Depth image super-resolution algorithm based on structural features and non-local means*

WANG Jing (王靖), **ZHANG Wei-zhong** (张维忠)**, **HUANG Bao-xiang** (黄宝香), **and YANG Huan** (杨环)
College of Computer Science and Technology, Qingdao University, Qingdao 266071, China

(Received 16 March 2018; Revised 9 April 2018)

The resolution and quality of the depth map captured by depth cameras are limited due to sensor hardware limitations, which becomes a roadblock for further computer vision applications. In order to solve this problem, we propose a new method to enhance low-resolution depth maps using high-resolution color images. The structural-aware term is introduced because of the availability of structural information in color images and the assumption of identical structural features within local neighborhoods of color images and depth images captured from the same scene. We integrate the structural-aware term with color similarity and depth similarity within local neighborhoods to design a local weighting filter based on structural features. To use non-local self-similarity of images, the local weighting filter is combined with the concept of non-local means, and then a non-local weighting filter based on structural features is designed. Some experimental results show that super-resolution depth image can be reconstructed well by the process of the non-local filter and the local filter based on structural features. The proposed method can reconstruct much better high-resolution depth images compared with previously reported methods.

Document code: A **Article ID:** 1673-1905(2018)05-0391-5
DOI https://doi.org/10.1007/s11801-018-8039-4

Nowadays, more and more RGB-D cameras with high performance and low price are available in the market. Devices such as the Microsoft's Kinect or the Asus's Xtion Pro can provide real time depth and color information with high quality. RGB-D technology has been used in many useful applications such as robotics object recognition, pose estimation and hand gesture recognition. But compared with color images of the same scene, the resolution and quality of depth images are limited due to hardware limitations and more accurate depth information cannot be provided for computer vision applications directly.

To enhance the resolution of depth maps, a lot of investigations[1-5] have been made on upsampling depth images. The super-resolution algorithms can be broadly classified as two: one class is based on the input of a depth image sequence, and the other is based on the input of a depth image and a higher-resolution color image. The algorithms in the first class have strict requirements of image translation among the depth image sequence, which can hardly be satisfied in some practical applications. The algorithms in the second class take advantage of the fact that the color image can provide significant information to enhance the raw depth map. So we concentrate on the algorithms based on a depth image and a color image of the same scene in this paper.

Yang et al[6] applied joint bilateral upsampling[7] with color information to enhance the resolution of depth images. This approach can obtain a depth map with sharper boundaries. However, it is sensitive to noise in the color image and a recovered depth map often contains some false edges. He et al[8] presented a guided image filter whose output is locally a linear transform of the guidance image. Tu et al[9] proposed an edge feature-guided super-resolution reconstruction method based on joint bilateral filter. The depth map is divided into different regions according to the edge feature and different color similarity weightings are calculated in different filtering regions. Yang et al[10] solved depth map super-resolution problem by developing an optimization framework, in which local structural features of color images are used to construct the regularization term. Zhang et al[11] modeled the depth image super-resolution as an energy function optimization problem via local and nonlocal prior. In Ref.[12], the depth super-resolution was also formulated as an optimization problem using a redescending m-estimator to measure the neighboring constraints for depth.

The bilateral filter is an edge-preserving filter, originally introduced by Tomasi and Manduchi[13]. The bilateral filter uses both a spatial filter kernel and a range filter kernel evaluated on input images. Let p denote one pixel at the input image I, and the filtered result is:

* This work has been supported by the National Natural Science Foundation of China (No.61602269), Shandong Province Science and Technology Development Project (No.2014GGX101048), China Postdoctoral Science Foundation (No.2015M571993), and Shandong Provincial Natural Science Foundation (No.ZR2017MD004).

** E-mail: cfyljnan@sohu.com

$$J_p=\frac{1}{k_p}\sum_{q\in\Omega}I_q f\left(\|p-q\|\right)g\left(\|I_p-I_q\|\right)\ , \tag{1}$$

where f is the spatial filter kernel, such as a Gaussian type centered over p, and g is the range filter kernel, centered at the image value at p. Ω is the spatial support of the kernel f , and k_p is a normalizing factor, the sum of the $f{\cdot}g$ filter weights. Edges are preserved since the bilateral filter $f{\cdot}g$ takes on smaller values as the range distance or the spatial distance increases. The bilateral filter has been used for various image processing tasks like image denoising[14].

