Modeling the effect of relative camera pose differences on change detection tasks

Rafał Muszyński
IMEC-IPI-URC research group, TELIN
University of Ghent
Ghent, Belgium
Rafal.Muszynski@UGent.be

Hiep Luong
IMEC-IPI-URC research group, TELIN
University of Ghent
Ghent, Belgium
Hiep.Luong@UGent.be

***Abstract*—Change detection algorithms have to be robust to unrelated changes in the observed scene such as changing lighting conditions or shot noise. When the camera is not static, image registration is often performed as a preprocessing step. Unfortunately, image resampling performed during registration can introduce additional artifacts which are a source of false positive detections. We present an approach to modeling changes resulting from observing the scene from a different pose. We reformulate the general problem to only moving the camera further away from the target object and model the effect of observing the scene from the new viewpoint using the camera's point spread function (PSF). Our proposed model significantly outperforms the baseline approach and we are able to characterize the performance of change detection algorithms more reliably. The proposed model can be applied to guide the user in acquiring good quality images at runtime for change detection or to design constraints on the camera pose to ensure required accuracy of change detection algorithms.**

***Index Terms*—camera pose, image quality, change detection, detectability, point spread function (PSF)**

I. Introduction

Change detection is the core of many image processing tasks. Given two images of the same scene or object over time, the task is to find relevant differences between them. It is important to distinguish between significant changes in the photographed scene and changes caused by changing conditions that are not of interest such as different lighting conditions or random changes caused by shot noise.

In many applications the images are acquired by a handheld camera or a mobile sensing platform. Popular applications include environmental monitoring via remote sensing [1], medical imaging [2], [3], infrastructure inspection [4], rephotography [5]. Image registration techniques are then used to account for different camera poses between image captures by spatially aligning the images.

However, change detection techniques are limited by the information present in the original images, which is dependent on the scene and the camera pose. Therefore, with a static scene, differences in camera pose alone may lead to false positives in change detection. More precisely, the following effects have to be considered when the camera pose changes:

1) The perceived brightness depends on the light source position, camera pose and observed object properties which can be modeled by the bidirectional reflectance distribution function (BRDF) [6]. Especially challenging are glossy surfaces.
2) The resolution of the camera sensor is constant, but the size of the object on the image can change. Therefore in one of the images, less information about the object is captured and some details are lost. This can lead to detecting differences between images even if the scene did not change.
3) Image registration uses resampling to align the images. The resampling algorithm has an effect on the produced image and therefore on the change detection algorithm.
4) Registration errors
5) Depth of field can cause different parts of the images to be in focus while capturing images from different poses. Difference in sharpness causes loss of details.
6) Some lens effects, for example fisheye lens effect, vignetting, or Seidel aberrations [7], as well as image processing pipeline can cause local changes to the image. Changing the camera pose changes how parts of the scene are affected.
7) Parallax in 3D scenes.

Out of the mentioned effects, we chose to concentrate on the second one.

PSF models the response of an imaging system to a point source. It is influenced by different factors such as the lens or the camera sensor. Often it is used to measure image sharpness [8] or to improve image details by deconvolution with the PSF.

We develop an approach to simulating moving the camera further away from the target object and capturing the image. We base our approach on the PSF, which we use to accurately estimate and simulate image sharpness. We can use the developed method, to model differences between registered images, resulting from camera pose change, which in turn could be used to reduce the number of false positives in change detection.

Moreover, we are interested in how camera pose differences affect the performance of change detection algorithms. To this end, we propose a model relating camera pose differences with change detectability. This allows us to quantify the quality of a camera pose for a followup image, given a reference image.

The remainder of this paper is structured as follows: in

Section II we briefly go over related work, in Section III we describe our model in detail, in Section IV we describe performed experiments with our model, and in Sections V, VI we respectively discuss the obtained results and draw final conclusions.

