Outside-in Monocular IR Camera based HMD Pose Estimation via Geometric Optimization

Pavel A. Savkin
Fove Inc./ Waseda University
pavel.savkin@fove-inc.com

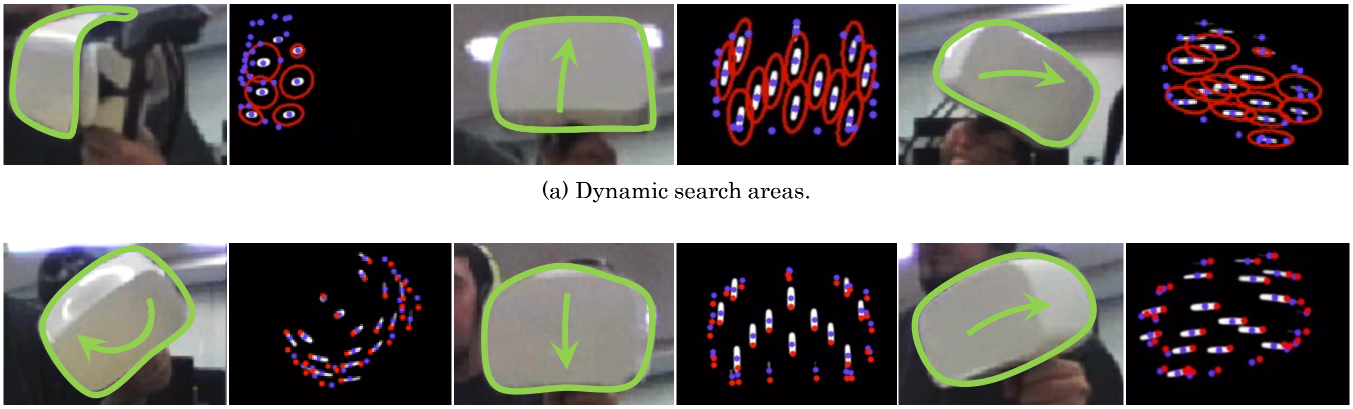
Shunsuke Saito
Fove Inc.
shunsuke.saito@fove-inc.com

Jarich Vansteenberge
Fove Inc.
jarich.vansteenberge@fove-inc.com

Tsukasa Fukusato
The University of Tokyo
tsukasafukusato@is.s.u-tokyo.ac.jp

Lochlainn Wilson
Fove Inc.
lochlainn.wilson@fove-inc.com

Shigeo Morishima
Waseda University
shigeo@waseda.jp



(a) Dynamic search areas.

(b) Pose deblurring.

Figure 1: Tracking sequences results - (a) the search area (red ellipse). The search areas are reduced in a self-occluded region and are updated by HMD motion (the direction of the green arrow). (b) Pose deblurring. The HMD's IR-LEDs (white blobs) are blurred due to fast motions and camera exposure time. Our method refines the estimated pose (blue dots) by determining the accurate location of IR-LEDs (red dots) in real time. For more results, please refer to the supplementary video.

ABSTRACT

Accurately tracking a Head Mounted Display (HMD) with a 6 degree of freedom is essential to achieve a comfortable and a nausea free experience in Virtual Reality. Existing commercial HMD systems using synchronized Infrared (IR) camera and blinking IR-LEDs can achieve highly accurate tracking. However, most of the off-the-shelf cameras do not support frame synchronization. In this paper, we propose a novel method for real time HMD pose estimation without using any camera synchronization or LED blinking. We extended over the state of the art pose estimation algorithm by introducing geometrically constrained optimization. In addition, we propose a novel system to increase robustness to the blurred IR-LEDs patterns appearing at high-velocity movements. The quantitative evaluations showed significant improvements in pose stability and accuracy over wide rotational movements as well as a decrease in runtime.

CCS CONCEPTS

•**Computing methodologies** → Tracking; Virtual Reality;

KEYWORDS

Perspective-n-Point problem Monocular IR camera Vision-based pose estimation Position tracking

ACM Reference format:
Pavel A. Savkin, Shunsuke Saito, Jarich Vansteenberge, Tsukasa Fukusato, Lochlainn Wilson, and Shigeo Morishima. 2017. Outside-in Monocular IR Camera based HMD Pose Estimation via Geometric Optimization. In *Proceedings of VRST '17, Gothenburg, Sweden, November 8–10, 2017,* 9 pages.
DOI: 10.1145/3139131.3139136

VRST '17, Gothenburg, Sweden
© 2017 ACM. 978-1-4503-5548-3/17/11…$15.00
DOI: 10.1145/3139131.3139136

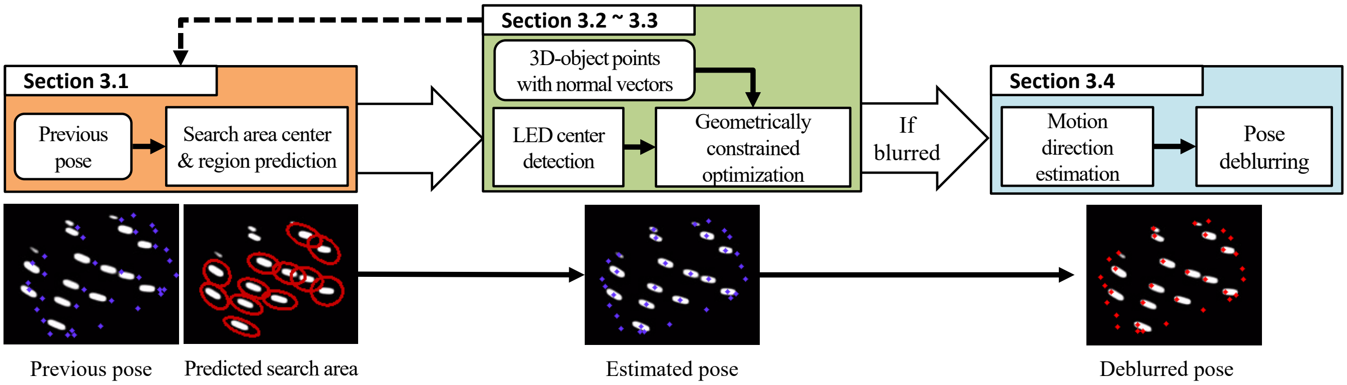


Figure 2: Our system workflow. The GMM component (search area) for next frame is predicted based on linear prediction with previous frames pose data. The 2D-image feature points are extracted and the correspondence candidate is chosen by Mahalanobis distance and geometric constraints. Then, the pose is updated with Kalman Filter to find next proper correspondence. For reducing blur effect caused by HMD fast movement, the pose is recalculated by computing the end points of the blurred IR-LEDs.

