement is. Therefore the *coverage*, a measure for covered surface, is an important objective function. We aim to optimize the input parameters (position and view direction of every camera in the setup) leading to the best possible coverage. The simulation data was used to determine the number of required cameras and their optimal position and orientation for the construction of the actual physical device, see Figure 1.

The *coverage* is the value we are maximizing, i.e., the quality measure of the camera placement. We define it as the relative car body surface area, which is seen by a camera and covered with a suitable pattern reflection. We combine the individual coverages for all cameras to obtain the total covered area per model. Then we combine the relative coverages of all models using

$$x_{\max} = \arg\max_{x \in D} \left(\min_{1 \le i \le N} \left\{ \frac{A_{\text{cov},i}(x)}{A_{\text{total},i}} \right\} + \frac{\lambda}{N} \sum_{i=1}^{N} \frac{A_{\text{cov},i}(x)}{A_{\text{total},i}} \right),$$

where D is the search space, as discussed in Section 3.1, and i iterates through the N car models. We use the penalty weight λ to increase the convergence rate. In this manner not only the worst case is optimized, but also the others. We use quantum annealing to compute $x_{\max}$ and thus maximize the coverage.

4.1 Computing the coverage

We use ray-tracing on commodity graphics hardware to simulate the recorded images for a set of cameras and then compute the covered area $A_{\text{cov},i}$ of each car. During this process, we estimate the total area of the car body surface in which the pattern reflection is clearly visible and which meets all constraints discussed in Section 3.1. Basically, we add up all pixels of the specular car surface for that the above constraints are satisfied. This value is then divided by the total externally visible area that is computed before starting the optimization.

Each car model is prepared in order to efficiently measure the covered area. The model is first split into reflective parts that need to be measured – i.e., the car body – and the remaining parts that may occlude the reflective ones or the screens. In addition to calculating per face and per vertex normals for the reflections, a texture map is created by unwrapping the car body using simple box mapping [4]. The Figure 5, left, shows the generated texture atlas of a car body. We use the texture map to determine if a pixel is externally visible: we simply trace several rays outward and check them for occlusion.

For each optimization pass the view of every camera is rendered to determine which parts are visible and contain usable reflections. Then each pixel of the camera view is mapped into the texture atlas. As the only information we need for this mapping is the texture coordinate of the pixel, we simply store that in the render buffer. For pixels that do not meet the constraints, we simply store invalid coordinates (e.g., $[-1,-1]$). Figure 6 shows the content of this render buffer for one of the cameras.

[1] Strictly speaking, there are two kinds of lateral shrinking of the pattern: if the surface curvature is too large and if the projected pattern is viewed under a too sharp angle. Both are important for our application, but in the following we discuss the one that is actually controllable with camera placement.

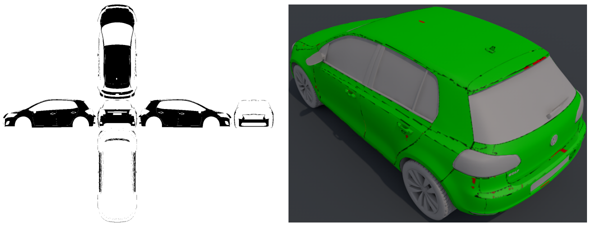

Figure 5: Texture atlas of a sample car (left) and computed coverage mapped onto the model (right). Almost everything is green, meaning sufficient coverage.

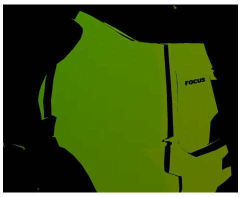

Figure 6: Intermediate buffer storing the texture coordinates of each pixel containing a suitable reflection of a pattern. These images are generated for every camera and mapped into the texture atlas.

4.2 Initial placement

All random-walk based methods, including QA, are *local* search methods. Good start values are crucially important for the quality of the result and the runtime. The first phase generates an initial placement of all cameras.

