Autonomous Surgical Lamps

Jörn Teuber[1], Rene Weller[1], Ron Kikinis[1], Karl-Jürgen Oldhafer[2], Michael J. Lipp[2], Gabriel Zachmann[1]

[1] *University of Bremen, Faculty of Computer Science, Bremen, Germany*
[2] *Asklepios Klinik Barmbek, Hamburg, Germany*

Kontakt: joern.teuber@cs.uni-bremen.de

Abstract:

We present a novel method for the autonomous positioning of surgical lamps in open surgeries. The basic idea is to use an inexpensive depth camera to track all objects and the surgical staff and also generate a dynamic online model of the operation situs. Based on this information, our algorithms continuously compute the optimal positions for all surgical lamps. These positions can then be communicated to robotic arms so that the lamps mounted on their end effectors will move autonomously. This will ensure optimal lighting of the operation situs at all times, while avoiding occlusions and shadows from obstacles. We tested our algorithm in a VR simulation using real-world depth camera data that was recorded during a real abdominal operation. Our results show that our method is robust and can ensure close-to-optimal lighting conditions in real-world surgeries with an update rate of 20 Hz.
Keywords: depth camera, open surgery, operating situs, optimal lighting.

1 Problem

Perfect lighting conditions are essential in surgery. According to basic guidelines in open abdominal surgery, the situs has to be illuminated as uniformly as possible with at least 40,000 lux [1]. While the pure luminance intensity can be easily achieved by modern surgical lamps, the positioning of the lamps to guarantee optimal lighting conditions is still a challenge. Usually, a set of surgical lamps is mounted above the operation table and the surgeon or the surgical staff have to adjust them manually. Obviously, this re-positioning interrupts the workflow, as further investigated by [2]. Moreover, it facilitates the need for low-hanging lamps with sterile handles, which have to be replaced before every surgery to maintain sterility. Even in state-of-the-art hybrid robotic operation rooms, the surgical lamps are still moved manually.

Surprisingly, there are only very few works published on the automatic positioning of surgical lamps. Most approaches simply replace the manual adjustment using handles by other interaction metaphors that still require the attention and input of a human operator. For instance, [3] proposed a remote control unit to manually change the rotation and some other parameters of otherwise stationary lamps. In [4], a voice recognition system was used to control a variety of operating room systems including surgery lamps.

To our knowledge, the only other approach for an autonomous positioning of the lamps was presented in [5]. The authors tracked an infrared marker mounted on the surgeons glove and the light cone simply followed the marker. However, usually not the hands of the surgeon but the operation situs should be perfectly illuminated. Moreover, this method does not avoid obstacles but simply relies on the lamps' shadow dilution property. This results in less-than-optimal lighting conditions at the situs, depending on the position of the staff and other obstacles.

None of these approaches considers an *optimal* positioning of the lamps, although this has been investigated for other instruments like the C-arm for aortic interventions [6]. Basically, this problem seems to be closely related to path planning tasks for virtual cameras that often arise in VR or computer games. [7] presents an overview on current state-of-the art in this field. However, the conditions in an operation room differ fundamentally; For instance, the movement of the lamps is restricted by physical constraints and we have a highly dynamic and interactive environment.

In this paper, we present the first autonomous method with optimized positioning of the surgical lamps. The main idea is to use a depth camera to track all elements in the environment, including the surgeon, the surgical staff and also the situs simultaneously. Based on this data we apply an optimization algorithm that operates directly on the depth images and does not require a time consuming polygonal reconstruction of the scene. The resulting new positions can be directly applied to the surgical lamps that can be mounted on robotic arms. Our algorithms are fast, robust and easy to implement. We tested our algorithm with real-world data from an abdominal operation and the results show a distraction-free positioning of the surgical lamps.

