LiTag: Localization and Posture Estimation with Passive Visible Light Tags

Pengjin Xie
xiepengjin@tsinghua.edu.cn
Tsinghua University

Lingkun Li
lilingk1@msu.edu
Michigan State University

Jiliang Wang
jiliangwang@tsinghua.edu.cn
Tsinghua University

Yunhao Liu
yunhao@tsinghua.edu.cn
Tsinghua University

ABSTRACT

The development of Internet of Things calls for ubiquitous and low-cost localization and posture estimation. We present LiTag, a visible light based localization and posture estimation solution with COTS cameras. The core of LiTag is based on the design of a chip-less and battery-less optical tag which can show different color patterns from different observation directions. After capturing a photo containing the tag, LiTag can calculate the tag position and posture by combining the color pattern and the geometry relation between the camera image plane and the real world. Unlike existing marker-based visible localization and posture estimation approaches, LiTag can work with a single camera without calibration, which significantly reduces the calibration overhead and deployment costs. We implement LiTag and evaluate its performance extensively. Results show that LiTag can provide the tag position with a median error of 1.6 *cm* in the 2D plane, a median error of 12 *cm* in the 3D space, and posture estimation with a median error of 1°. We believe that LiTag has a high potential to provide a low-cost and easy-to-use solution for ubiquitous localization and posture estimation with existing widely deployed cameras.

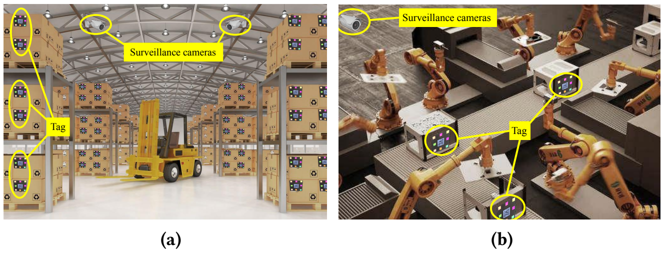
CCS CONCEPTS

• **Networks** → **Location based services**; • **Information systems** → **Location based services**; • **Computer systems organization** → Special purpose systems; • **Human-centered computing** → *Ubiquitous and mobile computing systems and tools.*

KEYWORDS

Visible Light Positioning, Indoor Localization, Visible Light Tag

ACM Reference Format:
Pengjin Xie, Lingkun Li, Jiliang Wang, and Yunhao Liu. 2020. LiTag: Localization and Posture Estimation with Passive Visible Light Tags. In *The 18th ACM Conference on Embedded Networked Sensor Systems (SenSys '20), November 16–19, 2020, Virtual Event, Japan.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3384419.3430777



(a)

(b)

Figure 1: The application scenarios that LiTag can be used: (a) a warehouse, and (b) a factory conveyor belt.

1 INTRODUCTION

Visible Light Positioning (VLP) has been shown as a promising technique for indoor localization due to its low cost and high precision. Visible light signals can be easily generated, e.g., by LED, and processed by camera or photodiode available on many mobile and embedded systems. Many VLP techniques leverage specially designed LED sources [1–4] or intrinsic light properties [5–8] to enable localization on devices with camera or photodiode sensor. Thus it usually requires the target to be equipped with cameras or special sensors for localization.

Marker-based visible light positioning techniques [9–12] localize the target by attaching markers (e.g. a QR-code or a special designed planar marker) on it. Those approaches usually leverage the geometry relationship between the marker and the image plane of the camera to derive the location. Based on precisely calibrated camera parameters, e.g., focal length, pixel size, etc., the relation between the camera and the real world coordinate systems can be used to calculate the markers' position. Those approaches typically require precise camera parameters as input for localization. However, camera parameters are not easy to obtain in practice, e.g., the parameter may change due to changes of focal length while the camera zooming in, zooming out, or autofocusing.

In this paper, we propose LiTag, a passive visible light positioning and posture estimation system. The core of LiTag is the design of a plastic optical tag which is totally *chip-less* and *battery-less*. The tag contains a retroreflection layer on the bottom which reflects lights back to the light source, just like RFID tags reflect radio-frequency signals. The reflected lights then pass through a birefringent chip caught in the middle of two polarizer chips. This results in a specific interference pattern for light passing through the tag, which shows different colors from different observation directions.

SenSys '20, November 16–19, 2020, Virtual Event, Japan
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7590-0/20/11…$15.00
https://doi.org/10.1145/3384419.3430777

After capturing a photo containing the tag, a camera in LiTag can derive the relative position by combining the color pattern and tag geometry. We design an algorithm to calculate the tag position and posture with high precision by the geometry relation between the camera image plane and the real world coordination. Unlike existing marker-based visible localization and posture estimation approaches, we show that LiTag can work with a single camera without precise camera parameters, which significantly reduces the calibration overhead and deployment costs.

Considering the widely deployment of surveillance cameras, ubiquitous localization and posture estimation can be made possible using LiTag at a very low cost. Based on its low cost features, we believe LiTag can be promising to be a novel and supplementary technique to RFID and visual-based markers in many application scenarios in the future. For example, as shown in Figure 1, in a warehouse and a factory, LiTag can support localization and posture estimation of objects with the readily deployed surveillance cameras. The location and posture information can be used to support applications in these environments like auto-robot warehouse objects management, and factory conveyor belt product operation, *etc.*

The contributions of this paper are as follows.

