Using CNN for Ellipse Estimation in An Infant Length Measurement Application

Maolong Tang* and Ming-Ting Sun*
*University of Washington, Seattle, WA, USA
E-mail: {mltang, mts}@uw.edu

Abstract— It is desirable to be able to measure an infant's length from a photo. This would make the measurement of the infant's length easy which is important for infant growth velocity monitoring. To make this possible, we developed a technique in which round stickers are put on the body joints of the infant before taking photos. The round stickers will be projected onto the image plane into ellipses with different parameters depending on the direction and distance from the camera. By estimating the parameters of the ellipses in the picture, we can calculate the 3D positions of the sticker centers and the length of the infant. A major difficulty with this technique is that the infant moves during the picture taking, which could cause severe motion blur. In this paper, we use a CNN (Convolutional Neural Network) to restore the ellipses for ellipse parameter estimation. To generate realistic training data which is needed for the CNN to accurately restore the ellipses for length measurements, we propose to simulate the whole ellipse formation pipeline. The whole training process includes generating ellipses of random sizes, motion blurs, and illumination changes, and adding noise, demosaicing, and gamma and inverse gamma corrections. Simulation results show that better accuracy of measurements can be achieved with the proposed training methods.

I. INTRODUCTION

It is important to monitor an infant's length to make sure that the infant is growing normally. Unlike adults who could listen to guidance and stand near reference object (for example height ruler), it is hard to ask infants to do the same. Traditionally, measuring an infant's length requires performing a manual measurement using an infantometer. However, the infant struggles in the process, and it often needs three trained assistants to hold the infant's head and feet and to adjust the infantometer, and so, constant measuring at home is not possible. It is very desirable to be able to measure an infant's length from a photo. This would make the measurement of the infant's length easy. To make this possible, we developed a technique in which round stickers are put on body joints of an infant before the pictures are taken, as shown in Fig. 1. Putting the stickers on the infant would only take a few seconds and is very easy to perform. After that, the whole process is fully automatic. The round shape stickers will become ellipses in the picture with different parameters depending on the pose of the camera. By estimating the parameters of the ellipses in the picture, our algorithm can calculate the 3D positions of the sticker centers and the length of the infant automatically [15].

We conducted field trials to test the method. We found it can provide accurate measurements, but several factors can affect the accuracy of the proposed technique. A major problem is that the infant moves during the picture taking, which could cause severe motion blur. The severe motion blur could cause the ellipse-parameter estimation inaccurate. One way to solve this problem is to deblur the image before the ellipse parameter estimation. However, traditional deblurring methods may introduce ringing effects [1][2], which makes accurate ellipse edge estimation difficult. Besides, most of the state of the art deblurring algorithms fail if the image contains strong noise. Applying a smoothing algorithm before deblurring usually affects the deblurring accuracy [3]. For accurate recovery of the ellipse parameters, the deblurring algorithm itself should be noise robust. Only a few publications explicitly considered noisy blurred images [3][4][5]. However, [3] relies on inverse Radon transform which requires straight edges in the latent image. In [4], it requires more than one image, and [5] assumes that the blur kernel is known.

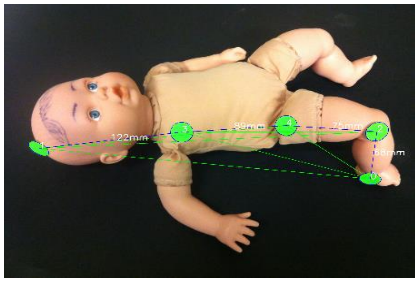

Fig. 1. An example of using rounds stickers to measure the infant length. The 3D distance of each blue dashed line is estimated. We could obtain the infant's length by summing up the lengths of all blue dashed lines.

Another problem is that sometimes because of poor lighting, the detected stickers' boundaries may not be continuous due to camera's sensor noise. Mild Gaussian smoothing could help to remove noise without causing serious edge shift. However, for motion-blurred and poor-lighting images, edge locations are not easy to recover. Almost all ellipse detection and fitting methods rely on accurate edges as input. Directly applying existing ellipse estimation methods will not result in good ellipse parameter estimations. Both edge-grouping and Hough-transform based methods failed. Errors caused by the motion blur under poor lighting could not be removed by outlier handling methods such as [6], since all edges' locations are affected.

