Assembling Convolution Neural Networks for Automatic Viewing Transformation

Haibin Cai, Lei Jiang, Bangli Liu, Yiqi Deng, and Qinggang Meng, Senior Member, IEEE

Abstract—Images taken under different camera poses are rotated or distorted, which leads to poor perception experiences. This article proposes a new framework to automatically transform the images to the conformable view setting by assembling different convolution neural networks. Specifically, a referential three-dimensional ground plane is first derived from the color image and a novel projection mapping algorithm is developed to achieve automatic viewing transformation. Extensive experimental results demonstrate that the proposed method outperforms the state-of-the-art vanishing points based methods by a large margin in terms of accuracy and robustness.

Index Terms—Automatic viewing transform, convolution neural networks, deep learning.

I. INTRODUCTION

HUMANS have the capability of automatically transforming scenes observed with rotated viewing angles into a particular comfortable viewing setting, for example, the orientation of a door will always look like vertical under different head poses. However, most of the existing image sensors are based on the pinhole model [1] and lack the built-in viewing transformation ability. As a result, images taken under different viewing angles might have an uncomfortable viewing experience for human beings. Fig. 1 shows an example of two images taken under different viewing angles. To this end, automatic viewing transformation aims to transform an image taken under different viewing angle to a common horizontal viewing setting.

Automatic viewing transformation is useful in applications such as photography, mixture reality, online shopping, and human–computer interaction. For example, with this technology, image sensors will be able to improve the imaging quality by automatic compensating the impact caused by shaking or rotating. In mixture reality scenarios, an intelligent system is expected to automatically adjust the viewing angle of the scene according to the current pose of a subject. Via automatically transforming the living scenes taken inside a shopping market to a human-centered viewing setting, customers will have a better online shopping experience. Moreover, the transforming procedure can also serve as a preprocessing step to overcome the viewing angle challenges for research such as human action recognition [2], image understanding [3], and multisensory system integration [4].

Existing viewing transformation methods are largely based on the calculation of vanishing points or line segments [5]–[10]. The cause of vanishing points is due to the perceptive projection, where parallel lines in three-dimensional (3-D) space will interact with each other at a point in the image plane. Using the geometric theory of the vanishing points, the corresponding transformation matrix can be modeled for viewing transformation. For example, Carroll et al. [10] proposed a nonlinear least-squares optimization-based warping model, which takes several user annotations including planar regions, straight lines, and associated vanishing points as constraints for viewing transformation. However, as mentioned by the authors, this method suffers from a complex user interface where a user has to understand the basic principles of perspective construction to be able to use it. Lee et al. [9] proposed an energy minimization framework to automatically correct the images via jointly modeling the camera parameters, vanishing points, and segments. The disadvantage of this type of methods is that they require high localization precision of the vanishing points, which is hard to achieve due to the uncertainty of the

Fig. 1. Images taken under different viewing angles. (a) Rotated viewing angle. (b) Horizontal viewing angle. It can be observed that the left image is much harder for human to interpret than the right image due to the rotated viewing angle.
scene. In fact, the edges of some rotated objects might even lead to false detection of the vanishing points or lines.

Taking advantage of the outstanding interpretation ability of deep learning models, this article proposes a new framework for automatic viewing transformation via assembling different convolution neural network (CNN). More specifically, the deep ordinal regression network (DORN) [11] is employed to recover the depth information of the color image and the PSPNet [12] trained on the ADE20K [13] dataset is utilized to semantically segment different parts of the input image. Inspired by the feeling that human beings are good at using the ground plane in judging the direction of objects, this article explores the possibility of using the ground plane as a reference to achieve automatic viewing transformation. We use the Ransac [14] Algorithm to estimate the 3-D ground plane and further propose a novel projection mapping algorithm to automatically transform the images to a conformable viewing setting. Thanks to the involvement of the 3-D structures, the proposed method also has the capability to recover the scene at any viewing angles. Compared to the existing methods, the proposed method does not require the detection of vanishing points, and thus makes it applicable in the scenarios where the detected vanishing points are insufficient or inaccurate. The contributions of this article are listed as follows.

1) A new deep learning based framework is proposed for automatic viewing transformation. The framework seeks the possibility of using a referential plane for the warping of the image, which is fundamentally different from the existing vanishing point based methods.

