Lisadスペース：スケール不変なライトフィールド分析の新しいフレームワーク

Lisad Spaces: A Novel Framework for Scale-Invariant Light Field Analysis

イバナ トシック* キャサリン バークナー*
Ivana TOSIC Kathrin BERKNER

要 旨

近年の携帯プレノプティックカメラの発展により，多くの実用的かつ低価格のイメージングアプリケーションにライトフィールド捕捉技術を取り入れることが可能となった．捕捉されたライトフィールドはカメラの開口部を通る光線の位置と角度両方の情報を含み，撮られたシーンの3D情報を本質的に捕らえている．しかし，ライトフィールドからの3Dシーンの再構築と分析は，高次元性とその特殊な構造のため，なお難しい問題である．この問題に取り組むため，ライトフィールドで記録された対象物のスケールと奥行きの両方からパラメーター化された***Li**ght field **s**cale-**a**nd-**d**epth* (Lisad) スペースを構築することで，ライトフィールドのスケール不変な3D分析のための新しい理論を提案する．

Lisadスペースの応用として，3Dキーポイントの検出と，稠密な奥行き予測の2つを示す．どちらのケースにおいても，Lisadスペースに基づくアルゴリズムはそれぞれ最先端の比較対象アルゴリズムを凌駕している．さらに，その実装はローカル処理しか必要としていないため，プレノプティックカメラのための将来のコンピュータビジョンアーキテクチャにLisadに基づいた手法を統合できる可能性が高い．

ABSTRACT

Recent development of hand-held plenoptic cameras has brought light field acquisition into many practical and low-cost imaging applications. The acquired light fields capture both spatial and angular information of light rays passing through the camera aperture. As such, they intrinsically capture 3D information of the imaged scene. However, 3D scene reconstruction and analysis from light fields still remains a challenge due to their high dimensionality and their particular structure. To address this problem, we propose a new theory for scale-invariant 3D analysis of light fields via the construction of ***Li**ght field **s**cale-**a**nd-**d**epth* (Lisad) spaces, which are parametrized both in terms of scale of objects recorded by a light field and in terms of objects' depth. We show two applications of Lisad spaces: 3D keypoint detection and dense depth estimation from light fields. In both cases, our algorithms based on Lisad spaces outperform their respective state-of-the-art counterparts. Moreover, their implementation requires only local processing, which shows great potential for integration of Lisad-based methods into future computer vision architectures for plenoptic cameras.

* リコーイノベーションズコーポレーション
Ricoh Innovations Corporation

1. Introduction

Compared to traditional imaging systems, plenoptic systems provide additional capabilities and functionalities such as single-snapshot multi-spectral imaging[1] refocusing[2] and 3D imaging[3]. This is achieved by inserting a micro-lens array in front of the imaging sensor. After calibration, plenoptic data can be demultiplexed to a set of multi-view images that form a 4-dimensional (4D) data structure called the light field (LF)[4]. Prior to development of plenoptic cameras, LFs have been acquired by camera arrays or by a moving camera rig, capturing images from regularly spaced viewpoints. Such acquisition leads to 3D, 4D or higher dimensional pixel arrays, which represent specific samplings of the 7D plenoptic function[5].

Recent development of plenoptic cameras outlined the need for efficient LF processing algorithms (e.g., to obtain 3D structure or depth of a scene), which is still a challenging problem because of the high dimensionality of LFs. To address this problem, we exploit the particular geometry of LFs obtained by plenoptic sensors or planar camera arrays, where viewpoints are regularly spaced on a planar surface. A parameterization of such 4D LFs is usually given in coordinates (x, y, u, v), where (x, y) are pixel coordinates for an image taken from a viewpoint (u, v). An example of a LF obtained from a plenoptic image is shown in Figure 1. We can see that a 2D $x-u$ slice of the LF, obtained by cutting the LF across views, has a "linear" or "ray" structure, where the angle of a ray corresponds to a different depth value of that point in a 3D scene. Bolles et al. observed this structure and named these slices epipolar plane images (EPIs)[6].

