Standard Plane Detection in 3D Fetal Ultrasound Using an Iterative Transformation Network

Yuanwei Li[1], Bishesh Khanal[2], Benjamin Hou[1], Amir Alansary[1],
Juan J. Cerrolaza[1], Matthew Sinclair[1], Jacqueline Matthew[2],
Chandni Gupta[2], Caroline Knight[2], Bernhard Kainz[1], and Daniel Rueckert[1]

[1] Biomedical Image Analysis Group, Imperial College London, UK
[2] School of Biomedical Engineering & Imaging Sciences, King's College London, UK

Abstract. Standard scan plane detection in fetal brain ultrasound (US) forms a crucial step in the assessment of fetal development. In clinical settings, this is done by manually manoeuvring a 2D probe to the desired scan plane. With the advent of 3D US, the entire fetal brain volume containing these standard planes can be easily acquired. However, manual standard plane identification in 3D volume is labour-intensive and requires expert knowledge of fetal anatomy. We propose a new Iterative Transformation Network (ITN) for the automatic detection of standard planes in 3D volumes. ITN uses a convolutional neural network to learn the relationship between a 2D plane image and the transformation parameters required to move that plane towards the location/orientation of the standard plane in the 3D volume. During inference, the current plane image is passed iteratively to the network until it converges to the standard plane location. We explore the effect of using different transformation representations as regression outputs of ITN. Under a multi-task learning framework, we introduce additional classification probability outputs to the network to act as confidence measures for the regressed transformation parameters in order to further improve the localisation accuracy. When evaluated on 72 US volumes of fetal brain, our method achieves an error of 3.83mm/12.7° and 3.80mm/12.6° for the transventricular and transcerebellar planes respectively and takes 0.46s per plane.

1 Introduction

Obstetric ultrasound (US) is conducted as a routine screening examination between 18-24 weeks of gestation. US imaging of the fetal head enables clinicians to assess fetal brain development and detect growth abnormalities. This requires the careful selection of standard scan planes such as the transventricular (TV) and transcerebellar (TC) plane that contain key anatomical structures [6]. However, it is challenging and time-consuming even for experienced sonographers to manually navigate a 2D US probe to find the correct standard plane. The task is highly operator-dependent and requires a great amount of expertise. With the advent of 3D fetal US, a volume of the entire fetal brain can be acquired quickly with little training. But the problem of locating diagnostically required standard planes for biometric measurements remains. There is a strong need to develop automatic methods for 2D standard plane extraction from 3D volumes to improve clinical workflow efficiency.

Related work: Recently, deep learning approaches have shown successes in many medical image analysis applications. Several works have applied deep learning techniques to standard plane detection in fetal US [1, 3, 2, 7]. Baumgartner *et al.* [1] use a convolutional neural network (CNN) for categorisation of 13 fetal standard views. Chen *et al.* [3] adopt a CNN-based image classification approach for detecting fetal abdominal standard planes, which they later combined with a recurrent neural network (RNN) that takes into account temporal information [2]. However, these methods identify standard planes from 2D US videos and not 3D volumes. Ryou *et al.* [7] attempt to detect fetal head and abdominal planes from 3D fetal US by breaking down the 3D volume into a stack of 2D slices which are then classified as head or abdomen using a CNN.

Plane detection is considered an image classification problem in the above works. In contrast, we approach the plane detection problem by regressing rigid transformation parameters that define the plane position and orientation. There are several works on using CNN to predict transformations. Kendall *et al.* [5] introduce PoseNet for regressing 6-DoF camera pose from RGB image with a loss function that uses quaternions to represent rotation. Hou *et al.* [4] propose SVRNet for predicting transformation from 2D image to 3D space and use anchor points as a new representation for rigid transformations. These works predict absolute transformation with respect to a known reference coordinate system with one pass of CNN. Our work is different as we use an iterative approach with multiple passes of CNN to predict relative transformation with respect to current plane coordinates, which change at each iteration. Relative transformation is used as our 3D volumes are not aligned to a reference coordinate system.