2) A novel projection mapping algorithm is proposed to enable the system to be transformed to a conformable viewing setting.

3) A dataset is collected to evaluate the performance of existing methods. Experimental results show that the proposed method achieves great performance improvements over the state-of-the-art methods.

The rest of this article is organized as follows. Section II reviews related work on image transformation. Section III presents the classic vanishing points based mapping method. The proposed method is described in detail in Section IV. Section V shows experimental evaluations with a variety of practical images. Finally, Section VI concludes this article.

II. RELATED WORK

The cause of the uncomfortable viewing images is mainly due to the rotation of the image sensors. This phenomenon is well identified and many methods have been proposed to address this issue in the literature.

The geometric property of the vanishing points makes it an ideal solution for the viewing transformation. Gallagher [15] used vanishing points to calculate the rotation toward the yaw angle of the camera and corrected the image via a simple back rotation of the image. Later, Chaudhury et al. [16] proposed a Ransac-based approach to estimate two vanishing points and aligned the closer vanishing point with the Y-axis of the image via a postmultiplication operation. Santana-Cedrés et al. [17] utilized several long lines in the image to locate the vanishing points and performed image rectification based on a camera motion simulation. Lee et al. [6], [9] proposed an optimization framework, which can simultaneously estimate the vanishing lines, vanishing points, and the camera parameters. Considering the difficulties and uncertainties in the accurate localization of the vanishing points, Carroll et al. [10] proposed a semiautomatic two-dimensional (2-D) image warping methods by asking the users to manually annotate several constraints such as planar regions of the scene, straight lines, and associated vanishing points. These constraints then serve as prior knowledge for the optimization of the image warping procedure. The main drawback of these methods is that they either require an accurate localization of vanishing points or an extra manual annotation operation.

There are some methods correcting the image by using the line segments [7], [18]–[20]. For example, He et al. [18] constructed an energy function to preserve the rotation of horizontal/vertical lines. By assuming that the users can give a rough rotation angle to create a perception rotation and the vertices should be on the upright rectangular boundary of the output, they managed to construct the energy function and further solved it by using a two-stage optimization procedure. Observing that the straight line segments are not sufficient for panoramic images, Li et al. [19] improved the above method by further modeling the geodesic appearance of line segments into the energy function. Similarly, An et al. [7] parameterized the homography with camera parameters and designed a cost function to encode the measure of line segment alignment for image warping. Although an accurate result can be achieved by perfectly aligning the lines to a horizontal or vertical boundary, this type of methods suffer from the misdetection of lines. This article differs greatly from these methods in that a 3-D wrapping solution is developed for the image transformation problem by using the estimated depth information.

III. VANISHING POINT BASED MAPPING (VPM) METHOD

The usage of vanishing points in correcting the image is a well-studied research topic [9], [15]–[17], [21], [22]. This section presents a VPM method for the calculation of the rotation matrix from three vanishing points. To ensure an accurate localization of the vanishing points, a manual procedure rather than automatical vanishing points detection algorithms [8], [17] is adopted.

Let \( \alpha \), \( \beta \), and \( \gamma \) be the rotation angles toward the X-axis, Y-axis, and Z-axis, respectively. Then, the rotation matrix can be:

\[
R = \begin{bmatrix}
R_{00}, & R_{01}, & R_{02} \\
R_{10}, & R_{11}, & R_{12} \\
R_{20}, & R_{21}, & R_{22}
\end{bmatrix}
= \begin{bmatrix}
\cos \beta \cos \gamma - \sin \alpha \sin \beta \sin \gamma & - \sin \gamma \cos \alpha & \sin \beta \cos \gamma + \cos \beta \sin \alpha \sin \gamma \\
\sin \beta \sin \alpha \cos \gamma + \cos \beta \sin \gamma & \cos \gamma \cos \alpha & \sin \beta \sin \gamma - \cos \beta \sin \alpha \cos \gamma \\
- \cos \alpha \sin \beta & \sin \alpha & \cos \alpha \cos \beta
\end{bmatrix}
\] (1)
formulated using (1) shown at the bottom of the previous page. Assuming that the positions of the three localized vanishing points in the image coordinate system are \((x_i, y_i), i \in [0, 2]\), they have the following relationship with their corresponding 3-D locations \((X_i, Y_i, Z_i)\) in the world coordinate system according to the 3-D projection rule [23]