II. Related work

Change detection is a well studied problem in image processing applications. In the context of our work, especially interesting are the recent works [9], [10]. In [9], super resolution is used to upscale a low resolution image before comparison with a high resolution image. Super resolution recreates some of the details, that would normally be missing from the upscaled image. Authors of [10] use registration parameters as another input to the change detection performing neural network. This information can be potentially used by the network to reduce the number of false positives.

In our work we assume, both images are captured with the same camera. In contrast, [3] explores problems with change detection when working with multiple modern mobile phones. While we analyze how the camera pose change affects the captured image details, authors of [3] concentrate on measuring image sharpness, spatial resolution, effects of sharpening algorithms, and difference in color reproduction between different phone models.

In [8], the PSF is used to characterize the performance of edge detection algorithms. Authors propose a method of estimating the PSF from a single image, which can be used with our algorithm for simulating image capture.

Our work on detectability is inspired by [11], [12]. Here, the detectability is used as a measure to optimize image quality in the task of denoising and image reconstruction in MRI.

Modern deep learning methods for super resolution require a training dataset consisting of pairs of high and low resolution photos of the same scene. Collecting such pairs is time-consuming. Preferably, synthetic data can be used instead. Until recently, synthetic datasets were created by downscaling high resolution images using bicubic resampling. As a result, models trained on synthetic datasets did not perform well in real world tasks. Many authors [13]–[15] improved over previous state of the art models by introducing more complex image degradation schemes. Our problem, of simulating observing an image from further away is closely related to the problem of image quality degradation for generating realistic super resolution datasets.

III. Model

In this section we explain our approach to model some of the changes in perceived scene caused by camera pose difference.

A. Modeling effects of camera pose difference

A camera can move with 6 degrees of freedom - 3 translational directions, and 3 rotations. Our goal is to model how any pose change influences the change detection problem. We will first analyze a scene consisting of a 2D surface, and then we will see how it generalizes to a 3D scene.

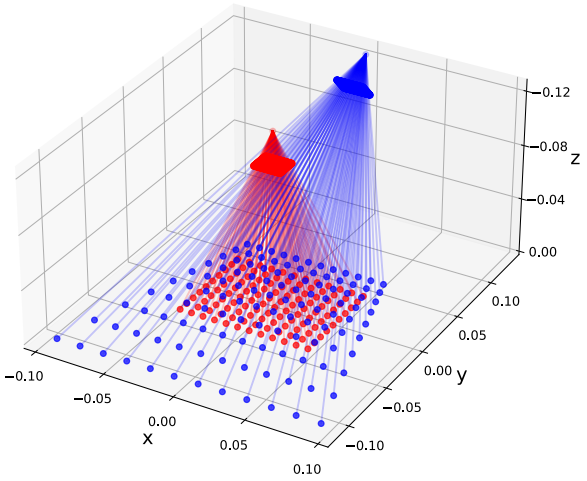

Fig. 1: The simplest scenario consisting of a 2D scene and a camera. When the camera pose changes, the target object's surface is sampled with different densities.

Fig. 1 depicts a simple scene, with ray casts through the camera sensor to the surface. Two images are taken of this scene. For the first image, the camera is positioned above the scene, so that the camera sensor is parallel to the scene surface.

We can see that the local sampling density changes. When a scene is a 2D plane, this sampling density change can be calculated using classical registration techniques and a homography. More complex movement models can be used to handle 3D scenes as well as lens distortions. Therefore, instead of reasoning about camera pose change in general, we can instead analyze moving the camera closer or further from the target surface, as this is enough to model all possible scene sampling densities change. We can focus on modeling only moving the camera further away (the camera zoom-out), as the case is symmetric when the camera gets closer to the surface.

B. Modeling effects of camera zoom out

In this subsection, we examine the relationship between a reference image, and a followup image captured from further away. We then propose a way to simulate the followup image, using the reference image as an input.

First we note, that image sharpness should be constant across zoom-outs, as sharpness of an in focus system only depends on the PSF. The PSF can be calculated through a camera calibration process or estimated from an image as in [8].