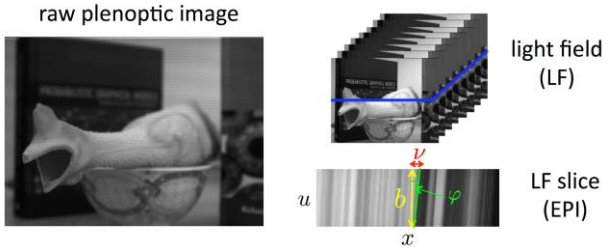


Fig. 1 Example of a LF extracted from a plenoptic image. The blue line indicates the pixels extracted in the horizontal $x-u$ slice (EPI) of LF, displayed below. Angle φ of ray at pixel x uniquely determines the disparity v of x.

For LFs with such EPI structure, extracting 3D scene information can be formulated as a problem of estimating the angle of rays for each pixel x in a given EPI. A typical EPI consists of discontinuities that we call ***ray edges***, and uniform regions (stripes) bounded by ray edges that we simply call ***rays*** throughout the rest of the paper. Each ray is parameterized by its position on x- axis, its width and its angle with the u-axis. In order to detect rays of different widths (i.e., scales) in a given EPI, we need a method for multi-scale analysis of light fields. For general signals, multi-scale representations are most commonly obtained via the scale-space theory, which is widely used to solve problems in computer vision, signal and image processing. It allows algorithms to detect or process image structures at different scales. The scale-space of an image is a family of smoothed images parameterized by the scale parameter of the chosen kernel[7]. Applications of building such a scale-space representation include image segmentation, keypoint detection and object recognition. A well known application of scale-space theory is the Scale-Invariant-Feature-Transform, where keypoint detection is based on finding extrema in scale spaces constructed using the difference-of-Gaussian function, which provides a close approximation to the normalized Laplacian of Gaussian scale-space[8].

Although scale-space theory has been widely studied for standard images, there is no prior work on scale-space construction for light fields. Here, we exploit the

particular structure of LFs for the construction of novel scale spaces for LFs, which are parametrized both in terms of scale of objects recorded by the LF and in terms of objects' depth. We show the scale invariance properties of new ***Lisad*** (***Li***ght field ***s***cale ***a***nd ***d***epth) spaces, based on which we can efficiently build multi-scale, multi-depth representations of LFs. Since our novel representations are parametrized in terms of scale and depth of 3D objects, we can detect and process objects of different scales and different depths in LF data and provide their depth estimates. Therefore, the proposed scale-depth spaces can be exploited in applications that require 3D scene estimation from light fields. We show here two examples of such applications: 1) 3D keypoint detection, where we find keypoints such as 3D edges and assign a depth value to each keypoint; and 2) dense depth estimation, where we assign a depth/disparity value to each pixel in the light field. We show that the detected keypoints outperform a widely used SURF algorithm[9] on a 3D structure estimation task. Moreover, our dense depth estimation method achieves the highest accuracy on the HCI (Heidelberg Collaboratory for Image Processing) LF benchmark database[10].

2. Light field scale-depth spaces

2-1 Basics of Gaussian scale-spaces

Research on scale-space theory has a long history, dating back to the seminal paper by Witkin[11]. There is a plethora of prior works that address the theory and construction of scale-spaces for representing signals by filtering them with kernels of different scales. Scale spaces have found many applications in analysis of images, videos, tomography data and medical images (see Witkin[12] for a review of literature on scale-spaces).

The most commonly used kernel for constructing scale spaces is the Gaussian kernel:

$$G_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$$

Its associated scale space in 1D case is defined as $\mathcal{L}(x,\sigma) = I(x) * G_\sigma(x)$, where $*$ denotes convolution. An important property of Gaussian scale-spaces is the scale invariance property: $(J * G_\sigma)(x) = (I * G_{s\sigma})(sx)$, where $J(x) = I(sx), s \in R$. This property says that a feature at scale σ elicits the same response as that feature at a larger scale $s\sigma$, which allows for scale-invariant processing of signals. This property is necessary for dealing with the object size variations in image processing[12]. Examples include edge detection by finding extrema in the scale-space built upon the normalized first derivative of the Gaussian $\sigma dG_\sigma/dx$ and blob detection by finding extrema in the scale-space built upon the normalized second derivative of the Gaussian $\sigma^2 d^2 G_\sigma/dx^2$ [13]. Our approach bears similarities to Gaussian scale-spaces, but is tailored specifically for analysis of LFs.

