Star-Effect Simulation for Photography Using Self-calibrated Stereo Vision

Dongwei Liu[1(✉)], Haokun Geng[1], and Reinhard Klette[2]

[1] Department of Computer Science, The University of Auckland, Auckland, New Zealand
dliu697@aucklanduni.ac.nz
[2] School of Engineering, Auckland University of Technology, Auckland, New Zealand

Abstract. Star effects are an important design factor for night photos. Progress in imaging technologies made it possible that night photos can be taken free-hand. For such camera settings, star effects are not achievable. We present a star-effect simulation method based on self-calibrated stereo vision. Given an uncalibrated stereo pair (i.e. a base image and a match image), which can be just two photos taken with a mobile phone with about the same pose, we follow a standard routine: Extract a family of feature-point pairs, calibrate the stereo pair by using the feature-point pairs, and obtain depth information by stereo matching. We detect highlight regions in the base image, estimate the luminance according to available depth information, and, finally, render star patterns with an input texture. Experiments show that our results are similar to real-world star effect photos, and that they are more natural than results of existing commercial applications. The paper reports for the first time research on automatically simulating photo-realistic star effects.

Keywords: Star effect · Computational photography · Stereo vision · Self-calibration

1 Introduction

Night photos with compelling lighting show sometimes star patterns around highlights, known as *star effect* in photography. Such star patterns are often essential for defining the aesthetic meaning of night photographs.

Recently, an increasing number of photos are taken with compact cameras or mobile phones. The technological progress (e.g. large aperture lenses or high-sensitivity image sensors) made it possible to shoot night photos free-hand.

Normally, to obtain a star effect, a photo has to be shot with a small aperture (e.g., f22). Such an aperture setting is unfit for free-hand shooting because it requires several seconds of exposure time; the photo would be largely blurred by hand shake. Alternatively, star effects can be achieved with a *star filter* in front of the lens. This is also not really appropriate for casual shooting with compact cameras or mobile phones because special equipment is needed. See Sect. 2.1 for

T. Bräunl et al. (Eds.): PSIVT 2015, LNCS 9431, pp. 228–240, 2016.
DOI: 10.1007/978-3-319-29451-3_19

details about generating star effects in photography. The question arises: How to obtain a star effect in case of casual hand-held photography?

Generating star effects by post-processing is a possible way, for example by adding star effects manually using available photo-editing applications. This might be a time consuming process for a complex scene because the size and color of each star should be carefully set, one by one. Professional skills are also needed. Dedicated applications such as *Topaz Star Effects plugins*, or instructions given in some online tutorial videos [7,17], draw uniform-sized star patterns to all the overexposed regions of a photo. Though it may also look beautiful, such effects typically look unnatural and different to the star effect generated by a small aperture. It is desirable to have an automatic star-effect generator which corresponds to the actual content of a photo.

Research on computational photography [16] explores the simulation of various photo effects such as fog [12], bokeh [13], or high dynamic range photos [23]. With this paper we present first time research on automatically simulating photo-realistic star effects.

This paper presents an automatic method for adding star effects to a photo using self-calibrated stereo vision. Given an uncalibrated stereo pair (e.g. two pictures of the same scene taken with a mobile phone at about "near-parallel" poses), we first detect a family of feature point-pairs and do a self-calibration. By stereo matching we estimate depth information of the scene. Next, "star-capable" highlights are detected, and luminance and color information (clipped by over-exposure) is recovered according to the available depth information. Finally, we render star patterns considering the luminance and color of the highlights using an input templet.

The rest of the paper is structured as follows. Section 2 briefly recalls theories and techniques related to our work. In Sect. 3 we provide details for our self-calibrated depth-estimation method. Section 4 discusses our method for the detection of highlights and the estimation of luminance and color of highlights. Then, Sect. 5 discusses our star effect rendering method. Experimental results are shown in Sect. 6. Section 7 concludes.

