3D Research manuscript No.
(will be inserted by the editor)

On Applications of Pyramid Doubly Joint Bilateral Filtering in Dense Disparity Propagation

Arash Abadpour

Received: date / Accepted: date

Abstract Stereopsis is the basis for numerous tasks in machine vision, robotics, and 3D data acquisition and processing. In order for the subsequent algorithms to function properly, it is important that an affordable method exists that, given a pair of images taken by two cameras, can produce a representation of disparity or depth. This topic has been an active research field since the early days of work on image processing problems and rich literature is available on the topic. Joint bilateral filters have been recently proposed as a more affordable alternative to anisotropic diffusion. This class of image operators utilizes correlation in multiple modalities for purposes such as interpolation and upscaling. In this work, we develop the application of bilateral filtering for converting a large set of sparse disparity measurements into a dense disparity map. This paper develops novel methods for utilizing bilateral filters in joint, pyramid, and doubly joint settings, for purposes including missing value estimation and upscaling. We utilize images of natural and man–made scenes in order to exhibit the possibilities offered through the use of pyramid doubly joint bilateral filtering for stereopsis.

Keywords Computational Stereopsis · Bilateral Filtering · Joint Bilateral Filtering · Pyramid Upscaling · Hole–Filling

1 Introduction

As per the taxonomy of dense stereopsis provided in [86], the approaches to this problem can be categorized into the two general classes of intensity–based and descriptor–based techniques. Intensity–based approaches, examples of which can be found in [15, 81], compare pixels, either directly [47] or through neighborhoods [89], in order to find correspondence. These approaches generally utilize a form of inference, such as Graph Cuts [18] or PDE–based methods [90], in order to compensate for the ambiguities present in the matches. Nevertheless, direct utilization of pixel information in images is known to be challenging in depth discontinuities and in texture–less areas [19]. The reader is referred to the work by Hirschmuller [42] for a state-of–the–art technique that utilizes Mutual Information–based matching smoothed through a semi–global constraint.

Historically, the first approaches to computational stereo utilized descriptor–based techniques. The first class of such algorithms selected points of interest in one image and then utilized block–based search methods in order to locate the corresponding points on the other image [39, 66, 7, 5, 54, 68, 71, 17, 63, 44, 16]. These methods produce a sparse depth representation, which could theoretically be converted into a dense map through regularization or other surface interpolation techniques [93]. Nevertheless, this next stage is not necessarily a component in a number of the published works (e.g. [54]). Barnard and Fischler [11] and Dhond and Aggarwal [24] provide reviews of related works published in the 70's and the 80's, respectively. A more recent review of the advances in computational stereo is written by Brown, Burschka, and Hager [19].

The core premise of a descriptor–based approach to stereopsis is that there are points in an image of a scene which are distinct and can be detected in other images of the same scene [23]. A generic descriptor–based stereopsis technique utilizes a local image descriptor such as SIFT [57, 58], SURF [13], DAISY [95], FAST [80], or

Arash Abadpour
Imaging Group, Epson Edge, Epson Canada Limited
Tel.: +1-905-944-3733
E-mail: arash_abadpour@ea.epson.com

GLOH [69], among others. Additionally, some works on the topic contain a proposal of novel descriptor models which are suggested to be favorable in terms of specificity, coverage, computational complexity, or other factors. For example, in [108] the authors suggest a new descriptor named Speeded-Up Local Descriptor (SULD) and report that it outperforms SURF and DAISY in the application developed in the paper.

Operators such as SIFT, SURF, and DAISY are often addressed as "strong" descriptors, because they are designed to be robust against changes in lighting and other environmental and imaging factors. They also provide different levels of robustness against changes in the viewpoint. These descriptors, however, are generally computationally expensive. More importantly, they emphasize distinction of the descriptor at the cost of lowering the number of feature points detected in an image. In essence, strong descriptors provide few reliable data points which are generally used as seed points [105] or in order to provide constraints for next levels of reconstruction which utilize less demanding approaches [90]. A comprehensive review of local descriptors is given in [69]. The reader is also refereed to [33] for an in–depth discussion of the topic.

As mentioned, in the complex compromise between distinction, density, and computational cost, strong descriptors favor distinction and act weaker on density and execution speed. As a remedy to this challenge, classes of "weak" descriptors have been designed which lean towards density and speed. Several research projects have tried primitive image features, such as corners [40], edges [102, 14], curves [87], collinear connected edge points [68], edge orientations [4], edges on epipolar lines [7, 71], waveforms in the scan–lines of the images [44], and invariant regions [98] in order to find correspondence. These weak descriptors cannot be reliably matched unless other pieces of information are included in the process. For example, the early work by Grimson [37] tried to match the zero–crossings of the Laplacian of Gaussian within a window–size. In effect, the utilization of descriptor–weak or descriptor–less features adds the requirement for heavy post–processing in order for the disambiguation of feature matches [67]. This additional step inflates the computational cost of the algorithm and is a potential source of error.

In this paper, we address the problem of producing a dense disparity map using sparse correspondences detected through the utilization of a low–strength and high–density descriptor. The problem to be solved in this setting is estimating the disparity for pixels which are not included in the sparse representation. We address this need through producing a dense disparity map at a base resolution and subsequently upscaling this representation to the full resolution of the input images. In this work, we automatically find the scale at which a quasi–dense representation can be created and then utilize joint bilateral filters for estimating missing values and also for upscaling the disparity map to full resolution. Here, we hybridize the bilateral filter with a Simulated Annealing process and also execute it within a pyramid upscaling procedure guided by two images. To the best of our understanding, the method described in this paper does not resemble any of the methods listed in Scharstein and Szeliski's taxonomy [86].

The input set of matching points in this work is collected through the execution of the Kanade–Lucas–Tomasi (KLT) Feature Tracker [59, 96]. Hence, we select the low–strength feature detector Harris Corner [40] and gain a high density of feature points on the input images. A byproduct of this process is the substantial reduction of computational complexity. However, we emphasize that speed of operation is not a decision factor in this design. We hypothesize that the results of this paper may be reproducible using some other descriptors which can produce a comparable density of matches. We have observed that for many conventional strong descriptors, this requires a significantly high–resolution input image, the handling of which is a contributor to inflated computational complexity.

In contrast to the framework described in [86], this work does not depend on the assumption that the two input images are rectified. However, in order to comply with the literature, we make the assumption of rectified input images and therefore produce a scalar disparity map. Hence, vertical disparity is assumed to be negligible and depth is assumed to be representable solely based on horizontal disparity. Furthermore, as a result of the process implemented in this work, dense disparity maps at some intermediate scales are produced as well. It is worth mentioning that the developed method allows for the pyramid bilateral filter to stop at any desired scale. This property is useful for applications which require a disparity map at dimensions lower than that of the original input images.

We emphasize that the intent of this work is not to develop a novel stereopsis method. Such a goal would require an extensive evaluation process against the state–of–the–art. In contrast, this paper proposes novel applications for bilateral filters, in different incarnations, within the framework of stereo matching. Hence, this work is compared to others in its class, including [78, 79, 104].

The rest of this paper is organized as follows: first, in Section 2, the literature is reviewed, then, in Section 3, the proposed method is presented and some experimen-

tal results are provided in Section 4. The concluding remarks follow in Section 5.

2 Literature Review

Marr and Poggio [62] provided a feature–based computational model for human stereopsis. Their model was later implemented and modified by Grimson [36]. The Marr–Poggio theory suggests that the human visual processor handles the computational stereo problem in five steps. These steps are combined in Grimson's implementation into feature extraction, feature matching, and match disambiguation. The Marr–Poggio theory also formulates two basic rules for stereo matching, i.e. each point in the left image can be assigned to one and only one disparity value and disparity varies smoothly except in depth discontinuities. Pollard, Mayhew, and Frisby [77] developed another theory which employs a limit on allowable disparity gradient.

There has been renewed interest in the use of local image descriptors for computational stereo in recent years. A key topic in these works is the treatment of unreliability of the matches found through the use of the selected descriptor. For example, Lim and Binford [54] describe a method and also cite previous literature which are based on a feature definition which allows for extremely low–confident matches and Veksler [101] utilizes the absolute difference in pixel values in left and right images along a scan–line in order to estimate disparity. In fact, in some cases any two feature points in the two images can be considered a match and the task of pruning erroneous matches is left to a subsequent consistency examination stage. For example, in a recent work, Smith, Zhang, and Jin consider each pixel as a feature vector and use Graph Cuts in order to estimate disparity values [88]. That method requires 15 minutes to complete in a sub–VGA image (360×480). Mu, Zhang and Li utilize Harris Corner points and employ Graph Cuts in order to perform matching [70]. Maciel and Costeira discuss the problem of correspondence when feature descriptors cannot be relied upon [60]. Recently, in 2006, Ulusoy and Hancock [99] presented an approach to finding the correspondence between oriented edges extracted using steerable filters. They produce a sparse depth representation using phase–similarity at multiple scales and cite the corresponding literature. Ayache and Faverjon [6] produce a neighborhood graph of line segments in the images and proceed with exploring the largest components of a disparity graph built from the descriptions of the two images. Some recent works, including the quasi–dense work by Lhuillier and Quan [52], mix concepts from sparse stereo matching with dense depth estimation in order to produce 3D models in matters of minutes. Elias [28] describes an approach which relies on matching features on the ground surface and then extends to features on other planes in the scene. Venkateswar and Chellappa [102] present the use of a hierarchical feature–matching scheme, which starts with matching surfaces and proceeds to match lines.

Disparity estimation is generally a midway step before depth estimation. In effect, the process necessary for determining the 3D point which corresponds to two points matched in the stereo pair depends on the model for and the placement of the two cameras. The complexity of this process can range from the more straightforward case of two cameras with parallel image planes to the general case in which the stereo system is non–parallel [6]. Determination of the depth for a pair of matching points, which is addressed as the reconstruction problem [19], is not a contribution of this paper. Here, we assume that given a disparity map, i.e. a 2D matrix which carries the disparity for each pixel, a depth map can be created. Detailed analysis of this procedure is outside the scope of this work.

Feature–based methods generally produce a sparse set of disparity representations which needs to be propagated to the rest of the image if a dense map is required. For example, Zhang and Shan [107] start with a small number of matching points, produced through feature matching or manual input, and then iteratively estimate depth values for more pixels in the input image. At each stage, in Zhang and Shan's work, the available set of depth estimations produces the search range for pixels for which depth has not been estimated yet. A similar approach is utilized by Lhuillier and Quan [50], where, instead of the simultaneous multi–point scheme utilized by Zhang and Shan, Lhuillier and Quan search for additional matches in the 5×5 neighborhood of each previously matched point in a sequential framework. A similar strategy is employed by Chen and Medioni [22] using a volumetric representation. Level Sets [29], PDE–based methods [2, 90], Dynamic Programing [7,35,84], Expectation Maximization [89], Graph Cuts [81,18,47], Freespace theorem [92], Region Growing [72,34,38,30], Surface Fitting [43], Markov Random Fields [89,55,56], Simulated Annealing [10], and Space Carving [49] are some of the methods utilized in order to propagate disparity to the surface of the image. Some of these approaches are not applicable to close–to–realtime settings. For example, a typical PDE–based method requires matter of minutes before convergence [91] and subsequent iterations of the operation in an algorithm which utilizes Markov Random Fields may take over five minutes to complete [55]. Other techniques may require several seconds to con-

verge on close–to–VGA resolutions [51]. The reader is referred to [83] for a review.