Recently, people have applied neural networks to deblur images. A complete review is out of the scope of this paper. We briefly summarize three representative works [7][8][9]. In [7], it uses blurred patches cropped from actual images to train a CNN that classifies blur kernel into limited number of types, and Markov Random Field (MRF) model is used to calculate a dense and smooth blur-kernel distribution. [8] deblurs in the frequency space domain, tt achieves the state of the art results with much higher processing speed. In [9], it uses a high-speed camera to take sharp video frames, then synthesizes motion blurred images by averaging consecutive clear frames. It adopts simplified residual network blocks [10] and a deep multiscale CNN structure. However, our goal is to deblur the ellipses for accurate parameter estimation which is different from debluring the image to make it sharp and appealing.

Through simulations, we found that for our infant-length measurement application, the data used in training the CNN affect the accuracy of the measurements significantly. In this paper, we propose to use a CNN structure, with a synthetic data generation pipeline that takes into account possible effects for generating realistic training data, to restore the ellipses for automatic infant length measurements. Since our task is to accurately recover the ellipse edge locations, not the whole latent image, our CNN structure is light-weighted. The results confirm the effectiveness of our proposed approach.

The organization of the rest of this paper is as follows. In Section II, we describe our CNN structure for restoring the ellipses for accurate parameter estimation in the infant length measurement application. In Section III, we discuss the synthetic data generation for training the CNN. In Section IV, we present our simulation results. In Section V, we conclude the paper.

II. PROPOSED CNN STRUCTURE FOR ELLIPSE RESTORATION

The proposed CNN structure for restoring the degraded ellipses is shown in Fig. 2. Our CNN structure is inspired by [11]. The differences are that we added sigmoid functions after most of the convolutions layers to increase the nonlinearity, and we do not use a very deep structure. Our use of the sigmoid function is inspired by [12]. It resembles the purpose of *tanh* function in the paper but gives us better results.

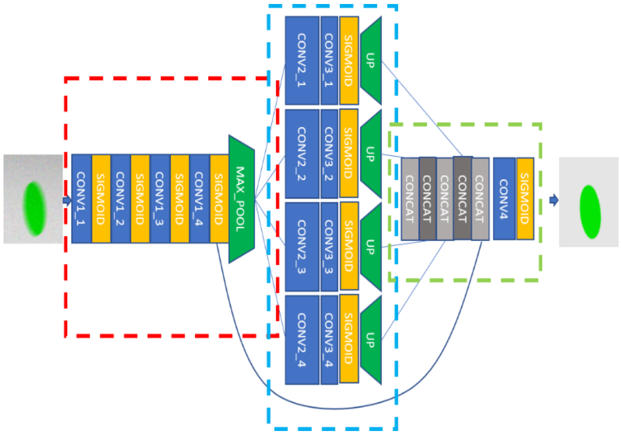

Fig. 2. Proposed CNN structure.

The structure could roughly be considered by the three parts enclosed by the red, blue, and green dashed rectangles as shown in Fig.2. The first part of the structure consists of four convolutional layers. After max pooling, the image's resolution is reduced to half in both horizontal and vertical dimensions of its original. The second part contains four parallel convolution layers to extract information from different scales. Since the resolution has been halved in both dimensions, four up-sampling layers are used at the end of the second part. The third part concatenates the results of part one and part two, then followed by a three-channel convolution layer and a sigmoid function. Parameters of each convolutional layers are listed in Table 1.

The motivation of using this structure is that an ellipse is much simpler than a general image. Thus, the network does not need to be too deep or too complicated as in [11], [9], or [12]. An ellipse has smoothly curved boundaries. To recover the edge location, there is no need to use the global information of the image. However, the receptive field still needs to be large enough to cover the blur-kernel size and to minimize the effect of noise. For our case, a 32 × 32 receptive field is large enough to handle the blurred ellipses. We could remove the max pooling and up-sampling layers. However, this will increase the computation time, and the results do not get better.

III. GENERATING TRAINING DATA

For deep learning, one of the most important parts is collecting and labeling sufficient amount of data for training.