- We present LiTag system which supports passive localization and posture estimation. We believe it could be a useful and low-cost approach for ubiquitous localization and posture estimation of everyday objects with widely available uncalibrated cameras.
- We prototype the LiTag system with the following components (1) a special chip-less and battery-less tag design, and (2) a novel localization and posture estimation algorithm which combines the interference pattern of the tag and the geometry relationship between the camera plane and the real world.
- We extensively evaluate the performance of LiTag in different scenarios. Results show that LiTag achieves posture estimation with a median error of 1°, 2D localization with a median error of 1.6 *cm* and 3D localization with a median error of 12 *cm* with COTS cameras like mobile phone cameras and surveillance cameras.

2 LITAG BASICS

LiTag system achieves localization and posture estimation by capturing a photo of optical tag on the target. LiTag leverages information from the following two aspects: (1) the geometry relationship between the target and its image in the photo, and (2) the spatial related light pattern from the optical tag, which is introduced by the birefringent interference phenomenon.

2.1 Geometry in camera imaging

Marker-based localization methods [9–12] and geometry-based localization methods [1, 2, 6] are based on the imaging principle of camera, which builds a geometry relationship between the object and the image. Geometry calculation such as bundle adjustment [13] can be used to derive the location of the object. In this section, we introduce the typical procedure of bundle adjustment to illustrate the geometry relationship in camera imaging system, and show the limitation of traditional geometry-based and marker-based methods.

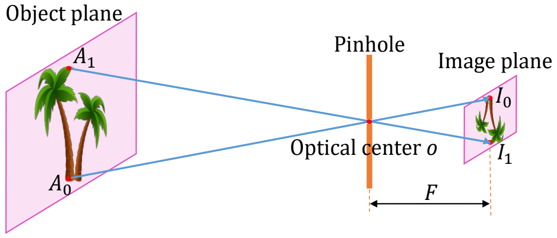


Figure 2: Pinhole camera model imaging system.

As shown in Figure 2, suppose we need to calculate the position of the tree. In the pinhole camera model, light beams from the tree pass through the pinhole and arrive at the image plane in the camera. Light beams on the image plane are further translated into an image by camera sensors such as CMOS sensor. For example, a light ray from A_1 on the object goes to I_1 and a light ray from A_0 goes to I_0 on the image plane.

Assume a 3D coordinate system with the optical center of the pinhole o as the origin. We name it as the coordinate system of the camera. The z-axis is the direction perpendicular to the image plane and pointing the forward direction of the camera. The distance between the image plane and the optical center represents the focal length F in pinhole camera model. The point on the object $A_i = (x_i, y_i, z_i)$ is mapped to point $I_i = (a_i, b_i, F)$ on the image plane. As A_i, o and I_i are on the same line, we have $\overrightarrow{oA_i} = k_i\overrightarrow{oI_i}$, where $k_i \in K$ is the scaling factor depending on the position of the object and the image plane. More specifically, we have $x_i = k_i a_i$, $y_i = k_i b_i$ and $z_i = k_i F$.

Assume the relative position of point A_i on the object is denoted as $B_i = (u_i, v_i, w_i)$ in the coordinate system of the object itself. The goal of the bundle adjustment method is to estimate the position of corresponding object points A_i in the camera coordinate system based on the coordinates of points on the image plane. The estimated position of A_i can also be denoted as $A_i = (k_i a_i, k_i b_i, k_i F)$. Suppose there are n points A_i ($1 \leq i \leq n$), the key idea is to estimate the position of those n points by estimating $K =< k_1, k_2, \cdots, k_n >$.

Denote d_{ij} as the distance between two points A_i and A_j, we have

$$\begin{aligned} d_{i,j} &= \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \\ &= \sqrt{(k_i a_i - k_j a_j)^2 + (k_i b_i - k_j b_j)^2 + (k_i F - k_j F)^2} \end{aligned} \tag{1}$$

Denote the distance between B_i and B_j as D_{ij}, we have

$$D_{i,j} = \sqrt{(u_i - u_j)^2 + (v_i - v_j)^2 + (w_i - w_j)^2} \tag{2}$$

It should be noted that D_{ij} can be measured in advance, e.g., based on the known geometry of the object such as a QR-code or a special designed marker. In bundle adjustment [13], we can obtain the estimated K by

$$K = \arg\min_{K} \sum_i \sum_j (D_{ij} - d_{ij})^2 \tag{3}$$

In practice, this is an optimization problem and there are different methods to solve Eq. (3), e.g., gradient descent [14]. Here we only show the basic principle of bundle adjustment and omit the details for different methods to solve the equation. Interested readers can refer to [14] for more details. It should be noted that the unit of D_{ij} and d_{ij} should be unified. That is to say, a_i, b_i and F should be measured in millimeter (or other units of Length) rather than in pixel. When the pixel size and focal length are unknown, some methods [12, 15–17] can estimate the focal length in pixels and achieve pose estimation, but cannot localize the target due to lacking of conversion scale between pixel and millimeter.

Geometry-based localization methods need precisely calibrated camera parameters to convert the geometry relationship in object coordinate into camera coordinate. It is difficult for an uncalibrated camera to obtain the target location using only the constraints in camera imaging. However, camera calibration in the application environment takes extra overhead. Even it can be conducted, the focal length of the camera may change in practice, e.g., adjusting focal length to obtain clear images of objects with different distance. This results in change of camera parameters. The camera needs to be re-calibrated each time after the change of focal length.

2.2 Birefringent interference

The geometry constraints in camera imaging are not enough for object localization and posture estimation using an uncalibrated camera. The tag used in this work must provide more spatial information. Birefringent interference is an ideal physical phenomenon. Before illustrating birefringent interference, we need to introduce some related physics concepts.

