Towards Contactless Patient Positioning

Srikrishna Karanam, *Member, IEEE*, Ren Li, Fan Yang, *Student Member, IEEE*, Wei Hu, Terrence Chen, *Senior Member, IEEE*, and Ziyan Wu, *Member, IEEE*

***Abstract*— The ongoing COVID-19 pandemic, caused by the highly contagious SARS-CoV-2 virus, has overwhelmed healthcare systems worldwide, putting medical professionals at a high risk of getting infected themselves due to a global shortage of personal protective equipment. This has in-turn led to understaffed hospitals unable to handle new patient influx. To help alleviate these problems, we design and develop a contactless patient positioning system that can enable scanning patients in a completely remote and contactless fashion. Our key design objective is to reduce the physical contact time with a patient as much as possible, which we achieve with our contactless workflow. Our system comprises automated calibration, positioning, and multi-view synthesis components that enable patient scan without physical proximity. Our calibration routine ensures system calibration at all times and can be executed without any manual intervention. Our patient positioning routine comprises a novel robust dynamic fusion (RDF) algorithm for accurate 3D patient body modeling. With its multi-modal inference capability, RDF can be trained once and used across different applications (without re-training) having various sensor choices, a key feature to enable system deployment at scale. Our multi-view synthesizer ensures multi-view positioning visualization for the technician to verify positioning accuracy prior to initiating the patient scan. We conduct extensive experiments with publicly available and proprietary datasets to demonstrate efficacy. Our system has already been used, and had a positive impact on, hospitals and technicians on the front lines of the COVID-19 pandemic, and we expect to see its use increase substantially globally.**

***Index Terms*— Covid-19, contactless, patient positioning, 3D pose, shape.**

I. Introduction

THE ongoing coronavirus disease 2019 (COVID-19) pandemic has resulted in over 2.9 million infections and 200,000 deaths as of April 25, 2020 across 210 countries and territories. This disease, caused by the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) virus, is highly contagious in nature, resulting in a dramatic increase, within a very short amount of time, in the number of patients seeking care in hospitals. This surge in patients has pushed our already overburdened hospitals, and the associated healthcare systems, to the brink of breaking down. In particular, we have noticed two issues that should be of particular concern. Due to the close proximity of healthcare providers (technicians, doctors, *etc.*) to the patients, many of these professionals are themselves getting infected, adding more pressure to an already overwhelmed hospital system. Furthermore, a global shortage in personal protective equipment (PPE) and other important supplies is only accentuating the risk our caregivers face as they treat patients. Consequently, there is an immediate need for solutions to these problems so that hospitals can operate at their full efficiency at all times.

The current patient examination workflow comprises several critical pre-scan events that involve physical interaction between patients and medical professionals. These include directing patients to the examination room, assisting them in lying down on the bed (to ensure readiness for scan) and helping position them correctly (*e.g.*, moving their limbs *etc.*), all of which are essential to ensure optimal scan parameters. With the objective of reducing these physical interaction events to the maximum extent possible, in this paper, we propose, design, and develop a remote-enabled and contactless system for accurate patient positioning, an important step in a CT examination. Note that while much work in diagnosing COVID-19 has been with CT [1]–[3], chest X-ray (CXR) is another modality that is widely used [4]–[6]. Given this, and our objective above, we emphasize that our system is not limited to just the CT modality. A useful by-product of our system design principle is that we can save scarcely available PPEs, potentially freeing them up for use in absolute emergency scenarios. This is because medical professionals can now conduct the examination while being physically seated in a room safe and far from where the patient might be housed. An illustration of this aspect is shown in Fig. 1, where we show our system deployed in a hospital in China.

There are several aspects that need to be considered as we go about designing such a system from the ground up. Assuming the patient is directed to the examination room (and the CT bed), the system must provide a visual cue (*e.g.*, an appropriately installed camera that can take an image or a sequence of images), analyze the contents of the image(s), and automatically assist the CT scanner in properly positioning (and hence preparing) the patient for a CT scan. This necessitates that the camera is accurately calibrated with respect to

Manuscript received April 12, 2020; revised April 27, 2020; accepted April 29, 2020. Date of publication May 6, 2020; date of current version July 30, 2020. *(Srikrishna Karanam, Ren Li, and Fan Yang contributed equally to this work.) (Corresponding author: Srikrishna Karanam.)*

Srikrishna Karanam, Ren Li, Fan Yang, Terrence Chen, and Ziyan Wu are with the United Imaging Intelligence, Cambridge, MA 02140 USA (e-mail: srikrishna.karanam@united-imaging.com; ren.li@united-imaging.com; fan.yang03@united-imaging.com; terrence.chen@united-imaging.com; ziyan.wu@united-imaging.com).

Wei Hu is with Shanghai United Imaging Healthcare Company, Ltd., Shanghai 201800, China (e-mail: wei.hu@united-imaging.com).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMI.2020.2991954

the coordinate system of the scanner. Furthermore, because we seek contactless functionality, any calibration faults (*e.g.*, accidental movement of either the camera or parts of the scanner) must be corrected efficiently, and preferably, without manual operations, requiring the system to have an automated camera-scanner calibration functionality that routinely monitors for and corrects any deviations. Next, the positioning algorithm itself must estimate a 3D representation of the patient that can then be used to perform positioning with respect to the scanner, which typically is specific to the modality and the scan protocol. Finally, because healthcare professionals can only look at the algorithm performance on a computer screen far away from ground zero, the system needs to provide them with as much information as possible to make sure they are comfortable with the positioning results. One way of doing this is by synthesizing the resulting 3D representation from various alternative viewpoints (*e.g.*, if the camera is placed at the top of the patient, synthesizing a side viewpoint of the positioning result to get a more accurate sense of thickness) so that there is as much redundant information available as possible that can be used to reliably verify positioning before proceeding to the next step in the examination process.

Much recent algorithmic work [7]–[9] in patient positioning has focused on estimating the 2D or 3D keypoint locations on the patient body. Such keypoints represent only a very sparse sampling of the full body mesh in the 3D space that defines the digital human body. However, in many relevant use-cases such as automated thickness estimation [10] and radiation dose optimization, one needs a full 3D mesh and not just the keypoint locations. While there is some recent progress [11] in addressing this problem, this method is limited to CT-specific poses and requires depth data. If we change either the application (*e.g.*, X-ray poses and protocols) or even the particular type of the camera (*e.g.*, some applications may be limited to RGB-only camera), this method will need (a) fresh collection and annotation of data, and (b) retraining the model with this new data, both of which may be prohibitively expensive to do repeatedly for each application separately. Given the extent of the impact of the COVID-19 pandemic, we need to be able to design a system that can quickly be developed and deployed at scale across various modalities and applications. These issues raise an important question that is of obvious practical concern: can we design generic models that can be trained just once and universally used across various scan protocols and application domains? Given that a scan modality or application may have its own needs (typically manifest in the form of the camera choice *e.g.*, RGB-only or RGB-thermal, or specific data scenario, *e.g.*, patient under the cover), a key consideration of our system is to equip the underlying positioning algorithm with what we call *dynamic multi-modal inference* capability. This ensures that the algorithm, and hence the system, can be trained just once and used across multiple applications, leading to scaling, a particularly important aspect as noted above. As a more concrete example, one hospital in country X may have an RGB-only camera, a second hospital in country Y may have a thermal-only camera, and a third hospital in country Z may have an RGB-thermal camera. With our design, the system (with the model trained with both RGB and thermal data) can be used in all the hospitals above without requiring any retraining in each individual hospital. Note that this proposed approach is substantially different than existing state-of-the-art 3D mesh modeling methods such as HMR [12], which shares the same multi-modal limitations as Singh *et al.* [11], *i.e.*, it can be trained only for one modality. A useful byproduct of our algorithm design is built-in redundancy to ensure system robustness. For instance, in the hospital in country Z, even if one modality in the RGB-thermal camera fails (*e.g.*, thermal stops working), the system above will still be able to perform 3D patient modeling with the remaining RGB-only data.