1 INTRODUCTION

Virtual Reality (VR) technologies are getting increasingly popular in various fields such as among others, gaming, CAD design, psychology, and medical science [LaValle 2017]. Accurately tracking the position and orientation of a user's head is extremely important to ensure a comfortable experience in VR.

Position tracking is typically done by fusing information from two different measurements. An estimation based on visual information obtained from a camera, and an estimation acquired from embedded sensors called inertial measurement units (IMUs). In this paper, we focus on providing fast and highly accurate 6 degrees of freedom (6DoF) pose estimation solemnly based on the visual information provided by the camera. Our objective is to improve the quality of the estimated pose before fusing it with IMUs measurements.

Estimating 6DoF pose from a vision data is a well-studied problem in a computer vision field [Marchand et al. 2016]. Methods used to solve these problems can be applied to various fields, such as simultaneous localization and mapping (SLAM) [Fuentes-Pacheco et al. 2015] in robotics. One of the most common ways of estimating poses from a monocular 2D camera is called Perspective-n-Point (PnP) problem [Marchand et al. 2016]. This technique requires 2D-image feature points to be extracted, and find a correspondence between 3D-object feature points of a target object. The 6DoF pose is estimated in form of camera position and orientation by solving a non-linear optimization. Then objects' pose can be calculated by changing coordinate systems.

In VR, reducing a mismatch in position and orientation compared to the real movement is essential. Otherwise, the user may experience nausea while playing in VR. Thus, to estimate highly accurate 6DoF pose, many VR products use infrared (IR) cameras and active markers such as IR-LEDs [LaValle 2017], and solve PnP problem. Since IR-LEDs have quite a similar appearance to each other compared to natural markers, it remains problematic to compute correspondence between 3D-object points and 2D-image points directly. Therefore, consumer products such as Oculus and OSVR avoid this by blinking LEDs with different patterns to differentiate each LED. While these systems employ camera frame synchronization, such technology is not available in off-the-shelf IR cameras or RGB cameras with IR filters. Thus, achieving highly accurate Head Mounted Display (HMD) pose estimation without camera frame synchronization is tricky.

In this paper, we propose a novel method to estimate a highly accurate HMD pose from monocular IR camera and IR-LEDs. Our contributions are as follows:

- We present a robust HMD pose estimation method which doesn't require camera frame synchronization or blinking LEDs.
- We propose geometric optimizations to achieve robust and real-time tracking performances over wide rotations and fast motions.
- We introduce a method to reduce the tracking sensitivity to motion blur.

In the next section, we review the related works. In Section 3, we detail our method and review and discuss results in Section 4. Finally, we conclude our work in Section 5.

2 RELATED WORK

2.1 IR based Pose Estimation

Several methods dealing with IR-LEDs mounted object tracking have been proposed over the years. Censi et al. [Censi et al. 2013] proposed a pose estimation method with IR-LEDs and Dynamic Vision Sensors (DVS). They achieved low-latency pose estimation and the use of DVS has achieved tracking object at extremely fast movements. However, the DVS is more expensive than full consumer VR systems, which is not suitable for casual use. Faessler et al. [Faessler et al. 2014] proposed a method to track a quadrotor with monocular IR camera and few IR-LEDs. The method achieved a high accuracy, fast computation, and the algorithm can be generally applied to common IR cameras with IR-LEDs. Also, Tjaden et al. [Tjaden et al. 2015] proposed a method which also can track an object at high accuracy and fast computation with few IR-LEDs. They proposed a marker structure and an algorithm which can find the 2D-3D correspondence robustly at speed of 2 [ms].

Although these methods enable to track the object accurately with few LEDs, the small number of LEDs restricts the ability to track the object for more than 90 [deg] of yaw and pitch with the monocular camera. Their algorithms have two major drawbacks; an increase in computation time along with the number of tracked LEDs as well as a significant drop in estimated pose accuracy due to occlusions and a large number of outliers. Thus, their method is ill suited to VR HMD tracking as self-occlusion and wide rotation are inevitable in such scenarios.

2.2 Simultaneous Pose and Correspondence Estimation

In computer vision field, several techniques have been proposed to estimate a pose which deal with outliers and occlusions where no correspondence between a large amount of 3D-object and 2D-image feature points is given. Although a RANSAC algorithm [Fischler and Bolles 1981] combined with fast PnP problem computation [Lepetit et al. 2009] may solve the problem, this type of random search quickly become computationally insufficient with large numbers of points. Therefore, David et al. [David et al. 2004] introduced an algorithm called SoftPosit which iteratively estimates the pose and the correspondence with high accuracy by optimizing a global cost function. They combined a pose estimation technique with an algorithm to assign correspondences which achieved an $O(MN^2)$ complexity, where M is the number of 3D-object points, and N is the number of 2D-image points. To allow faster pose estimation with higher accuracy, Moreno et al. [Moreno-Noguer et al. 2008] constructed a recursive system called BlindPnP, by (1) restricting the search area with Gaussian Mixture Models (GMM) which contains several means value and covariances, and (2) updating pose with Kalman Filter. The algorithm achieved $O(Gn^3 M \log N)$ complexity, where G is the number of GMM components and n is the number of 2D-image correspondences that can be potentially matched to each 3D-object point. In [Brown et al. 2015], Brown et al. proposed an updated method which can estimate the globally optimal pose. This method estimates both the correspondence and the pose by combining the SoftPosit algorithm with Branch and Bound (BnB) method to achieve a globally optimal solution. Nevertheless, their method is not suited for real-time applications such as VR.