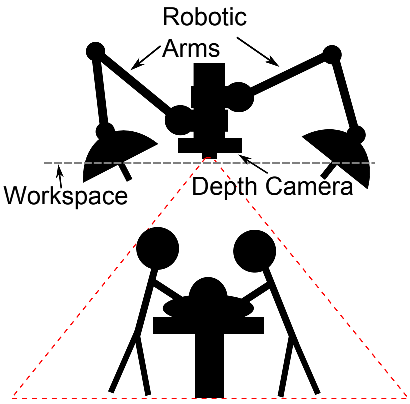

Figure 1: Setup of our system in the operating room. A depth camera looks down on the patient and surgical staff. The lamps can be moved to any given point on a predefined plane above the operating table called workspace and rotate to always face the situs.

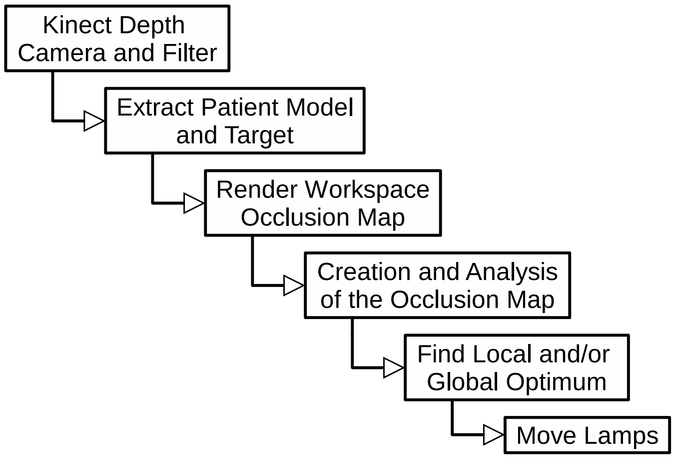

Figure 2: Overview of our pipeline which starts with the input of a depth camera and results in optimal positions for the lamps.

2 Materials and Methods

Our method is suitable for almost all open surgical scenarios. The initial situation in all these scenarios is very similar: The patient is fixed on the surgical table and the lamps are mounted directly above the table. In order to apply our method, we simply have to add an additional depth camera. The camera should be positioned directly above the patient. In order to move the lamps automatically, the typical hinged brackets of the surgical lamps have to be motorized or replaced by robotic arms. Figure 1 shows the basic setup of our system. We are fully aware that for many operations the situs is not completely opened up so that the region of feasible lamp positions is smaller than the complete reachable volume of the lamps' linkage. However, in this paper, we still consider the complete reachable volume of the lamps because any method must be able to optimize over that space.

The computation of the optimal lamp position from the depth image can be divided into four main steps: We have to filter the input image, find the target to be illuminated, compute and analyse an occlusion map to identify obstacles and finally, optimize the lamp positions. Figure 2 shows an overview of our pipeline. In the following, we will describe all individual steps in more detail.

2.1 Depth Image and Filtering

Depth images of all available depth cameras are typically very noisy. In our current setup we use a Kinect v2. Depth images from this particular device tend to oscillate even on static surfaces. In order to mitigate this effect we initially have to apply a special filter to the depth data before continuing with further steps in the pipeline. In our system, real-time capability is essential. Consequently, we should at least be able to preserve the 30Hz frame-rate of the Kinect to guarantee fast reactions in a dynamic environment. Unfortunately, most sophisticated filters like [8] and [9] that are able to correct invalid values, are too slow (with approx. 0.5s/frame, according to the papers). Therefore, we use a conditional Infinite Impulse Response (cIIR) approach. The idea is to reset the buffer of an IIR filter whenever the depth value changes beyond a certain threshold (See Algorithm 1). The advantage is that static or slow moving objects are filtered while fast movements are not delayed.

2.2 Extracting the Situs Position

In the second step, we have to find the interesting area that should be illuminated in the depth image, this means the situs. In open operations, the patient is usually fixed on the table and he does not move. Moreover, he is covered with a cloth except of the operation situs area. In other words, the situs is, in the depth image, a local minimum valley surrounded by a local high plateau. Finding such an area in a static depth image is relatively easy. However, in our scenario, the situs changes over time and it could be, partially or completely, occluded by the surgeons hands or instruments. In order to handle these challenges, we maintain a *global depth buffer* that stores the maximum depth values for each pixel over time, similar to a z-buffer known from computer

Algorithm 1: filterDepthPixel

Input: *current*, *last* and *history* pixel values, *threshold*
Output: filtered pixel value
begin
  if *current is valid* ***and*** $|current - last| < threshold$ **then**
    if *history is valid* **then**
      set *history* to $(current + history)/2$;
    else
      set *history* to $(current \cdot 2 + last)/3$;
    end if
    return *history*;
  else
    invalidate *history*;
    return *current*;
  end if
end

graphics. Obviously, temporal occlusions does not affect the search for the situs area in this global depth buffer. As a second advantage, we can use it for background subtraction. This reduces the complexity in our next step, the occlusion map construction.