2-2 Novel light field scale-depth spaces

In order to build scale spaces for LFs, we need a kernel that well represents 2D slices of the LF. To that end, we have defined in[14] the ***Ray-Gaussian*** kernel as:

$$\mathcal{R}_{\sigma,\varphi}(x,u) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x+u\tan\varphi)^2}{2\sigma^2}},$$

where x and u are coordinates of pixels in a 2D LF slice, φ is the angle that the Ray Gaussian forms with the u-axis and σ is the width parameter of the kernel. An example of a Ray Gaussian function is given in Figure 2. We can see that it is Gaussian in x-direction and a ridge in u-direction, where the slant of the ridge is $\tan\varphi$.

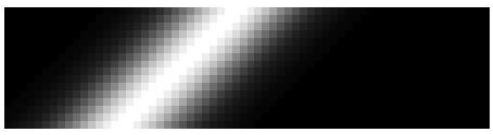


Fig. 2 Example of a ray Gayssian kernel with $\varphi = \pi/4$ and $\sigma = 6$.

We further use the Ray-Gaussian to construct the LF scale space $\mathcal{L}(x;\sigma,\varphi)$ in the following way:

$$\mathcal{L}(x;\sigma,\varphi) = \big(I * \mathcal{R}_{\sigma,\varphi}\big)(x,u)|_{u=0}$$

where $u=0$ is chosen because we need to evaluate convolution only over x (pixel domain) and not over views u. This is because the features will be present in all views (except in rare cases of occlusion), so we do not need to localize features within the views. That is,

$(f * g)(x,u)|_{u=0} = \int_{x'}\int_{u'} f(x-x',-u')g(x',u')dx'du'$.

Note also that $\mathcal{L}(x;\sigma,\varphi)$ does not depend on u since the convolution is only over x, and has both scale σ and angle φ as parameters. Since angle φ is uniquely associated with depth of a ray, we refer to $\mathcal{L}(x;\sigma,\varphi)$ as **Li**ght field **s**cale **a**nd **d**epth space, or ***Lisad*** space. In the following, we show certain properties of Lisad spaces, which have been proven in our prior work[14].

2-3 Properties of Lisad spaces

We start by showing the scale-invariance property of the Ray Gaussian (RG) kernel, which is important for building its associated Lisad spaces. Unlike in standard image scale-spaces, we analyze the scale-invariance by downsampling only in x since downsampling in u (dropping views) is usually undesirable.

Lemma 1 *[Scale-invariance of RG.]* The following equality holds: $\mathcal{R}_{\sigma,\varphi}(x,u) = s\mathcal{R}_{s\sigma,\varphi'}(sx,u)$, where $\varphi' = \tan^{-1}(s\tan\varphi), \varphi \in (-\pi/2,\pi/2), s>0$.

Proofs of Lemma 1 and of all following Propositions are in[14,15]. Lemma 1 shows that a RG with scale σ and angle φ is equal to its downsampled version at scale $s\sigma$ and angle $\varphi' = \tan^{-1}(s\tan\varphi)$, with values multiplied by s, for downsampling in x by factor s. Furthermore, we show another property of the RG, which relates to the angle invariance of its inner product with an LF slice.

Proposition 1. If we have a real function $f_\varphi(x,u) = h(x+u\tan\varphi)$, where $x,u \in R, \varphi \in (-\pi/2,\pi/2)$ and h is a 1D embedding of f_φ, then $\forall\varphi$ it holds:

$\langle f_\varphi, R_{\sigma,\varphi}\rangle = \langle f_0, R_{\sigma,0}\rangle = \iint_{-\infty}^{\infty} f_0(x,u)R_{\sigma,0}(x,u)dxdu$.

This property means that if we have a 2D slice of LF that can be embedded in a 1D space for a given φ, i.e., $I(x,u) = h(x+u\tan\varphi), \forall x \in R$, then its inner product with the RG of the same angle φ will always have the same value, independently of value of φ. Note that only LFs without occlusions satisfy this assumption, which is not always the case. However, we can assume that this requirement is satisfied locally. This is an important property of Lisad spaces because it assures that there is no angle (depth) bias. For example, when using Lisad spaces for keypoint detection (Sec. 3-1), it assures that keypoints at different depths are treated equally.

Using Lemma 1, we now show the scale-invariance property of the Lisad space and its derivatives.