Polarization. Polarization is a special feature describing the oscillation direction of light. Natural light, e.g, sun light, and light emitted from a lamp, has oscillation in any direction. Polarizer allows a light beam passing through it when the oscillation direction of the light is parallel to its *transmission axis*, and blocks the light with perpendicular oscillation direction. A polarizer with a single transmission axis is called *linear polarizer*.

Birefringence. Birefringence [18] is a feature of the optically anisotropic material such as plastics, calcite, and quartz. There is a special direction in birefringent material,which is called *optic axis*. When a light beam passes through a birefringent material, two rays of the refracted light could be observed. As shown in the right part of Figure 3, a beam of polarized light will be split into two rays in birefringent material, one ray with polarization direction perpendicular to the *optic axis* is called *ordinary ray* and the other with polarization direction parallel to the *optic axis* is called *extraordinary ray*. For a certain type of birefringent material, the *ordinary refractive index* is constant, while the *extraordinary refractive index* varies depending on the incident direction.

Interference. When two light beams have the same frequency, stable phase difference and same polarization direction, they can interfere with each other. For a beam of light composed of light with different wavelength, when the optical path difference Δ is fixed, the phase difference can be calculated as $\delta = 2\pi\Delta/\lambda$, which is related to the wavelength λ. This leads to different intensity for light of different wavelength in the interfered light. Given a path differece, the interference will lead to a certain light spectrum with different intensity at different wavelength. Further, different light spectrums lead to different observed colors.

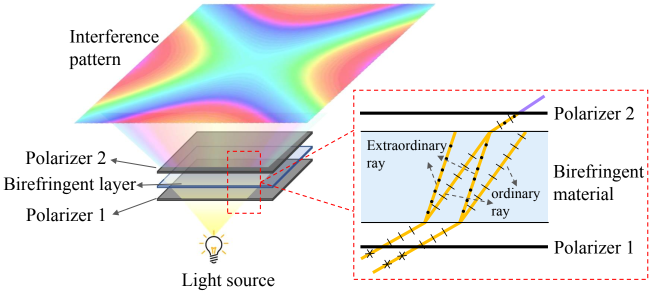


Figure 3: Birefringent interference phenomenon.

In our tag, there are a layer of birefringent material and two layers of polarizer as shown in Figure 3. When a beam of ambient light with all the polarization directions incident into birefringent interference system, the light first passes polarizer 1 and becomes a polarized light. The polarized light split into *ordinary ray* and *extraordinary ray* in the birefringent material. The *ordinary ray* polarized perpendicular with the *extraordinary ray*, and they have a fixed optical path difference when leaving the birefringent material. The second polarizer filters out the components polarized along its *transmission axis* from *ordinary ray* and *extraordinary ray*, the two rays now have the same polarization direction. They interfere with each other and change the spectrum, and show a specific color when getting out the birefringent interference system. Light with different incident directions shows different interference color due to different optical path difference. As shown in Figure 3, we can simulate the color of the interference result according to [8]. Due to interference, we have different colors from different observation directions to the tag. The color of the tag can be used to indicate the possible incident directions.

3 TAG DESIGN

In this section, we present the design of the chip-less and battery-less tag, which can be used to achieve localization and posture estimation of objects using uncalibrated camera.

3.1 Tag structure

From Section 2.2, we know that the color of birefringent interference color can provide spatial information and indicate the possible directions. However, in birefringent interference system, the light source must be on the opposite side with the light screen, i.e. the observation points. If we directly use birefringent interference phenomenon to build our optical tag, there must be a light source on the tag side to provide incident light. In order to make our optical tag totally *battery-less*, we use a layer of retroreflector in the structure of the tag. Retroreflector has a special structure and the light beam can be reflected to the source with a minimum of scattering [19], which is widely used on traffic signs.

As shown in Figure 4, a tag consists of two polarizers, P_1 and P_2, and one film of birefringent material S. The combination of polarizers and birefringent material is the typical structure of birefringent interference system, which is also used in [8]. A layer of

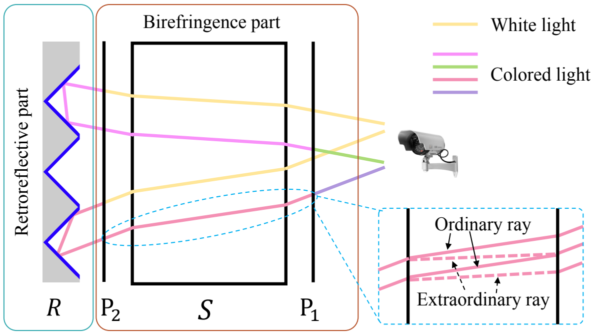


Figure 4: The design of the tag in LiTag system.

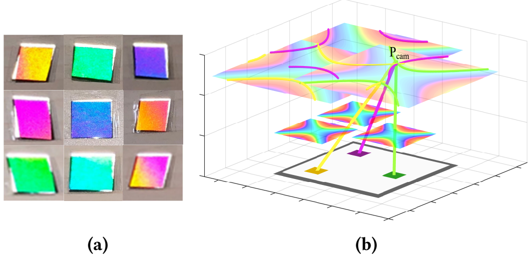


Figure 5: (a) The tag shows different colors in different observation directions. (b) Colors of anchor points can determine the location of observer.

retroreflector R is placed after P_2, which can reflect the light from birefringent interference system back to its incident direction.

We illustrate the light paths in Figure 4. When a light beam from the camera passes P_1, it becomes a polarized light beam. As shown in the right region in Figure 4, the light would split into ordinary ray and extraordinary ray in the birefringent material S. After passing the polarizer P_2, those two rays interfere with each other and exhibit a specific color. Then, the light will be reflected by the retroreflector R and pass through the birefringence part again. Then, the spectrum of the reflected light change again when it comes out from P_1 and returns to the camera, leading to a specific color observed by the camera.