Bilateral filtering [97] is an alternative to anisotropic diffusion [74] which has found many applications in image filtering. This is mainly because many conventional linear approaches to data filtering produce blurred edges in depth maps [48,8]. Common non–linear approaches, such as median or morphological filtering, on the other hand, take object boundaries into account but can cause severe image–depth misalignment [79]. These approaches, therefore, are not appropriate for depth maps and other measurement matrices which represent geometrical displacement. On the contrary, a bilateral filter is designed to perform interpolation while preserving edges. The mathematical model for bilateral filters can be related to Adaptive Smoothing [82,8], Weighted Gaussian Filtering [46], and Guided Filters [41]. The weight function concept from bilateral filtering is also utilized for upgrading the box filter used in common block–based stereo matching methods into a content–aware entity which emphasizes pixels in the blocks to be matched based on the central pixel(s) [64,65,53]. Ansar, Castano, and Matthies [3] utilize a bilateral filter as a pre–processing stage before block–matching in stereopsis. The reader is referred to two reviews of the theory and some of the applications of bilateral filtering for further details [27,73].

Joint bilateral filtering is the process of using a guidance image in order to filter an existing image. During this process, the guidance image provides additional constraints or pieces of information. For example, joint bilateral filters have been used for combining high–frequency components from one image with low–frequency components of another image [76,26]. This operator can be also utilized in order to upscale an image [48]. In this process the guidance image and the output image have higher dimensions than the input image. Kapf et al. [48] utilize a joint bilateral filter in order to upscale a depth map using one of the images in the pair. This approach is also used for, among others, aligning depth map and visible–range images through successive downscaling and upscaling [31] and for Time of Flight (TOF) upscaling [32]. A multi–dimensional generalization of this method is given by Dolson et al. [25]. Chan et al. [21] applied joint bilateral upscaling on low–resolution depth images using the visible–range image as guidance. Varekamp and Barenbrug [100] utilize joint bilateral filtering to propagate depth information to successive frames in order to adapt to motion in the scene.

Multi–level bilateral filtering has been investigated by the researchers as well. For example, Ramanath and Snyder [78] employ a bilateral filter in order to upscale up to a fixed small scale, but do not use a guidance image and Sawhney et al. [85] utilize a method similar to bilateral filtering in order to upscale stereoscopic pairs. Riemens et al. [79] proposed a variant on joint bilateral filtering for the cases in which the scale factor is 8. In contrast with the Kopf et al. [48] approach of single–stage upscaling, Riemens et al. proposed to perform the upscaling in three consecutive stages ($2\times2\times2$). They also utilized anti-aliasing pre–filtering in order to reduce artifacts and showed that the method can be deployed to consumer–grade DSPs [31]. In that work, they utilize a box filter instead of the more generic definition of the spatial weight function. Implications of this option are discussed in more detail by Buades et al. [20]. Reference to multi–scale bilateral filtering has also been made by Xiao et al. [104] in their work on upscaling texture content.

3 Proposed Method

The method developed in this paper starts with finding many pairs of matching points in the input stereo images. This set in effect produces a sparse disparity representation. Then, we find an optimal scale in which the sparse disparity representation can be converted to a quasi–dense disparity map. Subsequently, we estimate values for missing areas in the quasi–dense disparity representation and produce a dense disparity map. This map is at a lower scale than the input images. Here, we develop a novel missing value estimation process which utilizes a joint bilateral filter executed within a Simulated Annealing–style operation. Finally, we upscale the dense disparity map into the full resolution of the input images using a doubly joint pyramid bilateral filter developed in this paper. Figure 1 shows the flowchart of the method developed in this paper.

The rest of this section describes the building–blocks in Figure 1, as follows. First, Section 3.1 introduces the notations used in this paper. Then, Section 3.2 describes the utilization of a feature detection and matching framework as the provider of input to the rest of the process. Subsequently, the process of creating a sparse and then a quasi–dense disparity map are reviewed in Sections 3.3 and 3.4, respectively. The quasi–dense disparity map contains areas where the value of disparity is unknown. This issue is addressed through the method outlined in Section 3.5, where a dense disparity map is produced. Finally, the disparity map is upscaled to the required scale, as described in Section 3.6. We visualize the flow of the process through exhibiting the output of each step for the sample pair shown in Figure 2.

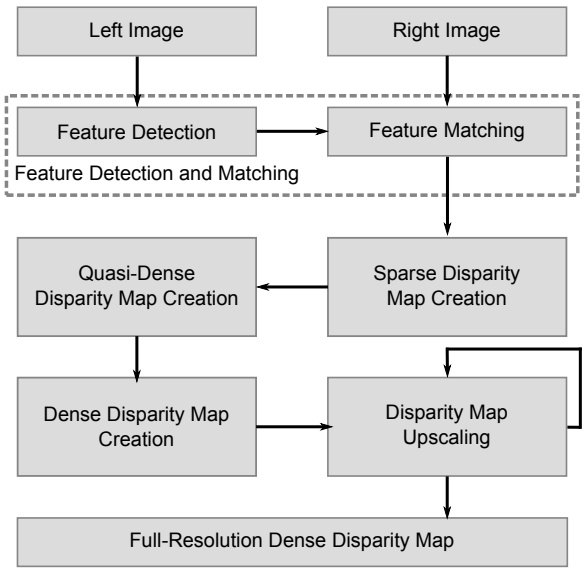
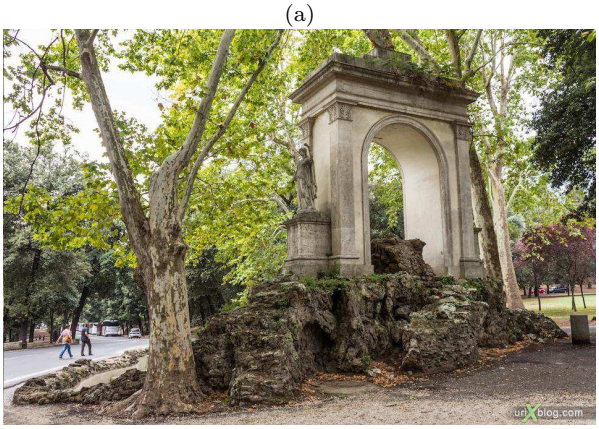


Fig. 1 Flowchart of the developed method. Details of the building blocks of this flowchart are given in Section 3.

(a)



(b)

Fig. 2 Sample input for the developed method. (a) Left image. (b) Right image. Original images by Yury Golubinsky, used under permission.

3.1 Notations

We assume that two same–sized input images, corresponding to a stereo pair, are given and denote them as $\mathbf{I}_L$ and $\mathbf{I}_R$, for the left and the right images, respectively. Both images have the dimensions of $H \times W \times C$. Here, C is the number of color channels, most likely to be either 1 or 3, but not restricted to these values. The acceptable values for C are determined by the two functions $F(\cdot)$ and $\tilde{F}(\cdot)$, described below. The method developed in this paper produces an $H \times W$ disparity map, denoted by $\mathbf{D}$.

We assume that there exists a function $F(\cdot)$, which when given an image, produces a set of points of interest, each potentially accompanied with a descriptor. This function effectively carries out feature detection in the left image. The function $F(\cdot)$ also produces the descriptor corresponding to each detected feature point. We further assume that there exists a function $\tilde{F}(\cdot)$ that takes as input an image, a set of feature points, and the corresponding descriptors. This set is assumed to have been produced through a separate call to the function $F(\cdot)$ for another image. The two images are assumed to be of the same scene taken from two slightly different vantage points. Here, we assume that the spatial and intrinsic variations between the two images comply with the requirements of $\tilde{F}(\cdot)$. In other words, for a particular $\tilde{F}(\cdot)$, which is, for example, capable of handling radiometric variations to a known extent, we allow the input images to vary accordingly.

Given the described input, the function $\tilde{F}(\cdot)$ produces a set of pairs of points, where each pair denotes that two points in the two images under consideration appear to correspond to the same physical location. Note that $\tilde{F}(\cdot)$ may perform feature matching or feature tracking and that it may internally implement geometrical constraints on points which are candidates to be potential matches. Details of the internal operations of the functions $F(\cdot)$ and $\tilde{F}(\cdot)$ are beyond the scope of this paper.

In this paper, we denote the downscaled version of an image $\mathbf{I}$ at the positive integer scale λ as $\mathbf{I}_{[\lambda]}$. Here, we assume that the height and the width of $\mathbf{I}$ are divisible by λ. We assume that a proper downscaling function exists which given the image $\mathbf{I}$ and an integer scale λ will produce $\mathbf{I}_{[\lambda]}$. We denote sparse maps using the superscript $\circ$. For example $\mathbf{I}^{\circ}$ is a sparse map.

When the integer value a divides the integer value b, we write $a|b$ and we denote the Greater Common Divisor of the two integer values a and b as $gcd(a, b)$. We denote the sorted set of descending positive integer values which divide the integer value a as $\Lambda(a)$.

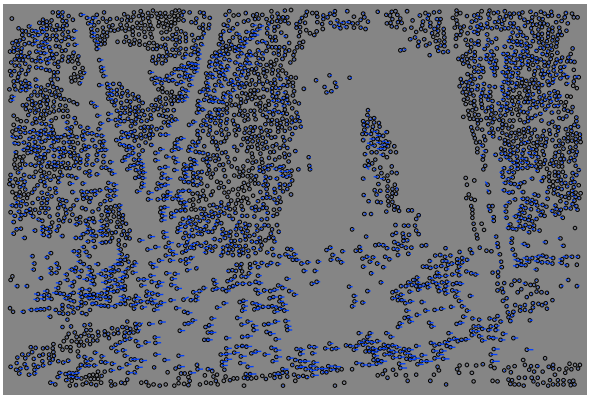


Fig. 3 Feature points detected and tracked on the two inp images shown in Figure 2.

Fig. 4 Sparse disparity representation corresponding to the two input images shown in Figure 2.

3.2 Feature Detection and Matching

This stage corresponds to the first block in Figure 1. Here, we input two images, $\mathbf{I}_L$ and $\mathbf{I}_R$, and produce a set of pairs of matching points in the left image and the right image. This set represents a sparse disparity map for the input images. The details of the process followed in this stage is outlined below.

We first pass $\mathbf{I}_L$ through $F(\cdot)$ and produce a set of feature points and the corresponding descriptors. Then, we pass $\mathbf{I}_R$ and the acquired feature representations through $\tilde{F}(\cdot)$ and produce a set of matching points in $\mathbf{I}_L$ and $\mathbf{I}_R$. We denote this set as $\mathbf{P} = \{(\mathbf{p}_L, \mathbf{p}_R)\}$. Here, we assume that the two images are rectified. Hence, vertical displacement between two matching points must be negligible. Therefore, we only accept a pair of points which are at most 1% of image height apart in the vertical direction. We also impose a maximum 20% of width constraints on the horizontal axis. Introduction of additional constraints, such as the method described in [106], may reduce the possibility of mismatch at this stage.