Designing a fully contactless system that enables robust deployment at scale has not been considered in existing research, where the focus primarily has been on laboratory-based algorithm development that is divorced from many of the practical issues and considerations noted above. Some of these problems remain unaddressed even in currently available industrial products. For instance, both Siemens Healthineers [13] and GE Healthcare [14] provide 3D camera solutions for patient positioning and isocentering but still require technicians to physically be present in the scanning room and touch panels mounted on the CT scanner to select/confirm scanning parameters. In this paper, we address these crucial gaps in the current art, which are all the more important given the lessons we are learning from the ongoing COVID-19 pandemic.

II. Contactless Automated Workflow

We begin by briefly describing the workflow associated with our proposed contactless patient positioning system. A block diagram illustration of the system is shown in Fig. 2. Before being put to use, the system executes an **automated camera-scanner calibration** routine (section II-A) that does not involve any manual intervention of a technician. This routine automatically detects locations of pre-defined markers on the scanner bed to establish 2D-3D correspondences that are used, with a standard off-the-shelf perspective-n-point algorithm, to calibrate the camera of the system with respect to the coordinate system of the scanner. Assuming the patient is directed to the correct examination room and the bed in the hospital, the system begins by first ensuring the patient is ready for the scan. This can be achieved by simply tracking key feature locations (*e.g.*, certain keypoints) temporally over a certain number of frames. Once the patient is determined to be ready for the scan, our system executes its patient positioning functionality that takes an image as input and produces a 3D mesh representation. As noted in Section I, this patient positioning module comprises our proposed algorithm capable of **dynamic multi-modal mesh inference** (section II-B). Next, our system comprises a **3D mesh optimizer** (Section II-C) module that, in an online fashion and at a substantially higher speed than competing methods, fine-tunes the mesh output by our positioning module with body keypoints from the detection module in Fig. 2, resulting in a more accurate 3D mesh. This resulting 3D mesh is then used, in conjunction with the calibration information, to accurately position the

Fig. 1. Our contactless positioning system being used in a hospital for diagnosing a patient affected by COVID-19.

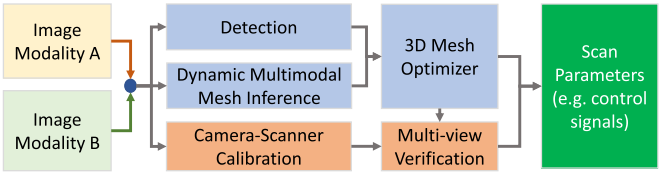

Fig. 2. Components and data flow of the proposed system.

patient on the scanner bed. For instance, in a CT examination, this involves ensuring correct isocentering so that the optimal amount of radiation dosage can be determined for the CT scan. Specifically, based on the estimated 3D representation and the given/selected scan protocol (*e.g.*, Thorax scan), our patient positioning component provides estimates for the scan range and isocenter location, which are used by the technician for verification and initiation of the scan. Finally, our system also comprises a **multi-view synthesis** module (section II-D) to synthesize the mesh from additional viewpoints (*e.g.*, side view if camera is looking down from a top view). This provides crucial information for a technician to verify the accuracy of the estimated mesh before initiating a medical scan.

A. Automated Camera-Scanner Calibration

In order to provide control signals to the scanner bed, we need to be able to go from 3D representation estimated in the camera coordinate system to the scanner coordinate system. For this purpose, we need to compute a spatial transformation from the image plane to scanner coordinate system by means of a system calibration process. Conventional methods calibrating the extrinsic parameters between the camera and scanner systems usually rely on precisely manufactured apparatus [15] and repetitive manual operations that are generally cumbersome to perform on a regular basis, let alone perform it manually during an ongoing pandemic. Consequently, to minimize human intervention and the maintenance effort needed, while ensuring the best possible positioning accuracy at all times, our system comprises an automated calibration process that does not depend on any extra/external apparatus or manual effort, and uses the standard pinhole camera model with the intrinsic matrix $\mathbf{K}$. If $\mathbf{P} = \begin{bmatrix} x & y & z \end{bmatrix}^T$ is a point in the scanner coordinate system, $\mathbf{p} = \begin{bmatrix} u & v & 1 \end{bmatrix}^T$ its corresponding undistorted projection on the image plane, and $\mathbf{R}$ and $\mathbf{T}$ are the rotation matrix and translation vector respectively, we have:

$$\mathbf{p} = \mathbf{K}(\mathbf{RP} + \mathbf{T}) \tag{1}$$

$\mathbf{K}$ is determined in an offline process using a standard checkerboard target [15] and remains unchanged after system installation. To compute $\mathbf{R}$ and $\mathbf{T}$, we establish 2D-3D correspondences with a marker on the patient support. This marker is used to calibrate the patient support to the scanner system using the horizontal and vertical laser beams through the isocenter of the gantry. The 3D location $\mathbf{M}$ of the marker in the scanner coordinate system is a function of the control parameters of the patient support. By moving the patient support with various horizontal and vertical perturbations, we obtain several 3D locations for the marker along with their corresponding 2D image projections $\mathbf{m}$, giving the 2D-3D correspondences set $(\mathbf{M}_i, \mathbf{m}_i)$. This set is then used to solve for $\mathbf{R}$ and $\mathbf{T}$ in Equation 1 using standard robust perspective-n-point solvers [16]. Once $\mathbf{R}$ and $\mathbf{T}$ are determined, they can be used to validate calibration accuracy by back-projecting marker locations $\mathbf{m}_i$ in the image plane to scanner coordinate system and comparing the result with the corresponding 3D ground-truth $\mathbf{M}_i$. Our system considers the calibration process to be successful if the reprojection error is smaller than a pre-defined threshold (we use 4mm). Once the success of the calibration process is determined, our system automatically updates the calibration parameters. Note that throughout this process, there is no need for any manual intervention/operation whatsoever, and hence it can be conducted at any user-desired frequency (*e.g.*, at a fixed time in the day/night).

B. Dynamic Multi-Modal Mesh Inference

As discussed in Section I, a key motivation for the patient positioning component of our system lies in scalability and generality. We seek to train a 3D mesh estimation model that can enable the system to be flexibly used across various applications, modalities, and even hospitals universally without requiring any extensive application- or hospital-specific fine-tuning or retraining. To this end, we propose a new 3D mesh estimation algorithm, called *robust dynamic fusion* (RDF). RDF enables the positioning model to be trained once with all the possible data modalities (*e.g.*, RGB and depth, or RGB and thermal) and used across applications having differing data requirements (*e.g.*, DR may need RGB only or CT may need RGB-D) without having to retrain. This plays a critical role in saving time (which is key given the ongoing COVID-19 pandemic) and quickly scaling up system deployment.