Contributions: In this paper, we propose the Iterative Transformation Network (ITN) that uses a CNN to detect standard planes in 3D fetal US. The network learns a mapping between a 2D plane and the transformation required to move that plane towards the standard plane within a 3D volume. Our contributions are threefold: **(1)** ITN is a general deep learning framework built for 2D plane detection in 3D volumes. The iterative approach regresses transformations that bring the plane closer to the standard plane. This reduces computation cost as ITN selectively samples only a few planes in the 3D volume unlike classification-based methods that require dense sampling [1, 3, 2, 7]. **(2)** We study the effect on plane detection accuracy using different transformation representations (quaternions, Euler angles, rotation matrix, anchor points) as CNN regression outputs. **(3)** We improve ITN performance by incorporating additional classification probability outputs as confidence measures of the regressed transformation parameters. At inference, the classification probabilities are used as confidence scores to yield more accurate localisation. During training, regression and classification outputs are learned in a multi-task learning framework, which improves the generalisation ability of the model and prevents overfitting.

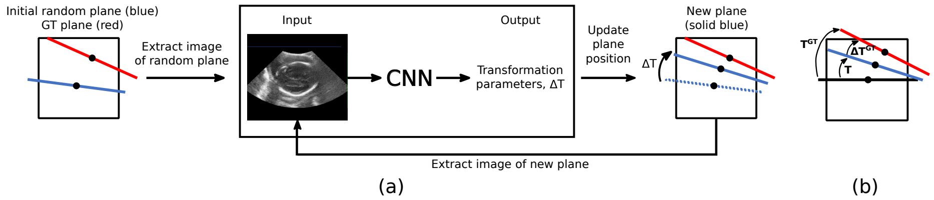
Initial random plane (blue)
GT plane (red)
Extract image of random plane
Input
Output
CNN
Transformation parameters, ΔT
Update plane position
ΔT
New plane (solid blue)
Extract image of new plane
T^{GT}
ΔT^{GT}
T
(a)
(b)

Fig. 1: (a) Overall plane detection framework using ITN. (b) Composition of transformations. Red: GT plane. Blue: Arbitrary plane. Black: Identity plane.

2 Method

Overall Framework: Fig. 1a presents the overall ITN framework for plane detection. Given a 3D volume V, the goal is to find the ground truth (GT) standard plane (red). Starting with a random plane initialisation (blue), the 2D image of the plane is extracted and input to a CNN which then predicts a 3D transformation $\mathit{\Delta T}$ that will move the plane to a new position closer to the GT plane. The image extracted at the new plane location is then passed to the CNN and the process is repeated until the plane reaches the GT plane.

Composition of Transformations: Transformation is defined with respect to a reference coordinate system. In Fig. 1b, we define an identity plane (black) with origin at the volume centre. T and T^{GT} are defined in the coordinate system of the identity plane and they move the identity plane to the arbitrary plane (blue) and GT plane (red) respectively. $\mathit{\Delta T}^{GT}$ is defined in the coordinate system of the arbitrary plane and $\mathit{\Delta T}^{GT}$ moves the arbitrary plane to the GT plane. Note that our ITN predicts $\mathit{\Delta T}^{GT}$ which is a relative transformation from the point of view of the current plane, and *not* from the identity plane. We compute these transformations from each other using $T^{GT} = T \oplus \mathit{\Delta T}^{GT}$ and $\mathit{\Delta T}^{GT} = T^{GT} \ominus T$ where $\oplus$ and $\ominus$ are the composition and inverse composition operators respectively. The computations defined by the operators are dependent on the choice of the transformation representation.