2 Basic and Notation

This section lists notation and techniques used. RGB color images I are defined on a rectangular set Ω of pixel locations, with $I(p) = (R(p), G(p), B(p))$ for $p \in \Omega$ and $0 \leq R(p), G(p), B(p) \leq G_{\max}$. Let N_{cols} and N_{rows} be width and height of Ω, respectively. Position O be the center of Ω (as an approximation of the principle point). We suppose lens distortion has been corrected in cameras, otherwise we can correct it by using the lens profile provided by the manufacturer. Altogether, we consider input image I as being generated by undistorted central projection. In the following, a *star-effect photo* refers to a photo taken of a night scene with a very small aperture, in which star patterns appear around highlights.

2.1 Star Effects in Photography

A star effect is normally caused by *Fraunhofer diffraction* [2]. Such a phenomenon is most visible when bright light from a "nearly infinite" distance passes through a narrow slit, causing the light to spread perpendicular to the slit. This spreads a point-like beam of light into a pair of streaks. Suppose a rectangular aperture A with width w_1 and height w_2, located in an x_1x_2 plane having its origin at the centroid of A; axes x_1 and x_2 are parallel to the edges with width w_1 and w_2, respectively. Suppose an axis y perpendicular to the x_1x_2 plane. When A is illuminated by a monochromatic plane wave of wavelength λ, the intensity $I(\theta_1, \theta_2)$ of the light through A forms a pattern that is described by

$$I(\theta_1, \theta_2) \propto \operatorname{sinc}^2\left(\frac{\pi w_1 \sin\theta_1}{\lambda}\right) \operatorname{sinc}^2\left(\frac{\pi w_2 \sin\theta_2}{\lambda}\right) \tag{1}$$

with $\tan\theta_1 = x_1/y$ and $\tan\theta_2 = x_2/y$. Similarly, for a parallelepiped aperture with edge lengths w_1, w_1, and w_1, we have a pattern defined by

$$I(\theta_1, \theta_2, \theta_3) \propto \operatorname{sinc}^2\left(\frac{\pi w_1 \sin\theta_1}{\lambda}\right) \operatorname{sinc}^2\left(\frac{\pi w_2 \sin\theta_2}{\lambda}\right) \operatorname{sinc}^2\left(\frac{\pi w_3 \sin\theta_3}{\lambda}\right) \tag{2}$$

Fig. 1. Relationship between aperture shape and star effect. From left to right: Five straight blades aperture, eight straight blades aperture, eight curved blades aperture, nine curved blades aperture, and a circular aperture with a star filter

Figure 1 shows some real-world star patterns shot with different aperture shapes. It can be seen that each blade of the aperture contributes a pair of streaks to the star pattern. Due to overlapping, in case of a lens with an even number a of blades, generated star patterns have the same number a of streaks.

In fact, every beam of light that goes to the image sensor through a small aperture, spreads a star pattern to its neighboring region due to diffraction. In normal-contrast regions of an image, the star pattern is very slight and not visible (it only causes a reduction of local sharpness). Only in very high-contrast cases, such as highlights in a night photo, the star pattern is noticeable.

As described above, the shape of the star pattern depends on the shape of the aperture, which is not a user-controllable parameter. A modern-designed lens normally uses a near-circular aperture (in order to obtain good *bokeh* quality [13]), which generates only weak and scattered streaks. Our method provides controllable star pattern styles, which is convenient for photo-art creation.

A *star filter* can be used to simulate a star effect as introduced by small aperture. Such filter embeds a very fine diffraction grating or some prisms. A star effect can be obtained even at large aperture when using such a filter in front of the lens. The star pattern generated by a star filter is visually a little different from that one generated by a small aperture. Our method can simulate both styles by introducing different templates.