In consideration of the accuracy and speed of the methods mentioned above, we decided to focus our work on the well balanced BlindPnP [Moreno-Noguer et al. 2008] algorithm. Specifically, we focus our attention on the improvement of its accuracy and runtime.

3 PROPOSED METHOD

In order to achieve fast and accurate HMD pose estimation with monocular IR camera and a larger number of IR-LEDs, we propose a novel method, which uses the modified BlindPnP algorithm. Under an assumption that the first frame was initialized, previous frames' results with only single GMM component is used, since multiple GMMs increase the computation time. To ensure the global optima, the current pose and its probable existing area are calculated based on previous pose and linear-based next pose prediction. Then the recursive function based optimization used in BlindPnP was carried. Here, the original algorithm suffered from self-occluded situation since there are no corresponding constraints. Thus, we add geometric constraints by using 3D-object point normal vectors to avoid local optima and increase of computation time in case of self-occlusion. To achieve even higher accuracy, we propose a pose deblurring algorithm to deal with fast motion, where the observed IR-LEDs are blurred in the moving direction due to the camera exposure. Fig. 2 describes our workflow. Key points of our method are as follows.

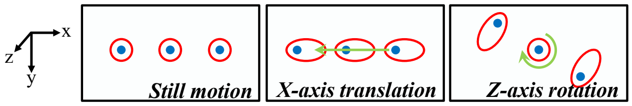


Figure 3: An example of dynamic search areas. Blue dots represent IR-LED markers observed in 2D-image. Red ellipses represent GMM covariance projected onto the 2D-image, which are treated as a search area. By applying our updating approach, the search area is automatically enlarged or reduced according to the predicted motion.

- Both robust and fast pose estimation have been achieved by applying modified BlindPnP [Moreno-Noguer et al. 2008] algorithm with single GMM component and previously calculated HMD pose. This was achieved by predicting the current frame pose with linear-based prediction and defining the probable search area.
- Self-occlusion has been properly considered in BlindPnP optimization function by introducing normal vector based geometric constraints. By calculating each 3D-object points' dot product with respect to the camera direction, the self-occlusion status was decided and some points were skipped in the optimization.
- The blur effect has been considered in case of fast motion where the IR-LEDs are blurred due to the camera exposure. The accurate positions of IR-LED were calculated based on ellipse fitting and estimated movement directions, and then used to deblur the pose.

Hereafter, we explain our method's workflow, which consists of four steps: (1) updating the single GMM component based on the previous frames' pose (Section 3.1); (2) extracting 2D-image feature points and finding the best correspondence between the 2D-image extracted points and the projected 3D-object feature points (Section 3.2); (3) updating pose and search area using Kalman Filter and repeating the process until the convergence (Section 3.3); and (4) re-estimating the pose using deblurred 2D feature points appearing in fast motion (Section 3.4). Here, we assume that the initialization step went correctly with the estimated pose (the initialization methodology will be briefly explained in Section 4).

3.1 GMM Component Update

To find the 2D-3D correspondence with a correct pose, the initial search area should be properly defined. In the original BlindPnP algorithm [Moreno-Noguer et al. 2008], 20 GMM components were used to initialize the search area, resulting in a high computation.

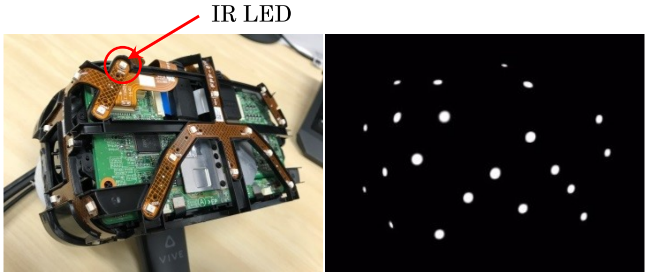


Figure 4: (Left) IR-LED layout. (Right) Observed IR-LEDs.

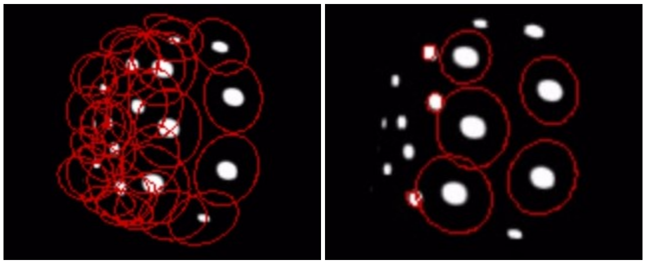


Figure 5: (Left) Search area without normal-based constraints, (Right) and with normal-based constraints. Note that the search area rendering the correspondence problem more complex without the constraints.

However, reducing the number of GMM component to one produce very poor search area resulting in a wrong estimated pose. To reduce this problem, the previous pose's result or a combination of the previous pose and the in-between frames IMU values could be used to predict a better search area. For example, a simple linear prediction based on two successive frames' pose could be used. Non-linear pose prediction can be done using techniques such as Extended Kalman Filter [Bleser and Stricker 2009][He et al. 2015] or state of the art Neural Networks [Rambach et al. 2016]. However, Neural Networks based approach has high CPU usage which prevents their use for low latency VR applications. For our application (≥ 100 [fps]), we decided to use a fast and simple linear-based approximation for pose prediction. For different environment setting, this prediction step can be easily replaced using Extended Kalman Filter based techniques. Let $\boldsymbol{p}_{f-1} = \{\boldsymbol{r}_{f-1}^T, \boldsymbol{t}_{f-1}^T\}$ be the previously estimated pose of a frame $f-1$, where $\boldsymbol{r}_{f-1}$ and $\boldsymbol{t}_{f-1}$ represent the axis-angle rotation and translation, respectively. The pose $\boldsymbol{p}'_f$ of a frame f is predicted with the following equation.

$$\boldsymbol{p'}_f = \boldsymbol{p}_{f-1} + \Delta T \cdot (\boldsymbol{p}_{f-1} - \boldsymbol{p}_{f-2}) \tag{1}$$

where ΔT is the timestep. If the camera frame update is constant, ΔT is set to 1.0.