2.3 Creating and Analyzing an Occlusion Map

Ideally, we want to find a position in the lamps *workspace* with a free sight on the operation situs, i.e., there are no obstacles in the line of sight from the situs to the lamps (See Figure 1). Hence, it does not matter if the obstacles appear in a sight from the lamps to the situs or vice versa. For sake of simplicity we choose to look from the situs' position as illustrated in Figure 3. This allows us to use traditional hardware accelerated rendering methods: We derive a point cloud from the Kinect's depth map, transform it into world coordinates, set the rendering screen to the situs' position with z-direction pointing to the lamps workspace, set the field of view to enclose the entire workspace and render the point cloud to an *occlusion map*. Each pixel in this occlusion map represents a small light cone and thus, corresponds to a small area in the lamps workspace. Consequently, we can perform our optimization directly on this occlusion map.

2.4 Local and Global Optimum Search

An optimal position and movement of the surgical lamps should fulfill several, partly contradictory, constraints:

- The ray of light of the lamps should not be blocked by any obstacle in the OR. If there is an object in the ray of light, the object should be small enough for the *shadow dilution* to mitigate the occlusion.
- The lamps should never collide with one another.
- The lamps should provide a uniform lighting of the situs by positioning themselves as uniformly as possible on the workspace.
- The lamps should only move when necessary to avoid distractions of the surgeon.

The information for the first three constraints can be, basically, directly derived from the occlusion map. However, a single pixel in the occlusion map is much smaller than the extension of the lamps and it represents only the single information, whether the pixel is occluded or not. Hence, performing an optimization directly on the occlusion map may result in a local minimum on a pixel that is not occluded but surrounded by a large area of occluded pixels. In order to overcome this limitation, we compute a *distance transform map* for the occlusion map: For each pixel we store the distance to the closest occluded pixel (See Figure 4).

The last constrain adds an additional dimension that can not be derived from a single static occlusion map because it is a temporal constraint. Consequently, we have to take the temporal progression into account. Actually, all objects in the scene move continuously and moreover, we know there are areas that are more likely occluded than other areas, like parts of the situs where the surgeon works with his hand. Hence, we can use temporal coherence to foresee areas in the map that will be occluded in the future with a higher probability than other areas. In order to identify these areas, we define two additional maps: an *average occlusion map* and an *activity map*. Our average occlusion map is an average over all previous occlusion maps while our activity

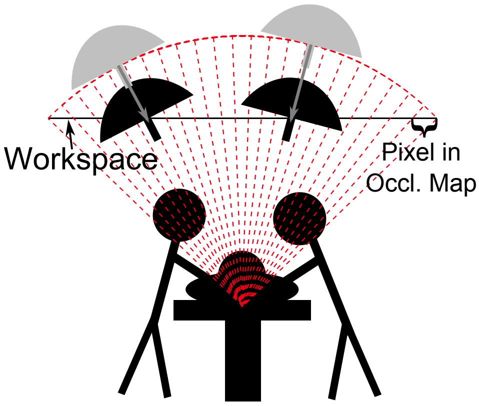

Figure 3: Illustration of the occlusion map in relation to the real world environment. The lamps can be seen as implicitly positioned on the surface of a sphere with the centre inside the situs. That is why we render from the situs towards the ceiling to obtain an occlusion map.