2.2 Stereo Vision

For stereo vision, we take a *stereo pair* (i.e. a *base image* I_b and a *match image* I_m) as input, and estimate depth information for a scene by searching for corresponding pixels in the stereo pair [10]. For reducing the complexity, the given stereo pair is normally geometrically rectified into *canonical stereo geometry* [10], in which I_b and I_m are as taken by identical cameras, only differing by translation distance b along the *base line* on the X-axis of the camera coordinate system of I_b. Rectification transform matrices $\mathbf{R}_b$ and $\mathbf{R}_m$ can be calculated in a stereo calibration process. Our proposed method is not very sensitive to accuracy of available depth values. For convenience, we use a self-calibration method with only a few assumed camera parameters. After rectification, a stereo matching method such as belief-propagation [3], or semi-global matching [5,6], can be used for calculating a disparity map. Depth information is then obtained by using standard *triangulation*.

Self-calibrated stereo vision is a convenient and easy to achieve setting for obtaining depth information of a scene (i.e. an image), compared to other depth sensors such as *structured lighting* as used in the *Kinect* [8], *depth from defocus (DFD)* [20], a *light-field camera* [18], or 3D laser imaging systems [1].

3 Depth Estimation

Given an uncalibrated stereo pair I_b and I_m, we do self-calibration by detecting a family of feature pairs, then we rectify the given stereo pair and obtain a disparity map through stereo matching. The disparity map is warped into the coordinates of the unrectified image I_b for avoiding any composition damage.

Feature detection and matching is the key step of the self-calibration method. A number of mainstream feature detectors are evaluated by [21], which suggests that the *oriented BRIEF* (ORB) (see [19] for an implementation) detector appears to be reasonably rotation invariant and noise resistant by also ensuring time efficiency.

First of all, two families of features $\mathcal{F}_b$ and $\mathcal{F}_m$ are collected from I_b and I_m using ORB. An ideal case would be that all the features are uniformly distributed in all the interesting regions of the input images.

The k-nearest neighbors (kNN) method is then used to detect best matches $\mathcal{F}_{\text{pairs}}$ between the two families of features. The feature pairs are filtered by a distance ratio test [15] and a cross-symmetry test:

$$\forall [p_b^{(i)}, p_m^{(i)}] \big(\; [p_b^{(i)}, p_m^{(i)}] \in \mathcal{F}_{\text{filtered}} \leftrightarrow [p_b^{(i)}, p_m^{(i)}] \in \mathcal{F}_{\text{pairs}}$$
$$\wedge \, \delta_1(p_b^{(i)}) < T_{\text{distance}} \cdot \delta_2(p_b^{(i)})$$
$$\wedge \, \|x_b^{(i)} - x_m^{(i)}\|_2 < T_{\text{cross,x}} \wedge \|y_b^{(i)} - y_m^{(i)}\|_2 < T_{\text{cross,y}} \; \big) \qquad (3)$$

where $\delta_1(p_b^{(i)})$ is the similarity distance to the closest neighbor of $p_b^{(i)}$ in kNN matching, and $\delta_2(p_b^{(i)})$ is the similarity distance to the second-closest neighbor. We use a ratio $T_{\text{distance}} = 0.8$. According to the statistics in [15], such a threshold eliminates 90 % of the false matches while discarding less than 5 % of the correct matches. $T_{\text{cross,x}}$ and $T_{\text{cross,y}}$ specify the tolerance thresholds. We use $T_{\text{cross,x}} = 0.15 \cdot N_{\text{cols}}$ and $T_{\text{cross,y}} = 0.01 \cdot N_{\text{rows}}$. Figure 2 shows results of the feature matching phase.