TABLE 1. CONVOLUTION LAYER PARAMETERS

Layer Name	CONV1_1	CONV1_2	CONV1_3	CONV1_4
Filter size	7 × 7	5 × 5	5 × 5	3 × 3
Filter numbers	16	32	32	64
Stride	(1,1,1,1)	(1,1,1,1)	(1,1,1,1)	(1,1,1,1)
Receptive field	7 × 7	11 × 11	15 × 15	17 × 17

Layer Name	CONV2_1	CONV2_2	CONV2_3	CONV2_4
Filter size	1 × 1	3 × 3	3 × 3	3 × 3
Filter numbers	16	16	16	16
Stride	(1,1,1,1)	(1,1,1,1)	(1,1,1,1)	(1,1,1,1)
Dilation	1	2	4	8
Receptive field	18 × 18	20 × 20	24 × 24	32 × 32

Layer Name	CONV3_1	CONV3_2	CONV3_3	CONV3_4	CONV4
Filter size	1 × 1	1 × 1	1 × 1	1 × 1	5 × 5
Filter numbers	1	1	1	1	3
Stride	(1,1,1,1)	(1,1,1,1)	(1,1,1,1)	(1,1,1,1)	(1,1,1,1)
Receptive field	18 × 18	20 × 20	24 × 24	32 × 32	NA

Collecting and labeling data could be very time-consuming. For us, pixel and even sub-pixel level edge location is desirable. We need the blurred ellipse and the corresponding clear ellipse pairs for training. The strategy of [9] to take several pictures to synthesize the blurred ones does not apply to our case.

To synthesize an ellipse and its degraded version close to that observed in the practical situations, we simulate the process of the ellipse formation and degradation. We limit our image size to 256 × 256, the stickers' color to green, and the background color to white. We synthesize the training data with the following four steps considering the actual ellipse generation and degradation process:

Step 1. Generate the motion-blurred ellipse mask. We first generate random ellipse parameters, motion directions, and motion magnitudes. We evenly divided the motion range to get 101 locations, then generate a clear 0/1 ellipse mask for each location. The motion-blurred ellipse mask is obtained by averaging all those 101 clear masks. The final ellipse mask has a value between 0 and 1. To account for the partial area effect [13], the mask is first generated in a high resolution (we use 2560×2560). It is then blurred by a Gaussian kernel of size 7x7 with standard deviation of 2, then downsampled to 256×256. An ellipse could be described as:

$$\boldsymbol{h(x,y,\theta,x_0,y_0,a,b)} = \boldsymbol{\frac{(cos\theta(x-x_0)+sin\theta(y-y_0))^2}{a^2}} + \boldsymbol{\frac{(-sin\theta(x-x_0)+cos\theta(y-y_0))^2}{b^2}} = \boldsymbol{1} \qquad \boldsymbol{(1)}$$

where θ is the ellipse's orientation, (x_0, y_0) is the ellipse's center, and a and b are parameters for major and minor axis. So, the clear ellipse mask can be written as:

$$f_{mask}(i,j,\theta,x_0,y_0,a,b) = \begin{cases} 1, h(i,j,\theta,x_0,y_0,a,b) \le 1 \\ 0, h(i,j,\theta,x_0,y_0,a,b) > 1 \end{cases} \qquad (2)$$

where (i,j) is the pixel's location. The motion blurred mask is:

$$f_{blurred}(i,j) = \frac{1}{2N+1} \sum_{n=-N}^{N} f_{mask}(i,j,\theta,x_0{}',y_0{}',a,b) \qquad (3)$$

$x_0^{'} = x_0 + n\delta cos\phi$ and $y_0^{'} = y_0 + n\delta sin\phi$,$(cos\phi, sin\phi)$ is the motion direction, and the total moved length is $(2N+1)\delta$ (in our case, *N=50*).

Step 2. Add noise and light intensity variation. We add zero-mean Gaussian noise with a magnitude of 0.2 and standard deviation of 1, and use Error function (integral of a Gaussian function) as the shape of light intensity variation (caused by shadow). Image pixel values first are multiplied by the light intensity profile. Then, each R, G, B channel is applied with independent Gaussian noise. We also randomly control the percentage of pixels that are affected by noise.