Equation (1) presents our model of image capture.

$$\boldsymbol{c_z} = \boldsymbol{b_\Sigma} * \boldsymbol{d_z} \tag{1}$$

Where $\boldsymbol{c_z}$ is an image captured from a pose $z \geq 1$ times further away from the target than the reference pose. Case $z = 1$ denotes the reference image. Lens blur is modeled as a convolution with the PSF ($\boldsymbol{b_\Sigma}$), and $\boldsymbol{d_z}$ is the downsampled hypothetical ideal image of the scene, without lens blur effects. Similar to [8], we model the PSF as a 2-dimensional Gaussian

curve, but we simplify the covariance to a diagonal matrix $\Sigma = \left(\begin{smallmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{smallmatrix}\right)$.

We observed that the downsampled reference image is sharper than the followup image. More specifically, we observed that resizing an image using bicubic resampling, by a factor $\frac{1}{z}$, changes the lens blur effect proportionally. Equation (2) presents this observation. r_z denotes the bicubic downsampling by a factor z.

$$r_z(\boldsymbol{c_1}) \approx \boldsymbol{b}_{\frac{\Sigma}{z^2}} * \boldsymbol{d_z} \tag{2}$$

Knowing the PSF, this observation is sufficient to correct the sharpness of the downsampled reference image. Based on (2), we can estimate $\boldsymbol{d_z}$ by performing a deconvolution, and then obtain $\boldsymbol{c_z}$ by a convolution with the PSF as in (1). Because we model the PSF as a Gaussian, those two operations can be combined into a single Gaussian kernel, with covariance $\Sigma' = \Sigma - \frac{\Sigma}{z^2}$. Equation (3) presents the final formula for simulating the followup image.

$$\boldsymbol{c_z} \approx \boldsymbol{b}_{\boldsymbol{\Sigma'}} * r_z(\boldsymbol{c_1}) \tag{3}$$

IV. Experimental results

In this section, we summarize the results of the conducted experiments. We describe our setup and explain how we calibrated the camera. Then, we compare our simulated zoomed out images with experimentally captured images. Finally, we use our model to estimate how detectability changes when the camera moves away from the target surface.

For the experiments we used an Allied Vision Manta G-046 machine vision camera, with a LMVZ4411 Kowa Lens 1/1.8" 4.4-11mm and a stand allowing steadily mounting a camera at different heights. The image resolution is 780x580. The camera is pointed down towards a target object. To reduce noise, each image is an averaging over 10 captures. We set the camera focus manually, with a help of squared gradient metric [16].

A. Camera calibration

We model the PSF as a Gaussian kernel with a diagonal covariance matrix $\left(\begin{smallmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{smallmatrix}\right)$. This is a simplified model. We calibrate σ_x and σ_y based on line spread function (LSF) estimations in horizontal and vertical directions. For the LSF estimation, we created a sharp edge using a dark and a white piece of paper. We placed the pieces unevenly, one on top of another, which created the edge. We took pictures of multiple rotations of the created edge, with the camera at different heights. We divided the captured images into 3 categories: vertical, horizontal and diagonal edge. Images with a diagonal edge were not used. We fitted a Gaussian curve to each LSF. Finally we estimate σ_x and σ_y as the median of fitted Gaussian standard deviations in appropriate directions.

B. Model validation

To validate our model, we test how well can we simulate taking an image from further away based on the reference image. We compare our approach with KernelGAN [17] used by [13]. For each camera zoom-out and reference image we trained a KernelGAN model and used it to downsample the reference image. We increased the number of training iterations from 3000 to 4000, which improved the results. Downsampling is achieved by blurring the input image with the trained kernel followed by nearest neighbor resampling. We based our implementation on the official code of the project [18]. We also include a simple baseline in our comparison, which simulates camera zoom-out with bicubic downsampling. For all of the resamplings we used the Pillow library [19].

method	correlation	MSE	normalized mutual information	PSNR	SSIM
Baseline	0.0059	22863	0.0256	53.8	0.062
KernelGAN	0.0017	5598	0.0062	16.1	0.013
Our model	**0.0004**	**2320**	**0.0052**	**9.7**	**0.005**

TABLE I: Mean square error between experimental and simulated curves, for each metric, averaged over all data points. From the tested models, ours is the best at modeling the relationship between zoom-out and artifacts.