Joint bilateral filter is a variant of bilateral filter in which the range filter is applied to a second guidance image GI. Thus, the filtered result is:

$$J_p=\frac{1}{k_p}\sum_{q\in\Omega}I_q f\left(\|p-q\|\right)g\left(\|GI_p-GI_q\|\right)\ . \tag{2}$$

The only difference to Eq.(1) is that the range filter uses GI instead of I.

Given a high resolution image GI, and a low resolution solution S, the joint bilateral upsampling[7] applies a spatial filter to the low resolution solution S, while a similar range filter is jointly applied on the full resolution image GI. Let p and q denote (integer) coordinates of pixels in GI, and p_{i} and q_{i} denote the corresponding (possibly fractional) coordinates in the low resolution solution S. The upsampled solution HS is then obtained as:

$$HS_p=\frac{1}{k_p}\sum_{q_{\downarrow}\in\Omega}S_{q_{\downarrow}}f\left(\|p_{\downarrow}-q_{\downarrow}\|\right)g\left(\|GI_p-GI_q\|\right). \tag{3}$$

Buades et al[15] proposed the non-local means algorithm which takes advantage of the high degree of redundancy of any natural image. They assumed that every small window in a natural image had many similar windows in the same image. One can define “neighborhood of a pixel i” as any set of pixels j in the image so that a window around j looks like a window around i. All pixels in that neighborhood can be used for predicting the value at i.

Given an image $v=\{v(i)\,|\,i\in I\}$, the value $NL(v)(i)$ estimated by the non-local means algorithm is computed as a weighted average of all the pixels in the image:

$$NL(v)(i)=\sum_{j\in I}\omega(i,j)v(j)\ , \tag{4}$$

where the weights $\{\omega(i,j)\}_j$ depend on the similarity between the pixels i and j and satisfy the usual conditions $0\le\omega(i,j)\le1$ and $\sum_j\omega(i,j)=1$. Let N_i denote the neighborhood of pixel i, and then the similarity between the pixels i and j will depend on the similarity of the intensity grey-level vectors $v(N_i)$ and $v(N_j)$. The pixels with a similar grey-level neighborhood to $v(N_i)$ will have larger weights on the average.

Given a high resolution color image GI, and a low resolution solution S, we first perform bilinear interpolation to S to obtain an initial high resolution depth image. Thus the depth image S has the same resolution with the color image GI.

Inspired by Ref.[16], we propose a local weighting filter which uses a range filter kernel evaluated on the interpolated depth image S together with a color filter kernel and a structure filter kernel calculated on the input color image GI. Let p and q denote coordinates of pixels in GI, and the filtered result HS is:

$$HS_p=\frac{\sum_{q\in N_p}S_q\omega_{\mathrm{d}}(p,q)\omega_{\mathrm{c}}(p,q)\omega_{\mathrm{s}}(p,q)}{\sum_{q\in N_p}\omega_{\mathrm{d}}(p,q)\omega_{\mathrm{c}}(p,q)\omega_{\mathrm{s}}(p,q)}\ , \tag{5}$$

where N_p denotes the neighborhood of pixel p which is a square patch around the pixel. And $\omega_{\mathrm{d}}(p,q)$ is the range similarity term:

$$\omega_{\mathrm{d}}(p,q)=\exp(-\frac{\|S_p-S_q\|^2}{\sigma_{\mathrm{d}}^2})\ . \tag{6}$$

$\omega_{\mathrm{c}}(p,q)$ is the color similarity term:

$$\omega_{\mathrm{c}}(p,q)=\exp(-\frac{\|GI_p-GI_q\|^2}{\sigma_{\mathrm{c}}^2}). \tag{7}$$

$\omega_{\mathrm{s}}(p,q)$ is the structure similarity term:

$$\omega_{\mathrm{s}}(p,q)=\frac{1}{2}(\exp(-(p-q)^{\mathrm{T}}T_p(p-q)/\sigma_{\mathrm{s}}^2)+$$
$$\exp(-(p-q)^{\mathrm{T}}T_q(p-q)/\sigma_{\mathrm{s}}^2)),$$
$$T_p=\frac{1}{|N_p|}\sum_{p'\in N_p}\nabla GI(p')^{\mathrm{T}}\nabla GI(p')\,. \tag{8}$$

If p and q are on the same structure, $\omega_{\mathrm{s}}(p,q)$ will be large. Here, $\nabla GI(p')=\{\nabla_x GI(p'),\ \nabla_y GI(p')\}$ is the x- and y- image gradient vector at p. $|N_p|$ is the number of pixels in neighborhood N_p. σ_{c}, σ_{d} and σ_{s} are the decay factors of the three similarity functions, which control the decay rate.