If we change the incident direction of the light, the direction of reflected light accordingly changes. As a result, different incident directions lead to different optical path between the ordinary ray and the extraordinary, which accordingly lead to different spectrums of interfered light. Intuitively, this means the tag shows different colors from different directions. Figure 5a shows photos of a tag captured by a camera from different directions. The results verify that the tag shows different colors in the photos from different directions.

3.2 Property of the tag

From each point of the tag, we can estimate the direction to the point from the observer. In order to obtain location of the observer, we need to choose multiple points on the the tag as anchor points. By combining the direction information provided by all anchor points, we can estimate the location of observer. Figure 5b shows the localization model of the tag. Without loss of generality, we consider three points on a tag. When a camera captures a photo at position P_{cam}, the direction from P_{cam} to each anchor different from each other, leading to a specific color for each anchor point. Those colors for multiple points form a color vector which can determine the position of the camera. Thus, the color on the tag can provide spatial information according to the property of birefringent interference phenomenon.

4 TAG LOCALIZATION AND POSTURE ESTIMATION

In this section, we introduce the algorithm for tag localization and posture estimation using an uncalibrated camera. The algorithm takes a photo containing a tag as the input, and derives the position and posture of the tag in the coordinate system of the camera.

4.1 Overview

As mentioned in section 2.1, it is difficult to achieve accurate localization and posture estimation with geometry information from an uncalibrated camera. We design an algorithm by combining the light properties of the tag with the geometry information of camera imaging. First, by leveraging the spatial information provided by the light properties of the tag, we can obtain relative location of the camera in the coordinate system of the tag. However, we cannot obtain the posture information of the camera. Then, we calculate camera posture based on the geometry information given the camera location. To the best of our knowledge, this is the first method which combines the location and geometry information to derive camera location and posture without camera parameters. Based on the camera location provided by the tag, we propose a posture estimation method. The computational complexity of our posture estimation method is much lower than traditional polynomial solving techniques [11, 15–17]. Then we calculate the posture and the position of the tag by coordinate system transform based on the posture and the position of the camera.

LiTag system mainly works as follows:

(1) Extract color on each anchor point. Calculate the location of camera P_{cam} in the coordinate system of the tag based on colors of multiple anchor points.
(2) Calculate the posture of camera in the coordinate system of the tag using camera location P_{cam} and geometry relationship between the anchor points and their images in the photo.
(3) Calculate the location and posture of the tag in the coordinate system of the camera.

Figure 6 shows two coordinate systems of LiTag system: A tag with its reference coordinate system XYZ and a camera with its reference coordinate system uvw. Denote three axes of reference coordination system of the tag as $\overrightarrow{X} = (1, 0, 0)$, $\overrightarrow{Y} = (0, 1, 0)$, $\overrightarrow{Z} = (0, 0, 1)$. Similarly, the coordinate system uvw of a camera also has three axes: (1) **forward direction** $\overrightarrow{u} = (1, 0, 0)$ represents the camera's pointing direction which is perpendicular to the camera's image plane, (2) **up direction** $\overrightarrow{v} = (0, 1, 0)$ represents camera's up direction, and (3) **side direction** $\overrightarrow{w} = \overrightarrow{u} \times \overrightarrow{v} = (0, 0, 1)$ is

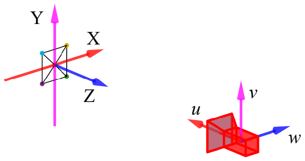


Figure 6: Tag coordinate system XYZ and camera coordinate system uvw.

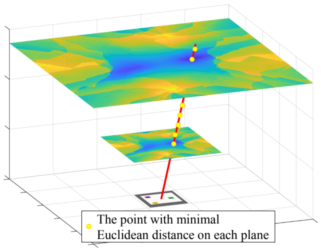
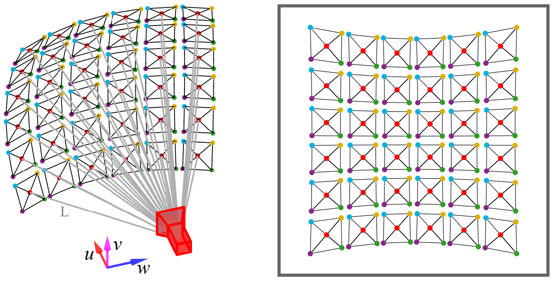


Figure 7: The minimal Euclidean distance point on each plane distributes along a straight line.



(a) (b)

Figure 8: (a) Camera posture dose not effect the observed color vector on the tag. (b) Different camera posture results in different geometry information in the image.

perpendicular with both $\overrightarrow{u}$ and $\overrightarrow{v}$. A notation in lowercase means a location or direction in uvw coordinate system, while an uppercase notation means it is in XYZ system. For simplicity, we assume there are four small anchor points in our figure and those four anchor points are the vertices of a square. The color of the vertex indicates the color captured by the camera.

We denote the position of the camera as P_{cam} in the tag's reference coordinate XYZ. We use the camera's optical center to denote the camera position. From the view of the coordinate system XYZ, the relative posture of the camera can be represented by three axes vectors as $\overrightarrow{U}$, $\overrightarrow{V}$, $\overrightarrow{W}$.

In the camera's reference coordinate system uvw, the tag's position is denoted as p_{tag}. From the view of the coordinate system uvw, the relative posture of tag can be represented by three axes vectors of the tag's coordinate system XYZ as $\overrightarrow{x}$, $\overrightarrow{y}$, $\overrightarrow{z}$.