Step 3. Image demosaicing. For a regular camera, at each pixel location only one of R, G, B is directly read from the sensor. Other values are estimated from its neighbors' values by interpolation. The reason to simulate demosaicing is because demosaicing will add artifact to otherwise smooth edges, and the noises of the neighbor pixels are no longer independent. Pixel or subpixel accuracy of the estimated ellipse is desirable. It is better to simulate those artifacts so that CNN can learn how to correct it.

Step 4. Gamma correction and pixel value quantization. Pixel values in all previous operations were represented using float numbers within the range [0,1]. For a JPEG image, modern digital cameras use the sRGB space where the RGB values went through a Gamma correction process with $\gamma = 2.2$ before the quantization and compression process. To make the synthesized data as close to those in the actual images as possible, we also perform the same Gamma correction process (i.e., the RGB colors were first gamma corrected in each channel, $R_\gamma = R^{1/\gamma}$, $G_\gamma = G^{1/\gamma}$, $B_\gamma = B^{1/\gamma}$). Then, all values are quantized to integers between 0 to 255, represented by eight-bit unsigned integers. When feeding the training data into our model, we first change the data type to float and rescale the pixel values to the range $[0, 1]$, then apply inverse gamma correction to make the pixel values proportional to the luminance. We show from simulations, the results are better with this Gamma correction.

The whole process of automatically generating the degraded ellipses for training the CNN is shown in Fig. 3. We could see that after applying the demosaicing effect, noise became less (because noise corresponds to the missing (RGB) color is dropped), and noise granularity become larger because of interpolation. Also, the ellipse boundary is less sharp than before. After quantization, the pixel values are not as accurate as before. However, this is not noticeable to human eyes. The resultant degraded ellipses look very similar to those observed in the practical images we encountered.

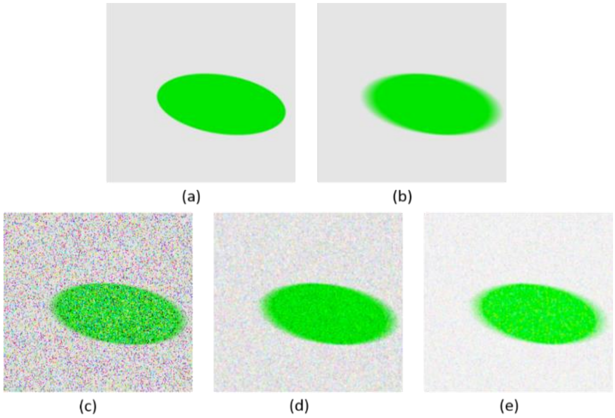

Fig. 3. (a) Clear ellipse, (b) blurred, no noise, (c) blurred image with noise, (d) after considering the demosaicing effect, (e) after gamma correction and pixel value quantization.

IV. Experiments and Results

To train the CNN model, we generated 30,000 blurred ellipses of random sizes, orientations, motion directions, and motion ranges. We choose the batch-size to be 16. Every time 16 images are randomly drawn from images containing the 30000 blurred ellipses, then they are applied with noise, random lighting variation, demosaicing effect, Gamma correction and pixel value quantization. All parameters in the CNN were randomly initialized. The energy function is chosen to be:

$$loss = \frac{\sum_{All\ pixels}\left(x_{recv}(i,j) - x_{gt}(i,j)\right)^2}{L_x \times L_y} \qquad (4)$$

where $x_{recv}(i,j)$ is the pixel value at the (i,j) position in the recovered image, $x_{gt}(i,j)$ is the pixel value at the (i,j) position in the ground truth image, and $L_x \times L_y$ is the image size. Since the shape of the sticker is simple, the training only takes about 4,000 steps with Adam optimizer. The initial learning rate is 0.0001. After 1000 steps, the learning rate was dropped to 0.00001. After another 1000 steps, learning rate is updated to 0.000001, and we will use 0.000001 as the learning rate for the rest of optimization. Training takes about 3 hours with i5-3570 CPU and GTX1070 graphic card. This is very fast due to the light weighted network. Removing max-pooling and up-sampling layers will increase the second part (Fig. 2, blue dashed rectangular) convolution layer's resolution, and training time will be longer. However, the result does not improve.