The single GMM consists of a mean, which is defined by six-dimensional rotation and translation values along with 6×6 covariance. The mean is set to $\boldsymbol{p}'_f$ which is computed as in (1), where the covariance $\Sigma_f^{\boldsymbol{p}'}$ is set as follows.

$$\Sigma_f^{\boldsymbol{p}'} = \begin{pmatrix} \Sigma_f^{rotate} & 0 \\ 0 & \Sigma_f^{translate} \end{pmatrix} \tag{2}$$

$$\Sigma_f^{rotate} = \alpha \boldsymbol{I} + (\boldsymbol{r}'_f - \boldsymbol{r}_{f-1})^T (\boldsymbol{r}'_f - \boldsymbol{r}_{f-1}) \tag{3}$$

$$\Sigma_f^{translate} = \alpha \boldsymbol{I} + (\boldsymbol{t}'_f - \boldsymbol{t}_{f-1})^T (\boldsymbol{t}'_f - \boldsymbol{t}_{f-1}) \tag{4}$$

where Σ_f^{rotate} and $\Sigma_f^{translate}$ represent covariance matrices of rotation and translation, respectively. $\boldsymbol{I}$ is an identity matrix and α is a constant value (in this paper we set $\alpha = 3.0e-02$). The first term in equations (3)(4), is a constant covariance on translational and rotational components that ensures a minimum search area in case of static and/or extremely slow motions. The covariance matrix is projected onto a 2D image plane to define search areas that reflects the movements of the HMD. For instance, if the predicted translational motion is predominant in the x-axis, the search area should be larger than the one on the y-axis. This is captured by the covariance matrix where the x-axis translational covariance would be higher that the covariance on the other axis. Once projected, the corresponding search area would end up elongated on the x-axis. An example is shown in Fig. 3. By doing so, we can avoid an unnecessary area search, which will cause local optima convergence and/or increase the computation time.

3.2 Blob Center Detection and Geometrically Constrained Optimization

After obtaining the predicted mean and covariance, we extract the 2D-image feature points from an observed camera frame. These 2D-image points can be treated as candidate locations for the 3D-object feature points. The 2D-image feature points $\boldsymbol{u}$ are extracted from the current frame using a blob detector. Since the IR-LEDs will appear as bright blobs in a frame (Fig. 4), we apply a simple intensity-based weighting blob center detector as presented in [Faessler et al. 2014][Tjaden et al. 2015]. The weighting is applied in order to detect the centroid of each blob with sub-pixel accuracy. To segment blobs, we use a binary thresholding proposed by Otsu et al. [Otsu 1979]. We group neighboring pixels into blobs. Each blob centroid $\boldsymbol{u}_i$ is calculated based on first image moments which is defined as

$$\boldsymbol{u}_i = (\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}}) \tag{5}$$

$$M_{pq} = \sum_x \sum_y x^p y^q \boldsymbol{L}(x, y) \tag{6}$$

where $\boldsymbol{L}(x, y)$ represents for a raw grayscale image. Now that we have a set of candidate IR-LED locations (blob centers) in the image, we project the 3D-object points $\boldsymbol{x}$ onto the same 2D image plane using the predicted pose $\boldsymbol{p}'_f$ and pre-calculated intrinsic camera parameters. Next, we look for the best matching between the detected blob centers and projected points. For a given projected 3D-object point $\boldsymbol{v}_j$, the best matching 2D-image feature points candidate $\boldsymbol{u}_i$ is determined using the Mahalanobis distance over the search area.

In some cases, the search areas of projected 3D-object points might fall close to each other or overlap on the 2D image. This can cause false correspondence and increase the algorithm computational cost. To prevent this, we redefine the search area based on the known normal vector of each 3D-object feature point. These normal vectors are obtained from the HMD CAD model. Let $\Sigma_j^{uv} = \boldsymbol{J}(\boldsymbol{x}_j)\Sigma_f^{\boldsymbol{p}'}\boldsymbol{J}^T(\boldsymbol{x}_j)$ be the covariance projected to the 2D plane, while $\boldsymbol{J}(\boldsymbol{x}_j)$ represents the Jacobian projecting the 6D covariance to the 2D space. The best matching 2D-image point $\boldsymbol{u}_i$ to $\boldsymbol{v}_j$ are found when the Mahalanobis distance is lower than the specific threshold M (in this paper we set $M = 2.0$).

$$(\boldsymbol{u}_i - \boldsymbol{v}_j)\Sigma_j^{uv'}(\boldsymbol{u}_i - \boldsymbol{v}_j)^T \leq M^2 \tag{7}$$

When the normal vector $\boldsymbol{n}_j$ is given for $\boldsymbol{x}_j$, the updated projected covariance $\Sigma_j^{uv'}$ is defined by

$$\Sigma_j^{uv'} = w_j^{\beta}\Sigma_j^{uv} \tag{8}$$

In other words, the projected search area is skewed based on the rotation of the corresponding 3D-object feature point. w_j is defined as

$$w_j = \begin{cases} \boldsymbol{n}_{cam} \cdot \exp_{SO(3)}([\boldsymbol{r'}_f]^{\wedge}) \cdot \boldsymbol{n}_j & if\ w_j < \gamma \\ \infty & otherwise \end{cases} \tag{9}$$

where $\boldsymbol{n}_{cam}$ represents constant camera direction, $exp_{SO(3)}$ represents the Rodrigues’s formula which converts an axis angle rotation to a rotation matrix, $[\boldsymbol{r'}_f]^{\wedge}$ represents the skew symmetric matrix of $\boldsymbol{r}'_f$, and β, γ are constant values (in this paper we set $\beta = 5.0$ and $\gamma = -5.0e-01$). Essentially, we compute the dot product between the camera normal vector and each rotated 3D-object’s normal vector. This give us an idea of whether a given 3D-object point is facing the camera. By applying equation (8) we can reduce the search area according to the 3D-feature point predicted rotation. When $w_j \geq \gamma$, the search area covariance is set to ∞ which completely discard the search area for the corresponding point $\boldsymbol{v}_j$. Doing so, we are able to increase the matching robustness and significantly reduce the computational time (see Fig. 5 for an example).