Proposition 2. *[Scale-invariance of the Lisad space.]* If we have a LF slice $J(x,u)$ such that $J(x,u) = I(sx,u)$ (depending on value of s, I is a downsampled or upsampled version of J over x), then it holds:

$$\big(J * \mathcal{R}_{\sigma,\varphi}\big)(x,u)|_{u=0} = (I * \mathcal{R}_{s\sigma,\varphi'})(sx,u)|_{u=0}$$

where $\varphi' = \tan^{-1}(s\tan\varphi), \varphi \in (-\pi/2,\pi/2), s>0$.

We show a similar equation holds for Lisad spaces built upon the "normalized" Ray Gaussian first derivative $\mathcal{R}'_{\sigma,\varphi} = \sigma\, d/dx\, \mathcal{R}_{\sigma,\varphi}$. We define the normalized first derivative Lisad space as:

$$\mathcal{L}'(x;\sigma,\varphi) = \frac{d}{dx}\mathcal{L}(x;\sigma,\varphi) = \left(I * \sigma\frac{d}{dx}\mathcal{R}_{\sigma,\varphi}\right)(x,u)|_{u=0}$$

and call it Lisad-1 space.

Proposition 3 *[Scale-invariance of the Lisad-1 space.]* If we have a LF slice $J(x,u)$ such that $J(x,u) = I(sx,u)$ (depending on value of s, I is a downsampled or upsampled version of J over x), then:

$$\left(J * \sigma\frac{d}{dx}\mathcal{R}_{\sigma,\varphi}\right)(x,u)|_{u=0} = \left(I * s\sigma\frac{d}{dx}\mathcal{R}_{s\sigma,\varphi'}\right)(sx,u)|_{u=0}$$

where: $\varphi' = \tan^{-1}(s\tan\varphi), \varphi \in (-\pi/2,\pi/2), s>0$.

Similarly, we have shown in[15] that the scale invariance property holds for the Lisad spaces built upon the "normalized" second derivative of the Ray Gaussian:

$$\mathcal{L}''(x;\sigma,\varphi) = \frac{d^2}{dx^2}\mathcal{L}(x;\sigma,\varphi) = \left(I * \sigma^2\frac{d^2}{dx^2}\mathcal{R}_{\sigma,\varphi}\right)(x,u)|_{u=0}.$$

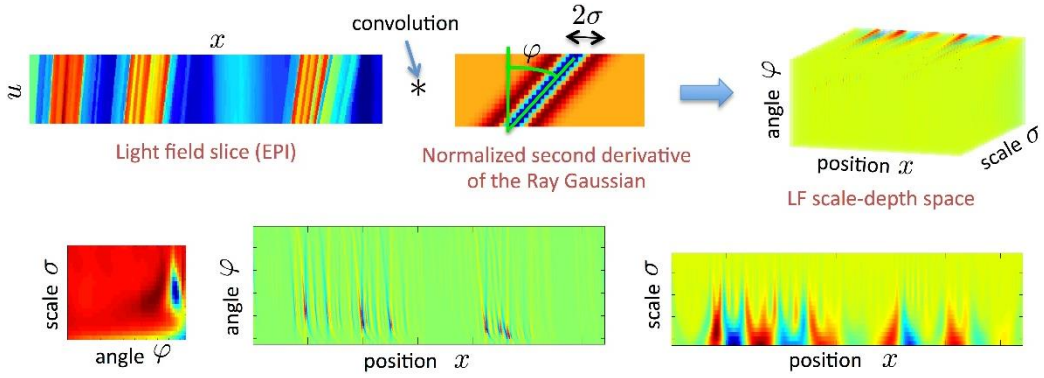


Fig. 3 Illustration of the scale-depth space of a light field slice, built upon the second derivative of a Ray Gaussian. Top panels: Convolution of an EPI with a normalized second derivative of the Ray Gaussian gives a 3D light field scale-depth space (Lisad-2).
Bottom panels show Lisad-2 slices; from left to right: angle-scale, position-angle and position-scale.

We call these Lisad-2 spaces. Finally, note that Proposition 1 holds also for the above defined derivatives of the Ray Gaussian.