We took pictures of different textures: a carpet, a printed image of a skin lesion and a printed calibration chart ISO_12233. Each object was captured multiple times with the camera mounted at different heights. For each texture, we selected 2 poses as a reference, and simulated image capture from further away based on them.

We only chose to model a single effect (Section I), but in the real world all of them influence the images. In our experimental data we noticed changes in the observed object brightness caused by the BRDF, shadows cast by the camera and changing exposure time. Moreover, due to lens effects, the image sharpness is not uniform and decreases gradually from the image center to the edges. Our model concerns changes in high frequencies, and the PSF is calibrated for the center of the image. Therefore, to evaluate our model, we apply processing steps to the images to reduce the effects outside of the scope of our model. First, we align the input image with the reference frame. For experimental data we use the `findHomography()` function from OpenCV [20] with SIFT [21] keypoints. Simulated images are simply upscaled. Then, we perform brightness normalization to a [0,255] range. Next, we remove low frequencies from the images, by applying a low-pass filter (Gaussian blur with $\sigma = 10$) and subtracting the filtered image from the input. Finally, we crop the center of the image, so that the original dimensions are halved.

Fig. 2 presents the preprocessed experimental and simulated images, with even smaller central part of the image cropped for visualization. The calibration chart (first two columns) is the most challenging target. Our model reproduces the artifacts only partially in that case. Overall, our method and KernelGAN produce similar results, while being much more realistic then the baseline approach.

Fig. 3 presents a quantitative comparison, for selected cases.