In the following, we first give a brief overview of the various components of the proposed RDF algorithm, summarized in Fig. 3. We then describe each component in greater detail in the next subsequent sections. Note that while all subsequent discussion assumes two pairs of modalities (RGB-thermal and RGB-depth), our approach can easily be extended to more than two modalities by simply adding more convolutional

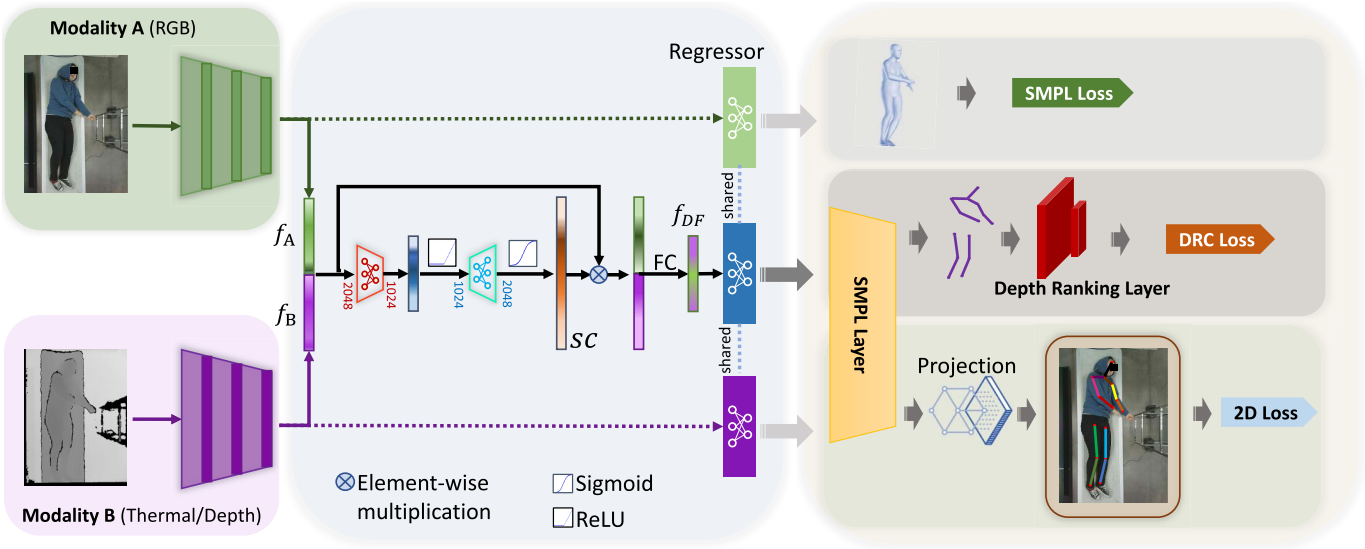

Fig. 3. RDF has multiple CNN branches to learn a joint feature representation, which, along with a fully-connected parameter regressor module, is used to estimate the 3D mesh parameters. "FC" refers to a fully connected unit.

branches in Fig. 3. Given this multi-modal data input, RDF first generates features in a *joint* multi-modal feature space. To ensure robustness of our method to situations involving absence of the two data inputs during inference (*e.g.*, thermal might be unavailable due to a sensor malfunction), we also propose a new training policy that adds noise to the input data while respecting all the possible multi-modal permutations (*e.g.*, "clean RGB and thermal", "RGB with noise and thermal without noise", and "no thermal only RGB"). Our intuition in training an RDF model with this strategy is to ensure the model has "seen" all these possible scenarios and hence is able to infer the correct 3D model parameters.

As can be noted from Fig. 3, given the input images in modalities A and B, RDF first generates feature representations for each of the two images with a two-branch convolutional neural network (CNN) architecture. These two feature vectors are then concatenated and processed with our dynamic feature fusion module to give the feature representation $\mathbf{f}_{\text{DF}}$ of the two input images in the joint feature space. Given the feature representation, RDF estimates the parameters of the 3D model that best describe the person shape and pose in the input with the parameter regressor module. This estimation process is supervised by means of objective functions we discuss next.

1) Multi-Modal Flexibility: Before describing the loss functions in training our RDF model, we first discuss how RDF achieves multi-modal inference flexibility that we discussed above. Given images $\mathbf{I}_A$ and $\mathbf{I}_B$ corresponding to the A and B modalities, we simulate multiple scenarios with a probabilistic data policy. Specifically, we first, with a probability p, select one of the two data streams A/B and replace the corresponding stream's input data array with an array of zeros. With this approach, over the training time, the model will have observed the following scenarios: A only (*i.e.*, simulating absence of B and hence setting $\mathbf{I}_B$ to zero), B only (*i.e.*simulating absence of A and hence $\mathbf{I}_A$ set to zero), and both A and B (*i.e.*, simulating presence of both A and B, hence neither $\mathbf{I}_A$ nor $\mathbf{I}_B$ is zero). This way, the model will be trained to infer the correct 3D mesh parameters under any of these scenarios. Given $\mathbf{I}_A$ and $\mathbf{I}_B$ (with or without the zero changes above), we first extract their individual feature representations with their corresponding CNN branches and concatenate them, giving $\mathbf{f}_{\text{cat}}$. Inspired by [17], we process $\mathbf{f}_{\text{cat}}$ with our feature fusion module, also shown in Fig. 3. This operation essentially generates a new representation $\mathbf{f}_{DF}$ that enables the model to learn which feature dimensions are important, as well as capturing interdependencies between the various input channels and modalities. Specifically, with a set of fully connected units, we output **sc**, a vector of weights highlighting the importance of each channel in the input feature vector $\mathbf{f}_{\text{cat}}$. We then element-wise multiply $\mathbf{f}_{\text{cat}}$ and **sc**, which is then followed by one more fully connected unit to give $\mathbf{f}_{DF}$.

2) Mesh Estimation: Given $\mathbf{f}_{DF}$, RDF estimates a 3D paramteric mesh model with its mesh parameter regressor, which is realized with a set of fully connected layers. In this work, we use the popular Skinned Multi-Person Linear (SMPL) model of Loper *et al.* [18], which is a parametric differentiable model parameterized by the following: shape $\beta \in \mathbb{R}^{10}$ (the first ten coefficients of a principal components analysis projection of the shape space) and pose $\theta \in \mathbb{R}^{72}$ (the three-dimensional axis-angle vector representing the orientation of each of the 24 keypoints defined in the SMPL model). This regressor module takes as input the fused feature vector $\mathbf{f}_{DF}$ and outputs the estimates for pose and shape $\hat{\theta}$ and $\hat{\beta}$ respectively. We use an l_1 distance loss, with the corresponding ground-truth parameters θ and β, to supervise these predictions:

$$L_{\text{mesh}}^{\text{DF}} = \left\| [\theta, \beta] - [\hat{\theta}, \hat{\beta}] \right\|_1 \tag{2}$$

Note that to further strengthen the representation capability of the features representation of each individual modality, we also enforce a loss on the parameters estimated directly from these feature vectors. Specifically, as shown in Fig. 3, we input $\mathbf{f}_A$ to

the regressor module, producing branch A's estimates for pose and shape; similarly we also input $\mathbf{f}_B$ to the regressor module, producing branch B's estimates for pose and shape. In each of these two cases, we enforce a loss on the estimated parameters using the same loss function as in Equation 2. We denote these terms $L^{\text{A}}_{\text{mesh}}$ and $L^{\text{B}}_{\text{mesh}}$, giving an overall mesh estimation loss function:

$$L_{\text{mesh}} = L^{\text{DF}}_{\text{mesh}} + L^{\text{A}}_{\text{mesh}} + L^{\text{B}}_{\text{mesh}} \tag{3}$$