\[
\begin{align*}
    x_i &= f \frac{R_{0i} X_i + R_{01} Y_i + R_{02} Z_i + t_0}{R_{30} X_i + R_{31} Y_i + R_{32} Z_i + t_2} + u_0 \\
    y_i &= f \frac{R_{1i} X_i + R_{11} Y_i + R_{12} Z_i + t_1}{R_{30} X_i + R_{31} Y_i + R_{32} Z_i + t_2} + v_0 \\
    \end{align*}
\]

\(i \in [0, 2]\)

where \(f\) is the focal length, \((u_0, v_0)\) is the coordinate of the principle point. The translation matrix is denoted as \(\begin{bmatrix} t_0 & t_1 & t_2 \end{bmatrix}^T\).

Considering that the vanishing point of axis \(X\) has an infinite coordinate value of \(X\), the value of \((x_0, y_0)\) can then be approximating to \((f \frac{X_0}{Z_0} + u_0, f \frac{Y_0}{Z_0} + v_0)\). The same rule applies to the other two vanishing points. Thus, (2) can be simplified into the following equation:

\[
\begin{align*}
    x_i &\approx f \frac{R_{0i}}{R_{30}} + u_0 \\
    y_i &\approx f \frac{R_{1i}}{R_{30}} + v_0 \\
    \end{align*}
\]

\(i \in [0, 2]\)

Combining (1) and (3), the following equation can be formed to solve the rotation matrix:

\[
\begin{align*}
    x_0 - x_1 &= \cos \gamma \sin \beta + \sin \gamma \cos \beta \sin \alpha \\
    x_1 - x_0 &= \cos \gamma \cos \beta \sin \alpha - \sin \gamma \sin \beta \\
    y_0 - y_1 &= \sin \gamma \sin \beta \sin \alpha - \cos \gamma \cos \beta \\
    x_2 - x_1 &= \cos \gamma \sin \beta \sin \alpha + \sin \gamma \cos \beta \\
    y_0 - y_2 &= -\sin \gamma \\
    x_2 - x_2 &= \cos \gamma \\
    \end{align*}
\]

Once the rotation matrix is determined, the image transformation can be performed using the following equation [5]:

\[
P' = K(KR)^{-1}P
\]

where \(P\) and \(P'\) represent the position of a pixel in the original image and transformed image, respectively, and \(K\) is the intrinsic parameter of the image sensor.

IV. PROPOSED METHOD

The proposed method mainly consists of the DORN for depth estimation, the PSPNet for image segmentation, a Ransac-based method for ground plane estimation, and a novel projection mapping method for the viewing transformation. The following sections describe the detail of these methods.

A. Framework Description

Fig. 2 shows the framework of the proposed method. The input of the framework is an RGB image, which can be captured via a variety of sensors such as a webcam, a Kinect sensor, or a GoPro camera. The aim is to achieve automatic coordinate transformation via the frequently occurred ground plane information. To this end, the DORN and the PSPNet are used for depth estimation and semantic segmentation, respectively. The 3-D point clouds of the ground plane can be obtained by fusing the outputs from these two CNNs. Although it is also possible to extract the 3-D ground plane directly from the estimated depth map, the combination of two methods can produce a more accurate 3-D ground plane-fitting result. Then, a Ransac-based algorithm is used to estimate the accurate plane surface whose normal vector is selected for the calculation of the rotation matrix. Finally, a novel 3-D projection mapping procedure is designed to achieve an automatic viewing transformation.

B. Depth Estimation

Estimating depth information from a single RGB image is an ill-posed problem. Recently, significant improvements have been achieved with the help of deep convolutional neural networks, indicating that it is possible to apply the estimated depth information to tasks such as 3-D reconstruction and scene understanding. In this article, DORN [11] is employed for depth estimation due to its advantages in both high accuracy and fast processing speed. Inside the DORN, a full-image encoder is designed and a spacing-increasing discretization strategy is developed to recast the depth network learning as an ordinary regression problem. The network is implemented on the Caffe [24] platform and trained on the NYU Depth V2 dataset [25].