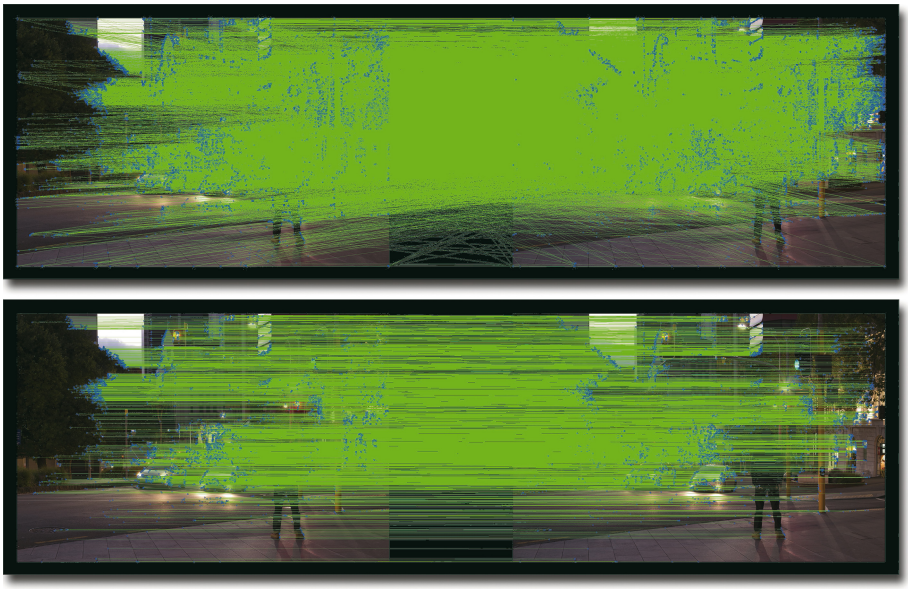

Fig. 2. Matched feature points before and after applying our filter

We use a *random sample consensus* (RANSAC) algorithm to calculate a fundamental matrix $\mathbf{F}$ according to $\mathcal{F}_{\text{filtered}}$. Both images I_b and I_m have been taken with the same camera. Thus, we do not need to create identical twins of cameras (as in the general stereo imaging case). Homography matrices $\mathbf{H}_b$ and $\mathbf{H}_m$ can be computed by using $\mathcal{F}_{\text{filtered}}$ and $\mathbf{F}$ [4]; those matrices transform I_b and I_m into planar perspective views $\hat{I}_b$ and $\hat{I}_m$.

We run an SGM stereo matcher [5] on $\hat{I}_b$ and $\hat{I}_m$, and obtain a disparity map $\hat{d}$ defined on the carrier $\hat{\Omega}$ of $\hat{I}_b$.

Note that without knowing the camera matrix and the relative position of the cameras, the transforms $\mathbf{H}_b$ and $\mathbf{H}_m$ obtained from self-calibration, are not standard rectification transforms. The rectified image $\hat{I}_b$ might be "largely warped",

Fig. 3. *Top*: A rectified image pair $\hat{I}_b$ and $\hat{I}_m$. *Bottom, left to right*: The resulting disparity map $\hat{d}$, the inversely transformed disparity map d, and the joint, bilaterally filtered result.

as shown in Fig. 3, and thus distort the attempted composition. We transform $\hat{d}$ into d according to the inverse transform H_b^{-1}.

A joint bilateral filter [9] is then applied for removing noise from d. See Fig. 3, bottom, for a result after inverse transformation and filtering of the disparity map. Because of lack of texture or limited availability of disparities, disparity information close to the border of the image might be incorrect. This does not affect our application because we only use disparity at highlights.

4 Highlight Registration

As mentioned in Sect. 2.1, star patterns appear only in very high contrast regions, typically at highlights in a night photo. Due to limitations of the dynamic range of the image sensor or of the image file format, luminance and color information in such regions normally gets clipped. We detect these regions in the original image I_b, and estimate the luminance and color information according to depth information and values at adjacent pixels.

4.1 Highlight Detection

To locate valid highlights, we first detect a family $\mathcal{F}_{\text{overexposed}}$ of 4-connected regions S in I_b which are formed by over-exposed pixels, satisfying

$$R_b(p) = G_b(p) = B_b(p) = G_{\max} \tag{4}$$

at all $p \in S$. Some regions in $\mathcal{F}_{\text{overexposed}}$ cannot be considered as a possible source for a star pattern, as occurring in real-world star-effect photos.