3) 2D Keypoints Prediction: To further ensure $\hat{\theta}$ and $\hat{\beta}$ give accurate 2D keypoints on the input images, RDF comprises an image projection operation that projects the resulting 3D keypoints (computed with $\hat{\theta}$ and $\hat{\beta}$ from $\mathbf{f}_{DF}$) to 2D keypoints on the images. To achieve this, RDF estimates a weak-perspective projection as in HMR [12], giving translation $\rho \in \mathbb{R}^2$ and scale $t \in \mathbb{R}$, which are then used to compute the 2D keypoints with an orthographic projection as $\hat{\mathbf{x}}_i = s\prod(\mathbf{X}_i) + \rho$, where $\mathbf{X}_i$ is the i^{th} 3D keypoint and $\hat{\mathbf{x}}_i$ is its correponding 2D projection. Given the corresponding ground-truth $\mathbf{x}_i$, we supervise this with an l_1 loss:

$$L_{2D} = \sum_i \left\| \mathbf{x}_i - \hat{\mathbf{x}}_i \right\|_1 \tag{4}$$

4) Depth Ranking Consistency: With RDF designed to work with any subset of the input modalities, it may so happen that in some applications the input is RGB-only data. Given the well-established challenges in inferring 3D information from an RGB-only image, in addition to the constraints described above, we propose a new depth ranking consistency (DRC) learning objective to provide more explicit 3D supervision for the model. The goal of DRC is to ensure the estimated 3D mesh parameters respect the relative configuration of the predicted keypoints on the depth data (e.g., one is closer than the other). Note that given the dependence on the depth modality for training with DRC, this component is particular to the scenario involving RGB and depth modalities (not RGB-thermal). The intuition of DRC is to ensure the relative ordering of the predicted joints locations (x_J, y_J, z_J) is consistent with the input joints. Given an input 2D keypoint (x, y), we obtain its corresponding depth z_d from the aligned depth map. While the raw depth value z_d is not directly comparable to the z_J (as the coordinate system definitions may be different), the relative depth orderings (*i.e.*, closer-farther) between each pair of joints z_J and z_d will have to be consistent. To this end, DRC explicitly enforces our network to predict θ and β that satisfies this relative depth ordering property. Specifically, for a pair of joints (p, q), we define its depth ranking relation $r_{p,q}$ as:

$$r_{p,q} = \begin{cases} 1, & \text{if } z^q_d - z^p_d > D \\ -1, & \text{if } z^p_d - z^q_d > D \\ 0, & \text{if } |z^p_d - z^q_d| \leq D \end{cases} \tag{5}$$

where D is the threshold to mitigate the effect of noise in depth maps. DRC penalizes the case when a pair of the inferred 3D joints, from the predicted θ and β, has relative depth relationship that is opposite to the relationship derived from the input depth. Our objective function can be expressed as:

$$L_{drc} = \sum_{(p,q)\in P} L_{p,q} \tag{6}$$

where P represents the set containing the non-repetitive pairs of joints and $L_{p,q} = \log(1 + \exp(r_{p,q} \cdot (z^p_J - z^q_J))$.

5) Overall Loss Function: Our RDF model is trained with an overall loss function that is simply a combination of the three loss functions discussed above:

$$L = L_{\text{mesh}} + L_{2D} + \lambda L_{drc} \tag{7}$$

where λ is an indicator that is 1 for an RGB-D application and 0 otherwise. As discussed in Section I, this approach is substantially different than existing state-of-the-art mesh estimation methods such as HMR [12]. While HMR also regresses mesh parameters from feature representations, it shares the same limitation as Singh *et al.* [11], *i.e.*, it can be trained only for one modality. Specifically, HMR is a one-branch architecture that estimates mesh parameters given data from a single modality. Given this, to use HMR in a multi-modal scenario will involve using two separate branches, one for each modality. Each branch follows the baseline HMR architecture, giving the corresponding modality's feature vector. We then concatenate the two feature vectors, one from each modality, giving the feature vector $\mathbf{f}_{\text{cat}}$, which is used in conjunction with a parameter regressor module to estimate the mesh parameters. Note, however, that this two-branch extension of HMR still does not solve the dynamic multi-modal inference problem. This is because it always assumes the availability of data from both modalities. If either of the two modalities is missing, this technique will produce a non-descriptive $\mathbf{f}_{\text{cat}}$ since it was not trained to handle this scenario (*i.e.*, during training, it assumes images in both branches are always available and neither is missing). On the other hand, our proposed RDF is able to address this limitation by means of the probabilistic data and training policy discussed in Section II-B.

C. 3D Mesh Optimizer

The θ and β estimated by an RDF model trained with the learning objective in Equation 7 can be further fine-tuned in an online fashion. Specifically, given a good initialization and a sufficient number of iterations, the work of Kolotouros *et al.* [19] noted that using an optimization-based iterative approach (*e.g.*, SMPLify [20]) to fit body keypoints (from detection in Fig. 2) typically leads to better results than regression-based approaches. An illustrative example is shown in Fig. 4, where we see by using the output of our RDF approach as a starting point, the SMPLify algorithm was able to substantially refine the 3D mesh (particularly with the hands/arms). Given the need for accurate positioning results for many of the applications discussed in Section I, such an optimizer module will help improve the robustness of our system. However, with their iterative nature, such optimization-based approaches tend to be very slow. This associated computational cost is typically quite high when one considers the near real-time performance requirements of our system. Therefore, the trade-off here is as follows: while we certainly want accurate 3D meshes, we do

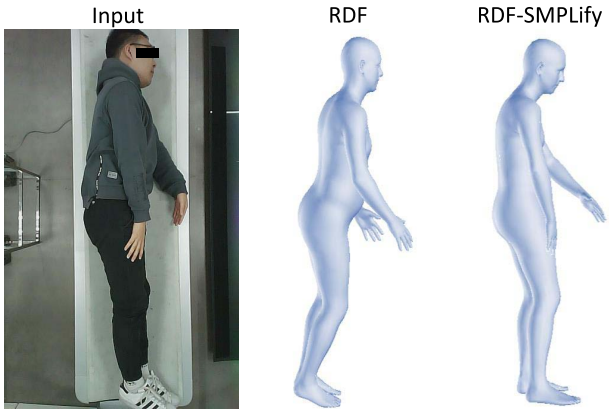

Fig. 4. An illustrative example of the improvement in the mesh estimate by using SMPLify [20] in conjunction with RDF.

not want the underlying operation to be so slow as to render it useless from a compute-time perspective.

To address this challenge, the 3D mesh optimizer component of our system presents an approximate solution that only takes a substantially small fraction of the time needed by the iterative approach. Specifically, we propose to replace the iterative SMPLify algorithm [20] with a fully learned regressor. Our regressor consists of two blocks, each block comprising multiple non-linearly activated (ReLU) fully connected units (with dropout during training). The input to the first block is a vector concatenation of the following quantities: the target 2D keypoints $\mathbf{x}$, the fused feature vector $\mathbf{f}_{\text{DF}}$, and the parameters $\theta = [\theta_{\text{glb}}, \theta_{\text{body}}]$, β, and $[s, \rho]$ estimated by RDF. Note that θ_{glb} represents the global orientation parameters (first three numbers) of the θ vector, whereas θ_{body} represents the vector of all the remaining 69 numbers. The output of this first block are new values for the global orientation $\hat{\theta}_{glb}$ and the camera parameters $\hat{s}$ and $\hat{\rho}$. Next, the second block takes as input a vector concatenation of $\mathbf{x}, \mathbf{f}, \hat{\theta}_{\text{glb}}, \theta_{\text{body}}, \beta, \hat{s}, \hat{\rho}$, and outputs new values for $\hat{\theta}_{\text{body}}$ and $\hat{\beta}$. Note that the idea of taking a two step approach- first estimating the global orientation and camera, followed by the remaining pose and shape parameters, is based on the same two-step approach of the iterative SMPLify algorithm. The $\hat{\theta}$, $\hat{\beta}$, $\hat{s}$, and $\hat{\rho}$ output by our regressor is supervised using the following objective function: $L = \sum_i \left\| \mathbf{x}_i - \hat{\mathbf{x}}_i \right\|_2 + E_\theta(\theta) + E_\beta(\beta)$, where $\mathbf{x}$ are the ground-truth 2D keypoints, $\hat{\mathbf{x}}$ are the estimated 2D keypoints based on the estimated mesh and camera parameters, and E_θ and E_β are the same priors (for pose and shape respectively) as in SMPLify. Once trained, the θ and β estimated by RDF can be fine-tuned by simply performing a forward pass with this learned regressor, which takes much less compute time when compared to the iterative optimization done in SMPLify.