According to the non-local means method, the objective pixel is not only related to adjacent pixels, but also related to other pixels in the image. Therefore we extend the local weighting filter Eq.(5), combining it with the non-local means method. That is, a pixel value is computed as a weighted average of all the pixels in the image. And the weights depend on the similarity between the patch centered by the objective pixel and the patches centered by the other pixels. More formally, using the non-local weighting filter based on structural features, the filtered result is:

$$HS_p=\frac{\sum_{q\in I}S_q\omega_{\mathrm{d}}^{\mathrm{N}}(p,q)\omega_{\mathrm{c}}^{\mathrm{N}}(p,q)\omega_{\mathrm{s}}^{\mathrm{N}}(p,q)}{\sum_{q\in I}\omega_{\mathrm{d}}^{\mathrm{N}}(p,q)\omega_{\mathrm{c}}^{\mathrm{N}}(p,q)\omega_{\mathrm{s}}^{\mathrm{N}}(p,q)}\ , \tag{9}$$

where $\omega_{\mathrm{d}}^{\mathrm{N}}(p,q)$ is the non-local range similarity term:

$$\omega_{\mathrm{d}}^{\mathrm{N}}(p,q)=\exp(-\frac{\|SN_p-SN_q\|^2}{\sigma_{\mathrm{d}}^2}). \tag{10}$$

$\omega_{\mathrm{c}}^{\mathrm{N}}(p,q)$ is the non-local color similarity term:

$$\omega_{\mathrm{c}}^{\mathrm{N}}(p,q)=\exp(-\frac{\|GIN_p-GIN_q\|^2}{\sigma_{\mathrm{c}}^2})\ , \tag{11}$$

where SN_p and GIN_p represent the patches centered by the pixel p in depth image S and color image GI respectively. The non-local structure similarity term $\omega_s^N(p,q)$ is the same with the local structure similarity term $\omega_s(p,q)$, which is determined by its mathematical formula.

To reduce computational complexity, a pixel value is commonly computed as a weighted average of pixels in a square search window instead of all the pixels in the image. Thus Eq.(9) can be rewritten into:

$$HS_p=\frac{\sum_{q\in SW_p} S_q\omega_d^N(p,q)\omega_c^N(p,q)\omega_s^N(p,q)}{\sum_{q\in SW_p}\omega_d^N(p,q)\omega_c^N(p,q)\omega_s^N(p,q)}, \tag{12}$$

where SW_p is the search window centered by pixel p.

The pseudo code of the depth image super-resolution algorithm using the above-mentioned filters is shown in Tab.1.

Tab.1 The pseudo code of our algorithm

Input: a low resolution depth image S and a high resolution color image GI of the same scene

Preprocessing: Perform bilinear interpolation to S to obtain an initial high resolution depth image.

Algorithm:

```
for each $p \in S$
    weight=0, sum=0;
    for each $q \in SW_p$
        sum+= $S_q\omega_d^N(p,q)\omega_c^N(p,q)\omega_s^N(p,q)$;
        weight=+ $\omega_d^N(p,q)\omega_c^N(p,q)\omega_s^N(p,q)$;
    end for
    $HS1_p$ = sum/weight;
end for

while $|PSNR^{k+1} - PSNR^k| \le \varepsilon$
    for each $p \in HS1$
        weight=0, sum=0;
        for each $q \in N_p$
            sum+= $HS1_q\omega_d(p,q)\omega_c(p,q)\omega_s(p,q)$;
            weight=+ $\omega_d(p,q)\omega_c(p,q)\omega_s(p,q)$;
        end for
        $HS1_p$ = sum/weight;
    end for
    $k$=$k$+1;
end while
$HS = HS1$;
```

Output: a high resolution depth image HS

Experiments are performed to investigate the performance of the proposed algorithm on depth image upsampling. Our results are compared with those of other state-of-the-art methods. The images for test are taken from the Middlebury stereo datasets[17] and Ref.[6].