For example, overexposed but non-luminous regions such as some metallic patches at buildings, large overexposed regions including the sky (e.g. with a backlight) or a white wall, or very small overexposed regions which are normally just local reflections (e.g. of eyes of a cat) or noisy pixels. To eliminate these items, we extract a subfamily $\mathcal{F}_{\text{highlight}} \subset \mathcal{F}_{\text{overexposed}}$ such that

$$\forall S\,[S \in \mathcal{F}_{\text{highlight}} \leftrightarrow S \in \mathcal{F}_{\text{overexposed}} \ \wedge\ \lambda_S < 1.5 \ \wedge\ 2 < R_S < 0.03 \cdot N_{\text{cols}}] \quad (5)$$

where λ_S is the aspect ratio of the bounding rectangle of S, and R_S is the circumradius of S. The *centroid* of $S \in \mathcal{F}_{\text{highlight}}$, denoted by p_S, is given by

$$p_S = \left(\frac{\mu_{10}(S)}{\mu_{00}(S)}, \frac{\mu_{01}(S)}{\mu_{10}(S)}\right) \quad \text{with} \quad \mu_{ab}(S) = \sum_{(x,y)\in S} x^a \cdot y^b \quad (6)$$

This filter removes a large part of distracting overexposed regions (see Fig. 4), while some tutorials [7,17] use all the overexposed regions to create star patterns. Experiments show that our strategy performs more naturally (see Sect. 6).

Fig. 4. Highlight detection. From left to right: a map of overexposed pixels for the scene shown in Fig. 3, top, detected overexposed regions, and remaining highlights after applying our filter

4.2 Color Recovery

Highlights are normally the most saturated parts in a night photo, for example, streetlights, traffic signals, or head- or taillights of a car. Though the center of a highlight is by our definition just overexposed without any color information, the adjacent region normally provides strong hints on color due to diffuse reflection or diffraction. Thus we estimate the color information of a highlight based on the color of adjacent pixels.

We first convert I_b into I_{HSV} in the HSV color space following [22]. For $I_b(p) = (R(p), G(p), B(p))$, $V(p) = \max\{R(p), G(p), B(p)\}$, and $\delta(p) = V(p) - \min\{R(p), G(p), B(p)\}$, we have that

$$S(p) = \begin{cases} \delta(p)/V(p) & \text{if} \quad V(p) \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{H}(p) = \begin{cases} 60 \cdot [G(p) - B(p)]/\delta(p) & \text{if} \quad V(p) = R(p) \\ 120 + 60 \cdot [B(p) - R(p)]/\delta(p) & \text{if} \quad V(p) = G(p) \\ 240 + 60 \cdot [R(p) - G(p)]/\delta(p) & \text{if} \quad V(p) = B(p) \end{cases} \tag{7}$$

$$H(p) = \begin{cases} \hat{H}(p) + 360 & \text{if} \quad \hat{H}(p) < 0 \\ \hat{H}(p) & \text{otherwise} \end{cases}$$

For a highlight region S, we select the hue H_S and saturation S_S by detecting the most saturated pixel p in a circular neighborhood Ω_S around S:

$$\forall q \, [q \in \Omega_S \rightarrow S(p) \geq S(q)] \tag{8}$$

Here $S(p)$ is the saturation value of p, and Ω_S is a circular region centered at the centroid p_S having the radius of $1.5 \cdot R_S$, where R_S is the circumradius of S.

4.3 Luminance Estimation

The size of a star pattern in a star-effect photo is related to the luminance value of the corresponding highlight. The real luminance values of highlight regions are lost in a photo due to the limited dynamic range of the image sensor or the image file format. We estimate a possible luminance value V_S for each highlight $S \in \mathcal{F}_{\text{highlight}}$. The goal of such an estimation is to render natural star effects; we do not go so far as understanding the real luminance of the image scene.