In the current implementation of this work, feature points are detected using Harris Corner Detector [40] and are tracked using the Kanade–Lucas–Tomasi (KLT) Feature Tracker [59,96]. The code for this paper is based on the implementation of these two modules contained in IVT [1].

Figure 3 shows the set of pairs of points produced through this process for the input images shown in Figure 2. Here, the points are visualized on the left image. In each pair, $\mathbf{p}_L$ is visualized as a circle and $\mathbf{p}_R - \mathbf{p}_L$ is drawn as a short blue line. This set contains 3738 pairs of points. Note that in certain areas no matching pairs are found. The process of finding feature points in the left image and tracking them in the right image in this experiment takes 262.76 milliseconds to complete. The bulk of this time is spent on feature tracking (212.81 milliseconds).

3.3 Sparse Disparity Map Creation

In this stage, the set of pairs of matching points produced in Section 3.2 is converted to a sparse disparity representation.

Given a pair of matching points denoted as $(\mathbf{p}_L, \mathbf{p}_R)$, we produce the new representation $(\mathbf{p}_L, \mathbf{d})$, where $\mathbf{d} = \mathbf{p}_L - \mathbf{p}_R$ denotes the disparity vector. As stated in Section 3.2, constraints on the the acceptable region for $\mathbf{d}$ are utilized in order to remove incorrect matches. In the current implementation, the camera system is composed of two parallel cameras at a fixed base–line and disparity is always parallel to the horizontal line. Therefore, we reduce the disparity representation into scalar disparity values, i.e. $(\mathbf{p}_L, d)$, where d corresponds to the horizontal coordinate of the disparity vector. Through this process, the following sparse representation is created,

$$\mathbf{P} = \{\, (\mathbf{p}_n, d_n) \,\}, n = 1, \cdots, N_{\mathbf{P}}. \tag{1}$$

In systems which maintain a more complicated setting, where $\mathbf{d}$ cannot be reduced to a scalar value, we propose to utilize the rest of the process on two disparity maps, i.e. vertical and horizontal, independently.

Figure 4 shows the sparse disparity representation corresponding to the images shown in Figure 2.

3.4 Quasi–Dense Disparity Map Creation

In this stage, the sparse disparity representation produced in Section 3.3 is converted to a quasi–dense disparity representation at a scale lower than the input

images. In effect, we seek a scale at which disparity values for a majority of the pixels can be inferred based on the available sparse disparity representation. In effect, we propose an algorithm which finds an optimal scale which is a proper compromise between coverage and detail. In other words, if scale was the only concern, the sparse representation would be considered as a quasi–dense representation with many missing data elements. On the contrary, if coverage was the only concern, the sparse disparity representation would be downscaled to a 1×1 dense representation which exhibits 100% coverage. The algorithm outlined in this stage finds the compromise between these two extreme cases.

We first note that for any given integer scale λ, which satisfies $\lambda \big| H$ and $\lambda \big| W$, the set $\mathbf{P}$, given in (1), can be downscaled in order to produce a disparity map of the size $\frac{1}{\lambda}H \times \frac{1}{\lambda}W$, denoted by $\mathbf{D}^{\circ}{}_{[\lambda]}$. The process for creating this sparse disparity map is to fill in any element in $\mathbf{D}^{\circ}{}_{[\lambda]}$ with the median of the disparity values of all elements of $\mathbf{P}$ which fall into that particular bin when $\mathbf{p}_n$ is scaled by a factor of λ. Following this procedure, however, some elements in $\mathbf{D}^{\circ}{}_{[\lambda]}$ will have no corresponding entity in $\mathbf{P}$ and are therefore assigned to a disparity value of UNKNOWN. We denote as $\mathbf{O}_{[\lambda]}$ the *occupancy map* of $\mathbf{D}^{\circ}{}_{[\lambda]}$, i.e. locations where disparity is not UNKNOWN (the use of UNKNOWN disparity or depth has precedence in the literature [107]).

Given that the input images have the dimension of $H \times W$, the number of non–UNKNOWN pixels in $\mathbf{D}^{\circ}{}_{[\lambda]}$ is an integer between 0 and $\frac{1}{\lambda^2}HW$. We map this number to $[0, 1]$ and address it as *occupancy* at scale λ, denoted as $\varphi(\lambda)$. Note that for the special case of $W = H$, $\varphi(W)$ is equal to 1.0, the maximum. This is assuming that at least one pair of matching features points has been detected in the input images. Note that, generally, $\varphi(\lambda)$ is an increasing function of λ.

We define the problem of finding the optimal scale for quasi–dense disparity map creation as finding the smallest value of λ for which $\varphi(\lambda)$ is bigger than a predefined threshold. In other words, defining $\lambda_{max} = gcd(W, H)$, the function $\varphi(\lambda)$ is defined for all positive integer values of λ for which $\lambda \big| \lambda_{max}$. The aim of the scale detection stage, then, is to find the smallest λ for which $\varphi(\lambda) \geq \varphi_0$, where φ_0 is a preselected value. Here, we outline the process of solving this problem.

We perform exhaustive search on the ordered set $\Lambda(\lambda_{max})$ until $\varphi(\lambda)$ drops below φ_0. In practice, the value of $\varphi_0 = 0.75$ is selected in order to guarantee than at least $\frac{3}{4}$ of the pixels have a disparity value assigned to them in the resulting quasi–dense disparity map. We address the selected scale as λ_0 and produce the corresponding sparse disparity map and occupancy map, denoted by $\mathbf{D}^{\circ}{}_{[\lambda_0]}$ and $\mathbf{O}_{[\lambda_0]}$, respectively.

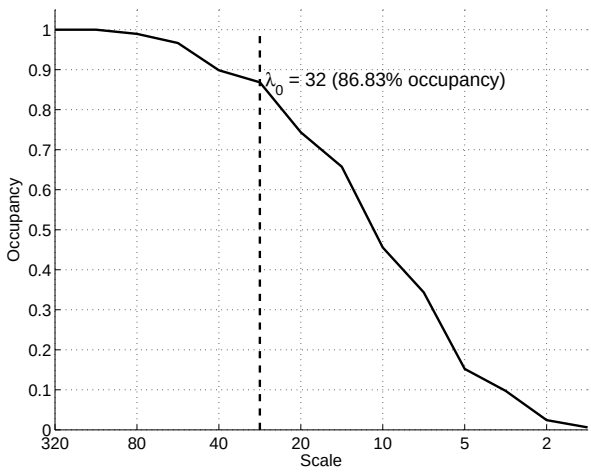


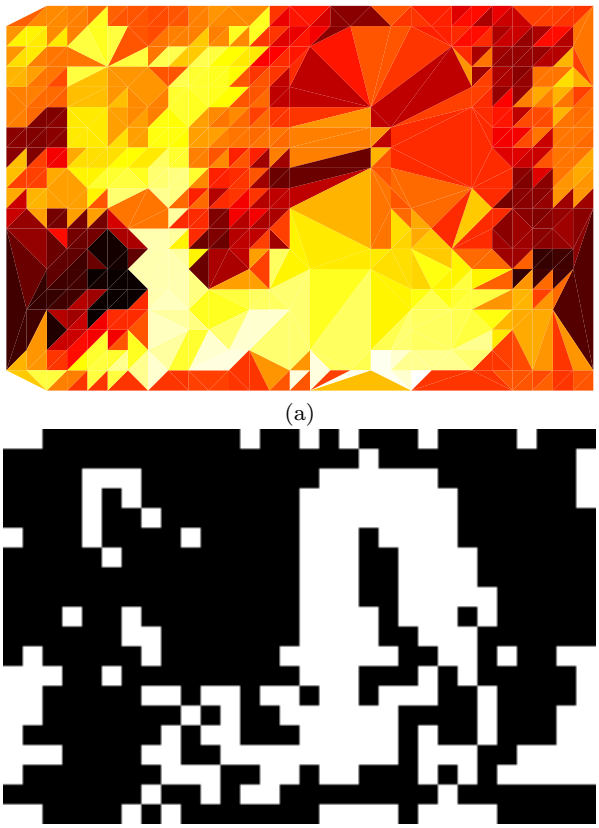
Fig. 5 Values of occupancy, $\varphi(\lambda)$, for the sparse representation shown in Figure 4. Here, $\lambda_0 = 32$ is selected by the algorithm, resulting in an occupancy of 86.83%.

Figure 5 shows values of $\varphi(\lambda)$ for the sparse disparity map shown in Figure 4. Here, the algorithm decides that the optimal scale for quasi–dense disparity map production is 32. This value corresponds to an occupancy of 86.83%. Figures 6–(a) and 6–(b) demonstrate the corresponding quasi–dense disparity map and occupancy map, respectively. The inputs images in this experiment are 640×960 pixels. The quasi–dense disparity map, therefore, has a resolution of 20×30. The process of producing this quasi–dense disparity map takes 1.31 milliseconds to complete in this experiment.

We address the zero sections in $\mathbf{O}_{[\lambda_0]}$ as *holes*. In the next stage of the algorithm, outlined in Section 3.5, we produce a dense $\mathbf{D}_{[\lambda_0]}$, i.e. a disparity map which corresponds to an all–one occupancy map. Before that, a note on the median filter utilized in the downscaling process is necessary.

As discussed above, the process of downscaling the sparse disparity representation involves applying a median operator on all elements which correspond to a particular entity in the quasi–dense disparity map at a particular scale. From the perspective of algorithmic efficiency, however, conventional median operators are not favored, as they require a list of all input values to be created first and then processed as a whole. Here, we present an approximation to the median operator which works on a stream of input variables at a computational complexity comparable to that of calculating the running average.

Fast implementations of the median filter for the purpose of window–based application on 2–D images are available in the literature [61]. Among these methods are the ones which require a histogram to be created [45] and the ones which rely on the exploitation



(a)

(b)

Fig. 6 Quasi–Dense disparity map corresponding to the sparse map shown in Figure 4. (a) Quasi–Dense disparity map. (b) Occupancy map. Unoccupied pixels are noted as white.

of the repeated references to the same elements during the calculation of filter output for consecutive pixels in the output image [103]. Such approaches are generally either tailored for integer–valued signals or, more specifically, for 2–D images [75]. Another class of median operators requires the inputs to the filter to be available in one accumulated list [12]. The requirements of the current algorithm, however, favor a method which can function on a stream of input values without the need to store the list in a temporary container. In this work, we utilize the idea presented in [94] and implement an approximation of the median filter which utilizes one pass through the data. We emphasize that this operator is applied on the separate pixels in the downscaled disparity representation independently.