There are several parameters that need to be tuned. We fix a search window of 15×15 and a square neighborhood of 5×5 in the calculation. In the non-local weighting filter, σ_c and σ_d are set to 2, while in the local weighting filter, they are set to 0.05. Setting σ_s to 0.3 has always given good results.

Both the spatial filter kernel f used in joint bilateral filter and the structure similarity term ω_s used in the proposed filter are the geometric closeness functions which control the effect of spatial domain on the weight. Fig.1 shows the difference between f and ω_s. Fig.1(a) is a part of image Art taken from Ref.[17]. The square area is the neighborhood of the point near the nose of the statue, which is enlarged in Fig.1(b). Fig.1(c) and Fig.1(d) show the values of f and ω_s in the neighborhood. It can be seen in Fig.1(c) that the less the distance between the pixel and the square center, the greater the value of f. In Fig.1(d), the more similar the structure between the pixel and the square center, the greater the value of ω_s.

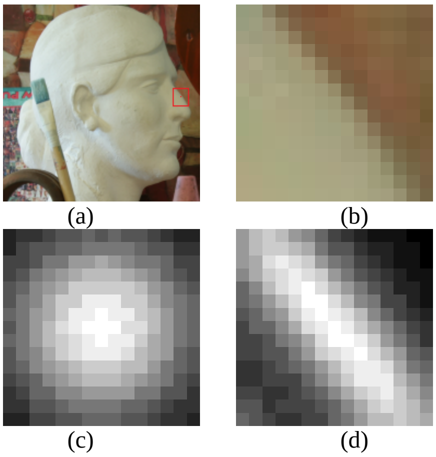

Fig.1 (a) Part of image Art; (b) Enlarged display of the square in (a); (c) Values of the spatial filter kernel f; (d) Values of the structure similarity term ω_s

In Fig.2, three depth and color image pairs taken from real-world scenes in Ref.[6] are tested to compare the upsampling performance of our algorithm with joint bilateral filter[7] and guided image filter[8]. The resolutions of the input RGB images and depth images are 640×640 and 64×64, respectively. RGB images and low-resolution depth images are shown in Fig.2(a) and Fig.2(b). The results of three methods are shown in Fig.2(c)—(e).

We also make comparison between our method and other methods on the Middlebury stereo datasets[17]. The resolutions of the RGB images and ground truth depth images are both 1 376×1 088. For each depth image, we downsample it using the scaling factors of 4 and 8 as the input low-resolution depth image. In order to better observe the detail of super-resolution construction, we zoom in the local parts of the original images and the results of these methods are in Fig.3.

It can be seen that the bilinear interpolation method produces the most blurred result. This is due to its simple use of low-resolution depth image, and thus it can be called a "blind" upsampling method. Joint bilateral filter[7], guided image filter[8] and our algorithm perform better than bilinear interpolation for the benefit of a high resolution prior to guiding the upsampling. But the depth image upsampled by joint bilateral filtering has obvious texture transfer effect because besides spatial distance, this method only considers the image intensity difference as the neighborhood similarity for depth propagation. The guided image filtering can do it better, but the effect is still noticeable in its results and the object edges become fuzzy. The method we proposed can further reduce texture transfer effect and make it imperceptible. And our method can also smooth small fluctuations and preserve sharp depth discontinuities. Obviously, our results are more visually similar to ground truth depth images.

We evaluate the performance of the algorithms not only through the subjective visual effect, but also by peak signal to noise ratio (*PSNR*). The *PSNR* metric is adopted as the quantitative similarity measurement between upsampled depth image and the ground truth depth image. The *PSNR* values of implementations of different methods on the Middlebury stereo datasets are presented in Tab.2. Compared with other methods, the proposed algorithm has a considerable improvement for *PSNR*.

We present an algorithm to upsample a low resolution depth map using an auxiliary high-resolution RGB image. A structure-aware weight is integrated with a range weight and a color weight to form a local weighting filter. Considering self-similarity in images, we propose a non-local weighting filter by combining non-local means and the local weighting filter. A low resolution depth map is first processed by the non-local weighting filter to inhibit jagged appearance of edges. Then the output result will be filtered several times by the local weighting filter based on structural features to obtain sharp depth boundaries. Both quantitative and qualitative experiments on the benchmark dataset demonstrate the effectiveness of our algorithm. Experimental results show that our method outperforms previous work in terms of both *PSNR* and visual quality.