The task of our LiTag localization and posture estimation algorithm is to determine P_{cam} and $\overrightarrow{U}$, $\overrightarrow{V}$, $\overrightarrow{W}$, and then obtain p_{tag} and $\overrightarrow{x}$, $\overrightarrow{y}$, $\overrightarrow{z}$.

4.2 Localize the camera

We first need to obtain the camera's location P_{cam} in the tag's coordinate system according to colors shown on the tag. From section 3, we know that from different observing directions, the camera captures different colors showing on the tag's anchor points. We therefore build a map from the camera observing direction to the color of each anchor point. Therefore, after the camera capturing a photo, we first extract the color vector from the photo, then obtain the camera location by the color vector and the mapping between observing direction and color.

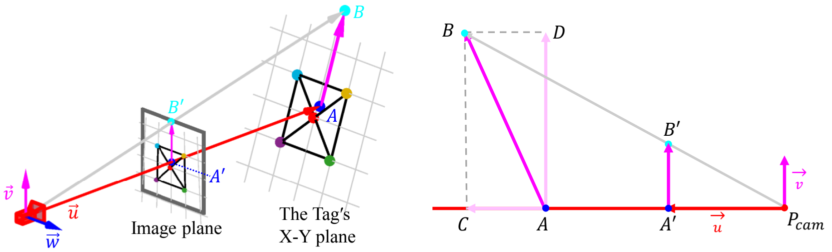
An intuitive solution for camera localization is to traverse the positions in the localization space and calculate a color vector for each point according to the mapping between observing direction and color. Then, we can calculate the Euclidean distance between the captured color vector and the color vector of each point in the localization space. The point which have a minimum Euclidean distance will be considered as the localization result. This method requires traversal of all the positions, and leads to high computational overhead.

In the experiment, we have an important observation which can help us to reduce the computation overhead. If we divide the positions into groups by their Z coordinate value, and highlight the point with minimal Euclidean distance in each group. We observe that these highlighted positions are approximately on a straight line, as shown in Figure 7. This inspires us to determine the line by checking the positions on a specific plane (e.g. the plane with $Z = 1\ m$) at first, then we can only traverse the positions on this line to find the position with minimal Euclidean distance value. The computational overhead is much lower than the brute force search in the entire space.

4.3 Estimate the posture of the camera

Then, we calculate the camera's posture $\overrightarrow{U}$, $\overrightarrow{V}$, $\overrightarrow{W}$ in the XYZ system. As long as the camera optical center is fixed, the observed color on each tag does not change regardless of the change of camera rotate, yaw, and pitch. In other words, the color of tag does not change when the camera change its posture, as long as the tag is in its field of view. This property tells us the following two things. On one hand, the camera's position, which is related to the color vector, can be derived regardless of the camera orientation. On the other hand, the camera's orientation cannot be derived from the color vector.

We show why the camera posture cannot be obtained by tag color in Figure 8. Based on the principle of relative motion, the change of the camera posture is equivalent to the tag's rotation around the camera's origin. In figure 8, we fix the coordinate system uvw, and transfer the camera's orientation change into the camera's point of view. For better understanding, intuitively, we consider that there is a line segment L between the tag's origin and camera's origin. The end of L and the tag compose a rigid body, and the other end of L can rotate at the camera side as shown in Figure 8a. Figure 8b shows the captured photo and corresponding images of the the tag while the camera changes its posture. To derive the posture of the camera, we leverage an important observation as shown in Figure 8b. Different camera postures result in different shapes and pixel position of the tag in the image. It is possible for us to derive the posture of the camera based on the imaging relation between the tag and its image in the photo. More specifically, this camera



(a)

(b)

Figure 9: (a) Camera's reference coordinate, image plane and the object plane. (b) The side view of the projection of vectors in (a).

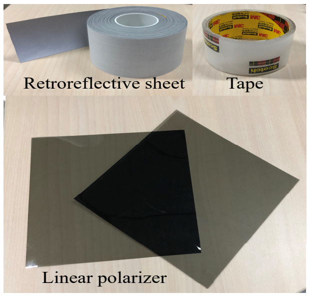


Figure 10: Materials used for the tag: retroreflective cloth, transparent tape and linear polarizers.

posture estimation approach does not require camera parameters such as focal length.

Figure 9a shows the camera's imaging process with a tag in 3D space and its image in the image plane. Without loss of generality, our discussion is based on the virtual projection surface rather than the real projection surface. The image plane is usually a rectangle perpendicular to the forward axis of camera that passes the center of the rectangle. The up axis V is parallel to the vector from the center of the image plane to the center of upper-side of the image rectangle. A' is the center of the photo, and B' is the middle point of the upper-side of the photo. A and B are two points on the tag's XY plane, their corresponding image points are A' and B'. A and A' are all on the U axis, and $\overrightarrow{A'B'}$ is parallel to V axis. Figure 9b shows the side view of the projection of vectors in Figure 9a. $\overrightarrow{A'B'}$ is parallel to $\overrightarrow{AD}$, which is the projection of $\overrightarrow{AB}$ on the image plane, i.e. VW plane. $\overrightarrow{AC}$ is the projection of $\overrightarrow{AB}$ on U axis.