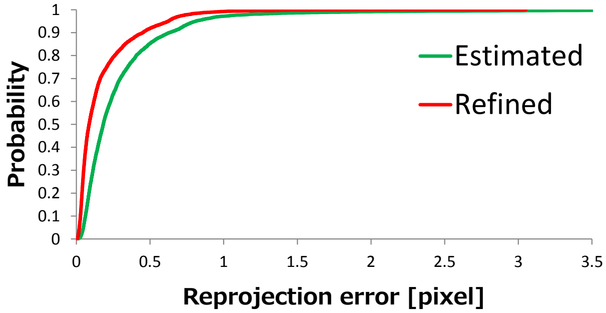


Figure 6: Cumulative distribution function result on the estimated pose and the refined pose. The reprojection error is smaller at the refined result which indicates that refinement step stabilizes the estimated pose.

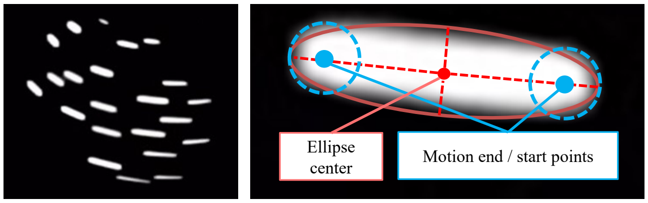


Figure 7: (Left) Observed IR-LEDs at fast motion. (Right) The methodology of estimating start and end point of a motion. These points can be found by calculating inscribed circles with the minor radius of a fitted ellipse lying along the major axis.

3.3 Kalman Filter Update

In the previous section, we described the method used to generate search areas for matching observed 2D-image feature points to projected 3D-object feature points. The matching process is performed iteratively, that is we find correspondence point by point until at three points are reliably matched. Between each iteration, we update our search areas using the newly gained information about the HMD pose. Following the BlindPnP algorithm [Moreno-Noguer et al. 2008], the update is performed using a Kalman Filter as follows

$$\boldsymbol{p}^+ = \boldsymbol{p} + \boldsymbol{K}(\boldsymbol{u}_i - \boldsymbol{v}_j) \tag{10}$$

$$\Sigma^{p+} = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{J}(\boldsymbol{x}_j)\Sigma_j^{uv'} \tag{11}$$

where $\boldsymbol{p}^+$ is the updated pose, Σ^{p+} the updated 6×6 covariance, $\boldsymbol{p}$ the HMD pose (GMM mean) of the current iteration, and $\boldsymbol{K}$ represents the Kalman gain. Updating the search area between each iteration helps us to reduce the potential search area for the next match. In the original BlindPnP algorithm [Moreno-Noguer et al. 2008], the above search and update process is repeated until the third match hypothesis is made. The final 2D-3D correspondence and pose are obtained once the total cost function converges under a threshold. In [Moreno-Noguer et al. 2008], an extra pose refinement step is carried by applying approximated nearest neighbor algorithm to find correspondence with the remaining unmatched points. The full set of correspondence is then used to solve the PnP problem to obtain the refined pose. For more details, please refer to the original paper [Moreno-Noguer et al. 2008]. Our experiments showed that this refinement step stabilizes the estimated pose (Fig. 6). Thus we also include this refinement in our approach.

3.4 Pose Deblurring

When an object moves over a large distance during the exposure time of a camera frame, it causes an effect called motion blur. The object then can be observed as a streak which is larger than its size. If the camera exposure time is not fast enough and the object velocity is high, these streaks can be observed. In VR the HMD and potentially handheld controllers are subject to high velocity motions which generates motion blur frequently. Fig. 7 shows an example of blur effects observed with our camera. Our tracking method is fairly robust to blur as we work with the computed blob centers from the images. However, to obtain more accurate pose estimation, we need to take into account the shape of the blur streak. That is, we need to find the actual location of the IR-LEDs which is not situated on the center of the blurred blob but at the end point of the streak. We call this process deblurring and the obtained pose "deblurred pose". This pose deblurring need to be fast enough to not become a bottleneck of our tracking method. There are several works in computer vision field dealing with blur [Wang and Tao 2014]. Unlike most application, in our case the whole image does not need to be deblurred completely. Only the IR-LED blobs needs to be deconvoluted which once again helps to keep computational costs low.

Rozumnyi et al. [Rozumnyi et al. 2017] proposed a method for tracking fast moving objects by using coordinate descent search algorithm [Wright 2015]. With fixed camera, the movement trajectory was calculated by solving an optimization function which detects the start point, end point, orientation, and trajectory length. Whereas the original paper involves time-consuming coordinate descent search algorithm, we simplify it with linear least square optimization and achieve real-time performance.

To obtain the deblurred pose, the PnP problem should be resolved using points calculated as the end point of the HMD motion. We carry the calculation for the 2D-image points found as 2D-3D correspondences through the process described in Section 3.2 and 3.3. We assume that the IR-LED blob contours can be modeled using an ellipse. By re-using the blob contours found in Section 3.2, we fit an ellipse with least square optimization. We can then determine the major and minor axis $\boldsymbol{e}_{maj}$, $\boldsymbol{e}_{min}$ and radii R_{maj}, R_{min} of the ellipse. We find the start and end point of blurred streak by looking for the center of two inscribed circles with radius R_{min} lying along the major axis (see Fig. 7). Since the ellipse center is $\boldsymbol{u}_i$, both end and start points $\boldsymbol{u}_i^1$ and $\boldsymbol{u}_i^2$ are given by

$$\boldsymbol{u}_i^{1,2} = \pm k \boldsymbol{e}_{maj} + \boldsymbol{u}_i \tag{12}$$

$$k = 1 - \frac{R_{min}}{R_{maj}} \tag{13}$$

The real location of the IR-LED in the current frame is located at the end point of the streak. We use the estimated HMD motion to differentiate between the start and end points of each blob. The end point can be formulated as

$$\boldsymbol{u}'_i = \underset{h=\{1,2\}}{\arg\max} f(\boldsymbol{u}_i^h) = \{\boldsymbol{u}_i^h | (\boldsymbol{v}_j^f - \boldsymbol{v}_j^{f-1}) \cdot (\boldsymbol{u}_i - \boldsymbol{u}_i^h)\} \tag{14}$$