A. Restoration with synthetic data:

A comparison of noisy blurred input, restored ellipses, and the ground truth ellipses is shown in Fig. 4. Subjectively, there is no obvious difference between the restored ellipses and the ground truths. The only noticeable difference is that the recovered ellipse edge is not as sharp as the ground truth. Fig. 5 (a) and (b) show the difference between blurred ellipses, the restored ellipses, and the ground truths. For the restored ellipses, the errors mainly locate at near the ellipse boundaries.

After obtaining the restored ellipses, we use the Taubin's method [14] with outlier removal [6] to estimate the parameters of all ellipses for comparisons. Fig.5(c) shows ellipse fitting results of the CNN processed image and the motion blurred image. Ellipses fitted from the restored images almost entirely overlap with the ellipse fitted from the ground truth images.

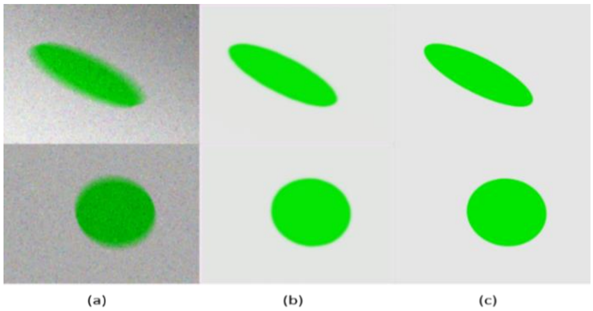

Fig. 4. (a) Noisy blurred input, (b) restored by CNN, (c) Ground Truth.

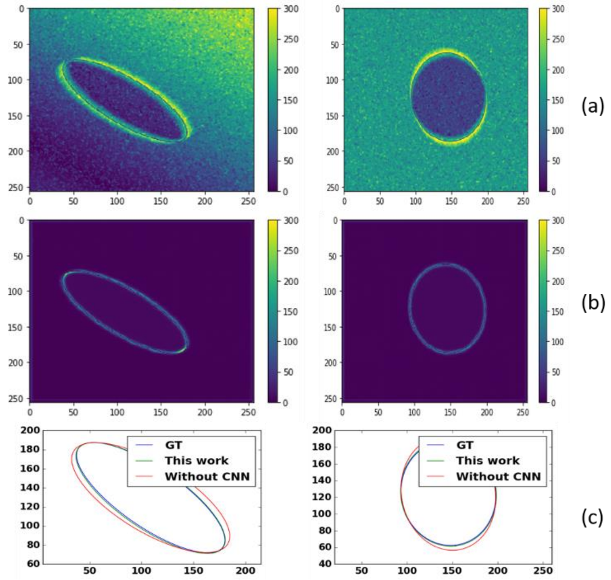

Fig. 5. (a) |blurred-GT|, (b) |recovered-GT|, (c) Fitted ellipses comparison.

B. Results with actual images:

To show the results in the practical situations, we scan the image patches near those green stickers. We used overlapped patches, each patch has a size 256 × 256. After recovery, only the center 200 × 200 pixels were stored and merged for ellipse detection. Circular stickers have a known diameter 19.05 mm.

We put the green circular stickers on the headboard and footboard of an infantometer as shown in Fig. 6, and take pictures of the stickers. The estimated footboard to headboard distance is 45.2 cm, compared to the infantometer reading of 45 cm. The difference is only 0.2 cm. This verifies that our method can accurately recover noisy motion-blurred circular sticker's parameters with the CNN.

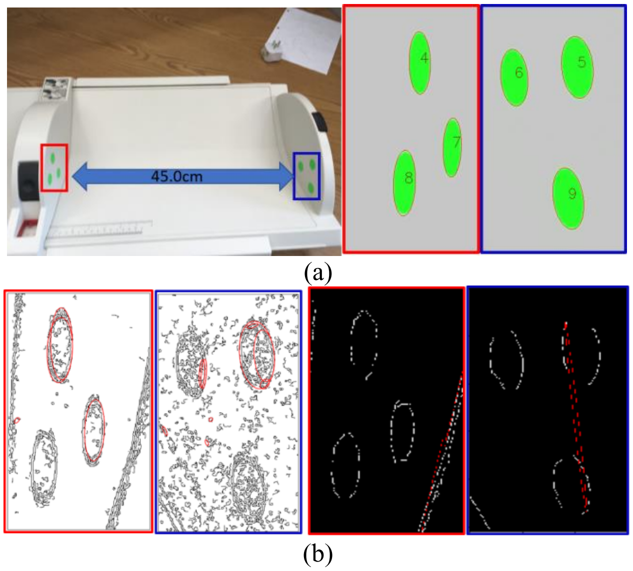

Fig. 6. (a) Infantometer with motion blurred circular stickers on the headboard and footboard, the distance reading from the infantometer is 45.0 cm, Recovered clean ellipses and fitting. The calculated two plane distance is 45.2 cm, compared to 45.0 cm read from the infantometer. (b) Ellipse detection using conventional methods from [15][16].