We make an assumption that all the light sources with a same color have the same energy. For example, all the streetlights along a street have the same energy, which is stronger than the energy of a traffic light. Thus, we classify the light sources by available color information, and assign a weight parameter $C_{\text{color}}(S)$ to the light source corresponding to S. Empirically, we have that

$$C_{color}(S) = \begin{cases} 0.3, & \text{if} \quad \text{color}(S) = \text{Red} \\ 1, & \text{if} \quad \text{color}(S) = \text{Yellow} \\ 0.4, & \text{if} \quad \text{color}(S) = \text{Green} \\ 0.2, & \text{if} \quad \text{color}(S) = \text{Others} \end{cases} \tag{9}$$

with

$$\text{color}(S) \; = \begin{cases} \text{Red} & \text{if} \quad H_S < 30 \vee H_S \geq 330 \\ \text{Yellow} & \text{if} \quad 30 \leq H_S < 75 \\ \text{Green} & \text{if} \quad 75 \leq H_S < 165 \\ \text{Others} & \text{otherwise} \end{cases} \tag{10}$$

This classification covers common light sources in night photos.

We also simply ignore the media between light source and lens, and estimate the luminance V_S of the highlight region according to the distance $D(p)$ between camera and light source, and color information:

$$V_S = \frac{E_S}{4\pi \cdot D(p)^2} \approx C_{\text{intensity}} \cdot C_{\text{color}}(p) \cdot d(p)^2 \tag{11}$$

Here E_S is the energy of the light source corresponding to S – which we do not know. For convenience, we use a global user parameter $C_{\text{intensity}}$ for controlling the "strength" of star effects.

Experiments show that this assumption works well in applications (see Sect. 6).

5 Star Pattern Rendering

The light diffracted by an aperture can be approximately modeled by the *Fraunhofer diffraction equation* [14] because the light sources are normally effectively at infinity. The diffracted light goes into (the system of) optical lenses before it arrives at the image sensor; this sensor introduces further complexity.

Equations (1) and (2) show that the diffraction distribution rule is not related to the strength of the input light. Thus, the pattern keeps the same shape under any luminance condition. A star of a weaker light source looks smaller because the outer-ring of the pattern is "too weak" to be visible. See a real-world example in Fig. 5. Due to the limitation of the dynamic range, the centers of the star patterns (i.e. the "peaks") look all the same. The pattern is stronger inside and weaker outside. Thus, we model the shape of a star pattern by loading an $N \times N$ texture α. The scale N is a user controlled parameter. We define α on a rectangular set Ω_α, having the center of Ω_α as its origin.