Note that, we have obtained the camera's location P_{cam} in the tag's reference frame XYZ in Section 4.2. Then, if we have the coordinate of point A in the XYZ system, we can obtain the camera's forward direction in the tag's reference frame by

$$\overrightarrow{U} = \frac{\overrightarrow{P_{cam}A}}{|\overrightarrow{P_{cam}A}|} \tag{4}$$

where $\overrightarrow{P_{cam}A}$ is a vector from the camera to point A. If we also have the position of point B in the tag's XYZ system, the camera's up direction can be derived by:

$$\begin{aligned} \overrightarrow{V} &= \frac{\overrightarrow{A'B'}}{|\overrightarrow{A'B'}|} = \frac{\overrightarrow{AD}}{|\overrightarrow{AD}|} \\ &= \frac{\overrightarrow{AB} - \overrightarrow{AC}}{|\overrightarrow{AB} - \overrightarrow{AC}|} \\ &= \frac{\overrightarrow{AB} - (\overrightarrow{AB} \cdot \overrightarrow{U})\overrightarrow{U}}{|\overrightarrow{AB} - (\overrightarrow{AB} \cdot \overrightarrow{U})\overrightarrow{U}|} \end{aligned} \tag{5}$$

Then we can obtain $\overrightarrow{W}$ by $\overrightarrow{W} = \overrightarrow{U} \times \overrightarrow{V}$.

The remaining question is to calculate the coordinates of A and B. We use perspective transformation from the object plane to the image plane. Specifically, we know the position of each anchor point in the tag coordinate system based on the tag geometry design. Meanwhile, we can measure the position of those tags in the image plane based on the image from the camera. Then we can build a transformation that maps the coordinate from the image plane to the tag's XY plane. Since we know the coordinates of A' and B' on the image plane, we can derive the coordinates of A and B in XYZ system.

4.4 Estimate the tag's location and posture

We have calculated the camera's location P_{cam} and posture ($\overrightarrow{U}$, $\overrightarrow{V}$ and $\overrightarrow{W}$) in tag's reference coordinate system XYZ in Section 4.2 and Section 4.3. Then, we transform tag's location $P_t = (0, 0, 0)$ in XYZ, three axes X, Y, and Z into camera's coordinate system uvw. In uvw, the tag's location can be calculated as

$$p_t = (\overrightarrow{P_{cam}P_t} \cdot \overrightarrow{U}, \overrightarrow{P_{cam}P_t} \cdot \overrightarrow{V}, \overrightarrow{P_{cam}P_t} \cdot \overrightarrow{W}) \tag{6}$$

where $\overrightarrow{P_{cam}P_t} \cdot \overrightarrow{U}$ is the length of the projection of $\overrightarrow{P_{cam}P_t}$ on $\overrightarrow{U}$. The axes of XYZ in camera's coordinate system uvw can be calculated as

$$\begin{aligned} \overrightarrow{x} &= (\overrightarrow{X} \cdot \overrightarrow{U}, \overrightarrow{X} \cdot \overrightarrow{V}, \overrightarrow{X} \cdot \overrightarrow{W}) \\ \overrightarrow{y} &= (\overrightarrow{Y} \cdot \overrightarrow{U}, \overrightarrow{Y} \cdot \overrightarrow{V}, \overrightarrow{Y} \cdot \overrightarrow{W}) \\ \overrightarrow{z} &= (\overrightarrow{Z} \cdot \overrightarrow{U}, \overrightarrow{Z} \cdot \overrightarrow{V}, \overrightarrow{Z} \cdot \overrightarrow{W}) \end{aligned} \tag{7}$$

Thus, we can derive the tag's location p_t and posture information $\overrightarrow{x}$, $\overrightarrow{y}$, and $\overrightarrow{z}$ under the camera's coordinate system. For better understanding, we illustrate the whole process of LiTag localization and posture estimation in Algorithm 1. We first obtain camera location P_{cam} in the XYZ system by the observed color vector on the tag. Then we leverages the projective relationship between the tag in the image and the real tag to derive the projection transformation matrix, and obtain coordinates of A and B. Next, we use P_{cam} and coordinates of A and B to calculate the camera's posture in the XYZ system. At last, the location and posture of the tag in the uvw system can be derived by coordinate system transformation using the camera location and posture in the XYZ system.

Algorithm 1: LiTag localization and posture estimation

```
Input:
n: The number of anchors on the tag.
$CO[i]$: Color vector on each anchor, $i \in \{1, \ldots, n\}$.
$(row_i, col_i)$: Pixel position of each anchor, $i \in \{1, \ldots, n\}$.
$(X_i, Y_i, 0)$: The coordinate of each anchor in $XYZ$ system, $i \in \{1, \ldots, n\}$.
$(row_A, col_A)$: Pixel position of A (photo center).
$(row_B, col_B)$: Pixel position of B (the middle point of the photo upper-side).
$P_t$: The coordinate of the tag in the $XYZ$ system.
$\overrightarrow{X}, \overrightarrow{Y}, \overrightarrow{Z}$: Axes of the $XYZ$ system.
Output:
$p_t$: The coordinate of the tag in the $uvw$ system.
$\overrightarrow{x}, \overrightarrow{y}, \overrightarrow{z}$: Axes of $XYZ$ in the $uvw$ system.
$P_{cam}$ = CameraLocalization($\{CO[1], \ldots, CO[n]\}$);
TForm = fitgeotrans($(row_{1,\ldots,n}, col_{1,\ldots,n})$,$((X_{1,\ldots,n}, Y_{1,\ldots,n})$,'projective');
$[X_A,Y_A]$ = transformPointsForward(TForm,$row_A$,$col_A$);
$[X_B,Y_B]$ = transformPointsForward(TForm,$row_B$,$col_B$);
$[\overrightarrow{U}, \overrightarrow{V}, \overrightarrow{W}]$ = CameraPostureEstimation($P_{cam}, A(X_A, Y_A, 0), B(X_B, Y_B, 0)$);
$[p_t]$ = TagLocalization($P_{cam}, P_t, \overrightarrow{U}, \overrightarrow{V}, \overrightarrow{W}$);
$[\overrightarrow{x}, \overrightarrow{y}, \overrightarrow{z}]$ = TagPostureEstimation($\overrightarrow{X}, \overrightarrow{Y}, \overrightarrow{Z}, \overrightarrow{U}, \overrightarrow{V}, \overrightarrow{W}$);
return $p_t$ and $\overrightarrow{x}, \overrightarrow{y}, \overrightarrow{z}$;
```