D. Multi-View Synthesis

The contactless pre-scan workflow discussed in Section II essentially isolates the technician from the patient by means of two physically separate rooms. Given this, the technician has to rely on the video stream displayed on the control panel display to monitor the patient, the progress of the current scan, and verify the estimated scan range and positioning/isocentering. With only one camera mounted on top of the ceiling (as shown in the top part of Fig. 5), it can be challenging for the technician to precisely gauge the scan range and isocenting accuracy due to perspective distortion. To address this issue, our system comprises a multi-view synthesis module that can synthesize the estimated positioning results from any desired alternative viewpoint. The motivation is to provide the technician with additional visualizations for more accurate positioning verification. Specifically, given the 2.5D data from an RGB-D camera, we transform it from the current viewpoint ($\mathbf{R}$, $\mathbf{T}$) determined in Section II-A to the user-desired viewpoint ($\mathbf{R}_{sv}, \mathbf{T}_{sv}$) in the scanner coordinate system. The choice of this new viewpoint is configurable in our system and can be easily specified by the user. Alternatively, given application requirements, our system also has the capability to itself suggest this viewpoint information to the user. We synthetically render the given ($\mathbf{R}_{sv}, \mathbf{T}_{sv}$) as: $\mathbf{P}_{sv} = \mathbf{R}_{sv}\mathbf{R}^{-1}(d\mathbf{K}^{-1}\mathbf{p} - \mathbf{T}) + \mathbf{T}_{sv}$. An example result in a CT application is shown in Fig. 5, where we present the top-down view (a) and an alternative viewpoint synthesized from the side (b). From this side-view image (which is essentially a rectified lateral viewpoint), the technician can clearly verify the scan range and isocenter, and make any necessary adjustments before initiating the actual scout scan.

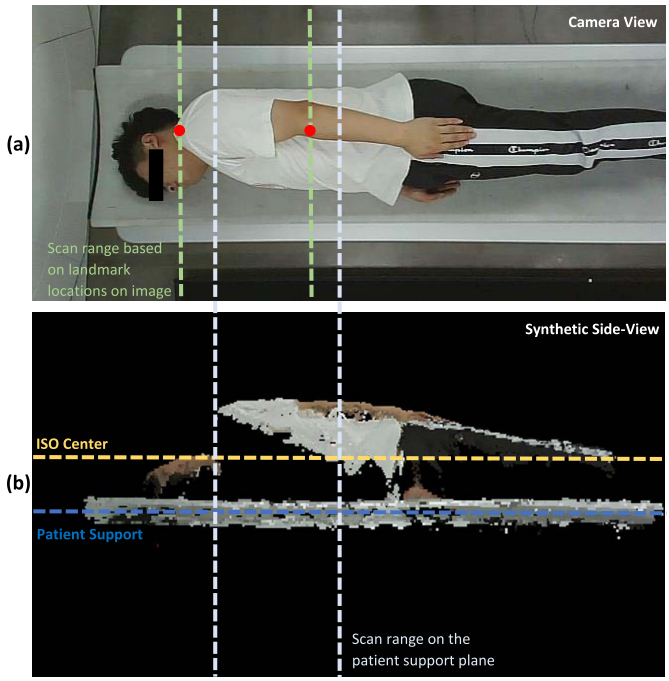

Fig. 5. An example synthetic side-view rendering.

III. Implementation Details

For all RGB and thermal experiments, we use the SLP [9] dataset, which comes with images of people lying on a bed and covered by a cloth under varying cover conditions: no cover (uncover), "light" cover (referred to as cover1), and "heavy" cover (cover2). For all RGB and depth experiments, we use the publicly available CAD [21] and PKU [22] datasets, along

with a proprietary medical scan patient setup (SCAN) dataset. SCAN comprises two parts: SCAN-RGB with 6000 RGB-only patient images in 8 different poses and SCAN-RGBD with 700 RGB-D images of 12 patients. For CAD and PKU, we use the standard train/test split, whereas we create a 350-image/6-patient and 5500-/500-image split SCAN-RGBD and SCAN-RGB respectively. We use ResNet50 [23] for both encoders of RDF, which, along with the parameter regressor, is pretrained on Human3.6M [24]. We set all loss weights in our loss function to 1.0 and train with the Adam optimizer with a batch size of 64, input size of 224×224, and a learning rate of 0.0001 (multiplied by 0.9 every 1,000 iterations). For evaluation, we use standard metrics [24]: 2D mean per joint position error (MPJPE) in pixels and 3D MPJPE in millimeters.

IV. Experiments

In this section, we conduct several experiments to evaluate the efficacy of the various components of our system. For the calibration component of the system, we ensure the reprojection error is within 4mm, as noted in Section II-A. The multi-view synthesis component of our system is a tool for qualitative visualization of an alternative viewpoint, and we provided an example result in Fig. 5. In the following, we evaluate, both quantitatively and qualitatively, the other two algorithmic components of our system, viz., the RDF approach of Section II-B and the optimizer approximation of Section II-C. Note that while we show results with two separate two-modality scenarios: RGB-T (A = RGB, B = thermal) and RGB-D (A = RGB, B = depth), our system is scalable and can be used with any number of input modalities (we accordingly need to modify the number of input branches in Fig. 3). In our evaluation, while we compare RDF's performance to that of a competing state-of-the-art mesh recovery algorithm, HMR [12], we emphasize that the crux of our study is in demonstrating RDF's flexibility with multi-modal inference. HMR, by design, can be used with only one data modality at a time. Therefore, as noted in Section II-B(e), we extend HMR to process two data modalities with a two-stream architecture, using $\mathbf{f}_{\text{cat}}$ to regress the mesh parameters.

A. Multi-Modal Inference Evaluation

We begin with multi-modal inference evaluation. Table I shows average 2D and 3D MPJPE, with standard deviation (std) in parentheses, across all test images in our five datasets. In the "HMR" row, the "RGB" sub-row indicates training and testing on RGB-only data (similarly for depth "D" and thermal "T"). The "RGB-T" (and "RGB-D") row indicates the two-stream baseline with with $\mathbf{f}_{\text{cat}}$ discussed above. Since RDF is trained with our probabilistic data policy, it processes, during training, both RGB and depth (or thermal) modalities, and hence we only see "RGB-T" or "RGB-D" in the "Train" column. During inference, while the baseline can only process the same kind of data it saw during training, RDF, by design, can work with any input modality (RGB only, D only, or RGB-D, and the corresponding cases with thermal).