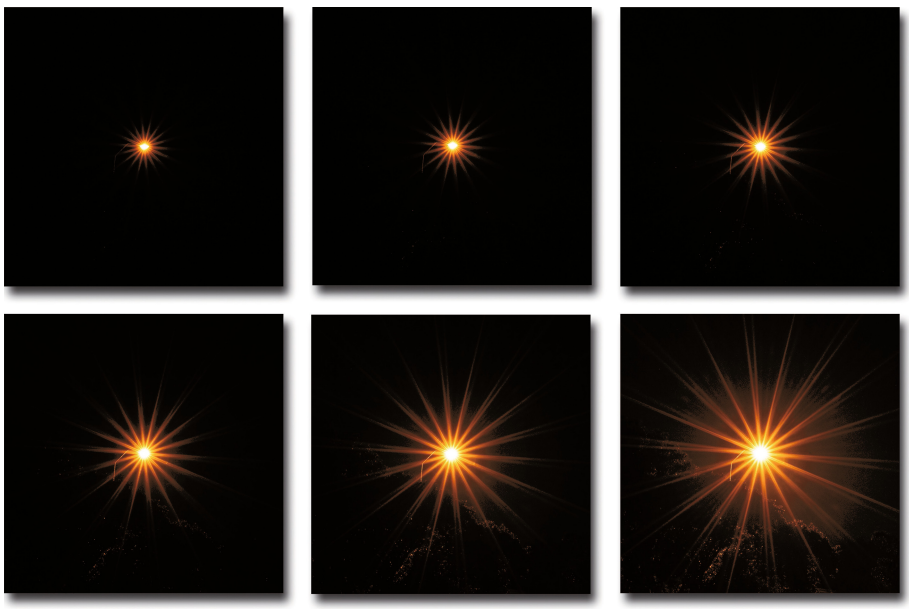

Fig. 5. Illustration of the relationship between star pattern and luminance. Photos shot with f/22 and ISO 800. Shutter speeds are, from top-left to bottom-right: 1/4s, 1/2s, 1s, 2s, 4s, and 8s

We render a star pattern for each highlight region S on I_b, and obtain the final result. For each position $p \in \Omega_\alpha$, let

$$I(p + p_S) = I(p + p_S) + \alpha(p)\,(R_S, G_S, B_S) \tag{12}$$

Here R_S, G_S, and B_S are the RGB color values corresponding to the previously calculated HSV color (H_S, S_S, V_S).

6 Experiments

We implemented our approach on a 3.30 GHz PC with 8.00 GB RAM and no graphics card. The run time is about 3 s for images of $2,400 \times 1,600$ resolution. Figure 6 illustrates star pattern rendering using different strength parameters. Figure 7 illustrates how to control the styles of the star effect using different textures. To control the shape of the star pattern by a camera, photographers need to use different lenses. Comparatively, our method is very convenient in controlling star effects, and thus convenient for artists to try different styles and obtain their favorite feeling.

Fig. 6. Use of different strengths for star-pattern rendering. *Left to right*: Original image, small scale star effect, and large scale star effect

Fig. 7. Application of different star textures on Fig. 6, left

Figure 8 compares our method, a uniform-size star effect, a random-size star effect, and a real-world star-effect photo. The uniform-size star effect is generated manually following a tutorial [17], the random-size star effect is generated using

our rendering method but without any use of depth information. The uniform-size star effect also looks appealing, but does not show any space feeling. Such an effect is similar to the style of a common star filter. It can also be noticed that all the overexposed regions are used in the manual process, including those regions that should not make a significant diffraction, as, for example, some sparkles on the water. The random-size star effect looks more flexible. Without understanding the image's content, this effect does not show any space feeling either, and involves false logic. For example, the further-away street light should define a smaller star. In comparison, our depth-aware method performs naturally, and the result is similar to the real-world star-effect photo. Another advantage of our method over manual processing [7,17] is that we can render colorful stars according to the content of the input photo, while the manual process only renders monochromatic stars because the color information of those overexposed regions is lost.

Fig. 8. Comparison between our method (*top-left*), uniform-size star effect (*top-right*), random-size star effect (*bottom-left*), and a real-world star-effect photo, i.e. the "ground truth" (*bottom-right*)

Our method can also simulate the style of a star filter, using a proper star texture (as demonstrated in Fig. 9). The photo on the left is shot at aperture f4, which is a relatively large aperture, and thus does not show any significant star effect. The photo in the middle is shot with a 4-point star filter and with

Fig. 9. Simulation of the style of a star filter. *Left to right*: A photo shot with large aperture (thus does not show any star effect), a photo shot with a star filter, and our result

the same camera parameters as used for the left photo. It can be seen that our result looks similar to the star-filter photo.

Variations in robustness of the applied stereo vision method can affect our star rendering. This is a limitation of our method. Figure 10 shows some failed cases due to failures in depth calculations. The car in the red rectangle of Fig. 10, right, is in an occlusion region (see Fig. 10, middle), thus the depth information is unavailable. As a result, star effect is not applied to the headlight of the car.

Fig. 10. Failed cases due to failures in depth calculations (Color figure online).

7 Conclusion

We present a star-effect simulation method. The key is to obtain depth information from self-calibrated stereo vision, and then estimate the luminance of the highlight regions according to the available depth information. Finally, do a content-aware star-pattern rendering. Experiments show that our results are similar to real-world star-effect photos. Our results are also more natural in their appearance than the results of existing commercial applications. To the best of our knowledge, this is the first research reported on automatically simulating photo-realistic star effects.