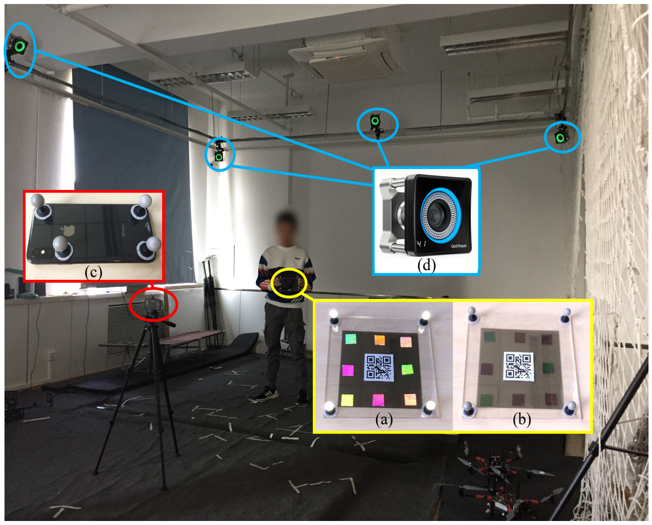


Figure 11: Experiment environment of LiTag system in the commercial tracking system OptiTrack. (a) The tag and motion capture markers with light source. (b) The tag and motion capture markers. Those markers are for OptiTrack to obtain the groud truth of location and posture. (c) A mobile phone with motion capture marker. (d) High precision motion capture cameras.

5 IMPLEMENTATION

5.1 The tag and camera

A tag in our system consists of birefringence part and retroreflection part as shown in Figure 4. The materials we use to make a tag are shown in Figure 10. For the birefringence part, we use transparent tape as the birefringent material, which is cheap and show good birefringence phenomenon. The tape layer is put between two layers of linear polarizer. For the retroreflective part, we use a kind of retroreflective cloth which can reflect light to its source. A tag with multiple anchor points is shown in Figure 11, the photo in (a) is taken with a light source on the camera side, and (b) is taken without light source. The tag in our experiment is small (about 15 × 15 *cm*) and can be made even smaller in practice. The tag is totally chip-less and battery-less, and it can be attached on object surface to provide localization and posture estimation.

LiTag can work with COTS cameras. As the tag is totally passive, it is better to have a light source near the camera. The light source along with the camera is common. For example, typically there is a flashlight along with most of cameras of mobile phone, and surveillance cameras are often equipped with light source for capturing images in dark environment. In localization and posture estimation, we do not require any camera parameters. Users can freely change the focus of the camera and zoom in/out the camera. We use the camera on iPhone X as shown in Figure 11 (c) in the experiments. We assume no pre-known intrinsic parameters in our experiments. The camera obtain color vector on the tag by detecting the tag in an image leveraging OpenCV.

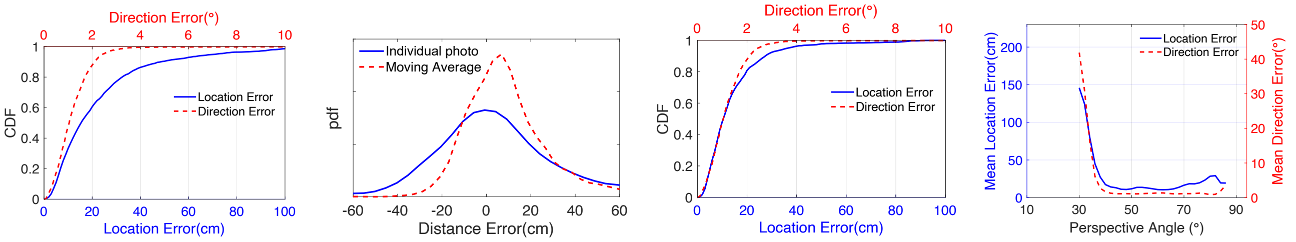
5.2 Ground truth and initial sampling

We leverage the high-precision and expensive commercial localization system, OptiTrack[20], to obtain the location and posture ground truth of the camera and the tag. The OptiTrack system is a motion capture system, which has eight high precision infrared camera with known location and orientation as shown in Figure 11. The cameras we use in this OptiTrack system is the "Prime 41" cameras, which is stated as "The most accurate motion tracking camera in the world" as shown in Figure 11 (d). Special designed reflective balls are used as markers for OptiTrack, which can reflect incident light back to its source and show very bright points in infrared cameras. Multiple reflective balls are attached to the tracking target in the motion track system. In order to obtain the location and posture ground truth of the camera and tag in the OptiTrack system (global coordinate system), we attach reflective balls on the camera and tag as shown in Figure 11 (a) (b) (c).

The initial sampling of the color distribution of the tag is also performed in OptiTrack system. We fix the camera and move the tag in camera's view as shown in Figure 11. We change the tag's location and posture to collect the mapping between color and direction. Then a fine-grained mapping between color and direction can be obtained by interpolation. In our sampling, the distance between the tag and the camera is within the interval [1.7 *m*, 2 *m*]. Note that for all tags of the same design and material, we only need to conduct the initial sampling for one time.