Network Training: During training, an arbitrary plane is randomly sampled from a volume V by applying a random transformation T to the identity plane. The corresponding 2D plane image X is then extracted. We define $X = I(V, T, s)$ where $I(\cdot)$ is the plane extraction function and s is the length of the square plane. We sample T such that the plane centre falls in the middle 60% of V and the rotation of the plane is within an angle of $\pm 45°$ about each coordinate axis. This avoids sampling of planes at the edges of the volume where there is no informative image data due to regions falling outside of the US imaging cone. A training sample is represented by $(X, \mathit{\Delta T}^{GT})$ and the training loss function can be formulated as the $L2$ norm of the error between the GT and predicted transformation parameters: $L = \left\|\mathit{\Delta T}^{GT} - \mathit{\Delta T}\right\|_2^2$

Network Inference: Algorithm 1 summarises the steps during network inference to detect a plane. The iterative approach gives rough estimates of the plane in the first few iterations and subsequently makes smaller and more accurate

Algorithm 1 Iterative Inference of Transformation

```
procedure PLANE TRANSFORMATION($V, s, N$)
    Initialise random plane with $T_1$
    for $i = 1$ to $N$ do
        $X_i \leftarrow I(V, T_i, s)$                    ▷ Sample plane image
        $\Delta T \leftarrow \mathrm{CNN}(X_i)$          ▷ CNN predicts relative transformation
        $T_{i+1} \leftarrow T_i \oplus \Delta T$         ▷ Update plane position
    return $T_N$
```

Table 1: Representations of rigid transformations and their loss functions.

Representation (Parameter count)	Loss function
Translation $\boldsymbol{t}$ (3) + Quaternion $\boldsymbol{q}$ (4)	$L = \alpha \left\| \boldsymbol{t}^{GT} - \boldsymbol{t} \right\|_2^2 + \beta \left\| \boldsymbol{q}^{GT} - \frac{\boldsymbol{q}}{\|\boldsymbol{q}\|} \right\|_2^2$
Translation $\boldsymbol{t}$ (3) + Euler angles $\boldsymbol{\theta}$ (3)	$L = \alpha \left\| \boldsymbol{t}^{GT} - \boldsymbol{t} \right\|_2^2 + \beta \left\| \boldsymbol{\theta}^{GT} - \boldsymbol{\theta} \right\|_2^2$
Translation $\boldsymbol{t}$ (3) + Rotation matrix $\boldsymbol{R}$ (9)	$L = \alpha \left\| \boldsymbol{t}^{GT} - \boldsymbol{t} \right\|_2^2 + \beta \left\| \boldsymbol{R}^{GT} - \boldsymbol{R} \right\|_2^2$
Anchor points $(\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3)$ (9)	$L = \sum_{i=1}^{3} \left\| \boldsymbol{A}_i^{GT} - \boldsymbol{A}_i \right\|_2^2$

refinements. This coarse-to-fine adjustment improves accuracy and is less susceptible to different initialisations. To improve accuracy and convergence, we repeat Algorithm 1 with 5 random plane initialisations per volume and average their final transformations T_N after N iterations.

Transformation Representations: In ITN, plane transformation ΔT is rigid, comprising only translation and rotation. We explore the effect of using different transformation representations as the CNN regression outputs (Table 1) since there are few comparative studies that investigate this on deep networks. The first three representations explicitly separate translation and rotation in which rotation is represented by quaternions, Euler angles and rotation matrix respectively. α and β are weightings given to the translation and rotation losses. Specifically, anchor points [4] are defined as the coordinates of three fixed points on the plane (we use: centre, bottom-left and bottom-right corner). The points uniquely and jointly represent any translation and rotation in 3D space. During inference, the predicted values of certain representations need to be constrained to give valid rotation. For instance, quaternions need to be normalised to unit quaternions and rotation matrices need to be orthogonalised. Anchor points need to be converted to valid rotation matrices as described in [4].