We assume that a set of N realizations of the random variable x are given as the set $\mathbf{X} = \{x_1, \cdots, x_N\}$, $x_n \in \mathbb{R}$. We denote the mean, the median, and the variance of x as μ, m, and σ and write,

$$|\mu - m| = \Big| E\{x - m\} \Big| \leq E\Big\{ |x - m| \Big\} \leq \tag{2}$$

$$E\Big\{ |x - \mu| \Big\} \leq \sqrt{E\Big\{ (x-\mu)^2 \Big\}} = \sqrt{\sigma}.$$

Hence,

$$\mu - \sqrt{\sigma} \leq m \leq \mu + \sqrt{\sigma}. \tag{3}$$

Here, we have used Jensen's inequality and the fact that the median is the minimizer of $G(y) = E\{|x - y|\}$.

Here, we propose to use the mean of all x_is which satisfy $|x_i - \mu| \leq \sqrt{\sigma}$ as an approximation for the sample median. Moreover, as the values of μ and σ are not known, we propose to use the values of sample mean and sample variance instead. Additionally, we utilize the running values of these identities. The details of this procedure are outlined below.

We propose to set up a state machine with the four state variables μ_n, σ_n, m_n, and $\tilde{n}_n$, all of which are initialized to zero at the beginning of the process. We will show that as the data is streamed through the filter, these state variables will carry sample mean, sample variance, an approximation of sample median, and the number of elements utilized in the calculation of the sample median, respectively. Here, we define the following update functions for the state variables,

$$\mu_{n+1} = \frac{n}{n+1}\mu_n + \frac{1}{n+1}x_{n+1}. \tag{4}$$

$$\sigma_{n+1} = \frac{n-1}{n}\sigma_n + \frac{1}{n+1}\left(x_{n+1} - \mu_n\right)^2, n \geq 1. \tag{5}$$

$$m_{n+1} = \begin{cases} \dfrac{\tilde{n}_n}{\tilde{n}_n + 1} m_n + \dfrac{1}{\tilde{n}_n + 1} x_{n+1}, & \left|x_{n+1} - \mu_n\right|^2 \leq \sigma_n \\ m_n, & \text{otherwise} \end{cases} \tag{6}$$

$$\tilde{n}_{n+1} = \begin{cases} \tilde{n}_n + 1, & \left|x_{n+1} - \mu_n\right|^2 \leq \sigma_n \\ \tilde{n}_n, & \text{otherwise} \end{cases} \tag{7}$$

One may use induction on (4) and (5) to prove that,

$$\mu_n = \frac{1}{n}\sum_{i=1}^{n} x_i. \tag{8}$$

$$\sigma_n = \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - \mu_n\right)^2. \tag{9}$$

Also, $\tilde{n}_{n+1}$, updated through (7), counts the number of elements in $\mathbf{X}$ which satisfy $\left|x_{n+1} - \mu_n\right|^2 \leq \sigma_n$. Hence, m_n is the mean of all input elements which satisfy $\left|x_{n+1} - \mu_n\right|^2 \leq \sigma_n$, as required.

A note on the response of this state machine to the cases where $\mathbf{X}$ contains too few elements is important. In fact, $\tilde{n}_n$ is zero for sets which contain 1 or 2 elements and for a set with 3 values the operator accepts a non–zero value for $\tilde{n}_n$ only if,

$$\left| x_3 - \frac{x_1 + x_2}{2} \right| \leq \frac{\sqrt{2}}{2}\left|x_2 - x_1\right|. \tag{10}$$

3.5 Dense Disparity Map Creation

The quasi–dense disparity map, $\mathbf{D}^{\circ}{}_{[\lambda_0]}$, produced through the process outlined in Section 3.4, contains areas where disparity is not known. These areas correspond to feature–less portions of the input images as well as to occluded areas. This section describes a developed process for converting $\mathbf{D}^{\circ}{}_{[\lambda_0]}$ to its dense version, which we denote as $\mathbf{D}_{[\lambda_0]}$. This process takes use of the downscaled version of the left image, i.e. $\mathbf{I}_{L[\lambda_0]}$, and utilizes an iterative Simulated Annealing–style joint bilateral filtering procedure, as follows.

Here, we use the discrete bilateral filter notation used by Barash [9],

$$\mathbf{I}^{\star}(\mathbf{x}) = \frac{\sum_{\boldsymbol{\delta}\in\mathbf{S}} \omega(\mathbf{x}, \mathbf{x}+\boldsymbol{\delta})\, \mathbf{I}(\mathbf{x}+\boldsymbol{\delta})}{\sum_{\boldsymbol{\delta}\in\mathbf{S}} \omega(\mathbf{x}, \mathbf{x}+\boldsymbol{\delta})}. \tag{11}$$

In this model, $\mathbf{I}$ is the image to be filtered and $\mathbf{I}^{\star}$ is the output. Here, $\mathbf{S}$ is the *domain* of the bilateral filter, also addressed as its *kernel*, or *window*. Generally, this set is defined as a square or circle of known dimensions. We denote image content at position $\mathbf{x}$ by $\mathbf{I}(\mathbf{x})$ and $\omega(\cdot)$ is the *weight function*. In general, this function depends on the difference between the image values at the two input points as well as their relative geometric positions, i.e. $\omega(\cdot)$ is a function of both $\|\boldsymbol{\delta}\|$ and $\|\mathbf{I}(\mathbf{x}+\boldsymbol{\delta}) - \mathbf{I}(\mathbf{x})\|$.

One commonly accepted category of weight functions considers an $\omega(\cdot)$ which is separable into an spatial component, $\omega_s(\cdot)$, and a range component, $\omega_r(\cdot)$, i.e., [48]

$$\omega(\mathbf{x}, \mathbf{x}+\boldsymbol{\delta}) = \omega_r(\|\mathbf{I}(\mathbf{x}+\boldsymbol{\delta}) - \mathbf{I}(\mathbf{x})\|)\, \omega_s(\|\boldsymbol{\delta}\|). \tag{12}$$

As seen in (12), the spatial component depends on the difference between image values while the range component attenuates the influence of pixels in the boundaries of the filter domain according to their distance to the central pixel.

Joint or cross bilateral filtering, shown in (13), is a particular bilateral filter which utilizes an additional guidance image. Here, it is assumed that the input image and the guidance image are correlated. For example, the input image may contain depth information and the guidance image may carry visible–range information corresponding to the same scene. Here, the guidance image is denoted as $\mathbf{J}$.

In this paper, we utilize a joint bilateral filter which can accept input images containing UNKNOWN values. As such, the output of the developed filter is non–UNKNOWN for a subset of the UNKNOWN input pixels in each iteration. We conclude the process when no UNKNOWN pixel remains in the image. We provide measures which guarantee that the process concludes in finite time. In effect, the variant of the joint bilateral filter developed here shares a structure similar to (13) with the variation that the weight function also contains a *known* component which is one iff $\mathbf{I}(\mathbf{x}+\boldsymbol{\delta})$ is not UNKNOWN. Note that here we treat $\frac{0}{0}$ as UNKNOWN, therefore, for an UNKNOWN pixel which does not have any non–UNKNOWN useable neighbours, the filter output is UNKNOWN.

The joint bilateral filter developed in this work for estimating values for the missing areas is formalized in (14). Here, we utilize the downscaled version of the left image as the guidance image and redefine the domain of the filter as any UNKNOWN pixel in $\mathbf{D}^{\circ}{}_{[\lambda_0]}$ which has a constrained difference to the central pixel in the guidance image. Note that in contrast to the generic bilateral filter, here the domain is pixel specific.

$$\mathbf{S}_{\mathbf{x}} = \left\{ \boldsymbol{\delta} \middle| \boldsymbol{\delta} \in \mathbf{S}, \mathbf{D}^{\circ}{}_{[\lambda_0]}(\mathbf{x}+\boldsymbol{\delta}) \neq UNKNOWN, \left| \mathbf{I}_{L[\lambda_0]}(\mathbf{x}+\boldsymbol{\delta}) - \mathbf{I}_{L[\lambda_0]}(\mathbf{x}) \right| < \kappa \right\}. \tag{16}$$

Here, κ is the threshold that is adjusted during the process, as outlined below. The purpose for including this variable threshold is to avoid closer pixels, which may have different content, from *invading* an UNKNOWN pixel, which has a better match at a farther position. In other words, the gradual increase of κ allows for UNKNOWN pixels to avoid early settlement and have the chance to find their potentially better matches in later iterations.

Here, we describe the process which governs κ. Assume that minimum, maximum, and a step factor for the threshold are given and denote them by κ_{min}, κ_{max}, κ_{step}. We start the procedure by setting $\kappa = \kappa_{min}$. Then, at each iteration, we apply the filter on any UNKNOWN pixel in $\mathbf{D}^{\circ}{}_{[\lambda_0]}$ once and produce the two Boolean operators of *change* and *can–change*. Note that, in practice, in order to avoid cascading effects, we keep changes to $\mathbf{D}^{\circ}{}_{[\lambda_0]}$ in a separate copy which is downloaded to it at the end of the operation. Here, *change* denotes that at least one pixel changed from UNKNOWN to a value. The identifier *can–change* denotes the presence of at least one UNKNOWN pixel which could have changed from UNKNOWN to a known value but that the current value of κ was inadequate for this change to happen. We repeat iterations without modifying κ as long as *change* is true. Then, if *can–change* is true we update κ to $\kappa_{step}\kappa$ and restart a new iteration. The procedure stops when both *change* and *can–change* are false. Note that the value of κ_{max} is selected so that it covers the dynamic range of the guidance image. The outcome of this process is the dense disparity map $\mathbf{D}_{[\lambda_0]}$.

$$\mathbf{I}^{\star}(\mathbf{x}) = \frac{\sum_{\boldsymbol{\delta}\in\mathbf{S}} \omega_r\left(\|\mathbf{J}(\mathbf{x}+\boldsymbol{\delta}) - \mathbf{J}(\mathbf{x})\|\right)\omega_s\left(\|\boldsymbol{\delta}\|\right)\mathbf{I}(\mathbf{x}+\boldsymbol{\delta})}{\sum_{\boldsymbol{\delta}\in\mathbf{S}} \omega_r\left(\|\mathbf{J}(\mathbf{x}+\boldsymbol{\delta}) - \mathbf{J}(\mathbf{x})\|\right)\omega_s\left(\|\boldsymbol{\delta}\|\right)}. \tag{13}$$

$$\mathbf{D}^{\circ\star}_{[\lambda_0]}(\mathbf{x}) = \frac{\sum_{\boldsymbol{\delta}\in\mathbf{S}_{\mathbf{x}}} \omega_r\left(\left\|\mathbf{I}_{L[\lambda_0]}(\mathbf{x}+\boldsymbol{\delta}) - \mathbf{I}_{L[\lambda_0]}(\mathbf{x})\right\|\right)\omega_s\left(\|\boldsymbol{\delta}\|\right)\mathbf{D}^{\circ}{}_{[\lambda_0]}(\mathbf{x}+\boldsymbol{\delta})}{\sum_{\boldsymbol{\delta}\in\mathbf{S}_{\mathbf{x}}} \omega_r\left(\left\|\mathbf{I}_{L[\lambda_0]}(\mathbf{x}+\boldsymbol{\delta}) - \mathbf{I}_{L[\lambda_0]}(\mathbf{x})\right\|\right)\omega_s\left(\|\boldsymbol{\delta}\|\right)}. \tag{14}$$

$$\mathbf{I}^{\star}(\mathbf{x}) = \frac{\sum_{\boldsymbol{\delta}\in\mathbf{S}} \omega_r\left(\left\|\mathbf{J}(\mathbf{x}) - \mathbf{J}_{[\lambda]}\left(\frac{1}{\lambda}\mathbf{x}+\boldsymbol{\delta}\right)\right\|\right)\omega_s\left(\|\boldsymbol{\delta}\|\right)\mathbf{I}\left(\frac{1}{\lambda}\mathbf{x}+\boldsymbol{\delta}\right)}{\sum_{\boldsymbol{\delta}\in\mathbf{S}} \omega_r\left(\left\|\mathbf{J}(\mathbf{x}) - \mathbf{J}_{[\lambda]}\left(\frac{1}{\lambda}\mathbf{x}+\boldsymbol{\delta}\right)\right\|\right)\omega_s\left(\|\boldsymbol{\delta}\|\right)}. \tag{15}$$