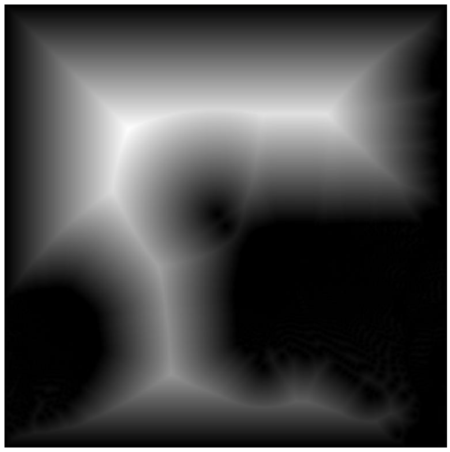

Figure 4: Example of a distance transformed occlusion map. Black pixels are occluded, white not. By using a distance transform it is easy to identify the optimal positions in this frame.

map stores the number of times each pixel has flipped its occlusion state in the past. In the following, we will call these three maps, the distance transform, the average occlusion and the activity map, our *workspace maps*.

We can formulate the search for the optimal positions of the lamps by defining a cost function on a combination of our workspace maps. This cost function is a simple weighted sum of pixel values of the workspace maps, with an additional component ensuring an even distribution of the lamps and preventing collisions. Usually, a single lamp can not fulfill all constraints, especially, if the occlusion changes suddenly. Hence we have to use a set of lamps as it is also standard in current operation rooms that are adjusted manually. A complete evaluation of the search space would require the computation of the cost function for each pixel for each combination of this set of lamps. The complexity of such a search is $O(m^n)$ with m being the resolution of our maps and n the number of lamps. Even in case of a moderate resolution of 256×256 pixels and 2 to 3 lamps it would be impossible to compute this in real time. We propose a two-level optimization scheme to still guarantee real-time performance: We start with a *local* search using *steepest descend* on the output of the cost function for quickly finding local optima based on the current lamp position. This can be done fast enough for real-time. If the lamps' position falls beneath a certain threshold, i.e., the lamps are probably caught in a bad local optimum, we start, in parallel, a *global optimization* using random sampling. Actually, this global optimization is much slower. However, we can simply choose the threshold in such a way that the current local optimum will be good enough until the global optimization has finished its computation.

3 Results

We have implemented a prototype of our pipeline in C++. Furthermore, we recorded a two part, seven hour surgery with the Kinect v2. In order to evaluate the performance and the quality of our system, we set up a virtual environment of an operation room where we can play back the recorded real-world data and set the positions of virtual lamps (See Figure 6). Our algorithm runs on a mobile workstation with an Intel I7, 16 GByte of main memory and a NVIDIA Quadro K2100M.

We started with an evaluation of the first two parts of our pipeline, i.e. the depth filtering, the rendering of the occlusion map, including also the computation of the workspace maps. The performance of the filtering and the computation of the point cloud depend only on the resolution of the depth image. In order to derive the dependency of the performance from the size of the occlusion map we conducted experiments with different resolutions of 256^2 up to 512^2 pixels (See Figure 5). The rendering to the occlusion map requires most of the time. However, it is almost independent of the resolution and depends mainly on the scene complexity and the download of the data from GPU to CPU memory. In contrast, the setup of the workspace maps, which is currently done on the CPU, depends linearly on the resolution.

Additionally, we measured the performance of our two-level optimization scheme for different resolutions of the workspace map. The local optimization using steepest descent is extremely fast: It requires less than two milliseconds to find a local minimum. Moreover, it is independent of the resolution (See Figure 7). The global search is, at least in its current not optimized implementation, much slower and takes around 0.2 seconds if we

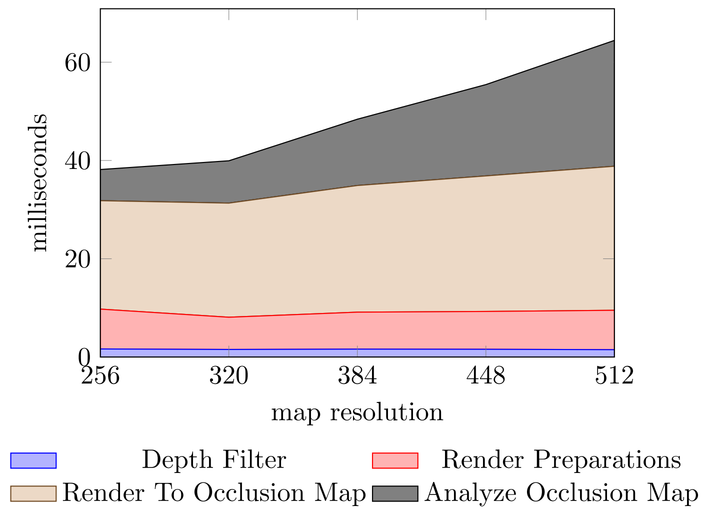

Figure 5: Overall performance of our method, including a breakdown of the running time by the first four stages of the pipeline depending on different occlusion map resolutions.