where $\boldsymbol{v}_j^f$ and $\boldsymbol{v}_{j-1}^{f-1}$ are projected 3D-object points of the current and the previous frame, respectively. We do not however deblur blobs whose corresponding 3D-object point moved less than a given threshold between two consecutive frames $\boldsymbol{p}^f$ and $\boldsymbol{p}^{f-1}$. Formally, the blob center $\boldsymbol{u}_i$ will be replaced by its end point $\boldsymbol{u}'_i$ if $\|\boldsymbol{x}_j'^f - \boldsymbol{x}_j'^{f-1}\| \geq \delta$ with $\boldsymbol{x}_j'^f$ and $\boldsymbol{x}_j'^{f-1}$ the 3D-object points and $\delta = 4.5$ [mm]. Using these deblurred blobs we can obtain a deblurred pose by solving a PnP problem. The resulting HMD pose could be further improved by fusing IMU information using techniques such as Extended Kalman Filter [Bleser and Stricker 2009][He et al. 2015] and/or Neural Networks[Rambach et al. 2016].

4 RESULT AND DISCUSSION

4.1 Environment Setting and Tracking Result

Here we describe our environment setting. For the very first frame we estimated the HMD pose using an object detector [Lienhart and Maydt 2002] to find the rough location of the HMD, then a cascade of regressors [Kazemi and Sullivan 2014] is used to locate the IR-LEDs rough centers, finally the BlindPnP [Moreno-Noguer et al. 2008] is carried with 20 GMM components to find the correct initial pose. Our monocular IR camera resolution is 640 × 480 [pixels] with 100 [fps] and 90 [deg] of the field of view. Our HMD has 34 IR-LEDs distributed on the front and sides.

We first compare the runtime speed of our algorithm and the original BlindPnP [Moreno-Noguer et al. 2008] algorithm. We reduced the number of GMM component used for BlindPnP from 20 to 1 because the required computation time increases proportionally to the number of GMM components. We used linearly predicted mean value and constant covariance for the GMM model used in the original BlindPnP.

The mean computation was compared using a single thread on the Intel Core i7-7700 3.6GHz CPU. Table 1 shows the mean runtime for four different sequences. We can see that the geometric constraints introduced in our algorithm significantly reduce the computational costs as we observe an average runtime of 7.03 [ms] against the 35.0 [ms] achieved by the BlindPnP. Especially, for "wide rotation and fast motion" our method is an order of magnitude faster.

Fig. 1(a) shows examples of tracking results. It can be seen that the motion based update and geometric constraints update of the search areas is working properly; the search areas are elongated along the dominant motion direction. Self-occluded IR-LEDs are discarded form the matching process altogether. Fig. 1(b) shows the effectiveness of our pose deblurring method. For more results, please refer to a supplementary video.

4.2 Accuracy Evaluation

In this section, we perform a quantitative evaluation of our tracking algorithm accuracy. We evaluate our system against the HTC Vive Lighthouse VR system known for its submillimeter tracking accuracy [lig 2017]. We match the HTC Vive system coordinate and our system coordinate using Hand-eye calibration [Dornaika and Horaud 1998].

We tested our algorithm against tracking sequences similar to those of Faessel et al. [Faessler et al. 2014] paper. Our sequences have a duration of 135 [sec] and contain regular translations (0.5 − 1.5 [m]) and wide range rotations (0 − 150 [deg]). The result can be seen in Fig. 8. Quantitative evaluations of the translational and rotational errors are reported in Table 2. Our experiments indicate that the translational mean error is under 1.0 [cm] while the rotational mean error is 1.01 [deg]. Faessel et al. [Faessler et al. 2014] propose a system similar to ours where a monocular IR camera (Resolution: 752 × 480 [pixel], Field of view: 90 [deg]) is used to track a quadrotor mounted with 5 IR-LEDs. They report a mean translational error of 7.4 [mm] and a rotational mean error of 0.79 [deg]. We can see here that we achieve comparable results. Our rotation performance of 1.01 [deg] is 28% higher than Faessel et al. [Faessler et al. 2014]. This can be explained by the fact that