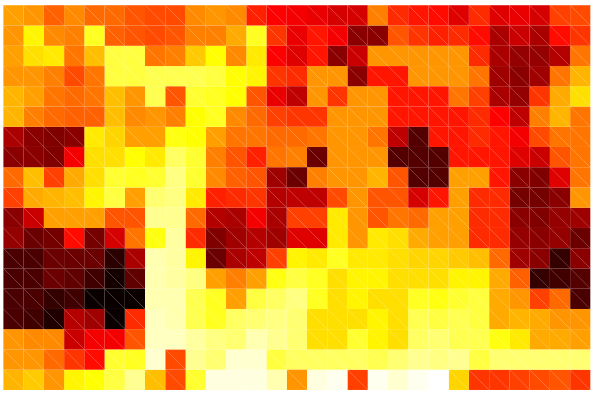


Fig. 7 Dense disparity map corresponding to the quasi–dense disparity map shown in Figure 6–(a).

In (16), $\mathbf{S}$ is the 3×3 window around $\mathbf{x}$ and excluding $\mathbf{x}$. Hence, in this case, $\|\boldsymbol{\delta}\| \in \{1, \sqrt{2}\}$, and we utilize,

$$\omega_s\left(\|\boldsymbol{\delta}\|\right) = \frac{1}{1+\|\boldsymbol{\delta}\|^2}. \tag{17}$$

$$\omega_r(\theta) = \frac{\kappa^2}{\kappa^2+\theta^2}. \tag{18}$$

Figure 7 shows the output of the developed method corresponding to the quasi–dense disparity map shown in Figure 6–(a). Note the regularity of the mesh in this figure compared to the missing areas in Figure 6, in which large irregular triangles are visible. The process of producing this dense disparity map takes 0.11 milliseconds to conclude.

3.6 Disparity Map Upscaling

In this section, the low–resolution disparity map $\mathbf{D}_{[\lambda_0]}$, produced through the process presented in Section 3.5, is upscaled to a larger size. Here, for convenience, we assume that the goal is to produce a disparity map at the same dimensions of $\mathbf{I}_L$. However, the same method is applicable if $\mathbf{D}_{[\acute{\lambda}]}$ is to be produced, given that $\acute{\lambda}|\lambda_0$. Here, we discuss the case of $\acute{\lambda} = 1$.

The procedure proposed in this section involves the process of performing upscaling at an arbitrary integer factor $\hat{\lambda}$. This upscaling procedure inputs a disparity map, denoted by $\mathbf{D}_{[\lambda\hat{\lambda}]}$, and a guidance image, denoted by $\mathbf{I}_{[\lambda]}$, and produces the upscaled disparity map $\mathbf{D}_{[\lambda]}$. Here, λ and $\hat{\lambda}$ are integers and $\hat{\lambda}$ is preferred to be a small prime number, for reasons given later. This operation can also be addressed as an upscaling with a factor of $\hat{\lambda}$ using a bilateral doubly joint filter, as described below.

We assume that the image to be upscaled is denoted by $\mathbf{I}_{[\lambda]}$. Here, we propose a process which takes use of the two guidance images $\mathbf{J}$ and $\mathbf{J}_{[\lambda]}$ in order to produce the upscaled image $\mathbf{I}$. Here, $\mathbf{J}$ and $\mathbf{I}$ have the same dimensions, which is the dimensions of $\mathbf{I}_{[\lambda]}$ and $\mathbf{J}_{[\lambda]}$ multiplied by the small prime number λ. We call this process a *Doubly* joint bilateral filter because it utilizes two guidance images.

In effect, for any given point $\mathbf{x}$ in the output image, i.e. $\mathbf{I}(\mathbf{x})$, we locate the corresponding point on the guidance images, i.e. $\mathbf{J}(\mathbf{x})$ and $\mathbf{J}_{[\lambda]}\left(\frac{1}{\lambda}\mathbf{x}\right)$, and the corresponding point on the input image, i.e. $\mathbf{I}_{[\lambda]}\left(\frac{1}{\lambda}\mathbf{x}\right)$. The doubly joint bilateral filter developed in this paper considers nine locations around $\frac{1}{\lambda}\mathbf{x}$ in order to produce $\mathbf{I}(\mathbf{x})$, as described in (15). Here, $\mathbf{S}$, the domain of the filter, is defined as the 3×3 square $[-1,0,1]^2$ in the input scale and the weight functions utilized in (15) are

given below,

$$\omega_s\left(\|\boldsymbol{\delta}\|\right) = \frac{1}{1+\|\boldsymbol{\delta}\|^2}. \quad (1$$

$$\omega_r(\theta) = \frac{C\kappa^2}{C\kappa^2+\theta^2}. \quad (2$$

Here, κ is a normalization factor and C is the numb of channels in the guidance image. We set κ equal 10 (dynamic range of the guidance images is 255). No that references to image locations with non–integer ordinates are rounded up to the nearest integer.

Here, we return to the necessity of λ being a pri number. We note that any non–prime λ can be deco posed into at least two prime factors, i.e. $\lambda = \lambda_1$ Hence, in this case, instead of performing the upscali for λ, we propose to perform the developed upscali process at the scales of λ_1 and λ_2 consecutively. We sert that the operation of (15) for a sequence of inte values produces less *Blockiness* than applying the sa operation for the product of the sequence.

Now, we address the general case of performing t upscaling for the non–prime integer scale λ_0. This p cess utilizes a pyramid upscaling scheme and produ $\mathbf{D}$ out of $\mathbf{D}_{[\lambda_0]}$ and $\mathbf{I}_{L[\lambda_0]}$.

We first factorize λ_0 into its prime elements as,

$$\lambda_0 = \prod_{i=1}^{m} {p_i}^{m_i}, p_1 < \cdots < p_m, m_i > 0, m > 0. \quad (2$$

Subsequently, we produce the following set of upscali factors.

$$p_m, p_m, \cdots p_m, p_{m-1}, \cdots, p_1. \quad (2$$

Here, each p_i is repeated m_i times. We propose to peat the upscaling procedure, described above, at t sequence of scales given in (22). It is evident that executing this process the disparity map at scale 1, i.e. $\mathbf{D}_{[1]} = \mathbf{D}$, is produced. This disparity map has the same dimensions as that of the inputs images, $\mathbf{I}_L$ and $\mathbf{I}_R$, i.e. $H \times W$.

Figure 8 shows the outputs of three consecutive executions of the developed upscaling algorithm applied on the input shown in Figure 7. Here, λ equals 2 in every iteration. Figure 9 shows the final output of the proposed upscaling procedure. This image represents the disparity map produced by the developed method for the stereo pair shown in Figure 2. It takes the algorithm 327.35 milliseconds to perform the developed upscaling. In total, the process of producing the output shown in Figure 9 from the images shown in Figure 2 takes 591.54 milliseconds.

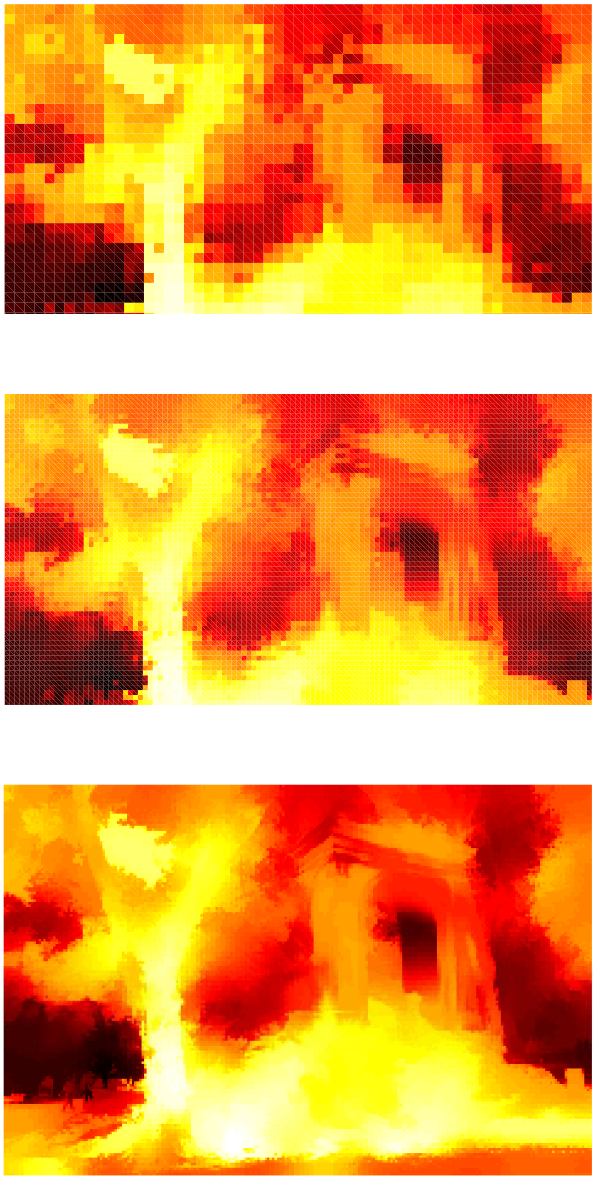


Fig. 8 Results of three consecutive executions of the developed upscaling procedures ($\lambda = 2$) on the dense disparity map shown in Figure 7.

4 Experimental Results

The proposed method is implemented as a C++ class named `Margo` and is executed on a personal computer running on Windows 7, 64bit, on an Intel Core i5–2400 CPU, 3.10GHz, with 8.00GB of RAM.

Figure 10 shows a sample output disparity map produced by the developed method. The input images in this experiment have dimensions of 704×960. The developed algorithm detects 3771 feature points in the left image and finds corresponding points for 3516 of them in the right image. The quasi–dense disparity map in this case is created at the automatically–detected scale

Fig. 9 Output disparity map produced by the developed method corresponding to the input images shown in Figure 2.

Fig. 10 Sample results of executing the developed method on a stereo pair. Left and right images courtesy of Yury Golubinsky, used under permission. Refer to text for more details.

of 64 (resolution of 11×15, occupancy of 86.06%). The total time required in order to complete the procedure in this case is 633.63 milliseconds. The time spent in the different stages is as follows: 29.88 milliseconds in feature detection, 231.19 milliseconds in feature matching, and 371.03 milliseconds in upscaling. The other operations take negligible time.