Tab.2 Quantitative evaluation under *PSNR* metric on the middle bury dataset

Image		Bilinear interpolation	Joint bilateral filter	Guided image filter	Ours
art	4×	37.531 7	38.045 2	37.622 6	38.348 8
	8×	35.454 2	35.700 5	35.535 4	36.604 8
books	4×	39.430 2	39.960 7	39.739 2	40.132 3
	8×	36.731 5	36.747 5	36.786 3	36.895 7
Moebius	4×	42.324 9	42.545 6	42.485 0	43.514 1
	8×	40.156 7	41.096 9	40.595 6	42.400 2
Reindeer	4×	36.968 6	37.263 6	36.976 4	37.633 7
	8×	34.883 8	35.225 3	35.023 3	36.421 1

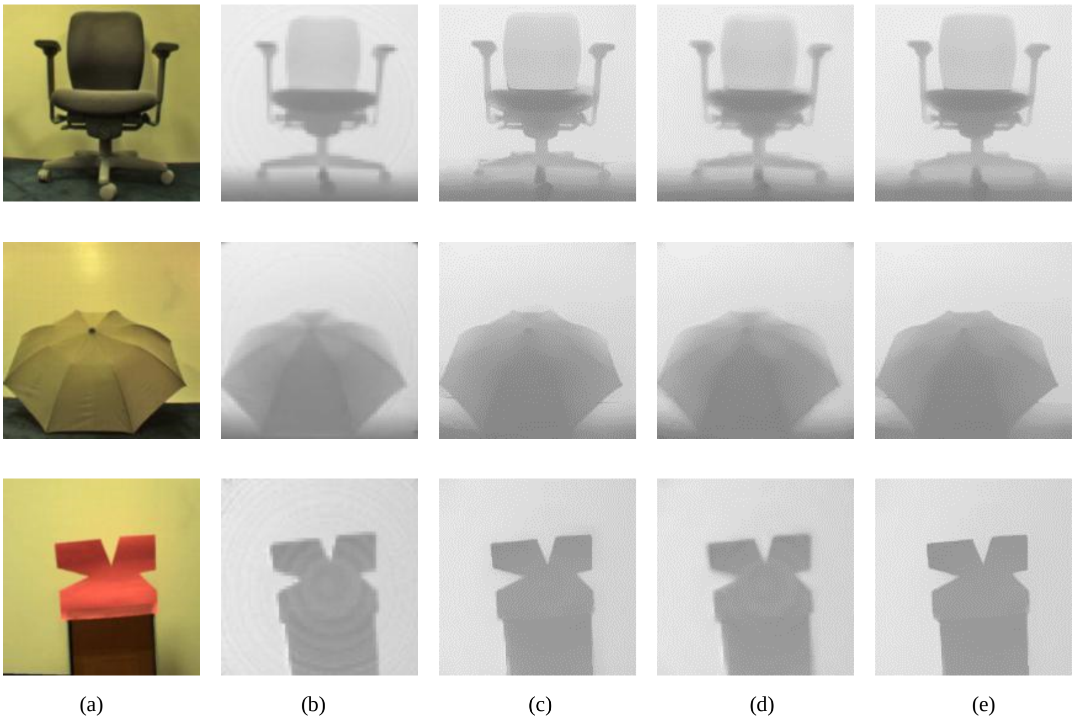

Fig.2 (a) RGB images; (b) Low-resolution depth images; (c) Results of joint bilateral filter; (d) Results of guided image filter; (e) Results of our algorithm