Classification Probability as Confidence Measure: We further extend our ITN by incorporating classification probability as a confidence measure for the regressed values of translation and rotation. The method can be applied to any transformation representation but we use quaternions since it yields the best results. In addition to the regression outputs $\boldsymbol{t}$ and $\boldsymbol{q}$, the CNN also predicts two classification probability outputs $\boldsymbol{P}$ and $\boldsymbol{Q}$ for translation and rotation respectively. We divide translation into 6 discrete classification categories: positive and negative translation along each coordinate axis. Denoting c as the translation

Algorithm 2 Compute relative transformation $\mathit{\Delta} T$

```
procedure COMPUTE TRANSFORM($\boldsymbol{t}, \boldsymbol{q}, \boldsymbol{P}, \boldsymbol{Q}$)
    $\boldsymbol{t}_{new} = \begin{pmatrix} \max(P_{c_1^+}, P_{c_1^-})t_1 \\ \max(P_{c_2^+}, P_{c_2^-})t_2 \\ \max(P_{c_3^+}, P_{c_3^-})t_3 \end{pmatrix}$        ▷ Compute weighted translation
    $Q_{max} = \max(\boldsymbol{Q})$        ▷ Compute weighted rotation
    if $Q_{max} = Q_{k_x^+}$ OR $Q_{k_x^-}$ then
        Convert $\boldsymbol{q}$ to Euler angles $(\theta_x, \theta_y, \theta_z)$ using convention ‘xyz’
        $\boldsymbol{r}_{new}$ ←Rotation about x-axis with magnitude $Q_{max}\theta_x$
    else if $Q_{max} = Q_{k_y^+}$ OR $Q_{k_y^-}$ then
        Convert $\boldsymbol{q}$ to Euler angles $(\theta_x, \theta_y, \theta_z)$ using convention ‘yxz’
        $\boldsymbol{r}_{new}$ ←Rotation about y-axis with magnitude $Q_{max}\theta_y$
    else if $Q_{max} = Q_{k_z^+}$ OR $Q_{k_z^-}$ then
        Convert $\boldsymbol{q}$ to Euler angles $(\theta_x, \theta_y, \theta_z)$ using convention ‘zxy’
        $\boldsymbol{r}_{new}$ ←Rotation about z-axis with magnitude $Q_{max}\theta_z$
    $\mathit{\Delta} T \leftarrow (\boldsymbol{t}_{new}, \boldsymbol{r}_{new})$
    return $\mathit{\Delta} T$
```

classification label, we have $c \in \{c_1^+, c_1^-, c_2^+, c_2^-, c_3^+, c_3^-\}$ where c_1^+ is the category representing translation along the positive x-axis. $\boldsymbol{P}$ is then a 6-element vector giving the probability of translation along each axis direction. Similarly, we divide rotation into 6 categories: clockwise and counter-clockwise rotation about each coordinate axis. Denoting k as the rotation classification label, we have $k \in \{k_1^+, k_1^-, k_2^+, k_2^-, k_3^+, k_3^-\}$ where k_1^+ is the category representing clockwise rotation about the x-axis. $\boldsymbol{Q}$ is then a 6-element vector giving the probability of rotation about each axis.

A training sample is represented by $(X, \boldsymbol{t}^{GT}, \boldsymbol{q}^{GT}, c^{GT}, k^{GT})$. c^{GT} gives the coordinate axis along which the current plane centre has the furthest absolute distance from the GT plane centre. Similarly, k^{GT} gives the coordinate axis about which the current plane will rotate the most to reach the GT plane. Appendix A derives the computations of c^{GT} and k^{GT} during training. The overall training loss function can then be written as:

$$L = \alpha \left\| \boldsymbol{t}^{GT} - \boldsymbol{t} \right\|_2^2 + \beta \left\| \boldsymbol{q}^{GT} - \frac{\boldsymbol{q}}{\|\boldsymbol{q}\|} \right\|_2^2 - \gamma \log P_{c^{GT}} - \delta \log Q_{k^{GT}} \tag{1}$$

The first and second terms are the $L2$ losses for translation and rotation regression while the third and fourth terms are the cross-entropy losses for translation and rotation classification. α, β, γ and δ are weights given to the losses.