Figure 11 shows more experimental results produced by the developed method. Details about these experiments are given in Table 1.

We note that utilization of the joint bilateral filter in the developed algorithm provides a strong basis for image to disparity correlation. For example, the sample output exhibited in Figure 12–(a) shows a passage where different disparity levels are present. In Figure 12–(b), a portion of the image on the top left is magnified. Figure 12–(c) shows the corresponding disparity values produced by the developed method. As seen here, the developed algorithm is capable of differentiating between the flower tree and the house behind it.

It is important, however, to emphasize that the method developed in this paper does produce disparity artifacts where conjoined objects have contrasting colors and where strong shadows are present. In effect, the premise of the bilateral filter, utilized in this work, is that objects are geometrically close to each other when they have similar presentations in the visual–range image and vice–versa. For example, the back of the car in the bottom right of Figure 13–(a) and the shadow on the tracks in Figure 13–(b) exhibit this artifact, where close objects with contrasting colors are considered to be far from each other by the algorithm. We argue that this deficiency is a byproduct of the utilization of joint bilateral filters, where variation in the guidance image is perceived as indication that difference in the output image is expected. In effect, one may argue that in the absence of the peripheral information about the continuity of the tracks and the structure of the car, a human observer, too, would be in doubt whether the differences in color in fact denote difference in the geometry of the scene.

One of the contributions of the present work is the decomposition of the scaling factor into prime components and then iterating through these elements. This is in contrast to the works by Ramanath and Snyder [78] and Kopf et al. [48], where the scale is utilized in one stage. It is worth to mention that the use of a guidance image, as included in the present work, is not applicable to [78]. The method developed in this paper in effect provides a generalization of the approach by Riemens et al. [79], where an upscaling by the factor of 8 is decomposed into three upscaling stages, each at the scale of 2.

Here, we exhibit the significance of the decomposition stage developed in this work for arbitrary scale factors. Figure 14–(a) shows the left image in a sample stereo pair. Here, an estimate of the depth map at the scale of 32 is available and is to be upscaled to the full resolution of the input image. Figure 14–(b) demonstrates the result of performing the upscaling at a single stage, as suggested by previous works in the literature. In contrast, Figure 14–(c) exhibits the output of the process developed in this work, wherein upsclaing is carried out in five consecutive stages, each at the scale of 2. Visual comparison of Figure 14–(c) with Figure 14–(b) shows that the former contains smoother edges while the latter carries visible discrepancies and jumpiness.

5 Conclusions

This paper develops novel applications of joint bilateral filtering for the purpose of dense disparity estimation. The input to this process is a large set of matching points collected through a feature point tracking procedure. This set is converted to a sparse disparity representation and a novel method for producing a quasi–dense disparity map is developed. This method finds an optimal scale at which a properly populated disparity map can be produced. Then, a novel joint bilateral filter is developed which operates within a Simulated Annealing–style iterative process and produces a dense disparity map at a scale lower than the input images. This disparity map is subsequently upscaled into the original dimensions of the input images using a novel pyramid doubly joint bilateral filter.

The main contributions of this paper include the different utilizations of bilateral filters for the purposes of missing value estimation and upscaling. We anticipate that the process developed in this paper can be utilized for other applications which require the production of a dense map containing geometrical terms using sparse measurements.

Acknowledgments

We acknowledge the reviews received on this paper from the respected anonymous referees. The author wishes to thank Sara Ayatollahzadeh and Seyed Mohsen Amiri for their help in locating critical pieces of literature needed for this work. We also acknowledge the gen-

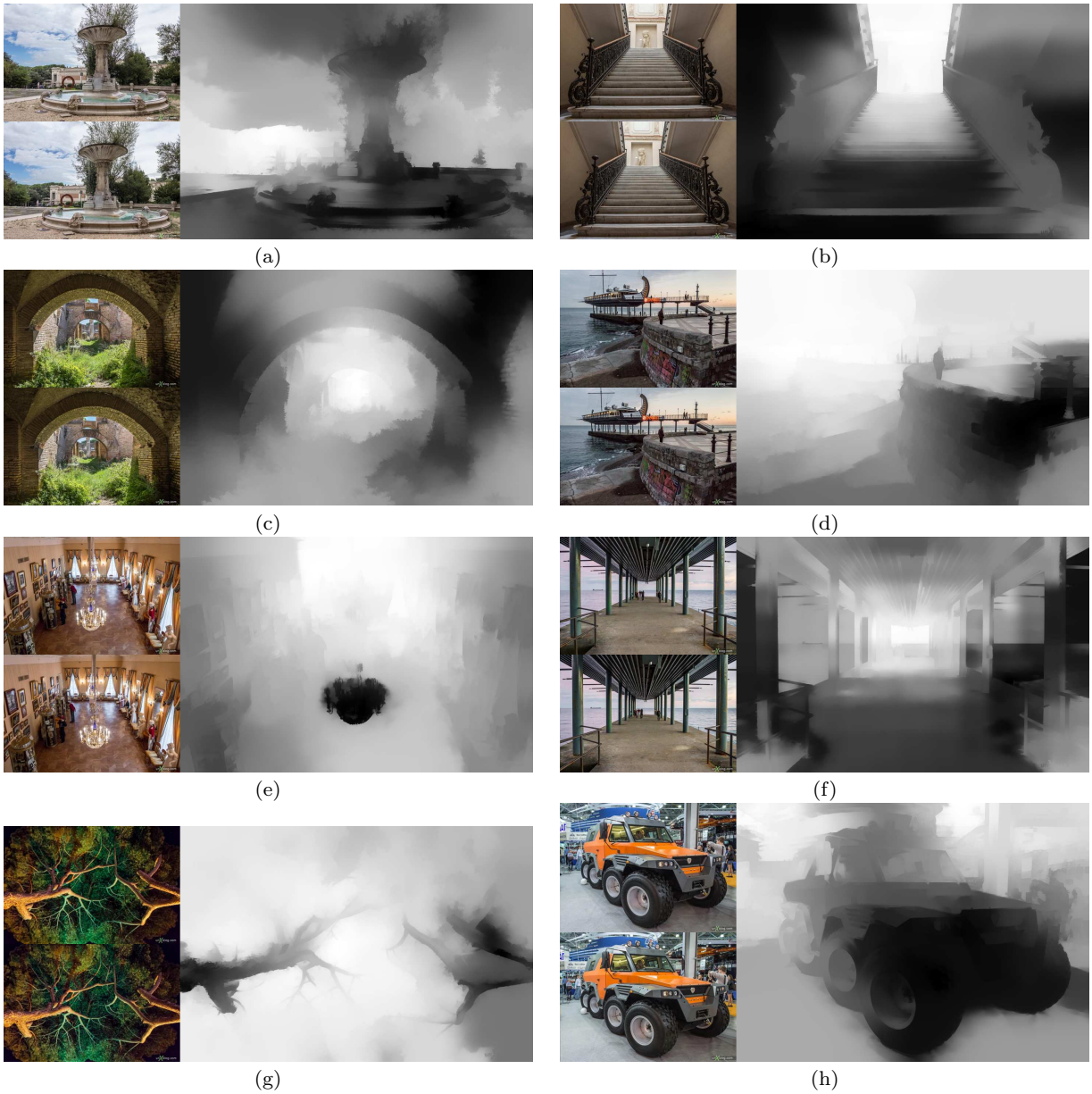


Fig. 11 Sample result of executing the developed method on stereo pairs. Left and right images courtesy of Yury Golubinsky, used under permission. Refer to Table 1 for numerical details.

erosity of Yury Golubinsky[1] for allowing us to use his photographic work to present the results of this work. The author thanks Mahsa Pezeshki for proofreading this manuscript.

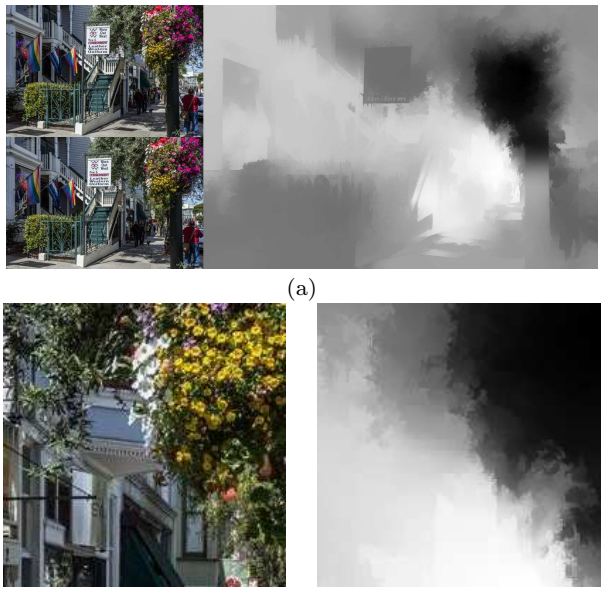
References

1. Integrating Vision Toolkit (IVT), version 1.3.20. http://ivt.sourceforge.net/ (February 2013)
2. Alvarez, L., Deriche, R., Sanchez, J., Weickert, J.: Dense disparity map estimation respecting image derivatives: a PDE and scale–space based approach. Journal of Visual Communication and Image Representation **13**(1/2), 2002 (3–21)
3. Ansar, A., Castano, A., Matthies, L.: Enhanced real–time stereo using bilateral filtering. In: 2nd International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT 2004), pp. 455–462 (2004)
4. Arnold, D.: Local context in matching edges for stereo vision. In: Image Understanding Workshop, pp. 65–72. Cambridge, Massachusetts (May 1978)
5. Arnold, R.D.: Automated stereo perception. Tech. Rep. Technical Report AIM–351, Artificial Intelligence Laboratory, Stanford University (1983)

[1] UrixBlog, Stereoscopic photographs from all over the world [RU,EN], *http://urixblog.com/*

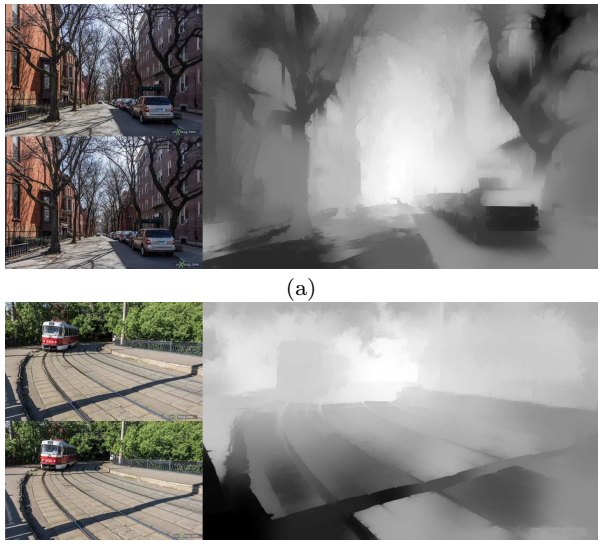
Table 1 Numerical values corresponding to the results shown in Figure 11.