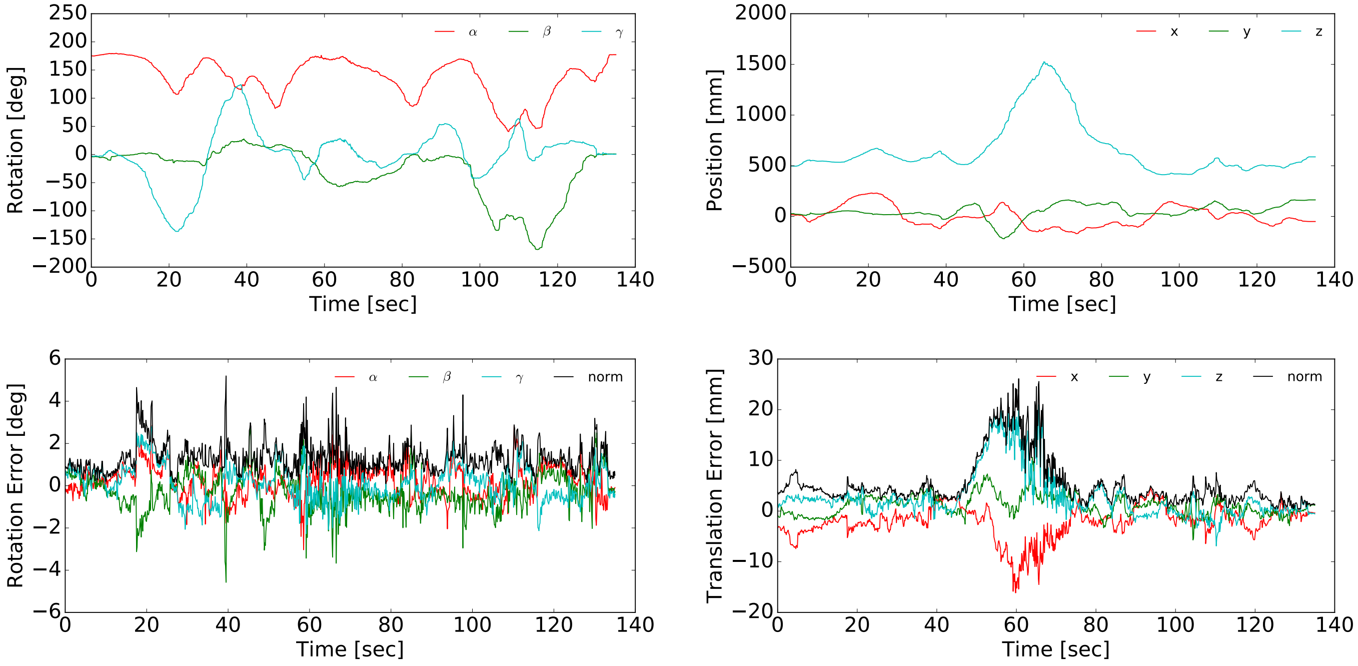


Figure 8: The estimated rotation and translation result by the proposed method, as well as its corresponding errors. Ground truth is not visualized because there is few difference at this scale. The rotation is parameterized with Euler angles, yaw (α), pitch (β), and roll (γ). During the sequence, no tracking loss was observed.

the average distance between visible LEDs on their target object is larger than ours. In addition, the resolution of their camera is 17.5% larger. Our system could be improved by revising the IR-LED layout to increase rotational accuracy. A higher camera resolution would be particularly helpful to compensate errors in a far distance.

4.3 Stability Evaluation

Stability of the HMD tracking is critical to achieve a comfortable VR experience. Instability in the estimated pose and recurrent tracking losses are a source of major discomfort and immersion breaking. We evaluated the stability of our algorithm relative to tracking loss and pose precision. We compute the ”Precision” and ”Recall” of our pose estimation by estimating the number of true positives, false positives and false negatives over multiple frames. To evaluate the stability of our tracking algorithm with relation to tracking losses, we used the ”Recall” which measures the ratio between the number of frames where the HMD was tracked over the total number of frames. On the other hand, the ”Precision” of the pose estimation is computed as the ratio of frames where the pose was accurately estimated over the total number of frames. Each frame in the test sequences was hand-annotated. We evaluate the validity of the estimated 3D pose by projecting 3D-object points onto the 2D image plane. Projected points are then compared to their ground truth location in the image. We require all visible feature points to fall within 6.5% (in our case 1.05 [cm]) of the HMD to consider the estimated pose as a true positive. The original BlindPnP tends to lose the tracking (fail to converge) for wide rotations. In such cases, we count the lost frame and the next hundred as false negatives. The tracking is then reinitialized with the ground truth. In our experiments, we employed two different types of HMD movement, ”narrow rotation with fast motion”, and ”wide rotation with fast motion”. Here narrow rotation indicates a yaw and pitch rotation of under 90 [deg], while wide rotation indicates a yaw and pitch rotation of over 90 [deg]. The average motion speed for both sequences was approximately 1.0 [m] per second. Such speeds were achieved via hand movements and can be considered as highly unlikely when it comes to actual head movements. Those could be seen in unusual cases such as head banging, or fast projectile dodging. For a fair comparison, we reduce the BlindPnP number of GMM components from 20 to 1 to retain real-time performances.

Table 3 indicates that our method provides higher stability than the previous method. Recalls for both sequences show that the HMD was never lost. Similarly, the Precisions for both sequences are very high at 99.7%. The ”narrow rotation and fast motion” is a fairly easy sequence to track. Both the BlindPnP and our method achieve very high accuracy. Nonetheless, we improve over the BlindPnP results, which might indicate that our search area updates based on movements direction are improving the quality of the estimated pose. Our method’s advantages are even more visible for the ”wide rotation and fast motion”. Indeed, we see an improvement of nearly 70% in Recall and more than 41% in Precision over the BlindPnP algorithm. An improvement over this sequence illustrates the effectiveness of our geometrically constrained matching.

Table 1: Mean runtime of four different sequences [ms].

Sequence category	**Ours**	BlindPnP (1GMM)
Narrow rotation and slow motion	**7.30**	13.5
Wide rotation and slow motion	**6.88**	47.2
Narrow rotation and fast motion	**7.11**	14.2
Wide rotation and fast motion	**6.82**	65.0
Average	**7.03**	35.0

Table 2: Translational and rotational mean error, standard deviation, and maximum error of proposed method.

	Mean	Standard deviation	Max error
Translation [mm]	5.29	4.79	26.1
Rotation [deg]	1.01	0.608	4.72

Table 3: Frame by frame Precision, and Recall for proposed method and the original BlindPnP [Moreno-Noguer et al. 2008] algorithm using a single GMM component (in percentage).

Sequence category		**Ours**	BlingPnP (1GMM)
Narrow rotation and fast motion	Precision	**99.7**	98.6
	Recall	**100**	93.3
Wide rotation and fast motion	Precision	**99.7**	58.6
	Recall	**100**	30.4

4.4 Limitations

Currently, our approach's initialization step does not satisfy real time processing requirements. A potential solution to this problem can be found in a recent machine learning approach proposed in [Brachmann et al. 2016]. Another potential approach would be to use SoftPosit with the BnB method [Brown et al. 2015] in combination with IMU information for real time initialization. In addition, there is also a problem with pose ambiguities when outliers such as sunlight are observed near to the HMD (see Fig. 9). Due to these outliers, the pose could be lost or the pose estimation results could be jittery. For such severe cases, the IMU data could be applied to bootstrap the optimization (e.g., combining the gravity information or known rotation of the IMU [Kukelova et al. 2010][Li et al. 2013][Sweeney et al. 2015]). These kinds of techniques could also be used in the PnP problem for pose refining and pose deblurring (see also Fig. 9). Finally, we would like to test our algorithm results against a very high-speed motion capture system for fine evaluation at very high-speed motion.