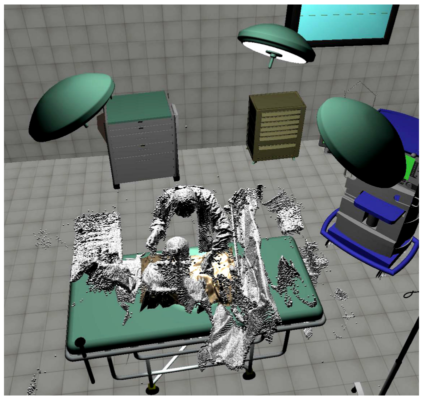

Figure 6: Screenshot of our virtual environment for testing our optimization methods. The grey shapes visualize point clouds recorded during a real-life operation. In addition, you can see the autonomous lamps (here without the robot booms) and part of the reconstructed patient model.

use a sample size of $512^2 = 262144$. Surprisingly, even if we use the same number of samples for all resolutions, its performance still seems to depend on the image size (See Figure 7). We assume caching effects on the CPU and we will further investigate this behavior in the future.

4 Discussion

The results of our measurements show that our algorithm is able to position a set of surgical lamps to illuminate a surgery situs uniformly. Any occlusion of the lamps rays of light is avoided and the lamps move accordingly. Basically, our algorithm is suited for all kinds of open operations.

However, there is still room for improvements. Currently, the heuristic to detect the situs relies on the global depth map that stores a global minimum for each depth value. In real operations, these values may also increase inside the situs. Moreover, the optimization function could also take the operation type into account, that depends on the situs' geometry: For instance, in case of deep surgical wounds the lamps can stay almost statically above the situs while shallow wounds allow a much larger work area. This can be achieved by analyzing the situs' exact geometry and subsequently changing the occlusion map to ensure the lamps to remain in a smaller area. Another challenge is the real distribution of the lights. At the moment, we assume a uniform radius and luminosity for the lamps light beam. This is not necessarily valid for real operation lamps (See Figure 8), though the geometry of the beam is adjustable in most cases. A robotic arm with an additional degree of freedom for the position of the lamps should be able to fix this. The appropriate optimization parameter can be easily added to our cost function. Additional depth cameras that are mounted directly to the lamps could help to enhance the accuracy of the pipelines input data and enable us to determine directly whether a lamp has free sight onto the situs.

The performance of our pipeline is already capable for real-time: It runs with 20-25 Hz, which should be fast enough for most situations, but does not use the Kinect's full potential of 30 Hz. Our measurements show that the most time consuming part is the rendering of the occlusion map and especially, the download of the map from the GPU to main memory. However, our algorithm can be easily parallelized. Hence, we can implement the whole pipeline using modern GPU programming languages and avoid the copying from GPU memory completely. Actually, this would gain a big performance boost also to all other parts, including the optimization.

Another interesting part of this project is the possibility of using the obtained data for other features. For instance, the tracking of the surgeons hands might be useful in determining the stage of the operation with AI algorithms. We could use this information to reset the average occlusion and the activity map on the start of a new operation stage.

Finally, we plan to build a real hardware mock-up of to evaluate the acceptance of our system by surgeons in a user study.