C. Semantic Segmentation

Semantic segmentation aims to recognize object categories in an image in the pixel level. By replacing the fully connected layer with a convolution layer, the fully convolutional network [26] has demonstrated the effectiveness of deep neural networks in semantic segmentation, inspiring many novel networks been proposed [27]. This article adopts the PSPNet [12] due to its good performance in segmenting the ground plane. The main novelty of this network lies in the design of the pyramid pooling module, which empirically proves to be essential in improving the segmentation accuracy. The model is trained on the ADE20K dataset [13], which contains more than 20 K images with dense annotations and wide distribution of scenes.

D. Ground Plane Fitting

The output of the depth estimation and semantic segmentation images is combined to reconstruct the 3-D point clouds of the ground plane. It should be noted that there is abnormal data in both the depth image and the segmented map, which leads to noises in the recovered 3-D points clouds. To reduce the effect of these noises, the Ransac algorithm is used to efficiently and robustly fit the ground plane. More specifically, it randomly selects three points to compute the hypothesis plane since three points determine a plane. A point is judged as an inlier if its distance to the hypothesis is smaller than a predefined threshold.

In practical, the threshold value of 600 mm is found to be robust for the plane detection task. After repeating this process several times, many hypotheses can be generated. Then, the
hypothesis with the most inner points is selected. The final plane is determined via a least square plane fitting over these inner points.

E. Projection Mapping

Once the ground plane is fitted, the viewing transformation can be accomplished via the proposed projection mapping method. First, the 3-D point clouds of the scene can be recovered using the following equation:

\[
\begin{align*}
X &= \frac{u - u_0}{f_x} \times Z \\
Y &= \frac{v - v_0}{f_y} \times Z
\end{align*}
\]  

(6)

where \((u, v)\) is the location of a pixel in the RGB image. \((u_0, v_0, f_x, f_y)\) stands for the intrinsic parameters of the camera and \((X, Y, Z)\) is the position of the reconstructed 3-D point.

Given the normal vector of the ground plane \(N = (n_x, n_y, n_z)\) and the target transformation vector \(V = (v_x, v_y, v_z)\), the rotation matrix can be determined using Rodrigues’ rotation formula, which is defined as follows:

\[
\begin{align*}
\theta &= \arccos \left( \frac{N \cdot V}{|N| \times |V|} \right) \\
\begin{bmatrix}
    c_x \\
    c_y \\
    c_z \\
    K \\
    R
\end{bmatrix} &= \begin{bmatrix}
    n_y \times v_z - n_z \times v_y \\
    n_z \times v_x - n_x \times v_z \\
    n_x \times v_y - n_y \times v_x \\
    0, -c_z, c_y \\
    c_z, 0, -c_x \\
    -c_y, c_x, 0
\end{bmatrix} \\
R &= \cos(\theta)I + \sin(\theta)K + (1 - \cos(\theta))K^2
\end{align*}
\]  

(7)

where \(\theta\) is the angle between the normal vector and transformation vector. Their cross product vector \((c_x, c_y, c_z)\) is used as the rotation axis. \(K\) and \(R\) are the cross product matrix and rotation matrix, respectively. Considering that human tend to use the ground plane as a reference in the perception system and the power of gravity makes objects pointing toward the ground plane, the target transformation vector \(V\) is set to be \((0, 1, 0)\) to create the comfortable viewing setting.

Using the calculated rotation matrix, the reconstructed 3-D point clouds can be rotated and projected to a specific viewing angle and position through the following equation:

\[
\begin{align*}
P' &= R \times P + T \\
u' &= u_0 + f_x \times \frac{X'}{Z'} \\
v' &= v_0 + f_y \times \frac{Y'}{Z'} \\
R &= \begin{bmatrix}
    R_0, R_1, R_2 \\
    R_3, R_4, R_5 \\
    R_6, R_7, R_8
\end{bmatrix}
\end{align*}
\]  

(8)

where \(R\) is the rotation matrix to transfer the points from the camera coordinate system to the world coordinate system. \(T\) is a user-specified transition matrix to set the location of the camera. \(P'\) is the position of a transferred 3-D point \((X', Y', Z')\) in the world coordinate system. \(P\) stands for the original 3-D point \((X, Y, Z)\) in the camera’s coordinate system. \((u', v')\) stands for the projection position of point \(P'\) in the transformed image.