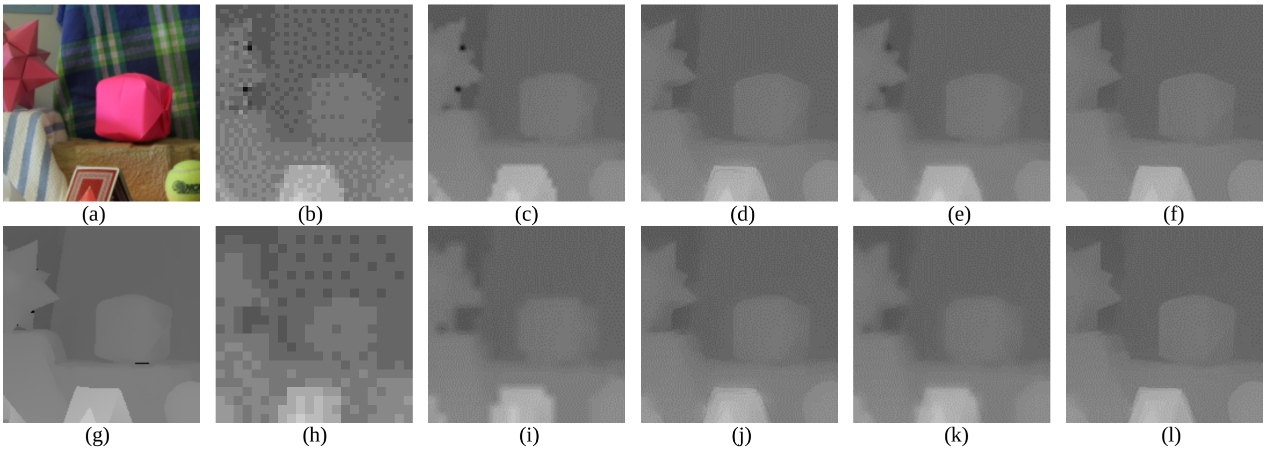

Fig.3 (a) RGB image; (b) Downsampled depth image with the scaling factor of 4; (c)—(f) Results of bilinear interpolation, joint bilateral filter, guided image filter and our algorithm using (b) as the input depth image; (g) Ground truth depth image; (h) Downsampled depth image with the scaling factor of 8; (i)—(l) Results of bilinear interpolation, joint bilateral filter, guided image filter and our algorithm using (h) as the input depth image

References

[1] Yanjie Li, Tianfan Xue, Lifeng Sun and Jianzhuang Liu, IEEE International Conference on Multimedia and Expo, IEEE Computer Society, 152 (2012).

[2] K. R. Lee, R. Khoshabeh and T. Nguyen, Signal Processing Conference IEEE, 1124 (2012).

[3] Weiwei Di, Xudong Zhang, Liangmei Hu and Linlin Duan, Journal of Image and Graphics **19**, 1162 (2014). (in Chinese)

[4] J. Xie, R. S. Feris and M. T. Sun, IEEE International Conference on Image Processing **25**, 3773 (2015).

[5] Jaesik Park, Hyeongwoo Kim, Yu-Wing Tai, Michael S. Brown and Inso Kweon, IEEE International Conference on Computer Vision **24**, 1623 (2011).

[6] Qingxiong Yang, Ruigang Yang, James Davis and David Nister, IEEE Conference on Computer Vision and Pattern Recognition, 1 (2007).

[7] J. Kopf, M. F. Cohen, D. Lischinski and M. Uyttendaele, ACM Transactions on Graphics (TOG) **26**, 96 (2007).

[8] K. He, J. Sun and X. Tang, IEEE Transactions on Pattern Analysis and Machine Intelligence **35**, 1397 (2013).

[9] Y. F. Tu, X. D. Zhang, J. Zhang and L. M. Hu, Computer Applications and Software **34**, 220 (2017). (in Chinese)

[10] Y. X. Yang and Z. F. Wang, Pattern Recognition and Artificial Intelligence **26**, 454 (2013). (in Chinese)

[11] J. Zhang, Dissertation for Doctor’s Degree, University of Science and Technology of China, 2015. (in Chinese)

[12] H. X. Yuan, P. An, S. Q. Wu, Y. Zheng and C. Y. Tong, Journal of Computer-Aided Design and Computer Graphics **28**, 2195 (2016). (in Chinese)

[13] C. Tomasi and R. Manduchi, International Conference on Computer Vision, 839 (1998).

[14] F. L. Liu, M. R. Sun and W. N. Cai, Optoelectronics Letters **13**, 237 (2017).

[15] A. Buades, B. Coll and J. M. Morel, Siam Journal on Multiscale Modeling and Simulation **4**, 490 (2005).

[16] J. Chen, C. K. Tang and J. Wang, ACM Transaction on Graphics **28**, 1 (2009).

[17] D. Scharstein and C. Pal, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1 (2007).