TABLE I
MPJPE Results for Various Datasets: We Show the Average (Std in Parentheses) MPJPE Across All Test Samples

	Method	Train	Test	2D MPJPE	3D MPJPE
SLP	HMR [12]	RGB	RGB	37.2 (14.2)	155 (76)
		T	T	34.2 (14.1)	149 (75)
		RGB-T	RGB-T	34.1 (18.8)	143 (81)
	RDF	RGB-T	RGB	36.6 (14.5)	144 (75)
			T	34.7 (14.6)	138 (74)
			RGB-T	**32.7** (14.2)	**137** (78)
CAD	HMR [12]	RGB	RGB	7.9 (4.5)	120 (43)
		D	D	9.2 (7.6)	118 (39)
		RGB-D	RGB-D	6.7 (3.6)	103 (27)
	RDF	RGB-D	RGB	6.1 (3.2)	106 (40)
			D	7.2 (6.4)	104 (29)
			RGB-D	**5.7** (2.5)	**97** (29)
PKU	HMR [12]	RGB	RGB	8.8 (5.0)	127 (49)
		D	D	13.2 (8.7)	150 (50)
		RGB-D	RGB-D	8.2 (5.7)	118 (47)
	RDF	RGB-D	RGB	7.7 (5.0)	123 (45)
			D	11.8 (7.0)	133 (51)
			RGB-D	**8.1** (5.0)	**106** (44)
SCAN-RGBD	HMR [12]	RGB	RGB	25.6 (10.0)	168 (46)
		D	D	23.7 (12.6)	150 (45)
		RGB-D	RGB-D	21.8 (9.4)	144 (44)
	RDF	RGB-D	RGB	17.8 (7.1)	117 (41)
			D	21.6 (12.5)	116 (42)
			RGB-D	**16.2** (6.9)	**103** (38)
SCAN-RGB	HMR [12]	RGB	RGB	12.1 (5.9)	84 (44)
	RDF	RGB-D	RGB	**11.6** (5.1)	**82** (42)

Given this background, we make several observations. In the RGB-T scenario, our method with RGB-only data (144mm average 3D MPJPE) is better than the corresponding baseline result (155mm average 3D MPJPE) because it has the ability to use the available privileged thermal data, helping improve its RGB-only performance. Thermal-only inference with our method (138mm) is also better than the baseline (149mm) for similar reasons (RGB acting as the extra supervision source). Finally, the RGB-T inference performance of our method is better (137mm) than the corresponding baseline (143mm), demonstrating the impact of our feature fusion as opposed to simple concatenation ($\mathbf{f}_{\text{DF}}$ vs. $\mathbf{f}_{\text{cat}}$). We note similar observations from the RGB-D results as well in Table I.

B. Under-the-Cover Evaluation

Some applications of our system involve positioning when the patient is covered by a cloth. To evaluate the system's performance in such challenging scenarios, we conduct a more detailed patient cloth coverage study, with results shown in Table II. In this experiment, we use the available labels (uncover, cover1, cover2 above) of the SLP dataset and present results on each sub-dataset individually. As expected, as cloth coverage increases (uncover to cover2), the performance generally drops (increasing MPJPE). Next, also along expected lines, with the RGB modality being less able (compared to thermal) to "see under the cover", the inference performance

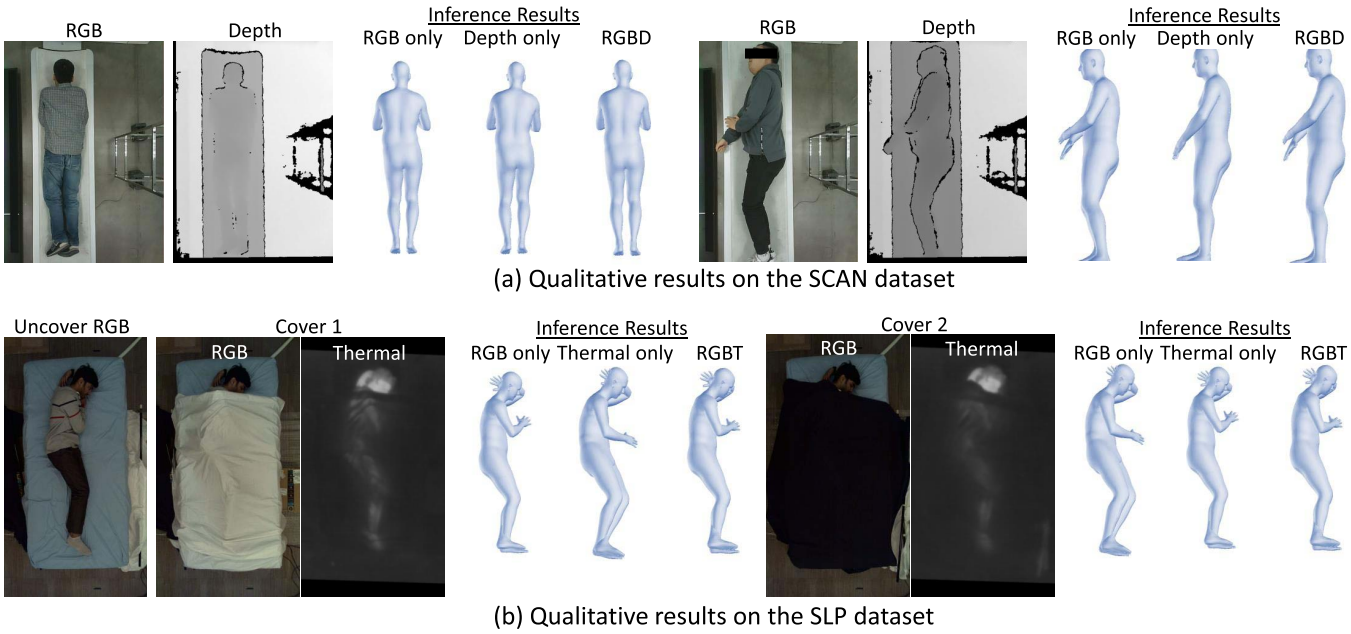

Fig. 6. Qualitative results of the proposed approach on the SCAN and SLP datasets.

TABLE II
AVERAGE 3D MPJPE, WITH STD IN PARENTHESES, RESULTS FOR THE SLP DATASET UNDER VARIOUS COVER SCENARIOS

Test modality	RGB			Thermal			RGB-T		
Cover condition	uncover	cover1	cover2	uncover	cover1	cover2	uncover	cover1	cover2
HMR [12]	139 (74)	150 (73)	154 (80)	145 (76)	149 (75)	151 (74)	141 (93)	145 (85)	143 (84)
RDF	**137** (75)	**146** (76)	**150** (77)	**135** (68)	**138** (76)	**140** (79)	**134** (79)	**137** (78)	**141** (77)

of RDF under RGB is generally lower. However, an interesting aspect of these numbers is that under the highest intensity of cloth coverage (cover2), the performance under the RGB-T scenario is generally better than that under thermal alone. This suggests that even under heavy cloth coverage, the RGB modality still has the potential to provide complementary information to improve thermal-only performance. Finally, we note that RDF's performance is generally better than HMR, across all modalities and cloth conditions, further substantiating our claims above. We also show some qualitative renderings of RDF's mesh output in Fig. 6, where we provide mesh outputs in all three cases. Note that the results in the RGB-D scenario is relatively more consistent across the three inputs when compared to the RGB-T scenario. This is likely due to the particular challenges in the covered scenario, where the RGB modality does not provide as much information as in the case when there is no cover (*e.g.*, first row in Fig. 6).

C. Additional Algorithmic Evaluation

1) Impact of Depth Ranking Consistency: Table III shows results of our RDF approach with and without the proposed DRC learning objective. One can note adding L_{drc} to the training objective results in a consistent decrease (across all test modalities) in MPJPE. The performance improvements are more notable with 3D MPJPE, which is expected since DRC essentially attempts to address the depth ambiguity problem by means of explicit constraints on 3D keypoints.