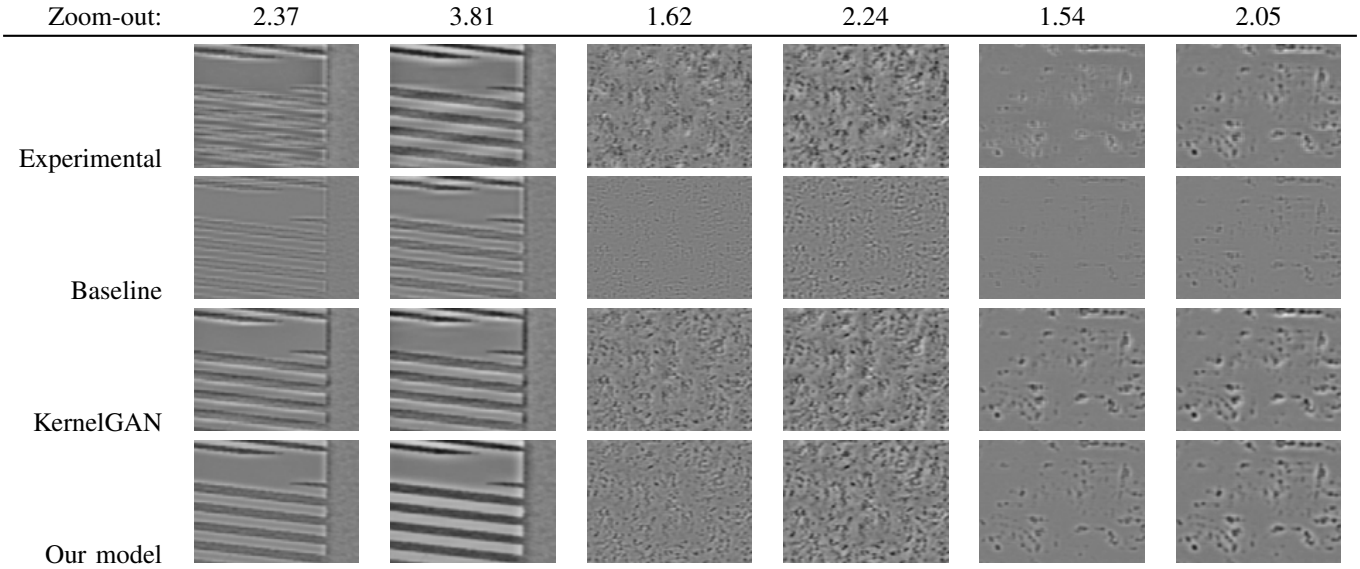

Fig. 2: Processed experimental and simulated images of various textures at different camera zoom-outs. The images are aligned with the reference image, the brightness is normalized, low frequencies are removed and the images are cropped. The image of the calibration chart (first two columns) is the most challenging target. Overall, our method achieves similar results to KernelGAN and outperforms the baseline approach.

model	$\overline{\text{correlation}}$	$\overline{\text{MSE}}$	$\overline{\text{normalized mutual information}}$	$\overline{\text{PSNR}}$	$\overline{\text{SSIM}}$
Baseline	0.953	86	1.256	36.6	0.733
KernelGAN	0.944	104	1.213	34.8	0.677
Our model	**0.968**	**64**	**1.275**	**37.8**	**0.760**

TABLE II: Results of comparing processed experimental and simulated images from different models. We compare each image pair and average the metrics over the whole dataset. Our model achieves the best results across all metrics.

We compared the reference images with images simulated using different models, and experimental images separately, to test whether we can simulate artifacts of similar properties across different zoom-outs. We compared the images across multiple metrics: PSNR, MSE, normalized mutual information, correlation and SSIM. The differences resulting from our model manage to follow a similar trend as the experimental data. Moreover, our model significantly outperforms the baseline approach. In this test, KernelGAN achieves good results in some cases, but is not consistent across zoom-outs. The instability might be caused by the relatively small resolution of the input images, which limits the training dataset of the model. Possibly, even longer training could solve this issue. We tried to improve stability, by training the model only on the center crops of the images, where sharpness is consistent, but the results were worse. Table I summarizes the experiment in terms of mean squared error calculated between experimental and simulated curves. The error is averaged over all experiments. Our model is the best at capturing the relationship between zoom-out and artifacts.

We also compared the processed experimental images with simulated counterparts directly. In Table II we present multiple metrics averaged over the whole dataset. Our model achieves the best results across all metrics.

C. Modeling detectability

Using the camera calibration data and our model, we calculate the detectability similarly as in [11]. We simulate a real life scenario, where we compare two images, $\boldsymbol{c'}$ and $\boldsymbol{c''}$, captured at different times. Before capturing $\boldsymbol{c''}$, the camera is moved further away from the target object. We start with an image $\boldsymbol{c}$, which contains all the information about the scene. The reference image $\boldsymbol{c'}$ has half the resolution of $\boldsymbol{c}$, and is obtained by simulating double camera zoom-out. We then add a known change signal to the $\boldsymbol{c}$ image, and simulate camera zoom-out, relative to the reference image, to obtain $\boldsymbol{c''}$. Gaussian noise defined by σ_{noise} is added to both images before aligning $\boldsymbol{c''}$ with the reference image and calculating the difference image. We model the resulting change signal, and use statistical tests to determine whether the signal is still distinguishable from noise, in the difference image. For this test, we used selected images from the HAM10000 skin lesion dataset [22].

As in the previous test, we compare our method with the baseline approach which simulates camera zoom-out with bicubic downsampling.

Fig. 4 presents the results of our experiments. The test signal is a symmetric Gaussian signal as in [11], with a $\sigma_{\bar{x}} = 2$ and contrast $= 12$. We see that if we use our model instead of the baseline approach, the detectability is much lower. This is because, with our model, the artifacts in the difference image resulting from camera pose changes, are stronger. Because