During inference, the CNN outputs $\boldsymbol{t}$, $\boldsymbol{q}$, $\boldsymbol{P}$ and $\boldsymbol{Q}$ are combined to compute the relative transformation $\mathit{\Delta} T$ (Algorithm 2). For translation, each component of the regressed translation $\boldsymbol{t}$ is weighted by the corresponding probabilities in the vector $\boldsymbol{P}$. For rotation, we only rotate the plane about the most confident rotation axis as predicted by $\boldsymbol{Q}$. In order to determine the magnitude of that rotation, the regressed quaternion $\boldsymbol{q}$ needs to be broken down into Euler angles using the appropriate convention in order to determine the rotation angle about that most confident rotation axis. An Euler angle representation using convention 'xyz' means a rotation about x-axis first followed by y-axis and finally z-axis. Hence, $\boldsymbol{P}$ and $\boldsymbol{Q}$ are used as confidence weighting for $\boldsymbol{t}$ and $\boldsymbol{q}$, allowing the plane to translate and rotate to a greater extent along the more confident axis.

Network Architecture: ITN utilises a multi-task learning framework for predictions of multiple outputs. The architecture differs according to the number of outputs that the CNN predicts. All our networks comprise 5 convolution layers, each followed by a max-pooling layer. These layers contain shared features for all outputs. After the 5th pooling layer, the network branches into fully-connected layers to learn the specific features for each output. Details of all network architectures are described in Appendix B.

3 Experiments and Results

Data and Experiments: ITN is evaluated on 3D US volumes of fetal brain from 72 subjects. For each volume, TV and TC standard planes are manually selected by a clinical expert. 70% of the dataset is randomly selected for training and the rest 30% used for testing. All volumes are processed to be isotropic with mean dimensions of 324×207×279 voxels. ITN is implemented using Tensorflow running on a machine with Intel Xeon CPU E5-1630 at 3.70 GHz and one NVIDIA Titan Xp 12GB GPU. We set plane size s=225, N=10 and α=β=γ=δ=1. During training, we use a batch size of 64. Weights are initialised randomly from a distribution with zero mean and 0.1 standard deviation. Optimisation is carried out for 100,000 iterations using the Adam algorithm with learning rate=0.001, β_1=0.9 and β_2=0.999. The predicted plane is evaluated against the GT using distance between the plane centres (δx) and rotation angle between the planes ($\delta\theta$). Image similarity of the planes is also measured using peak signal-to-noise ratio (PSNR) and structural similarity (SSIM).

Results: Table 2 compares the plane detection results when different transformation representations are used by ITN. In general, there is little difference in the translation error. This is because all translation representations are the same, which use the three Cartesian axes except for anchor points which have slightly greater translation error. The rotation errors on TC plane suggest that quaternions are a good representation. Rotation matrices and anchor points over-parameterise rotation and can make network learning more difficult with greater degree of freedom. Since these parameters are not constrained, it is also harder to convert them back into valid rotations during inference. Quaternions have fewer parameters and slightly-off quaternion can still be easily normalised to give valid rotation. Compared to Euler angles, quaternions avoid the problem of gimbal lock. For TV plane, there is little difference in rotation error. This is because sonographers use the TV plane as a visual reference when acquiring 3D volumes. This causes the TV plane to lie roughly in the central plane of the volume with lower rotation variances, thus making the choice of rotation representation less important. Table 3 compares the performance of ITN

Table 2: Evaluation of ITN with different transformation representations for standard plane detection. Results presented as (Mean ± Standard Deviation).