2) Impact of Feature Fusion: In Table IV, we show results of RDF with and without our feature fusion module, where

TABLE III
IMPACT OF DEPTH RANKING CONSISTENCY (DRC): AVERAGE (STD IN PARENTHESES) MPJPE RESULTS WITHOUT DRC (RDF) AND WITH DRC (RDF-DRC)

	Train	Test	CAD		SCAN-RGBD	
			2D MPJPE	3D MPJPE	2D MPJPE	3D MPJPE
RDF	RGB-D	RGB	6.1 (3.2)	106 (40)	16.8 (6.8)	106 (47)
		D	7.2 (6.4)	104 (29)	15.1 (7.9)	99 (50)
		RGB-D	5.7 (2.5)	97 (29)	14.2 (5.7)	100 (51)
RDF-DRC	RGB-D	RGB	6.0 (2.9)	100 (31)	14.4 (8.1)	**96** (47)
		D	6.7 (4.7)	98 (29)	14.9 (6.8)	**96** (39)
		RGB-D	**5.6** (2.4)	**90** (25)	**13.4** (6.0)	97 (45)

(a) CAD and SCAN-RGBD

	Train	Test	2D MPJPE	3D MPJPE
RDF	RGB-D	RGB	11.6 (5.1)	82 (42)
RDF-DRC	RGB-D	RGB	**11.4** (5.0)	**80** (40)

(b) SCAN-RGB

RDF (w/o DF) refers to using $\mathbf{f}_{\text{cat}}$ for parameter regression. From these results, we note that $\mathbf{f}_{\text{DF}}$ gives better performance (lower MPJPE) when compared to $\mathbf{f}_{\text{cat}}$, further demonstrating the efficacy of the feature fusion component of RDF.

D. Robustness to Noise

To evaluate and quantify robustness to noise, we vary the probability of a particular modality missing at test time, *i.e.*, with probability p, we replace each $\mathbf{I}_{m_i}$ with a zero array. We disregard the case when both modalities are missing. We then infer the resulting 3D mesh and compute the 2D

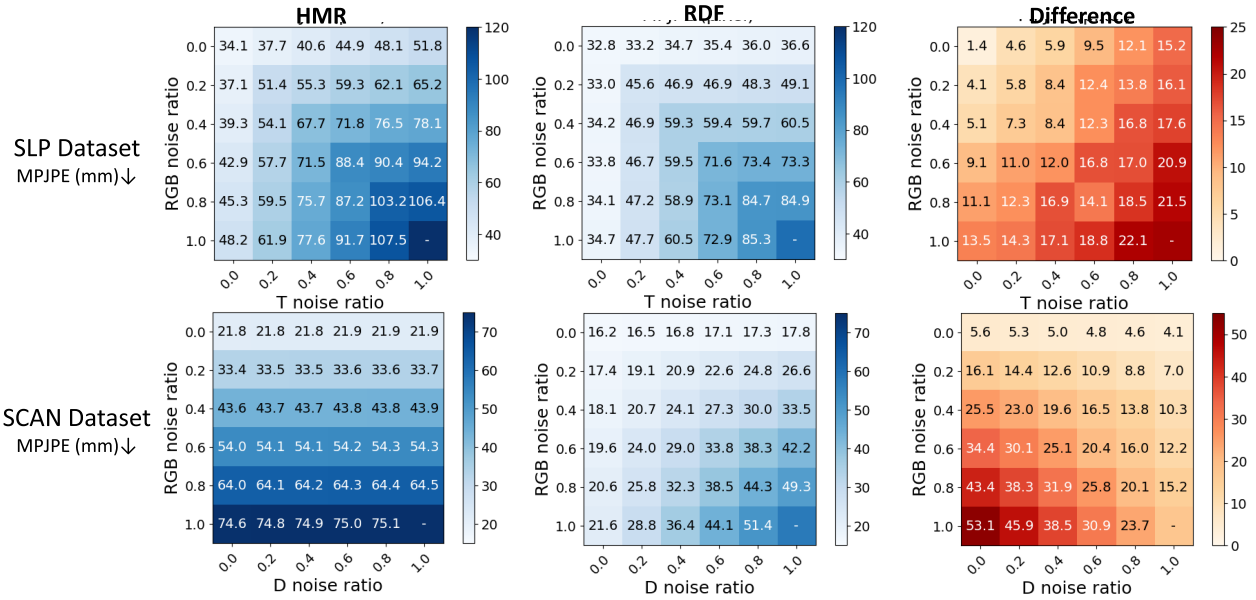

Fig. 7. MPJPE performance comparison of RDF and HMR at various noise levels. "T": thermal,"D": depth.

TABLE IV
IMPACT OF FEATURE FUSION (DF): AVERAGE (STD IN PARENTHESES) MPJPE RESULTS WITHOUT AND WITH DF

	Train	Test	CAD		PKU	
			2D MPJPE	3D MPJPE	2D MPJPE	3D MPJPE
RDF (w/o DF)	RGB-D	RGB	6.1 (2.3)	109 (34)	7.9 (5.0)	130 (47)
		D	8.4 (4.8)	122 (39)	12.9 (7.5)	152 (47)
		RGB-D	6.0 (2.2)	99 (31)	8.2 (4.9)	111 (47)
RDF	RGB-D	RGB	6.1 (3.2)	106 (40)	7.7 (5.0)	123 (45)
		D	7.2 (6.4)	104 (29)	11.8 (7.0)	133 (51)
		RGB-D	**5.7** (2.5)	**97** (29)	**8.1** (5.0)	**106** (44)

(a) CAD and PKU

	Train	Test	2D MPJPE	3D MPJPE
RDF (w/o DF)	RGB-T	RGB	37.4 (15.6)	145 (78)
		T	35.6 (15.4)	134 (76)
		RGB-T	33.2 (16.9)	137 (80)
RDF	RGB-T	RGB	36.6 (14.5)	144 (75)
		T	34.7 (14.6)	138 (74)
		RGB-T	**32.7** (14.2)	**137** (78)

(b) SLP

MPJPE. Fig. 7 shows a heatmap of the 2D MPJPE of both RDF and the baseline HMR, where we note performance of both RDF and HMR go down as the noise level increases (as expected). However, this degradation is much less for RDF compared to HMR (also shown in the difference matrix), suggesting better robustness of RDF and demonstrating an important practical aspect for robust system deployment.

E. Evaluating the 3D Mesh Optimizer

We quantify the importance of the mesh optimizer module, discussed in Section II-C, with particular emphasis on the associated run-time efficiency. Specifically, we conduct three experiments; first, using RDF without any optimizer, second, using RDF with SMPLify [20] as the optimizer, and finally, using RDF with our method of Section II-C as the optimizer.

TABLE V
AN MPJPE AND FPS COMPARISON OF RDF, RDF-SMPLIFY, AND RDF-OPT

Method	SLP			SCAN		
	2D MPJPE	3D MPJPE	FPS	2D MPJPE	3D MPJPE	FPS
RDF	32.7	137	66.7	13.4	97	66.7
RDF-SMPLify	21.6	99	0.8	11.3	83	0.7
RDF-OPT	25.2	107	50.8	12.7	84	58.

Method	CAD			PKU		
	2D MPJPE	3D MPJPE	FPS	2D MPJPE	3D MPJPE	FPS
RDF	5.6	90	66.7	8.1	106	66.7
RDF-SMPLify	2.8	80	0.4	4.0	106	0.4
RDF-OPT	3.8	79	62.5	5.2	95	58.8

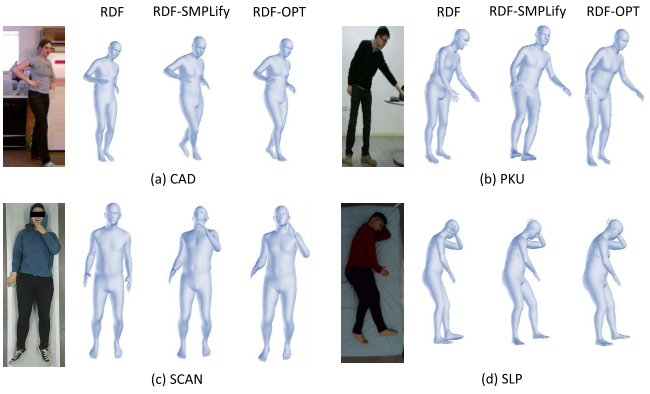

Fig. 8. Mesh estimation comparison of RDF, RDF-OPT, and RDF-SMPLify.