Figure 3 shows an example of generating a Lisad-2 space for a given 2D EPI. The $x-u$ EPI (top left panel) is convolved over dimension u with the normalized second derivative of a Ray Gaussian kernel (top middle panel), giving a 3D Lisad-2 space with coordinates (x,σ,φ) (top right panel). In the bottom panels, we show examples of 2D slices through the Lisad-2 volume when fixing a location x_0, a scale σ_0, and an angle φ_0. From left to right panels, we show slices $(x_0,\sigma,\varphi),(x,\sigma_0,\varphi)$ and (x,σ,φ_0). Extrema in those slices are located in the blue and red regions. The exact coordinates of the extrema are found through search over the entire 3D volume. Whereas the (x,σ,φ_0) looks like typical scale-space visualization with extrema located at the bottom of figure along the x-axis, the slices $(x_0,\sigma,\varphi),(x,\sigma_0,\varphi)$ exhibit a different structure, with extrema being located inside the volume.

Note here that calculation of the Lisad spaces involves only local processing, since it is done by convolutions. Likewise, search for extrema in this space can be done via computations only in the local neighborhood of each point in the Lisad space.

3. Applications of Lisad spaces

In this section we show two applications of Lisad spaces: 1) 3D keypoint detection and 2) dense depth estimation from light fields.

3-1 3D keypoint detection

Gaussian scale-spaces have been extensively used in the past for detecting interest points in images, where among most known approaches are SIFT (Scale-invariant feature transform)[8] and SURF (Speeded-UP Robust features)[9]. These keypoints are blob features that are found as extrema in an approximation of second derivative Gaussian scale-spaces. Though blob detectors give more stable keypoints in 2D images[16], we have chosen to compute edge keypoints in LFs since edges are important features in 3D scenes. In the following, we briefly explain the Lisad keypoint detection method, while for more details we refer the reader to[14].

In order to detect 3D keypoints in LFs, we first have to construct the Lisad-1 space. Once we have $\mathcal{L}'(x;\sigma,\varphi)$, we detect 3D keypoints in LF slices and estimate depth values for those keypoints by finding extrema in $\mathcal{L}'(x;\sigma,\varphi)$. The parameters of Q extrema

points $\{(x_q, \sigma_q, \varphi_q)\}, q = 1, \ldots, Q$, will give us the following information about each keypoint q:

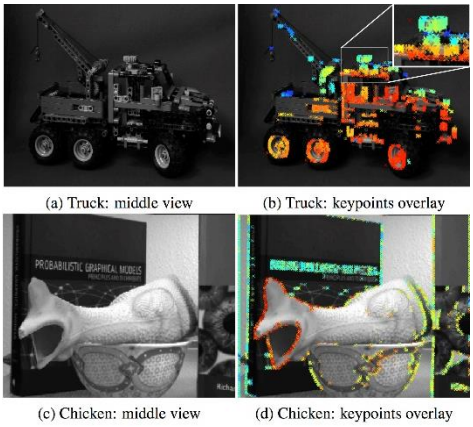


Fig. 4 3D keypoint detection results for Truck and Chicken scenes.

(i) position of the keypoint x_q, (ii) scale of the keypoint σ_q and (iii) angle of the keypoint φ_q. Since we get a depth value immediately assigned to each keypoint, our method effectively performs 3D keypoint detection.

Figure 4 shows our 3D keypoint detection method on two 4D LFs: "truck" LF captured by a moving camera rig[17] and "chicken" LF captured by Ricoh plenoptic camera prototype, which is a DLSR-type system with a microlens array mounted on a monochrome CCD sensor.

Multi-view extraction is performed using the publicly available LFtoolbox[18]. Middle views for truck and chicken LFs are shown in Fig. 4a) and c), respectively. We display the results by plotting keypoints overlaid on a reference image (Fig. 4b) and d) for truck and chicken, respectively). The depth of each keypoint is indicated by marker color, where hotter colors mean closer keypoints. We can see that there are only few keypoints with large depth errors.

For a quantitive evaluation of our keypoints on a 3D estimation task, we use multi-view images from the Middlebury database[19], which have ground truth disparity maps. We compare the disparity estimation accuracy of SURF keypoints and the proposed Lisad keypoints. We find SURF keypoints in each of 9 multi-view images (using opensurf [20]) and then find best matching pairs between a reference image and all other images. Keypoints are then matched across views using SURF feature descriptors. Since Lisad keypoints have by construction a disparity value associated with each keypoint, we do not need to construct the descriptors. We simply map φ to its disparity value $v = b\,tan(\varphi)$, where b is the camera baseline for which the disparity map is given. Table 1 gives the disparity estimation accuracy and computational efficiency of SURF keypoints and Lisad keypoints with angle sampling step corresponding to 1 pixel disparity (Lisad (a)) and corresponding to 2 pixels disparity (Lisad (b)), for Teddy and Cones datasets. The accuracy is calculated as the percentage of keypoints with disparity error less than ε, where ε varies from 1 to 4 pixels. All experiments were done on a Linux® machine with one six-core 2.6 GHz Intel® CPU and 64 GB of RAM. We can see that our keypoint detection always gives more keypoints with accurate disparity values. Moreover, disparity values using Lisad keypoints are more efficient to compute since we do not need to evaluate descriptors.