Acknowledgements. This project is supported by the China Scholarship Council.

References

1. Blais, F.: Review of 20 years of range sensor development. J. Electron. Imaging **13**, 231–240 (2004)
2. Born, M., Wolf, E.: Principles of Optics. Cambridge University Press, Cambridge (1999)
3. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient belief propagation for early vision. Int. J. Comput. Vis. **70**, 41–54 (2006)
4. Hartley, R.I.: Theory and practice of projective rectification. Int. J. Comput. Vis. **35**, 115–127 (1999)
5. Hirschmüller, H.: Accurate and efficient stereo processing by semi-global matching and mutual information. In: Proceedings of CVPR, pp. 807–814 (2005)
6. Hermann, S., Klette, R.: Iterative semi-global matching for robust driver assistance systems. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012, Part III. LNCS, vol. 7726, pp. 465–478. Springer, Heidelberg (2013)
7. Hoey, G.: Advanced photoshop starburst filter effect - Week 45 (2009). www.youtube.com/watch?v=lRKp4_EkIvc
8. Khoshelham, K., Elberink, S.O.: Accuracy and resolution of Kinect depth data for indoor mapping applications. Sensors **12**, 1437–1454 (2012)
9. Kopf, J., Cohen, M.F., Lischinski, D., Uyttendaele, M.: Joint bilateral upsampling. ACM Trans. Graph. **26**(3), 96 (2007)
10. Klette, R.: Concise Computer Vision: An Introduction into Theory and Algorithms. Springer, London (2014)
11. Klette, R., Rosenfeld, A.: Digital Geometry: Geometric Methods for Digital Picture Analysis. Morgan Kaufmann, San Francisco (2004)
12. Liu, D., Klette, R.: Fog effect for photography using stereo vision. Vis. Comput. 1–11 (2015). doi:10.1007/s00371-014-1058-7
13. Klette, R., Liu, D., Nicolescu, R.: Bokeh effects based on stereo vision. In: Azzopardi, G., Petkov, N., Yamagiwa, S. (eds.) CAIP 2015. LNCS, vol. 9256, pp. 198–210. Springer, Heidelberg (2015). doi:10.1007/978-3-319-23192-1_17
14. Lipson, A., Lipson, S.G., Lipson, H.: Optical Physics. Cambridge University Press, Cambridge (2010)
15. Lowe, D.G.: Distinctive image features from scale-invariant keypoints Int. J. Comput. Vis. **60**, 91–110 (2004)
16. Lukac, R.: Computational Photography: Methods and Applications. CRC Press, Boca Raton (2010)
17. mahalodotcom: How to create a lens flare effect in Photoshop (2011). www.youtube.com/watch?v=qmL0ct2Ries
18. Ng, R., Levoy, M., Brédif, M., Duval, G., Horowitz, M., Hanrahan, P.: Light field photography with a hand-held plenoptic camera. Stanford University, CSTR 2005–02 (2005)
19. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF. In: Proceedings of ICCV, pp. 2564–2571 (2011)
20. Schechner, Y.Y., Kiryati, N.: Depth from defocus vs. stereo: how different really are they? Int. J. Comput. Vis. **39**, 141–162 (2000)
21. Klette, R., Song, Z.: Robustness of point feature detection. In: Wilson, R., Hancock, E., Bors, A., Smith, W. (eds.) CAIP 2013, Part II. LNCS, vol. 8048, pp. 91–99. Springer, Heidelberg (2013)
22. Smith, A.R.: Color gamut transform pairs. ACM Siggraph Comput. Graph. **12**, 12–19 (1978)
23. Wang, L., Wei, L.Y., Zhou, K., Guo, B., Shum, H.Y.: High dynamic range image hallucination. In: Proceedings of Eurograph, pp. 321–326 (2007)