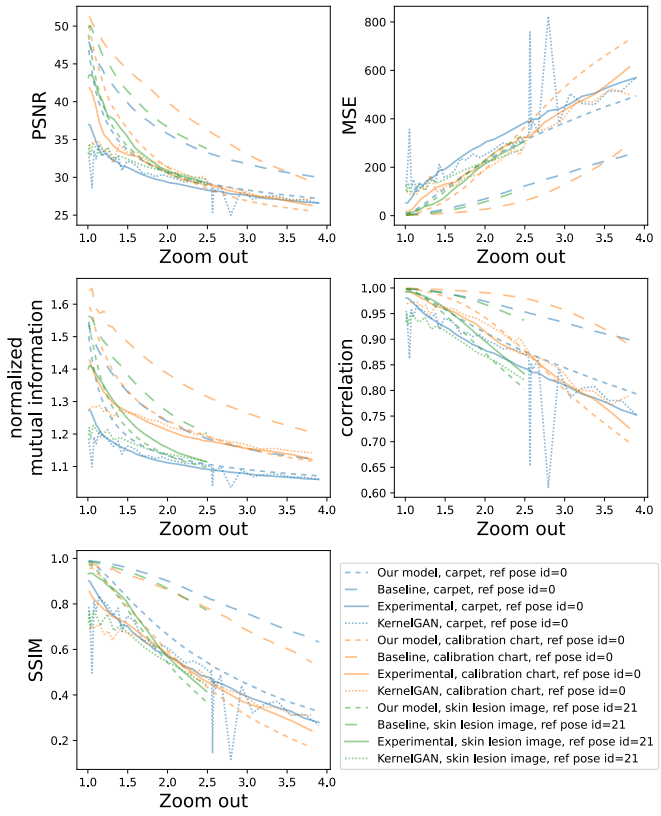

Fig. 3: Comparing reference frames to experimental and simulated results. Images are warped to the reference, normalized, low frequencies are removed, finally the images are cropped before comparison. We tested multiple target objects and on all of them our model successfully follows the trend observed in the experimental data. Our approach outperforms the baseline model. KernelGAN achieves good results in many cases but is inconsistent.

our new approach is more accurate, the obtained detectability curves should better model real life results.

Fig. 5 presents what happens to the detectability when our test signal gets stronger. We kept the noise $\sigma_{\text{noise}} = 2$ and test signal $\sigma_{\bar{x}} = 2$ in this experiment. The graph shows how we can use our model to determine restrictions on the relative camera pose differences depending on system requirements.

V. Discussion

In this section we further discuss the results of our work and talk about possible applications.

As shown in the previous section, our model does not produce results that match the experimental data completely, but only provides an upper bound on the image quality. Although our model is very simple, it is a significant improvement over naive approaches. Once calibrated, our model can be used reliably across different zoom-outs, and on average achieves better results than KernelGAN.

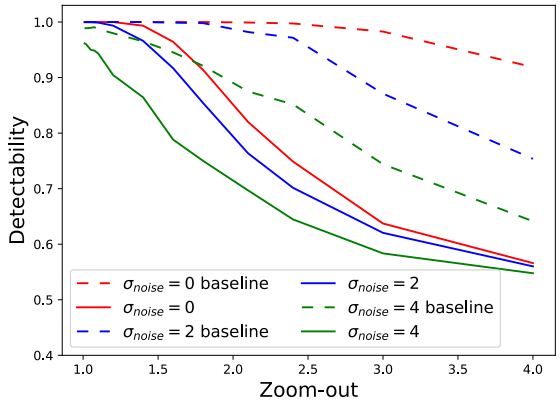

Fig. 4: Detectability as a function of zoom-out. The test signal is a symmetric Gaussian with a $\sigma_{\bar{x}} = 2$ and contrast $= 12$. We compare baseline results of using a bicubic downscaling vs simulating zoom-out with our method. We tested different levels of noise in the images.