Sample	Resolution	#Features		Scale	Elapsed Time			
		Left	Right		Detection	Tracking	Upscaling	Total
Figure 11–(a)	640 × 960	4000	3647	64	43.20ms	206.79ms	335.56ms	587.05ms
Figure 11–(b)	640 × 960	1474	1337	160	17.64ms	119.08ms	331.75ms	469.66ms
Figure 11–(c)	640 × 960	4000	3375	64	41.44ms	282.40ms	334.06ms	659.34ms
Figure 11–(d)	640 × 960	2077	2031	64	19.22ms	130.47ms	334.68ms	485.68ms
Figure 11–(e)	640 × 960	2200	2093	64	21.15ms	147.47ms	362.56ms	532.55ms
Figure 11–(f)	640 × 960	1268	1080	80	13.90ms	85.72ms	331.19ms	432.02ms
Figure 11–(g)	640 × 960	4000	3897	32	27.84ms	233.24ms	325.87ms	588.91ms
Figure 11–(h)	704 × 960	2083	1939	64	20.66ms	127.56ms	369.06ms	518.42ms



(a) (b) (c)

Fig. 12 Image to disparity correlation in the output of the developed method. Left and right images courtesy of Yury Golubinsky, used under permission.

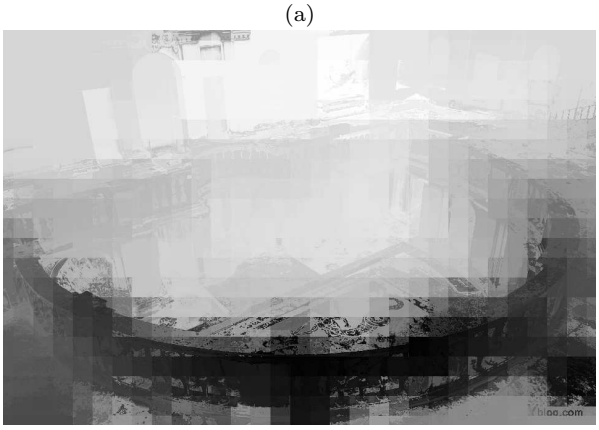


(a) (b)

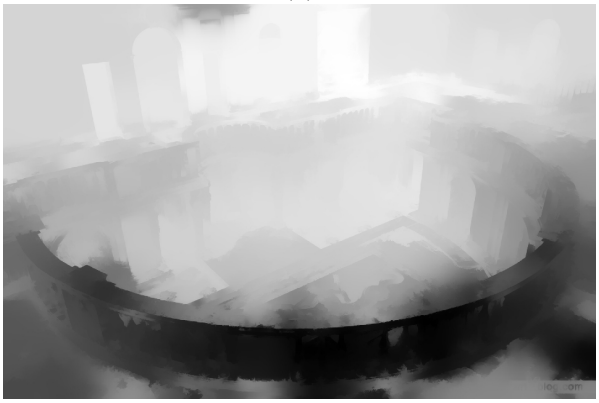
Fig. 13 Disparity artifacts produced by the developed method. (a) Different disparity for conjoined objects with contrasting colors. (b) Disparity variation in the presence of strong shadows. Left and right images courtesy of Yury Golubinsky, used under permission.

6. Ayache, N., Faverjon, B.: Efficient registration of stereo images by matching graph descriptions of edge segments. International Journal of Computer Vision **1**(2), 1987 (107–131)
7. Baker, H.H., Binford, T.O.: Depth from edge and intensity based stereo. In: International Joint Conference on Artificial Intelligence (IJCAI 1981), pp. 631–636. Vancouver, British Columbia, Canada (1981)
8. Barash, D.: Fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(6), 844–847 (2002)
9. Barash, D.: Bilateral filtering and anisotropic diffusion: towards a unified viewpoint. Tech. Rep. HPL-2000-18 (R.1), HP Laboratories Israel Technical Report, Technion City, Haifa 32000 Israel (August 2000)
10. Barnard, S.T.: A stochastic approach to stereo vision. Tech. Rep. Technical Note 373, SRI International, Menlo Park, California (April 1986)
11. Barnard, S.T., Fischler, M.A.: Computational stereo. Computer Surveys **14**(4), December 1982 (553–572)
12. Battiato, S., Cantone, D., Catalano, D., Cincotti, G., Hofri, M.: An efficient algorithm for the approximate median selection problem. In: G. Bongiovanni, R. Petreschi, G. Gambosi (eds.) Algorithms and Complexity, *Lecture Notes in Computer Science*, vol. 1767, pp. 226–238. Springer (2000)
13. Bay, H., Tuytelaars, T., Gool, L.V.: Surf: Speeded up robust features. In: European Conference on Computer Vision (ECCV 2006), pp. 404–417. Graz, Austria (2006)
14. Bignone, F., Henricsson, O., Fua, P., Stricker, M.: Automatic extraction of generic house roofs from high resolution aerial imagery. In: B. Buxton, R. Cipolla (eds.) Proceedings of the European Conference on Computer Vision (ECCV 1996), vol. 1064, pp. 83–96 (1996)
15. Birchfield, S., Tomasi, C.: A pixel dissimilarity measure that is insensitive to image sampling. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(4), 401–406 (1998)
16. Bolles, R.C., Baker, H.H., Hannah, M.J.: The JISCT stereo evaluation. DARPA Image Understanding Workshop (22nd) pp. 263–274 (1993)

(a)



(b)



(c)

Fig. 14 Comparison of the iterative upscaling process developed in this work with single–stage upscaling schemes suggested in the literature. (a) Left Image. (b) Result of performing a single–stage upscaling at the scale of 32. (c) Result of performing five consecutive upscaling stages, each at the scale of 2. Left image and the right image, not shown here, courtesy of Yury Golubinsky, used under permission.

17. Bolles, R.C., Baker, H.H., Marimont, D.H.: Epipolar–plane image analysis: An approach to determining structure from motion. International Journal of Computer Vision **1**, 7–55 (1987)
18. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence **23**(11), 1222–1239 (2001)
19. Brown, M.Z., Hager, D.B.G.D.: Advances in computational stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence **25**(8), 993–1008 (2003)
20. Buades, A., Coll, B., Morel, J.: A review of image denoising algorithms, with a new one. Multiscale Modeling & Simulation **4**(2), 490–530 (2005)
21. Chan, D., Buisman, H., Theobalt, C., Thrun, S.: A noise–aware filter for real–time depth upsampling. In: Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications (M2SFA2). Marseille, France (2008)
22. Chen, Q., Medioni, G.: A volumetric stereo matching method: application to image-based modeling. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 1999), vol. 1, pp. 29–34. Colorado (1999)
23. Collins, R.T.: A space-sweep approach to true multi-image matching. In: In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 1996), pp. 358–363. San Francisco (1996)
24. Dhond, U.R., Aggarwal, J.: Structure from stereo – a review. IEEE Transactions on Systems, Man and Cybernetics **19**(6), 1489–1510 (1989)
25. Dolson, J., Baek, J., Plagemann, C., Thrun, S.: Upsampling range data in dynamic environments. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010), pp. 1141–1148 (2010)
26. Eisemann, E., Durand, F.: Flash photography enhancement via intrinsic relighting. ACM Transactions on Graphics **23**(3), 673–678 (August 2004)
27. Elad, M.: On the origin of the bilateral filter and ways to improve it. IEEE Transactions on Image Processing **11**(10), 1141–1151 (2002)
28. Elias, R.: Sparse view stereo matching. Pattern Recognition Letters **28**(13), 1667—1678 (2007)
29. Faugeras, O., Keriven, R.: Complete dense stereovision using level set methods. In: H. Burkhardt, B. Neumann (eds.) Proceedings of the European Conference on Computer Vision (ECCV 1998), *Lecture Notes in Computer Science*, vol. 1406, pp. 379–393. Springer Berlin Heidelberg (1998)
30. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(8), 1362–1376 (2010)
31. Gangwal, O., Berretty, R.P.: Depth map post-processing for 3D–TV. In: Digest of Technical Papers International Conference on Consumer Electronics (ICCE 2009), pp. 1–2 (2009)
32. Garcia, F., Mirbach, B., Ottersten, B., Grandidier, F., Cuesta, A.: Pixel weighted average strategy for depth sensor data fusion. In: 17th IEEE International Conference on Image Processing (ICIP 2010), pp. 2805–2808 (2010)
33. Gauglitz, S., Hollerer, T., Turk, M.: Evaluation of interest point detectors and feature descriptors for visual tracking. International Journal of Computer Vision **94**(3), 1–26 (2011)