By letting the transition matrix \(T\) be \((0, 0, 0)\) and combining (6) and (8), the projection mapping function can be simplified as (9) shown at the bottom of this page.

After feeding the rotation matrix \(R\) calculated in (7)–(9), the transformed image can be obtained. It should be noted that
although the system is designed to transform the scene into the ground plane guided viewing setting, it can also perform a transformation to any viewing angle settings via giving a specific rotation and transition matrix.

The main novelty of the projection mapping lies in the derivation of the transformation matrix using the ground plane information and the applying of the derived matrix for the automatic viewing transformation. The scene observed under different viewing angles often has different contents, which will probably result in some black holes or pixels in the transformed image. To deal with these holes and create a smooth image, the bilinear interpolation filter is used in the final 2-D image.

V. EXPERIMENTS

The algorithms are implemented on a computer with an Intel Core i7-7700K CPU and a GTX 1080Ti GPU. Each algorithm has been run for more than 1000 times and their average processing time is given in Table I. The resolution of the estimated depth map from DORN is fixed at 353 × 257 and further resized to 640 × 480, which is the resolution of the RGB image. During the testing, we use the same set of model and camera parameters for all the images. It can be seen from the table that the large computation cost mainly lies in the depth estimation and semantic segmentation. Although the 3-D plane-fitting algorithm and the projection mapping algorithm are conducted only on the CPU, they are able to achieve real-time performance.

A. Perspective Transformation Dataset (PTD)

To evaluate the performance of the proposed method, the PTD that contains 280 images captured under ten different indoor scenes is collected. It employs three different webcams, a Kinect sensor, and a GoPro sensor to ensure the diversity in the hardware sensors. Among them, the GoPro sensor has a wider viewing angle than the other sensors, which results in some distortions in the captured images. During the capturing process, the sensors are randomly rotated to a certain angle (ranges from 0° to 30°) to make the images various in rotations.

B. Accuracy Measurement

The transformation results are usually measured by conducting a user study to judge the effectiveness of the methods or simply looking at the transformed images [6], [9], [16], [17]. Apart from directly comparing the transformed results, this article further performs a quantitative comparison by measuring the angle difference of a specific line that should be horizontal after the transformation. The specific line can be determined by manually selecting two points inside the image. Then, the angle error can be determined using the following equation:

\begin{equation}
\theta = \frac{y_2 - y_1}{x_2 - x_1}
\end{equation}

where \(\theta\) stands for the angle difference, \((x_1, y_1)\) and \((x_2, y_2)\) are the coordinate of two manually selected points on the line that should be horizontal in the transformed image.

C. Performance of 3-D Map

The proposed method has an advantage over the 2-D image wrapping algorithms in which it can perform a transformation to any viewing settings given a target rotation matrix and transition matrix. Fig. 3 presents some snapshots of the 3-D map under specific viewing condition. Looking at the images in the same row, we can easily find out the influence of the \(\alpha\) angle. The effect of changing in the \(\beta\) angle can be found by comparing images in each column.

D. Performance of the VPM Method

This section presents the performance of the VPM method in transforming images. Due to the projection, the parallel lines in the 3-D world will intersect at a point in the image. Thus, three different axis directions will result in three corresponding vanishing points. Fig. 4(a) shows the manual procedure to localize the three vanishing points whose position can be used to calculate the rotation matrix according to (4). The transformed result is shown in Fig. 4(b). As pointed out by many researchers, some photographs do not have enough content for the localization of three vanishing points [6], [17], which limits the application of the VPM method. Fig. 4(c) shows one of these images where the third vanishing point cannot be localized.
Fig. 4. Illustrations for the VPM method. (a) Way to retrieve the vanishing points via manually clicking points and calculating the intersection point of the selected lines. (b) Transformed results using the accurately calculated vanishing points and the theory presented in Section III. (c) Example of the images where the third vanishing point cannot be determined.

Fig. 5. Comparison of viewing transformation results with images captured under different angles and illuminations. The TV object in the second and third columns is enlarged to facilitate reading. The enlarged parts are marked with red rectangles.