The results are shown in Table V. The performance of RDF as is without any optimizer (row 1 in all tables) is the worst among the three, and this is not surprising since the optimizer is expected to only improve the estimated mesh. In terms of MPJPE, RDF-SMPLify gives the best performance, and this is also not surprising since given sufficient iterations, SMPLify is expected to give a very good mesh fit. However,

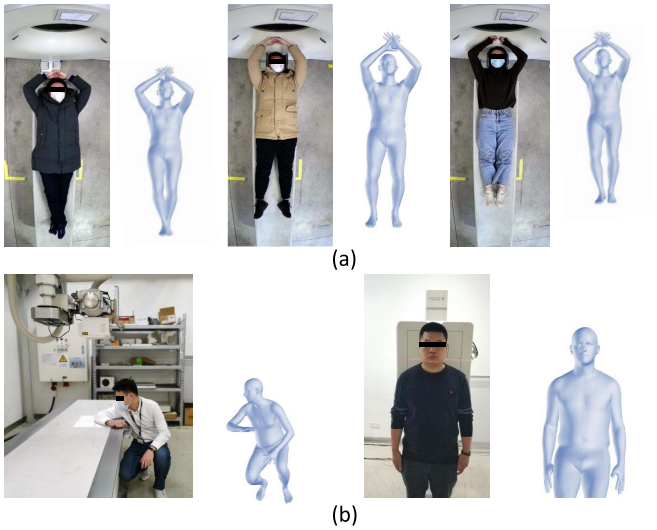

Fig. 9. RDF-OPT mesh estimation results with mask images and an X-ray application.

crucially: RDF-OPT (our proposed approximation) runs at a substantially higher frame rate (*e.g.*, 62.5 fps vs. 0.54 fps for RDF-SMPLify) while performing better than RDF and reasonably close to, and in some cases even outperforming, RDF-SMPLify (*e.g.*, 79mm 3D MPJPE for RDF-OPT vs. 80mm for RDF-SMPLify, see also Fig. 8). As can be noted from Fig. 8, while RDF is able to provide a reasonable estimate of the mesh, the accuracy of some parts (*e.g.*, legs in left column and right hand in right column of first row) is not satisfactory. This issue is addressed by using the mesh optimizer in conjunction with RDF, with RDF-OPT giving (approximately) similar results as RDF-SMPLify. These results provide evidence for our approximation achieving a good trade-off between speed and accuracy (from Section II-C). Finally, in Fig. 9, we show additional qualitative results with patients wearing a face mask as well as an X-ray application.

V. Summary

The COVID-19 pandemic has resulted in a substantial shortage in personal protective equipment and increased the likelihood of medical professionals getting infected. In this paper, we presented the design and development of a contactless patient positioning system that took a step towards addressing these problems. We presented several components of the system including automated calibration, positioning, and multi-view synthesis routines that we showed enabled the possibility of remotely scanning a patient without physical proximity. We evaluated our system with extensive experiments on public as well as proprietary datasets, and showed how it can be used for a variety of applications without significant re-training, thus enabling deployment at scale. While the proposed method provides an efficient and contactless workflow for medical scans, it does not restrict or limit medical professionals from performing the scan in close proximity with patients if that is desired in a non-pandemic scenario.

References

[1] F. Pan *et al.*, "Time course of lung changes on chest CT during recovery from 2019 novel coronavirus (COVID-19) pneumonia," *Radiology*, Feb. 2020, Art. no. 200370, doi: 10.1148/radiol.2020200370.

[2] T. Ai *et al.*, "Correlation of chest CT and RT-PCR testing in coronavirus disease 2019 (COVID-19) in China: A report of 1014 cases," *Radiology*, Feb. 2020, Art. no. 200642, doi: 10.1148/radiol.2020200642.

[3] Y. Fang *et al.*, "Sensitivity of chest CT for COVID-19: Comparison to RT-PCR," *Radiology*, Feb. 2020, Art. no. 200432, doi: 10.1148/radiol.2020200432.

[4] A. Narin, C. Kaya, and Z. Pamuk, "Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks," *CoRR*, vol. abs/2003.10849, pp. 1–17, Mar. 2020.

[5] I. D. Apostolopoulos and T. Bessiana, "Covid-19: Automatic detection from X-ray images utilizing transfer learning with convolutional neural networks," *CoRR*, vol. abs/2003.11617, pp. 1–8, Mar. 2020.

[6] L. Wang and A. Wong, "Covid-net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest radiography images," *CoRR*, vol. abs/2003.09871, pp. 1–7, Mar. 2020.

[7] V. Srivastav, T. Issenhuth, A. Kadkhodamohammadi, M. de Mathelin, A. Gangi, and N. Padoy, "MVOR: A multi-view RGB-D operating room dataset for 2D and 3D human pose estimation," *CoRR*, vol. abs/1808.08180, pp. 1–10, Aug. 2018.

[8] V. Srivastav, A. Gangi, and N. Padoy, "Human pose estimation on privacy-preserving low-resolution depth images," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, Shenzhen, China, 2019, pp. 583–591.

[9] S. Liu and S. Ostadabbas, "Seeing under the cover: A physics guided learning approach for in-bed pose estimation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, Shenzhen, China, 2019, pp. 236–245.

[10] J. Li, U. K. Udayasankar, T. L. Toth, J. Seamans, W. C. Small, and M. K. Kalra, "Automatic patient centering for MDCT: Effect on radiation dose," *Amer. J. Roentgenol.*, vol. 188, no. 2, pp. 547–552, Feb. 2007.

[11] V. Singh *et al.*, "DARWIN: Deformable patient avatar representation with deep image network," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, Montreal, QC, Canada, 2017, pp. 497–504.

[12] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *Proc. IEEE Conf. Comput. Vis. Pattern Regognition*, Salt Lake City, UT, Jun. 2018, pp. 7122–7131.

[13] Siemens Healthineers. (Apr. 12, 2020). *FAST Integrated Workflow*. [Online]. Available: https://www.siemens-healthineers.com/computedtomography/technologies-and-innovations/fast-integrated-workflow

[14] GE Healthcare. (Apr. 12, 2020). *GE Revolution Maxima*. [Online]. Available: https://www.gehealthcare.com/products/computedtomography/revolution-maxima

[15] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.

[16] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate O(n) solution to the PnP problem," *Int. J. Comput. Vis.*, vol. 81, no. 2, pp. 155–166, Feb. 2009.

[17] H. Jie, S. Li, and S. Gang, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Regognition*, Salt Lake City, UT, USA, Jun. 2018, pp. 7132–7141.

[18] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 1–16, Nov. 2015.

[19] N. Kolotouros, G. Pavlakos, M. Black, and K. Daniilidis, "Learning to reconstruct 3D human pose and shape via model-fitting in the loop," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 2252–2261.

[20] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, "Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, 2016, pp. 561–578.

[21] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Human activity detection from RGBD images," in *Proc. AAAI Conf. Artif. Intell. Workshops*, San Francisco, CA, USA, 2011, pp. 47–55.

[22] C. Liu, Y. Hu, Y. Li, S. Song, and J. Liu, "PKU-MMD: A large scale benchmark for continuous multi-modal human action understanding," *CoRR*, vol. abs/1703.07475, pp. 1–10, Mar. 2017.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[24] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1325–1339, Jul. 2014.