34. Goesele, M., Snavely, N., Curless, B., Hoppe, H., Seitz, S.: Multi–view stereo for community photo collections. In: IEEE 11th International Conference on Computer Vision (ICCV 2007), pp. 1–8 (2007)
35. Gong, M., Yang, Y.H.: Fast unambiguous stereo matching using reliability–based dynamic programming. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(6), 998–1003 (2005)
36. Grimson, W.E.L.: Computational experiments with a feature based stereo algorithm. Pattern Analysis and Machine Intelligence, IEEE Transactions on **PAMI–7**(1), 17–34 (1985)
37. Grimson, W.E.L.: A computer implementation of a theory of human stereo vision. Tech. Rep. A.I. Memo No. 565, Massachusetts Institute of Technology (January 1980)
38. Habbecke, M., Kobbelt, L.: A surface-growing approach to multi–view stereo reconstruction. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007), pp. 1–8 (2007)
39. Hannah, M.J.: Computer matching of areas in stereo images. Ph.D. thesis, Stanford University (1974)
40. Harris, C., Stephens, M.: A combined corner and edge detector. In: 4th Alvey Vision Conference, pp. 147–151 (1988)
41. He, K., Sun, J., Tang, X.: Guided image filtering. In: Proceedings of the 11th European conference on Computer vision: Part I (ECCV 2010), pp. 1–14 (2010)
42. Hirschmuller, H.: Stereo processing by semiglobal matching and mutual information. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(2), 328–341 (2008)
43. Hornung, A., Kobbelt, L.: Robust and efficient photo-consistency estimation for volumetric 3d reconstruction. In: The 9th European conference on Computer Vision – Volume II (ECCV 2006), pp. 179–190. Berlin, Heidelberg (2006)
44. Hsieh, Y.C., McKeown, D., Perlant, F.P.: Performance evaluation of scene registration and stereo matching for cartographic feature extraction. IEEE Transactions on Pattern Analysis and Machine Intelligence **14**(2), 214–238 (1992)
45. Huang, T.S., Yang, G.Y., Tang, G.Y.: A fast two-dimensional median filtering algorithm. IEEE Transactions on Acoustics, Speech and Signal Processing **27**(1), 13–18 (1979)
46. Izquierdo M., E., Ghanbari, M.: Texture smoothing and object segmentation using feature–adaptive weighted Gaussian filtering. In: SBT/IEEE International Telecommunications Symposium (ITS 1998), vol. 2, pp. 650–655 (1998)
47. Kolmogorov, V., Zabih, R.: Multi–camera scene reconstruction via graph cuts. In: Proceedings of the 7th European Conference on Computer Vision – Part III (ECCV 2002), pp. 82–96. London, UK, UK (2002)
48. Kopf, J., Cohen, M.F., Lischinski, D., Uyttendaele, M.: Joint bilateral upsampling. ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007) **26**(3) (July 2007)
49. Kutulakos, K.N., Seitz, S.M.: A theory of shape by space carving. International Journal of Computer Vision **38**(3), 199–218 (2000)
50. Lhuillier, M., Quan, L.: Image interpolation by joint view triangulation. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 1999), vol. 2, pp. 139–145. Colorado (1999)
51. Lhuillier, M., Quan, L.: Match propagation for image–based modeling and rendering. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(8), 1140–1146 (2002)
52. Lhuillier, M., Quan, L.: A quasi–dense approach to surface reconstruction from uncalibrated images. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(3), 418–433 (2005)
53. Li, L., Zhang, C.M., Yan, H.: Cost aggregation strategy for stereo matching based on a generalized bilateral filter model. In: R. Zhu, Y. Zhang, B. Liu, C. Liu (eds.) Information Computing and Applications, *Communications in Computer and Information Science*, vol. 105, pp. 193–200. Springer Berlin Heidelberg (2011)
54. Lim, H.S., Binford, T.O.: Stereo correspondence: Features and constraints. In: DARPA Image Understanding Workshop, pp. 373–380. Miami Beach, Florida (December 9–10, 1985)
55. Louw, M., Nicolls, F.: Improved sparse correspondence resolution using loopy belief propagation with MRF clique based structure preservation. In: The Seventh IASTED International Conference on Visualization, Imaging and Image Processing, pp. 88–93. Palma de Mallorca, Spain (2007)
56. Louw, M., Nicolls, F.: Accelerated sparse feature correspondence resolution using loopy belief propagation with MRF clique based structure preservation. In: The Eleventh IASTED International Conference on Computer Graphics and Imaging (CGIM 2010), pp. 109–116. Innsbruck, Austria (February 2010)
57. Lowe, D.G.: Object recognition from local scale–invariant features. In: IEEE International Conference on Computer Vision (ICCV), pp. 1150–1157. Toronto, Canada (1999)
58. Lowe, D.G.: Distinctive image features from scale–invariant keypoints. International Journal of Computer Vision **60**(2), 91–110 (2004)
59. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: International Joint Conference on Artificial Intelligence (IJCAI 1981), pp. 674–679. Vancouver, British Columbia, Canada (1981)
60. Maciel, J., Costeira, J.P.: A global solution to sparse correspondence problems. IEEE Transactions on Pattern Analysis and Machine Intelligence **25**(2), 187–199 (2003)
61. Marcus, R.C., Ward, W.C.: DP: a fast median filter approximation. Tech. Rep. LA–UR–13–25331, Los Alamos National Laboratory (2013)
62. Marr, D., Poggio, T.: A computational theory of human stereo vision. Proceedings of Royal Society of London **B204**, 301–328 (1979)
63. Matthies, L., Kanade, T., Szeliski, R.: Kalman filter–based algorithms for estimating depth from image sequences. International Journal of Computer Vision **3**(3), 209–236 (1989)
64. Mattoccia, S., Giardino, S., Gambini, A.: Accurate and efficient cost aggregation strategy for stereo correspondence based on approximated joint bilateral filtering. In: The 9th Asian conference on Computer Vision – Part II (ACCV 2009), pp. 371—380. Springer–Verlag, Berlin, Heidelberg (2010)
65. Mattoccia, S., Viti, M., Ries, F.: Near real–time fast bilateral stereo on the GPU. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 2011), pp. 136–143 (2011)

66. Mayhew, J.E.W., Frisby, J.P.: The computation of binocular edges. Perception **9**, 69–87 (1980)
67. Mayhew, J.E.W., Frisby, J.P.: Psychophysical and computational studies towards a theory of human stereopsis. Artificial Intelligence **17**, 349–385 (1981)
68. Medioni, G., Nevatia, R.: Segment–based stereo matching. Computer Vision, Graphics, and Image Processing (CVGIP) **31**(1), 2–18 (July 1985)
69. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(10), 1615–1630 (2005)
70. Mu, Y., Zhang, H., Li, J.: A global sparse stereo matching method under structure tensor constraint. In: International Conference on Information Technology and Computer Science (ITCS 2009), vol. 1, pp. 609–612 (2009)
71. Ohta, Y., Kanade, T.: Stereo by intra– and inter–scanline search using dynamic programming. IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI–7**(2), 139–154 (1985)
72. Otto, G.P., Chau, T.K.W.: "region–growing" algorithm for matching of terrain images. Image and Vision Computing **7**(2), 83–94 (May 1989)
73. Paris, S., Kornprobst, P., Tumblin, J., Durand, F.: Bilateral filtering: Theory and applications. Foundations and Trends in Computer Graphics and Vision **4**(1), 1–73 (2009)
74. Perona, P., Malik, J.: Scale–space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence **12**(7), 629–639 (1990)
75. Perreault, S., Hebert, P.: Median filtering in constant time. IEEE Transactions on Image Processing **16**(9), 2389–2394 (2007)
76. Petschnigg, G., Szeliski, R., Agrawala, M., Cohen, M., Hoppe, H., Toyama, K.: Digital photography with flash and no–flash image pairs. ACM Transactions on Graphics **23**(3), 664–672 (August 2004)
77. Pollard, S.B., Mayhew, J.W., Frisby, J.P.: PMF: a stereo correspondence algorithm using a disparity gradient limit. Perception **14**, 449–70 (1985)
78. Ramanath, R., Snyder, W.E.: Adaptive demosaicking. Journal of Electronic Imaging **12**(4), 633–642 (2003)
79. Riemens, A.K., Gangwal, O.P., Barenbrug, B., Berretty, R.P.M.: Multistep joint bilateral depth upsampling. In: Visual Communications and Image Processing. San Jose, CA (2009)
80. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: European Conference on Computer Vision (ECCV 2006), pp. 430–443. Graz, Austria (2006)
81. Roy, S., Cox, I.J.: A maximum–flow formulation of the N–camera stereo correspondence problem. In: Proceedings of Sixth International Conference on Computer Vision, (ICCV 1998), pp. 492–499 (1998)
82. Saint-Marc, P., Chen, J.S., Medioni, G.: Adaptive smoothing: a general tool for early vision. IEEE Transactions on Pattern Analysis and Machine Intelligence **13**(6), 514–529 (1991)
83. Sarkis, M., Diepold, K.: Sparse stereo matching using belief propagation. In: 15th IEEE International Conference on Image Processing (ICIP 2008), pp. 1780–1783 (2008)
84. Sarkis, M., Diepold, K.: Towards real–time stereo using non–uniform image sampling and sparse dynamic programming. In: The Fourth International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT 2008). Atlanta, GA, USA (June 2008)
85. Sawhney, H.S., Guo, Y., Hanna, K., Kumar, R., Adkins, S., Zhou, S.: Hybrid stereo camera: An IBR approach for synthesis of very high resolution stereoscopic image sequences. In: SIGGRAPH 2001, pp. 451–460. Los Angeles, California (2001)
86. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two–frame stereo correspondence algorithms. International Journal of Computer Vision **47**(1–3), 7–42 (April–June 2002)
87. Schmid, C., Zisserman, A.: The geometry and matching of lines and curves over multiple views. International Journal of Computer Vision **40**(3), 199–233 (2000)
88. Smith, B.M., Zhang, L., Jin, H.: Stereo matching with nonparametric smoothness priors in feature space. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009), pp. 485–492 (2009)
89. Strecha, C., Fransens, R., Gool, L.V.: Combined depth and outlier estimation in multi–view stereo. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 2394–2401 (2006)
90. Strecha, C., Tuytelaars, T., Gool, L.V.: Dense matching of multiple wide–baseline views. In: Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2 (ICCV 2003), pp. 1194–1201. Washington, DC, USA (2003)
91. Strecha, C., Tuytelaars, T., Gool, L.V.: Dense matching of multiple wide–baseline views. In: The Ninth IEEE International Conference on Computer Vision (ICCV 2003), vol. 2, pp. 1194–1201. Nice, France (2003)
92. Taylor, C.J.: Surface reconstruction from feature based stereo. In: Ninth IEEE International Conference on Computer Vision (ICCV 2003), vol. 1, pp. 184–190. Nice, France (2003)
93. Terzopoulos, D.: Regularization of inverse visual problems involving discontinuities. IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI–8**(4), 413–424 (1986)
94. Tibshirani, R.J.: Fast computation of the median by successive binning (2008). ArXiv: 0806.3301
95. Tola, E., Lepetit, V., Fua, P.: A fast local descriptor for dense matching. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008), pp. 1–8 (2008)
96. Tomasi, C., Kanade, T.: Detection and tracking of point features. Tech. Rep. CMU–CS–91–132, Carnegie Mellon University (1991)
97. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Sixth International Conference on Computer Vision (ICCV 1998), pp. 839–846 (1998)
98. Tuytelaars, T., , Gool, L.V.: Wide baseline stereo matching based on local, affinely invariant regions. In: roceedings of The Eleventh British Machine Vision Conference (BMVC 2000), pp. 412–422. University of Bristol (2000)
99. Ulusoy, I., Hancock, E.: A statistical approach to sparse multi-scale phase-based stereo. Pattern Recognition **40**(9), 2504–2520 (2007)
100. Varekamp, C., Barenbrug, B.: Improved depth propagation for 2D to 3D video conversion using key–frames. In: 4th European Conference on Visual Media Production (IETCVMP 2007), pp. 1–7 (2007)
101. Veksler, O.: Semi–dense stereo correspondence with dense features. In: The 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), vol. 2, pp. II490–II497 (2001)

102. Venkateswar, V., Chellappa, R.: Hierarchical stereo and motion correspondence using feature groupings. International Journal of Computer Vision **15**(3), 245–269 (1995)
103. Weiss, B.: Fast median and bilateral filtering. ACM Transactions on Graphics **25**(3), 519–526 (2006)
104. Xiao, C., Nie, Y., Hua, W., Zheng, W.: Fast multi–scale joint bilateral texture upsampling. The Visual Computer **26**(4), 263–275 (2010)
105. Yao, J., Cham, W.K.: 3D modeling and rendering from multiple wide–baseline images by match propagation. Image Communication Signal Processing **21**(6), 506–518 (2006)
106. Zhang, Z., Deriche, R., Faugeras, O., Luong, Q.T.: A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. Artificial Intelligence. **78**(1–2), 87–119 (October 1995)
107. Zhang, Z., Shan, Y.: A progressive scheme for stereo matching. In: Second EuropeanWorkshop on 3D Structure from Multiple Images of Large–Scale Environments (SMILE 2000), pp. 68–85. Dublin, Ireland (2000)
108. Zhao, G., Chen, L., Chen, G.: A speeded–up local descriptor for dense stereo matching. In: Proceedings of 16th IEEE International Conference on Image Processing (ICIP 2009), pp. 2101–2104 (2009)