E. Comparison to the State of the Art

This section compares the transformation results of the proposed method with the state-of-the-art methods. Although some related works [7], [17], [20], [22] have been proposed recently, none of these papers have open sourced their implementation, which causes the difficulty in further comparisons. To boost further research, the binary version of this software will be made publicly available.

Fig. 5 shows the performance of the methods with images taken under different angles and illuminations. The original images are shown in the first column. The results of the proposed method and Google’s method [16] are positioned in the second and third columns, respectively. Although both methods achieve satisfactory results in these circumstances, the details from the enlarged TV object show that the proposed method outperforms Google’s method in most of the cases. For example, in the first row of Fig. 5, the boundary of the TV in the middle column looks like a rhombus, whereas it is a rectangle in the third column, showing that the transformation result of the proposed method is closer to the realistic situation.

Fig. 6 compares the viewing transformation results of the two methods with images captured by the GoPro camera. Due to the wide angle lens, images taken by this sensor have some distortions that might lead to unstable performance for Google’s method. The first two rows demonstrate that both methods can accurately transform the images in some circumstances, whereas images in the last five rows show that the transformation result of the proposed method is more stable and accurate.

Fig. 7 presents the transformation results of the two methods under some other scenes with different angles. These images
Fig. 7. Comparison of viewing transformation results with images captured under some other scenes with different angles.

are captured by normal webcams and the Kinect RGB sensor. The images are not cropped to give an overall view of the transformation results. It can be seen from the second row of the figure that the method in [16] sometimes fails to transform the images even when they are not distorted. The cause of this might due to the wrongly detected vanishing points. Though the proposed method achieves accurate and robust performance over [16] in most of the cases, it relies on the reconstruction of the 3-D ground plane, which might hinder its broad application for the images where the ground information is not presented.

Fig. 8 shows the accuracy of the three methods using the measurement presented in Section V-B. The red line and blue line indicate the accuracy of the proposed method and [16], respectively. As can be seen from the figure, the performance of the VPM method (green line) is much inferior to the other two methods. The cause of this phenomenon is either due to the missing of the third vanishing point or the inaccurately localized vanishing points. Note that the precise position of the vanishing points is extremely difficult to achieve when the lines are nearly parallel in the image.

Table II lists a quantitative comparison of the accuracy of the three methods in the PTD database. The proposed method and the method in [16] both achieve around 57.1% under the condition that the tolerance is within 2°. When the tolerance is bigger than 2°, the proposed method outperforms the method in [16] by a large extent. For instance, over 30% improvements have been achieved by the proposed method when the tolerance is within 6°.

VI. CONCLUSION

This article achieved a significant improvement in automatic viewing transformation, not only does it get rid of the general requirement of detecting vanishing points or lines but it also outperformed the state-of-the-art methods in terms of accuracy and robustness. Based on the assumption that the ground plane can be observed in the captured images, a deep learning based viewing transformation framework and a novel projection mapping algorithm were designed to adjust the perspective. Experimental evaluations demonstrated that the proposed algorithm can perform automatic viewing transformation for images taken under different positions and rotations, paving the way for video analysis applications such as mixture reality, human behavior recognition, and calibration-free multicamera system integration.

Future works have been targeted as follows.
1) Collecting and annotating a large PTD via synthesizing images using 3-D models of the scene.
2) Exploring the possibility of using an end-to-end network for the viewing transformation task.
3) Designing intelligent multicamera human–machine interaction applications based on the proposed techniques.
References

[1] J. Heikkilä, “Geometric camera calibration using circular control points,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 10, pp. 1066–1077, Oct. 2000.

[2] Y. Guo, Y.-F. Li, and Z. Shao, “On multiscale self-similarities description for effective three-dimensional/six-dimensional motion trajectory recognition,” IEEE Trans. Ind. Inform., vol. 13, no. 6, pp. 3017–3026, Dec. 2017.

[3] H. Liu, S. Chen, and N. Kubota, “Intelligent video systems and analytics: A survey,” IEEE Trans. Ind. Inform., vol. 9, no. 3, pp. 1222–1233, Aug. 2013.