5 CONCLUSION

In this paper, we proposed a novel approach for HMD pose estimation in VR context. Our system uses a monocular IR camera, and IR-LEDs mounted onto the HMD. No camera synchronization and/or LED blinking is required. To achieve high accuracy and real-time estimation, we based our method on the BlindPnP [Moreno-Noguer et al. 2008] algorithm and extended it with dynamic search area updates and normal-based geometric constraints for 2D-3D correspondence. In addition, we proposed an efficient method to increase our method's robustness to strong motion blur artifacts observed in captures images. Our quantitative results indicate that our method achieves high accuracy tracking with high stability. In addition, we show that our algorithm runtime is an order of magnitude faster compared to the widely used BlindPnP method. For future works, we intend to focus on disambiguating the estimated pose by introducing IMU information within our approach. We also intend to investigate the most effective IR-LED layout to make a guideline for accurate pose estimation in VR scene.

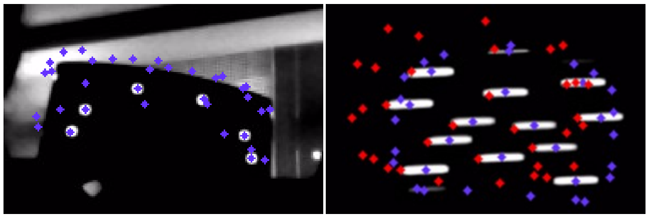


Figure 9: Limitation (Blue dots: the estimated pose, and Red dots: the deblurred pose). (Left) Pose ambiguity against the severe sunlight observed around IR-LEDs, and (Right) pose ambiguity against the pose deblurring.

ACKNOWLEDGEMENT

Pavel A. Savkin is the inventor and applicant of a patent application relevant to this work (JP-2016-253835). This work was supported by JST ACCEL Grant Number JPMJAC1602, Japan.

REFERENCES

Online; accessed 25-June-2017. Lighthouse tracking examined. http://doc-ok.org/?p=1478.

Gabriele Bleser and Didier Stricker. 2009. Advanced tracking through efficient image processing and visual–inertial sensor fusion. *Computers & Graphics* 33, 1 (2009), 59–72.

Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, et al. 2016. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 3364–3372.

Mark Brown, David Windridge, and Jean-Yves Guillemaut. 2015. Globally optimal 2D-3D registration from points or lines without correspondences. In *Proceedings of the IEEE International Conference on Computer Vision.* 2111–2119.

Andrea Censi, Jonas Strubel, Christian Brandli, Tobi Delbruck, and Davide Scaramuzza. 2013. Low-latency localization by Active LED Markers tracking using a Dynamic Vision Sensor. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 891–898.

Philip David, Daniel Dementhon, Ramani Duraiswami, and Hanan Samet. 2004. SoftPOSIT: Simultaneous pose and correspondence determination. *International Journal of Computer Vision* 59, 3 (2004), 259–284.

Fadi Dornaika and Radu Horaud. 1998. Simultaneous robot-world and hand-eye calibration. *IEEE transactions on Robotics and Automation* 14, 4 (1998), 617–622.

Matthias Faessler, Elias Mueggler, Karl Schwabe, and Davide Scaramuzza. 2014. A monocular pose estimation system based on infrared leds. In *IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 907–913.

Martin A Fischler and Robert C Bolles. 1981. Random Sample Consensus: a Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM* 24, 6 (1981), 381–395.

Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. 2015. Visual simultaneous localization and mapping: a survey. *Artificial Intelligence Review* (2015), 1–27.

Changyu He, Peter Kazanzides, Hasan Tutkun Sen, Sungmin Kim, and Yue Liu. 2015. An inertial and optical sensor fusion approach for six degree-of-freedom pose estimation. *Sensors* 15, 7 (2015), 16448–16465.

Vahid Kazemi and Josephine Sullivan. 2014. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1867–1874.

Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. 2010. Closed-form solutions to minimal absolute pose problems with known vertical direction. *Proceedings of the 10th Asian Conference on Computer Vision* (2010), 216–229.

Steven M. LaValle. 2017. *Virtual reality*. Cambridge University Press.

Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. 2009. Epnp: An accurate o(n) solution to the pnp problem. *International Journal of Computer Vision* 81, 2 (2009), 155–166.

Bo Li, Lionel Heng, Gim Hee Lee, and Marc Pollefeys. 2013. A 4-point algorithm for relative pose estimation of a calibrated camera with a known relative rotation angle. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 1595–1601.

Rainer Lienhart and Jochen Maydt. 2002. An extended set of haar-like features for rapid object detection. In *Proceedings of the International Conference on Image Processing*, Vol. 1. IEEE, I–I.

Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. 2016. Pose estimation for augmented reality: a hands-on survey. *IEEE Transactions on Visualization and Computer Graphics* 22, 12 (2016), 2633–2651.

Francesc Moreno-Noguer, Vincent Lepetit, and Pascal Fua. 2008. Pose priors for simultaneously solving alignment and correspondence. *Proceedings of the 10th European Conference on Computer Vision* (2008), 405–418.

Nobuyuki Otsu. 1979. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics* 9, 1 (1979), 62–66.

Jason R Rambach, Aditya Tewari, Alain Pagani, and Didier Stricker. 2016. Learning to fuse: A deep learning approach to visual-inertial camera pose estimation. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 71–76.

Denys Rozumnyi, Jan Kotera, Filip Sroubek, Lukas Novotn, and Jiri Matas1. 2017. The World of Fast Moving Objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chris Sweeney, John Flynn, Benjamin Nuernberger, Matthew Turk, and Tobias Höllerer. 2015. Efficient computation of absolute pose for gravity-aware augmented reality. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 19–24.

Henning Tjaden, Ulrich Schwanecke, Frédéric Stein, and Elmar Schömer. 2015. High-Speed and Robust Monocular Tracking.. In *Proceedings of the 12th International Conference on Computer Vision Theory and Applications*. 462–471.

Ruxin Wang and Dacheng Tao. 2014. Recent progress in image deblurring. *arXiv preprint arXiv:1409.6838* (2014).

Stephen J Wright. 2015. Coordinate descent algorithms. *arXiv preprint arXiv:1502.04759* (2015).