CNN	TV plane				TC plane			
outputs	δx (mm)	$\delta\theta$ (°)	PSNR	SSIM	δx (mm)	$\delta\theta$ (°)	PSNR	SSIM
$\boldsymbol{t}, \boldsymbol{q}$	6.29±5.33	17.0±12.0	15.3±2.0	0.375±0.081	6.23±6.99	14.9±7.5	15.5±2.1	0.383±0.100
$\boldsymbol{t}, \boldsymbol{\theta}$	5.69±5.85	17.0±8.5	15.2±1.7	0.372±0.084	7.13±9.00	16.0±5.9	14.6±2.4	0.357±0.119
$\boldsymbol{t}, \boldsymbol{R}$	5.79±6.10	17.7±11.6	15.8±1.9	0.389±0.091	6.39±7.39	17.3±15.4	15.5±2.4	0.385±0.118
$\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3$	6.64±8.66	17.0±10.4	15.9±2.4	0.399±0.099	7.88±10.0	16.3±12.6	15.0±2.7	0.351±0.124

Table 3: Evaluation of ITN with/without confidence probability for standard plane detection. Results presented as (Mean ± Standard Deviation).

CNN	TV plane				TC plane			
outputs	δx (mm)	$\delta\theta$ (°)	PSNR	SSIM	δx (mm)	$\delta\theta$ (°)	PSNR	SSIM
M1: $\boldsymbol{t}, \boldsymbol{q}$	6.29±5.33	17.0±12.0	15.3±2.0	0.375±0.081	6.23±6.99	14.9±7.5	15.5±2.1	0.383±0.100
M2: $\boldsymbol{t}, \boldsymbol{q}, \boldsymbol{P}$	5.14±5.37	16.8±9.9	16.0±2.1	0.408±0.092	5.12±5.50	13.9±7.1	15.8±2.2	0.393±0.115
M3: $\boldsymbol{t}, \boldsymbol{q}, \boldsymbol{Q}$	6.07±6.32	14.0±8.1	15.7±2.5	0.399±0.108	7.66±7.14	12.7±6.0	15.5±3.0	0.386±0.123
M4: $\boldsymbol{t}, \boldsymbol{q}, \boldsymbol{P}, \boldsymbol{Q}$	3.83±2.10	12.7±7.7	16.4±1.9	**0.419±0.092**	3.80±1.85	12.6±6.1	16.5±2.1	0.407±0.110
M4+: $\boldsymbol{t}, \boldsymbol{q}, \boldsymbol{P}, \boldsymbol{Q}$	**3.49±1.81**	**10.7±5.7**	**16.6±1.8**	0.413±0.082	**3.39±2.13**	**11.4±6.3**	**16.8±2.1**	**0.437±0.110**

with/without classification probability outputs. Given a baseline model (M1) that only has regression outputs $\boldsymbol{t}$, $\boldsymbol{q}$, the addition of classification probabilities $\boldsymbol{P}$, $\boldsymbol{Q}$ improves the translation and rotation accuracy respectively (M2-M4). The classification probabilities act as confidence weights for the regression outputs to improve plane detection accuracy. Furthermore, the classification and regression outputs are trained in a multi-task learning fashion, which allows feature sharing and enables more generic features to be learned, thus preventing model overfitting. M1-M4 use one plane image as CNN input. We further improve our results by using three orthogonal plane images instead as this provides more information about the 3D volume (M4+). M4 and M4+ take 0.46s and 1.35s to predict one plane per volume. The supplementary material provides videos showing the update of a randomly initialised plane and its extracted image through 10 inference iterations.

Fig. 2 shows a visual comparison between the GT planes and the planes predicted by M4. To evaluate the clinical relevance of the predicted planes, a clinical expert manually measures the head circumference (HC) on both the predicted and GT planes and computes the standard deviation of the measurement error to be 1.05mm (TV) and 1.25mm (TC). This is similar to the intraobserver variability of 2.65mm reported for HC measurements on TC plane [8]. Thus, accurate biometrics can be extracted from our predicted planes.