[4] J. Chen, K. Li, Q. Deng, K. Li, and P. S. Yu, “Distributed deep learning model for intelligent video surveillance systems with edge computing,” IEEE Trans. Ind. Inform., to be published, doi: 10.1109/TII.2019.2909473.

[5] R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[6] H. Lee, E. Shechtman, J. Wang, and S. Lee, “Automatic upright adjustment of photographs,” in Proc. IEEE Conf. Comput. Vision Pattern Recognit., Jun. 2012, pp. 877–884.

[7] J. An, H. I. Koo, and N. I. Cho, “Rectification of planar targets using line segments,” Mach. Vision Appl., vol. 28, no. 1/2, pp. 91–100, 2017.

[8] F. M. Mirzaei and S. I. Roumeliotis, “Optimal estimation of vanishing points in a Manhattan world,” in Proc. IEEE Int. Conf. Comput. Vision, Oct. 2011, pp. 2454–2461.

[9] H. Lee, E. Shechtman, J. Wang, and S. Lee, “Automatic upright adjustment of photographs with robust camera calibration,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 5, pp. 833–844, May 2014.

[10] R. Carroll, A. Agarwala, and M. Agrawala, “Image warps for artistic perspective manipulation,” ACM Trans. Graph., vol. 29, no. 4, 2010, Art. no. 127.

[11] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, “Deep ordinal regression network for monocular depth estimation,” in Proc. IEEE Conf. Comput. Vision Pattern Recognit., Jun. 2018, pp. 2002–2011.

[12] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in Proc. IEEE Conf. Comput. Vision Pattern Recognit., Jul. 2017, pp. 6230–6239.

[13] B. Zhou et al., “Semantic understanding of scenes through the ADE20K dataset,” Int. J. Comput. Vision, vol. 127, no. 3, pp. 302–321, 2019.

[14] M. A. Fischler and R. C. Bolles, “Random sample consensus. A paradigm for model fitting with applications to image analysis and automated cartography,” Graph. Image Process., vol. 24, no. 6, pp. 381–395, 1981.

[15] A. C. Gallagher, “Using vanishing points to correct camera rotation in images,” in Proc. Conf. Comput. Robot Vision, 2005, pp. 460–467.

[16] K. Chaudhury, S. DiVerdi, and S. Ioffe, “Auto-rectification of user photos,” in Proc. Int. Conf. Image Process., 2014, pp. 3479–3483.

[17] D. Santana-Cedrés et al., “Automatic correction of perspective and optical distortions,” Comput. Vision Image Understanding, vol. 161, pp. 1–10, 2017.

[18] K. He, H. Chang, and J. Sun, “Content-aware rotation,” in Proc. IEEE Int. Conf. Comput. Vision, Dec. 2013, pp. 553–560.

[19] D. Li, K. He, J. Sun, and K. Zhou, “A geodesic-preserving method for image warping,” in Proc. IEEE Conf. Comput. Vision Pattern Recognit., 2015, pp. 213–221.

[20] J. Jung, B. Kim, J. Y. Lee, B. Kim, and S. Lee, “Robust upright adjustment of 360 spherical panoramas,” Vis. Comput., vol. 33, no. 6–8, pp. 737–747, 2017.

[21] C. Hughes, P. Denny, M. Glavin, and E. Jones, “Equidistant fish-eye calibration and rectification by vanishing point extraction,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 12, pp. 2289–2296, Dec. 2010.

[22] B. Kim, J.-Y. Lee, J. Jung, and G. Miller, “Automatic orientation adjustment of spherical panorama digital images,” U.S. Patent Appl. 16/150614, Oct. 3, 2018.

[23] R. M. Haralick, “Using perspective transformations in scene analysis,” Comput. Graph. Image Process., vol. 13, no. 3, pp. 191–221, 1980.

[24] Y. Jia et al., “Caffe: Convolutional architecture for fast feature embedding,” in Proc. ACM Int. Conf. Multimedia, 2014, pp. 675–678.

[25] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor segmentation and support inference from RGBD images,” in Proc. Eur. Conf. Comput. Vision, 2012, pp. 746–760.

[26] E. Shehhamer, J. Long, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proc. IEEE Conf. Comput. Vision Pattern Recognit., 2014, pp. 3431–3440.

[27] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 4, pp. 834–848, Apr. 2018.