We place a camera, to cover the largest "hole". These are detected by low pass filtering the texture atlas using a Gaussian filter of radius 0.1 of the texture size. Then we search for the local maxima and keep the largest 10. To place a camera, we choose randomly one of these and determine position and normal at the closest texel containing a part of the surface. Based on position and normal, we first place the camera 1 m above that point. Then we choose a random position within a distance of 1 m from that point and project the camera to the closest free location on a beam. This process is repeated 100 times for each camera and the placement with the highest coverage is chosen. Note that the filtering and search are performed once for each camera, so the additional overhead is relatively low compared to QA iterations.

Note that the initialization search phase is not *required*, as QA would have found good solutions anyway, but greatly improves the speed of the computation.

4.3 Output

To visualize the quality of the currently found solution and build the actual device, we produce several outputs. These are a texture depicting the current coverage, a text output of the minimum coverage, and the camera placement itself. Using color-coding, we can show each point's characteristics. Pixels that are covered are green and the others are red. In addition, pixels that are not visible from above or beside the car are black. The latter is, e. g., the case for pixels covered by the license plate, in joints, or in the bottom-facing parts of the car. The right image in Figure 5 shows such a visualization.

100 cameras, 1000 iterations

Method	RW	SA	QA	IP+QA*
Time, s.	8344.34	7851.55	6995.35	3256.38

Table 1: Left columns: run time for three optimization methods, right: mean run time for the performance-tuned QA method with initial placement phase, designated "IP+QA*". All approaches use 100 cameras, we measure the time for 1000 iterations. Time is in seconds.

The textual output of the simulation is the list of cameras with their position in the construction coordinates and their look-at point. We also implemented a script to import these in 3D-modeling/CAD software to visualize the cameras and for the construction of the actual device.

5 RESULTS

We optimized the camera positions for *twelve different* car models. These models were chosen to be as variative as possible to ensure covering maximal shape space.[2] The cameras we use have a 6.9 mm $\times$ 5.5 mm CCD sensor with a resolution of 1280×1024 pixels. The cameras are capable of producing images at 100 fps. These parameters where chosen because of the sharp capture time limitations. The focal lengths of the lenses are 9 mm, 12 mm, 16 mm, 25 mm, and 35 mm. As desired resolution on the car body surface, we set a minimum of one pixel per mm. Relatively small resolution of the cameras is compensated by their speed; more cameras are added to achieve the desired resolution on the car surface. We found the minimum number of cameras required to achieve the desired coverage of 94% to be 99 for our test set. While the minimum coverage was 94.0006% (Chevrolet Lumina), the peak coverage was 97.0595% (Volkswagen Golf 6). Still remaining red spots (see Figure 7) are outliers outside the desired coverage. The high number of cameras stems from the desired accuracy of ~ 0.5 mm for the detection.

The optimization was performed on an Intel Core-i7-3770K at 3.5 GHz with 16 GB RAM and NVIDIA GeForce GTX 680, running Windows 7. For the quantum annealing we used $\lambda = 10^{-3}$. The number of iterations is 10000, disregarding heat-up. In each iteration we either changed the position of a single camera or moved all cameras. In the first case the distance of the camera "jitter" ranged from 50 cm to 0.1 mm, and in the

[2] The models were: Audi R8, BMW X5, Chevrolet Lumina, Fiat Grande Punto, Ford Focus ST, Volkswagen Golf 6, Mazda MX5, Mercedes C, Peugeot 107, Porsche Carrera GT, Renault Clio 2, Toyota Auris.

second case from 5 m to 1 mm. The cool-down factor was $\varepsilon^- \approx 0.99915$, the heat-up factor was $\varepsilon^+ \approx 1.1857$. Each time, we found a better solution, the temperature was increased by that factor. In total, the optimization took 25.5 hours. The optimization time is less than the three days required to set up the rest of the device before the cameras could be mounted.

We compared the number of required cameras for all cars against the number required when optimizing for each car separately. Figure 7 shows the coverage–number of cameras relation. For all 12 cars in a static setup, the number of cameras (99) roughly doubles the number required for each car alone (35–67).