Unlike previous deblur algorithms which try to recover all textures of an image, our model suppresses textures that do not belong to label and background. The reason is that we only trained the model with green circular stickers and white background, the ground truths are synthesized ideal ellipses. All other unrelated texture will be considered as noise by the CNN model. We could compare the red and blue rectangles in Fig 6(b) with the original image Fig 6(b), dark edges of the background were gone in the recovered images. This does not invalidate our method, because the goal is to recover the edge location, not a good visual result. The model tries to recover ellipses with smooth and clear edge boundaries that can facilitate edge tracing. Such effect could not be achieved if actual images are used as ground truths of training data due to the unavoidable noise and artifacts.

Table 2 shows the comparison of 3D distance recovery using parameters of the detected ellipses. The relation between 3D positions and ellipse parameters could be found in [17]. The 3D distance estimation is very sensitive to ellipse parameters. It is impossible to obtain ground truth parameters from the blurred images, thus, the accuracy of recovered 3D distance could serve as a very good metric to evaluate the algorithm. The ground truth of sticker's 3D distances could be obtained using physical tools. Here image blur was caused by camera motion. The distance estimated without CNN processing is 42.3 cm, 2.7 cm shorter than the infantometer reading. The distance estimation with our proposed CNN is 45.2 cm, only 0.2 cm longer than the infantometer reading. The results also show the necessity of simulating gamma and inverse gamma correction during training: the error decreases from 1 cm to 0.2 cm. Notes about Avg, Med, and STD: for one image we have six length estimations, three headboard stickers' distances to the footboard, and 3 footboard stickers' distances to the headboard. Avg means the average value of those six estimations; Med means the median value; STD means standard deviation of those six values. The result validates our synthetic training data generation method. The good performance on synthetic data is transferable to realistic images.

TABLE 2. RESULTS OF APPLYING DIFFERENT METHODS ON AN IMAGE WHERE THE ELLIPSES ARE BLURRED. "THIS WORK*" MEANS "THIS WORK WITHOUT GAMMA CORRECTION."

Estimation	This work	This work*	Without CNN processing
1	45.1 cm	45.8 cm	42.6 cm
2	45.3 cm	46.3 cm	41.9 cm
3	45.0 cm	45.8 cm	42.2 cm
4	45.0 cm	45.8 cm	41.9 cm
5	45.4 cm	46.2 cm	42.6 cm
6	45.3 cm	46.1 cm	42.7 cm
Avg	**45.2 cm**	46.0 cm	42.3 cm
Med	**45.2 cm**	46.0 cm	42.4 cm
STD	**0.1 cm**	0.2 cm	0.3 cm
Err	**0.2 cm**	1.0 cm	2.6 cm

Fig. 7 shows the recovered clear ellipses from the image that contains motion-blurred circular stickers. Here motion blur was simulated by moving the baby doll's leg while taking the picture. The ground truth distance is 6.70 cm. The estimated distance by the method proposed in this paper is 6.78 cm. The difference is less than 1 mm.

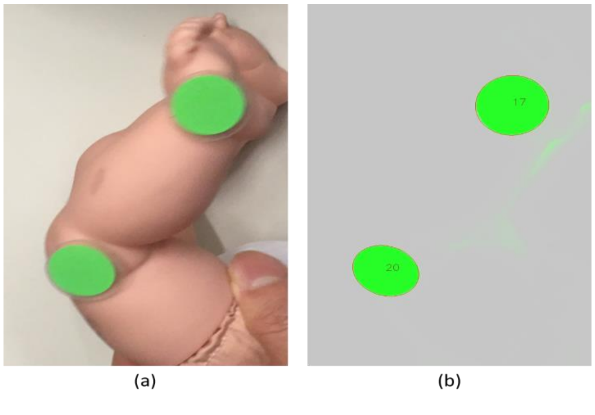

Fig. 7. Blurred stickers. (a) Circular stickers were blurred due to foot motion, (b) recovered clear ellipses. Conventional methods (such as [15][16]) failed when directly applied to blurred images.