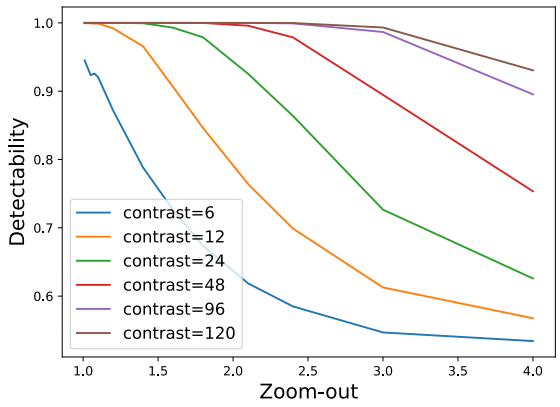

Fig. 5: Figure shows how test signal strength affects the relationship between camera zoom-out and detectability. For all cases $\sigma_{\text{noise}} = 2$ and test signal $\sigma_{\bar{x}} = 2$.

A. Application #1: real-time camera pose quality assessment

Given a reference image, a user adjusts the camera pose to capture a followup image. The end goal is to find changes between the two images. The more freedom the user has with capturing the followup image, the more user-friendly the system is, but at the same time it has to handle a bigger range of camera poses.

For this use case we can calculate the camera zoom-out - detectability curve in advance. This should be possible if the camera's PSF can be calibrated beforehand. At runtime, we can infer the relative pose differences. Based on the local sampling density change, we can model local detectability. A followup camera pose is then accepted or rejected based on an aggregated detectability criterion, for example thresholding minimum detectability.

To estimate detectability, aspects such as relevant change characteristics, scene contrast, camera resolution, and image capture noise levels have to be taken into account. The exact parameters are application specific.

Concrete examples: telemedicine in dermatology, industrial inspections, satellite / drone images analysis.

B. Application #2: determining change detection system constraints

In this task, we would like to calculate the camera movement range during a system design process. The only difference between the first application is that this time the camera motion is not known. This is not a problem if we can approximate the scene using a 2D surface and can describe the relationship between the reference and followup images using a homography. In such case, we can simulate the local sampling density change for any 2 poses of the camera relative to the surface.

C. Application #3: detecting false positives in change detection

During the change detection process, we compare a reference image with an aligned followup image. The difference image, will contain relevant and irrelevant changes between those images. Some of the irrelevant changes, are a result of camera pose change. We can detect them, by simulating the followup image with our model, aligning it and calculating the difference image by comparing it with the reference image. Because the simulated difference image contains only irrelevant changes, we can use it to filter out some of the irrelevant changes in the difference image.

D. Application #4: training super resolution models

As discussed in Section II, image degradation is an important problem in super resolution research. In future work, our model could be used to generate a training data specific to a chosen target device, with a calibrated PSF.

VI. Conclusions

We analyzed the selected effects of relative camera pose differences on the change detection problem. Our model provides an upper bound for the quality of the camera pose for the followup image, which can be used during system design or at runtime. Moreover, our model has the potential to improve change detection accuracy or super resolution synthetic datasets generation.

Our model can be further improved by more accurate camera capture simulations. From our perspective, effects of image processing algorithms such as sharpening, improved camera lens simulations and incorporating a BRDF model are the most interesting.

References

[1] K. S. Willis, "Remote sensing change detection for ecological monitoring in United States protected areas," *Biological Conservation*, vol. 182, pp. 233–242, Feb. 2015. https://doi.org/10.1016/j.biocon.2014.12.006

[2] B. Hibler, Q. Qi, and A. Rossi, "Current state of imaging in dermatology," *Seminars in Cutaneous Medicine and Surgery*, vol. 35, no. 1, pp. 2–8, Mar. 2016. https://doi.org/10.12788/j.sder.2016.001

[3] B. Dugonik, A. Dugonik, M. Marovt, and M. Golob, "Image Quality Assessment of Digital Image Capturing Devices for Melanoma Detection," *Applied Sciences*, vol. 10, no. 8, p. 2876, Jan. 2020, number: 8 Publisher: Multidisciplinary Digital Publishing Institute. https://doi.org/10.3390/app10082876

[4] K. Gopalakrishnan, H. Gholami, A. Vidyadharan, A. Choudhary, and A. Agrawal, "Crack damage detection in unmanned aerial vehicle images of civil infrastructure using pre-trained deep learning model," *International Journal for Traffic and Transport Engineering (IJTTE)*, vol. 8, pp. 1–14, Feb. 2018. https://doi.org/10.7708/ijtte.2018.8(1).01