Table 1 3D keypoint detection: comparison of Lisad and SURF.

data	method	% of keypoints with $\lvert\Delta d\rvert < \epsilon$				time
		$\epsilon = 1$	$\epsilon = 2$	$\epsilon = 3$	$\epsilon = 4$	(sec)
Teddy	SURF	51.05	66.41	72.94	77.93	14.31
	Lisad (a)	66.61	74.54	78.54	81.09	8.57
	Lisad (b)	60.04	71.74	76.92	79.01	5.51
Connes	SURF	46.16	64.42	75.00	79.14	18.16
	Lisad (a)	63.57	71.87	78.17	83.05	8.91
	Lisad (b)	57.66	71.53	77.38	82.53	4.71

3-2 Dense depth estimation

For LFs with the linear EPI structure, as the ones obtained with plenoptic sensors, the task of dense depth estimation can be formulated as a problem of estimating the angle of rays for each pixel x in a given EPI. While depth at ray edges can be reliably estimated via for example line fitting, estimating depth within rays is

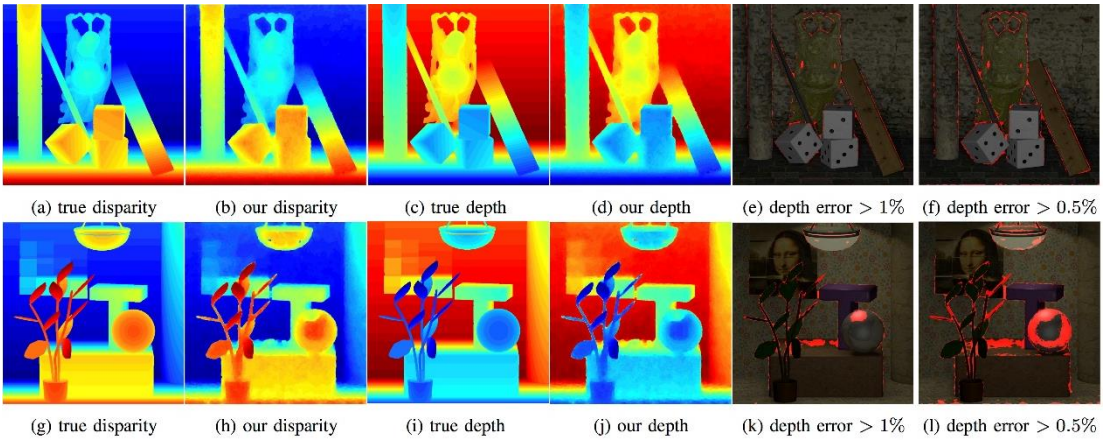
challenging, as there is an ambiguity in angle estimation due to the same value of pixels within the ray. Similar ambiguities exist in stereo and multi-view approaches, where most solutions impose smoothness constraints in a global optimization[21] or in a variational framework[22]. We take a different approach by formulating the problem as a ray detection problem where the goal is to simultaneously detect rays and determine their positions, angles and widths[15]. Once we have detected rays and estimated their widths, we assign the same depth value to all points within the same ray. This way, we obtain a dense depth map without performing global optimization.