Our current trained CNN cannot handle stickers of general shapes. However, our CNN model and the data generation method do not make assumptions about the shape of the stickers. For different shapes of stickers, we just need to synthesize the training data for this specific shape of sticker. For sticker-based pose estimation or many other 3D reconstruction problems, accuracy of the edge location is very important. If we could integrate the smoothed curve deblurring problem into the deep learning structure, this could facilitate 3D reconstruction with low-quality images.

V. CONCLUSION

In this paper, we proposed a CNN-based structure for ellipse restoration in an infant-length measurement application. We also proposed a fully synthetic training data generation method which can generate realistic data for training the CNN. The method considers different artificial effect during the image formation. We demonstrated the effectiveness of our method using actual images taken by a regular cellphone camera. Our CNN model and the training data generation method do not make assumptions about the shape of stickers. Thus, they can be directly trained to recover other types of stickers.

REFERENCES

[1] W.-S. Lai *et al.*, "A Comparative Study for Single Image Blind Deblurring," *2016 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1701–1709, Jun. 2016.

[2] L. Xu, J. S. Ren, C. Liu, and J. Jia, "Deep Convolutional Neural Network for Image Deconvolution," in *Advances in neural information processing systems*, 2014, no. 413113, pp. 1–9.

[3] L. Zhong, S. Cho, D. Metaxas, S. Paris, and J. Wang, "Handling noise in single image deblurring using directional filters," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2013, pp. 612–619.

[4] L. Yuan *et al.*, "Image deblurring with blurred/noisy image pairs," in *ACM Transactions on Graphics*, 2007, vol. 26, no. 3, p. 1.

[5] M. Jin, S. Roth, and P. Favaro, "Noise-Blind Image Deblurring," pp. 3510–3518, 2017.

[6] M. Shao, Y. Ijiri, and K. Hattori, "Grouped outlier removal for robust ellipse fitting," *Proc. 14th IAPR Int. Conf. Mach. Vis. Appl. MVA 2015*, pp. 138–141, May 2015.

[7] J. Sun, W. Cao, Z. Xu, and J. Ponce, "Learning a Convolutional Neural Network for Non-uniform Motion Blur Removal."

[8] A. Chakrabarti, "A neural approach to blind motion deblurring," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9907 LNCS, pp. 221–235, Oct. 2016.

[9] S. Nah, T. H. Kim, and K. M. Lee, "Deep Multi-scale Convolutional Neural Network for Dynamic Scene Deblurring," *Comput. Vis. Pattern Recognit. (CVPR), 2017 IEEE Conf.*, pp. 3883–3891, 2016.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 770–778, 2016.

[11] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing Network," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[12] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Scholkopf, "Learning to Deblur," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1439–1451, 2016.

[13] A. Trujillo-Pino, K. Krissian, M. Alemán-Flores, and D. Santana-Cedrés, "Accurate subpixel edge location based on partial area effect," *Image Vis. Comput.*, vol. 31, no. 1, pp. 72–90, 2013.

[14] G. Taubin, "Estimation of Planar Curves, Surfaces, and Nonplanar Space Curves Defined by Implicit Equations with Applications to Edge and Range Image Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 11, 1991.

[15] Y. Xie and Q. J. Q. Ji, "A new efficient ellipse detection method," *Proceedings. 16th Int. Conf. Pattern Recognition, 2002.*, vol. 2, no. c, pp. 957–960, 2002.

[16] D. K. Prasad, M. K. H. Leung, and S. Y. Cho, "Edge curvature and convexity based ellipse detection method," *Pattern Recognit.*, vol. 45, no. 9, pp. 3204–3221, 2012.

[17] Q. Chen, H. Wu, and T. Wada, "Camera calibration with two arbitrary coplanar circles," in *European Conference on Computer Vision (ECCV)*, 2004, pp. 521–532.