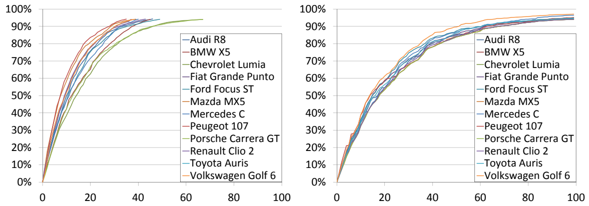

Figure 7: Comparison of coverage when optimizing for each car independently (left) vs. optimizing for all cars together (right). The horizontal axis represents the number of cameras, the vertical axis shows the coverage.

Figure 8 shows the coverage visualized on some of the car models. Most important here is the top view as hail dents are much more common on horizontal surfaces. The rightmost image shows an experiment to test our approach. After optimizing the cameras for the 12 cars, we used this setup to compute the coverage of the first generation Pontiac Firebird, which was not included in the optimization set. We reach a coverage of 93.87%. Hence, an acceptable coverage is also possible for cars that were not considered during the optimization of the camera placement. This makes it easy to use the camera setup for cars that are newly introduced to the market, because the shape space changes only marginally in short time. Still, the setup needs to be revised at some point as those chances in shape space can add up and need to be covered. The bigger the shape space is during optimization, the less revisions are due. This has also a positive effect on custom car alterations, e. g., spoilers. Surface parts occluded by these are less prone to dents.

6 CONCLUSION & FUTURE WORK

We have presented an approach to build a deflectometric measurement system for large specular objects using a multi-camera setup. The objects measurable by our setup are complete cars – a quite large field of work. The device we describe here was actually built and is now operational at AXA, Bern. We use quantum annealing combined with an effective initial camera placement to determine the optimal camera placement. We consistently reached high coverage rates for multiple simulated vehicles using a single configuration – the set of camera positions, orientations and focal lengths, shared for all vehicles. Approximately 48% more cameras are required for all cars than for the most difficult one alone.

There are some issues to investigate in our future work. Other build forms for the supportive constructions for projection panels and camera mounting could be viable. It could make sense to try some further optimization methods on the whole setup, not only for the cameras. The cameras also have further advanced parameters one could change during the optimization, like the tilt for the selective focus. We will further investigate the possibility to dynamically group the cameras for 3D reconstruction without reference models. It would also be possible to use multiple *camera triples* for a 3D reconstruction. We have tested this approach in our simulation, but do not use it in the actual device due to the higher number of cameras and availability of reference models. While 3D reconstruction is not necessary in the current system, it could be of interest for future projects.

ACKNOWLEDGMENTS

Most of this research has been conducted while the authors were with University Marburg, supported by HMWK LOEWE KMU program and PHIcom GmbH.

REFERENCES

[1] B. Apolloni, C. Carvalho, and D. de Falco. Quantum stochastic optimization. *Stoch. Proc. Appl.*, 33(2):233–244, 1989.

[2] J. Balzer, S. Höfer, and J. Beyerer. Multiview specular stereo reconstruction of large mirror surfaces. In *Proc. CVPR ’11*, pages 2537–2544. IEEE.

[3] J. Balzer and S. Werling. Principles of shape from specular reflection. *Measurement*, 43(10):1305–1317, 2010.

[4] E. A. Bier and K. R. Sloan. Two-part texture mappings. *IEEE Comput. Graph. and Appl.*, 6(9):40–53, 1986.

[5] P. Bovet and S. Benhamou. Spatial analysis of animals’ movements using a correlated random walk model. *J. Theor. Biol.*, 131(4):419–433, 1988.

[6] S. Chandrasekhar. Stochastic problems in physics and astronomy. *Rev. Mod. Phys.*, 15:1–89, 1943.

[7] B. Denkena, H. Ahlers, F. Berg, T. Wolf, and H. K. Tönshoff. Fast inspection of larger sized curved surfaces by stripe projection. *CIRP Ann. Manuf. Technol.*, 51(1):499–502, 2002.

[8] U. M. Erdem and S. Sclaroff. Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements. *Comput. Vis. Image Und.*, 103(3):156–169, 2006.