Similar to using second derivative Gaussian scale-spaces for blob detection, we use the normalized second derivative Ray-Gaussian Lisad spaces to find rays in the EPIs. Namely, it can be easily shown that an extremum in Lisad-2 space will be located exactly in the middle of the ray, where the width of the ray is exactly 2σ of that extremum. Parameters of P extrema points $\{(x_p, \sigma_p, \varphi_p)\}, p = 1, \dots, P$, give us the following information about each ray p: 1) position of the center of the ray x_p; 2) width of the ray $2\sigma_p$; 3) angle of the ray φ_p. From the angle φ_p we get depth of that ray by using the camera calibration parameters. For example, for light fields obtained by a camera array, the depth is calculated as $d_p = f\,b/\tan\varphi_p$, where f is camera focal length and b is the distance between neighboring cameras (baseline). After we have detected rays and found their parameters, we need to resolve occlusion conflicts between overlapping rays. Since we have the position and width of each ray, we can easily find pairs that overlap. Once we have found overlapping rays, we need to decide on their ordering from foreground to background. Because larger angle of rays indicates smaller depth (closer objects, larger parallax), rays with larger angles should always be in the foreground. Due to noise in images, detected rays sometimes have an opposite ordering, which is physically impossible. When we find such cases of overlapping rays, we remove the occluded ray from the rays set. Besides rays eliminated due to occlusion conflicts, we also remove rays that have one weak edge. Those rays are sometimes detected next to object boundaries, meaning that one side of the ray is an object edge while the other side is within a uniform background. To solve this problem, we use the normalized first derivative Ray-Gaussian Lisad space (Lisad-1) to detect ray edges and then keep the rays that have ray edges on both sides. Lisad space construction (for first and second derivatives), ray detection and occlusion detection are performed separately on horizontal and vertical EPIs. After we have eliminated the weak rays, we convert information about each ray (its position, scale, angle and value of the scale- depth space) into a dataterm for depth estimation and assign depth values to pixels according to the maximum of the dataterm. Finally, we apply some post-processing to the obtained depth map in order to fill out missing regions and remove the noise. For more details, please see[15].

We have evaluated our depth estimation method on the LF benchmark database hosted by the Heidelberg Collaboratory for Image Processing (HCI)[23], where LFs are simulated by a virtual camera array. We compared the depth estimation accuracy to the best reported method of the benchmark. Table 2 shows the obtained disparity and depth accuracy for Buddha and Mona datasets, compared to the previously best reported results on the benchmark. We can see that our method outperforms the best prior art[22,24], both in terms of disparity MSE and percentage of pixels with depth error larger than 1%.

Table 2 Results for Buddha and Mona datasets in terms of the mean squared error (MSE) of disparity values and the percentage of pixels with depth error less than 1%.

Dataset	Metric	Lisad-depth	Best prior work
Buddha	disparity MSE > 1% depth error	**0.0048** **1.2%**	0.0055 [22] 3.5% [24]
Mona	disparity MSE > 1% depth error	**0.0061** **2.4%**	0.0082 [22] 4.6% [24]



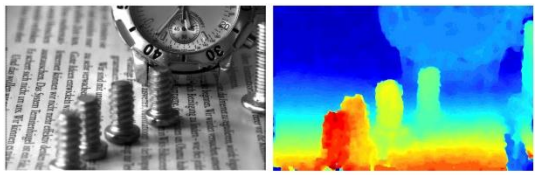
(a) true disparity (b) our disparity (c) true depth (d) our depth (e) depth error > 1% (f) depth error > 0.5%

(g) true disparity (h) our disparity (i) true depth (j) our depth (k) depth error > 1% (l) depth error > 0.5%

Fig. 5 Visual comparison of estimated depth/disparity maps to the ground truth depth/disparity maps for Buddha (a-f), Mona (g-l).

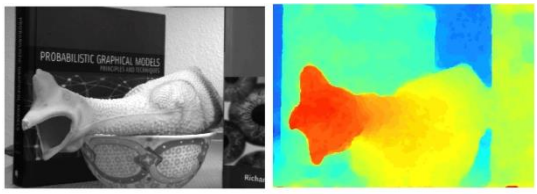
We show visually the ground truth and obtained disparity and depth maps for Buddha in Figure 5(a-d) and Mona in Figure 5(g-j). To see the pixels whose depth errors are larger than a certain threshold (1% and 0.5%) we display them in red overlaid on the original images in Figures: 5(e-f) and (k-l), for Buddha and Mona respectively.

Finally, we show disparity estimation results for the "watch" image from the RAYTRIX® camera, obtained from the HCI database, and for the "chicken" image from our own plenoptic camera prototype, in Figures 6 and 7, respectively.



(a) Middle view (b) Disparity map

Fig. 6 Disparity estimation for the RAYTRIX® "watch" image.



(a) Middle view (b) Disparity map

Fig. 7 Disparity estimation for the "chicken" image from our plenoptic camera prototype.