[5] S. Bae, A. Agarwala, and F. Durand, "Computational rephotography," *ACM Transactions on Graphics*, vol. 29, no. 3, pp. 1–15, Jun. 2010. https://doi.org/10.1145/1805964.1805968

[6] F. E. Nicodemus, "Directional Reflectance and Emissivity of an Opaque Surface," *Applied Optics*, vol. 4, no. 7, pp. 767–775, Jul. 1965, publisher: Optica Publishing Group. https://doi.org/10.1364/AO.4.000767

[7] Chris G. Berger, "Seidel Aberration Imaging, Optics: The Website," accessed on 5/1/2023. https://www.opticsthewebsite.com/SeidelSimulation

[8] M. Luxen and W. Forstner, "Characterizing image quality: Blind estimation of the point spread function from a single image," *Proc. Photogrammetric Computer Vision*, pp. 205–210, 01 2002.

[9] J. Tian, D. Peng, H. Guan, and H. Ding, "RACDNet: Resolution- and Alignment-Aware Change Detection Network for Optical Remote Sensing Imagery," *Remote Sensing*, vol. 14, p. 4527, Sep. 2022. https://doi.org/10.3390/rs14184527

[10] S. Bu, Q. Li, P. Han, P. Leng, and K. Li, "Mask-CDNet: A mask based pixel change detection network," *Neurocomputing*, vol. 378, pp. 166–178, Feb. 2020. https://doi.org/10.1016/j.neucom.2019.10.022

[11] B. Goossens, H. Luong, L. Platisa, and W. Philips, "Optimizing image quality using test signals: Trading off blur, noise and contrast," in *2012 Fourth International Workshop on Quality of Multimedia Experience*. Melbourne, Australia: IEEE, Jul. 2012, pp. 260–265. https://doi.org/10.1109/QoMEX.2012.6263867

[12] H. Luong, B. Goossens, J. Aelterman, L. Platiša, and W. Philips, "Optimizing image quality in MRI: On the evaluation of k-space trajectories for under-sampled MR acquisition," in *2012 Fourth International Workshop on Quality of Multimedia Experience*, Jul. 2012, pp. 25–26. https://doi.org/10.1109/QoMEX.2012.6263881

[13] X. Ji, Y. Cao, Y. Tai, C. Wang, J. Li, and F. Huang, "Real-world super-resolution via kernel estimation and noise injection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 1914–1923. https://doi.org/10.1109/CVPRW50498.2020.00241

[14] X. Wang, L. Xie, C. Dong, and Y. Shan, "Real-esrgan: Training real-world blind super-resolution with pure synthetic data," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2021, pp. 1905–1914. https://doi.org/10.1109/10.1109/ICCVW54120.2021.00217

[15] A. Bulat, J. Yang, and G. Tzimiropoulos, "To learn image super-resolution, use a GAN to learn how to do image degradation first," Jul. 2018, arXiv:1807.11458 [cs].

[16] F. C. Groen, I. T. Young, and G. Ligthart, "A comparison of different focus functions for use in autofocus algorithms," *Cytometry*, vol. 6, no. 2, pp. 81–91, Mar. 1985. https://doi.org/10.1002/cyto.990060202

[17] S. Bell-Kligler, A. Shocher, and M. Irani, "Blind Super-Resolution Kernel Estimation using an Internal-GAN," Jan. 2020, arXiv:1909.06581 [cs].

[18] S. Kligler, "Blind Super-Resolution Kernel Estimation using an Internal-GAN," Jan. 2023, original-date: 2019-08-13T08:15:01Z. https://github.com/sefibk/KernelGAN

[19] A. Clark, "Pillow (PIL Fork) Documentation." https://pillow.readthedocs.io/en/stable/

[20] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[21] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004. https://doi.org/10.1023/B:VISI.0000029664.99615.94

[22] P. Tschandl, "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," 2018. https://doi.org/10.7910/DVN/DBW86T