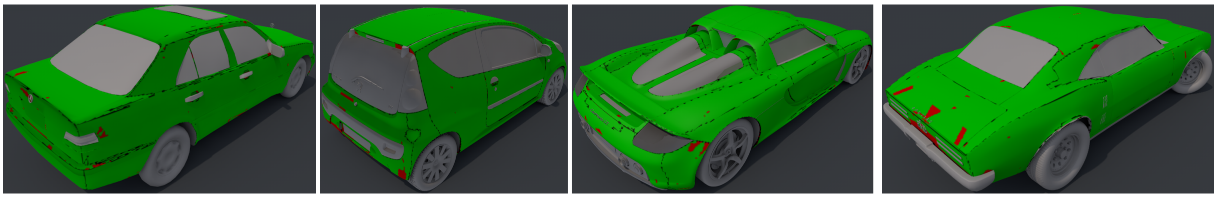

Figure 8: Visualization of the coverage for three of the models used during the optimization. Green is covered, red is not covered, black is not externally visible. The rightmost image shows the coverage of a car that was not part of this set. Nevertheless, we still achieve a good coverage.

[9] O. Faugeras and R. Keriven. Complete dense stereovision using level set methods. In *Computer Vision*, ECCV '98, pages 379–393, 1998.

[10] A. B. Finnila, M. A. Gomez, C. Sebenik, C. Stenson, and J. D. Doll. Quantum annealing: A new method for minimizing multidimensional functions. *Chem. Phys. Lett.*, 219(5–6):343–348, 1994.

[11] S. Fleishman, D. Cohen-Or, and D. Lischinski. Automatic camera placement for image-based modelling. *Comput. Graph. Forum*, 5:231–239, 2000.

[12] L. Grady. Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(11):1768–1783, 2006.

[13] N. Halper and P. Olivier. Camplan: A camera planning agent. In *Smart Graphics 2000 AAAI Spring Symposium*, pages 92–100, 2000.

[14] D. Hong, H. Park, and H. Cho. Design of a multiscreen deflectometer for shape measurement of solder joints on a PCB. In *Proc. ISIE '09*, pages 127–132. IEEE, 2009.

[15] I. Ihrke, K. N. Kutulakos, H. P. A. Lensch, M. Magnor, and W. Heidrich. Transparent and specular object reconstruction. *Comput. Graph. Forum*, 29(8):2400–2426, 2010.

[16] S. Kammel and F. P. León. Deflectometric measurement of specular surfaces. *IEEE Trans. Instrum. Meas.*, 57(4):763–769, 2008.

[17] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[18] M. C. Knauer, J. Kaminski, and G. Häusler. Phase measuring deflectometry: A new approach to specular free-form surfaces. In *P. Soc. Photo-Opt. Ins.*, 2004.

[19] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital Michelangelo project: 3D scanning of large statues. In *Proc. SIGGRAPH '00*, pages 131–144. ACM, 2000.

[20] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(5828):133–135, 1981.

[21] R. Martoňák, G. E. Santoro, and E. Tosatti. Quantum annealing of the traveling-salesman problem. *Phys. Rev. E*, 70:057701, 2004.

[22] M. Meila and J. Shi. Learning segmentation with random walk. In *Proc. NIPS '01*, 2001.

[23] A. T. Murray and K. Kim. Coverage optimization to support security monitoring. *Comput. Environ. Urban.*, 31:133–147, 2007.

[24] G. Olague and R. Mohr. Optimal camera placement for accurate reconstruction. *Pattern Recognit.*, 35(4):927–944, 2002.

[25] M. Piccardi. Background subtraction techniques: a review. In *Proc. Sys. Man. Cybern.*, pages 3099–3104. IEEE, 2004.

[26] N. Qian. Binocular disparity review and the perception of depth. *Neuron*, 18(3):359–368, 1997.

[27] A. Sanderson, L. Weiss, and S. Nayar. Structured highlight inspection of specular surfaces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(1):44–55, 1988.

[28] H. Schultz. Retrieving shape information from multiple images of a specular surface. *IEEE T. Pattern. Anal.*, 16(2):195–201, Feb 1994.

[29] F. Spitzer. *Principles of random walk*. Springer-Verlag, 2001.

[30] J. Zhao, S. Cheung, and T. Nguyen. *Optimal visual sensor network configuration*. AP, 2009.

[31] J. Y. Zheng and A. Murata. Acquiring a complete 3D model from specular motion under the illumination of circular-shaped light sources. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):913–920, 2000.