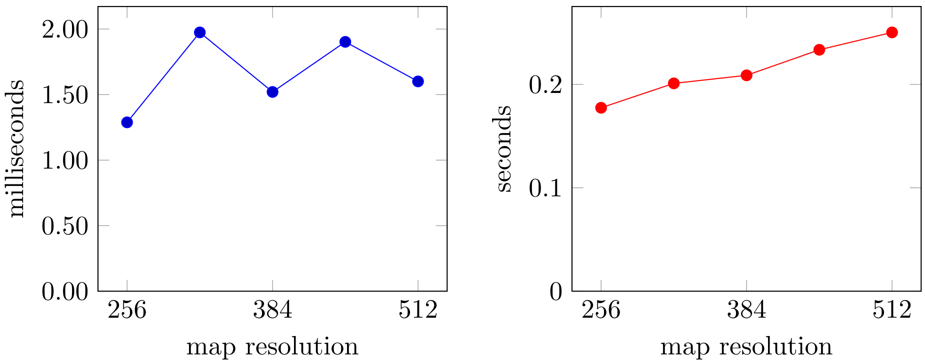

Figure 7: Time taken by the local (left) and global (right) optimum search on varying workspace map resolutions. Note the different scales on the time axis.

Figure 8: Converging-Diverging light beam of a typical surgical lamp. Source: [1]

5 Summary

In this paper, we presented the first method that allows for autonomous, optimized, continuous positioning of surgical lamps during open operations. It should be fairly easy to integrate into real products and install them in hospitals, because we use an inexpensive, off-the-shelf depth camera (e.g., the Kinect v2). This allows us to track and acquire all elements in the environment of the operating table, including the surgeon, surgical staff, and also the situs simultaneously. Based on this data, we have developed an optimization algorithm that operates directly on the depth images and, thus, does not require a time consuming polygonal reconstruction of the scene. The resulting new positions can be directly applied to the surgical lamps that can be mounted on robotic arms. Our algorithms are fast, robust, and easy to implement. The overall method can update the positions of three lamps with 20 Hz. We tested our algorithm with real-world data from an abdominal operation and the results show a distraction-free positioning of the surgical lamps.

6 Acknowledgements

This work was partially supported by the grand “Creative Unit – Intra Operative Information”. We would like to thank the Asklepios Klinik Barmbek and the International Neuroscience Institute for their support.

7 References

[1] U.S. Department of Energy, *Technical Guidance Document: LED Surgical Task Lighting — Commercial Buildings Resource Database*, August 2011. [Online]. Available: `https://buildingdata.energy.gov/cbrd/resource/1081`. [Accessed 17 June 2015].

[2] Arjan J Knulst, Rik Mooijweer, Frank W Jansen, Laurents PS Stassen, and Jenny Dankelman, *Indicating shortcomings in surgical lighting systems*, in Minimally Invasive Therapy & Allied Technologies, **vol. 20 (no. 5)** pp. 267–275 (2011).

[3] D. W. Davis, J. K. Brownell and S. C. Brownell, *Lighting system for medical procedures.* Patent US5526245 A, 11 June 1996.

[4] Y. Wang, C. S. Jordan, D. R. Uecker and C. C. Wooters, *General purpose distributed operating room control system.* Patent US6642836 B1, 4 November 2003.

[5] G. M. Kim and T. L. Chen, *Automated surgical illumination system.* Patent US20120259178 A1, 5 April 2012.

[6] S. Virga, V. Dogeanu, P. Fallavollita, R. Ghotbi, N. Navab and S. Demirci, *Optimal C-arm Positioning for Aortic Interventions*, in Bildverarbeitung fr die Medizin 2015, H. Handels, T. M. Deserno, H. Meinzer and T. Tolxdorff, Eds., Springer Berlin Heidelberg, pp. 53–58 (2015).

[7] M. Christie, R. Machap, J.-M. Normand, P. Olivier and J. Pickering, *Virtual camera planning: A survey*, in Smart Graphics (2005).

[8] Min, Dongbo, Jiangbo Lu, and Minh N. Do, *Depth video enhancement based on weighted mode filtering*, Image Processing, IEEE Transactions on, **vol. 21 (no. 3)**, pp. 1176-1190 (2012).

[9] Kim, S. Y., Cho, J. H., Koschan, A., and Abidi, M., *Spatial and temporal enhancement of depth images captured by a time-of-flight depth sensor*, Pattern Recognition (ICPR), 2010 20th International Conference on. IEEE, 2010.