4. Conclusions

We have presented a theoretical formulation and construction of Light field scale and depth (Lisad) spaces based on a new Ray-Gaussian kernel and shown that such spaces satisfy the scale invariance property. Unlike 2D image scale spaces, Lisad spaces are parametrized both in terms of scale and depth of objects in a 3D scene. By finding extrema in Lisad spaces, we have shown that we can find 3D keypoints in scenes both accurately and efficiently. Moreover, we have presented a novel method for dense depth estimation from light fields, based on Lisad spaces, which outperforms state-of-the art methods in terms of depth accuracy.

References

1) R. Horstmeyer, G. Euliss, R. Athale, M. Levoy: Flexible multimodal camera using a light field architecture, *Proceedings of the IEEE International Conference on Computational Photography* (2009).
2) R. Ng et al.: Light field photography with a hand-held plenoptic camera, *Technical Report CSTR* (2005).
3) C. Perwass, L. Wietzke: Single lens 3d-camera with extended depth-of-field, *Proceedings of SPIE Electronic Imaging* (2012).
4) M. Levoy, P. Hanrahan: Light field rendering, *Proceedings of the 23rd annual conference on Computer Graphics and Interactive Techniques* (1996).
5) E. H. Adelson, J. R. Bergen: The plenoptic function and the elements of early vision, *Computational Models of Visual Processing*, Vol. 1, No. 2, pp. 3-20 (1991).
6) R. C. Bolles, H. H. Baker, D. H. Marimont: Epipolar-plane image analysis: An approach to determining structure from motion, *International Journal of Computer Vision*, Vol. 1, No. 1, pp. 7-55 (1987).
7) T. Lindeberg: Scale-Space. Hoboken, NJ, USA, John Wiley & Sons, Inc. (2007).
8) D. G. Lowe: Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91-110 (2004).
9) H. Bay, A. Ess, T. Tuytelaars, L. Van Gool: Speeded-Up Robust Features (SURF), *Computer Vision and Image Understanding*, Vol. 110, No. 3 (2008).
10) S. Wanner, S. Meister, B. Goldluecke: Datasets and benchmarks for densely sampled 4d light fields, *Proceedings of Vision, Modelling and Visualization (VMV)* (2013).
11) A. Witkin: Scale-space filtering: A new approach to multi-scale description, *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (1984).
12) T. Lindeberg: Generalized Gaussian Scale-Space Axiomatics Comprising Linear Scale-Space, Affine Scale-Space and Spatio-Temporal Scale-Space, *Journal of Mathematical Imaging and Vision*, Vol. 40, No. 1, pp. 36-81 (2010).
13) T. Lindeberg: Edge detection and ridge detection with automatic scale selection, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (1996).
14) I. Tošić, K. Berkner: 3d keypoint detection by light field scale-depth space analysis, *Proceedings of the IEEE International Conference on Image Processing* (2014).
15) I. Tošić, K. Berkner: Light Field Scale-Depth Space Transform for Dense Depth Estimation, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2014).
16) K. Mikolajczyk: Detection of local features invariant to affine transformations, Ph.D. dissertation, Institut National Polytechnique de Grenoble, France (2002).
17) V. Vaish: The (New) Stanford Light Field Archive, http://lightfield.stanford.edu (accessed 2014-06).
18) D. G. Dansereau, O. Pizarro, S. B. Williams: Decoding, calibration and rectification for lenselet-based plenoptic cameras, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2013).
19) D. Scharstein, R. Szeliski: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, *International Journal of Computer Vision*, Vol. 47, No. (1/2/3), pp. 7-42 (2002).
20) D.-J. Kroon: OpenSURF (including Image Warp), http://www.mathworks.com/matlabcentral/fileexchange/28300- opensurf-including-image-warp (accessed 2014-06).
21) V. Kolmogorov, R. Zabih: Multi-camera scene reconstruction via graph cuts, *Proceedings of the European Conference on Computer Vision* (2002).

22) S. Wanner, B. Goldluecke: Variational Light Field Analysis for Disparity Estimation and Super-Resolution, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 36, No. 3, pp. 606-619 (2014).
23) S. Wanner, C. Straehle, B. Goldluecke: Globally consistent multi-label assignment on the ray space of 4d light fields, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2013).
24) B. Goldluecke, S. Wanner: The variational structure of disparity and regularization of 4d light fields, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2013).

1) Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.
2) Intel® is a trademark of Intel Corporation in the U.S. and/or other countries.
3) Raytrix is a registered trademark of the Raytrix GmbH.