4 Conclusion

We presented ITN, a new approach for standard plane detection in 3D fetal US by using a CNN to regress rigid transformation iteratively. We compare the use of different transformation representations and show quaternions to be a good representation for iterative pose estimation. Additional classification probabili-

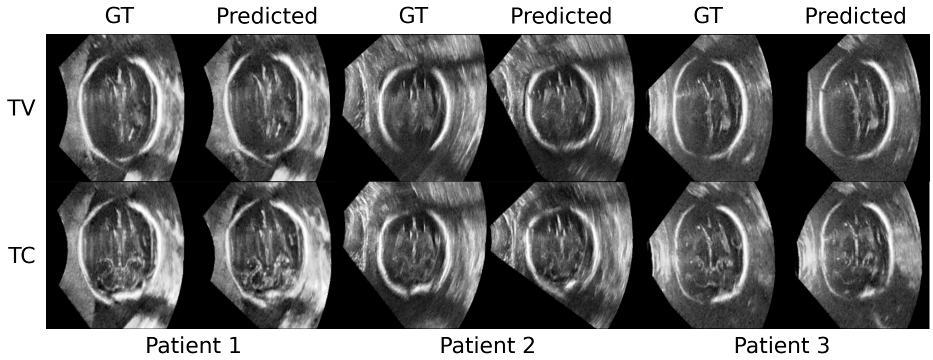

Fig. 2: Visualisation of GT planes and planes predicted by M4.

ties are learned via multi-task learning which act as confidence weights for the regressed transformation parameters to improve plane detection accuracy. As future work, we are evaluating ITN on other plane detection tasks (*eg.* view plane selection in cardiac MRI). It is also worthwhile to explore new transformation representations and extend ITN to simultaneous detection of multiple planes.

Acknowledgments. Supported by the Wellcome Trust IEH Award [102431]. The authors thank Nvidia Corporation for the donation of a Titan Xp GPU.

References

1. Baumgartner, C.F., Kamnitsas, K., Matthew, J., Fletcher, T.P., Smith, S., Koch, L.M., Kainz, B., Rueckert, D.: Sononet: Real-time detection and localisation of fetal standard scan planes in freehand ultrasound. IEEE TMI 36(11), 2204–2215 (2017)
2. Chen, H., Dou, Q., Ni, D., Cheng, J.Z., Qin, J., Li, S., Heng, P.A.: Automatic fetal ultrasound standard plane detection using knowledge transferred recurrent neural networks. In: MICCAI 2015. pp. 507–514. Springer (2015)
3. Chen, H., Ni, D., Qin, J., Li, S., Yang, X., Wang, T., Heng, P.A.: Standard plane localization in fetal ultrasound via domain transferred deep neural networks. IEEE journal of biomedical and health informatics 19(5), 1627–1636 (2015)
4. Hou, B., Alansary, A., McDonagh, S., Davidson, A., Rutherford, M., Hajnal, J.V., Rueckert, D., Glocker, B., Kainz, B.: Predicting slice-to-volume transformation in presence of arbitrary subject motion. In: MICCAI. pp. 296–304. Springer (2017)
5. Kendall, A., Grimes, M., Cipolla, R.: Posenet: A convolutional network for real-time 6-dof camera relocalization. In: ICCV 2015. pp. 2938–2946. IEEE (2015)
6. NHS: Fetal anomaly screening programme: programme handbook June 2015. Public Health England (2015)
7. Ryou, H., Yaqub, M., Cavallaro, A., Roseman, F., Papageorghiou, A., Noble, J.A.: Automated 3d ultrasound biometry planes extraction for first trimester fetal assessment. In: International Workshop on MLMI. pp. 196–204. Springer (2016)
8. Sarris, I., Ioannou, C., Chamberlain, P., Ohuma, E., Roseman, F., Hoch, L., Altman, D., Papageorghiou, A.: Intra-and interobserver variability in fetal ultrasound measurements. Ultrasound in Obstetrics & Gynecology 39(3), 266–273 (2012)