5.3 Identification of the tag

To distinguish different tags, we can use the tag material to encode the tag's id as in [8], which use orthogonal polarizer directions to represent '0' and '1', respectively. We can also learn from FerroTag [21], and arrange the birefringent material to form special colorful patterns on the tag. The patterns can help us to identify different tags. We can also attach different QR-codes on each tag to support identification as shown in Figure 11. Note that identification is different from localization and we do not use those codes in localization.



(a) (b) (c) (d)

Figure 12: (a) Camera localization accuracy using a single photo: CDF of location error and direction error. (b) Camera location error distribution using an individual photo and using moving average. (c) Camera localization accuracy using moving average. (d) Camera localization accuracy with different perspective angle.

6 EVALUATION

We mainly measure the basic performance and validate the feasibility of LiTag system from the following aspects:

- The localization accuracy of the camera in the coordinate system of the tag.
- The posture estimation accuracy of the camera in the coordinate system of the tag.
- The localization and posture estimation accuracy of tag in the coordinate system of the camera.
- The system performance under different scenarios. It involves the camera's perspective angle, distance, pixel position of the tag in the photo, image resolutions, and different types of ambient light.

In all experiments, we obtain the ground truth of localization and posture estimation based on the OptiTrack.

6.1 Camera localization accuracy

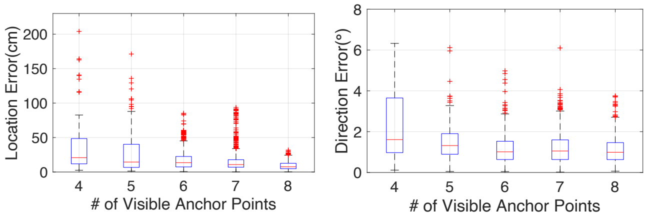
In this section, we examine the accuracy of the first step of LiTag system, i.e., using the observed tag color to localize the camera in the tag coordinate system. In this experiment, we fix the tag and change the position of the camera. We characterize the camera localization accuracy with two metrics, **location error** and **direction error**. Location error is the distance between localization result P_{cam_r} and ground truth location P_{cam_g}. Direction error is the angle between two vectors: $\overrightarrow{OP_{cam\_r}}$ and $\overrightarrow{OP_{cam\_g}}$, where O is the origin of the XYZ coordinate system.

6.1.1 Basic performance. In order to test the basic performance of camera localization, we put the tag at the origin and move the camera, and keep the distance between them within [1.7 m, 2 m], which is the initial sampling distance range. Figure 12a shows the CDF of camera localization error using one photo. We can see that the median location error of the camera is about 15 *cm* and 80% location error is lower than 33 *cm*. The median direction error is about 1° and 90% of the direction error is lower than 2.1°. The result shows that the direction accuracy is very high, but the location accuracy is lower than the direction accuracy.

Considering the camera direction result has high accuracy, it is possible for us to use the moving average of the distance from the camera to the tag to reduce the location error of camera. In this case, we assume the camera's moving is not fast, and it captures the photo continuously. During localization, the system stores a few distances of latest camera location results. After the camera capturing a new photo, the system firstly derives the camera position and its direction, then calculates the average distance with the stored distances. With an accurate estimated direction, we can refine the position by moving the point from the tag toward the estimated direction by the average distance. We start from the distribution of distance errors to clarify it is reasonable. The solid line in Figure 12b shows the distribution of the distance error using an individual photo. We can see the distance error distribution is similar to the Gaussian distribution, and the peak appears when the error is zero. After moving average, it has a high possibility to get a lower distance error than the error from the individual photo, which provides us the basis to refine the location error by moving average. The dashed line in Figure 12b shows the distribution of the distance error after applying the moving average. To verify its effectiveness, we show the camera location accuracy after using moving average in Figure 12c. We can see the location error decreases significantly compared with that in Figure 12a, i.e., the median location error of camera is about 11 *cm* and 80% location error is lower than 20 *cm*. In Section 6.2 and Section 6.3, we use the result after applying moving average as the camera localization result for camera posture estimation and the tag localization.

6.1.2 Perspective angle. We investigate the impacts of the perspective angle ψ between $\overrightarrow{OP_{cam\_g}}$ (P_{cam_g} is the location ground truth of camera) and the tag's surface (XY plane of the tag's coordinate system) on localization performance in this experiment. Figure 12d shows the performance of camera localization under different angle ψ. When $\psi > 35°$, the location accuracy and direction accuracy are both high. LiTag system can provide accurate localization for camera with perspective angle $\psi > 35°$. The performance decreases significantly when ψ is lower than 35°. This is because when ψ is lower than 35°, the brightness of reflection becomes much lower. Therefore, the color captured from anchor points is not accurate enough for localization. In the following section, we will show the specific impact of the brightness of anchor points.

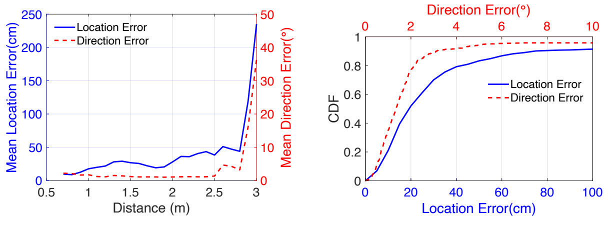
We also see that when ψ increases beyond 75°, the distance error and direction error both increase slightly. This is due to the distribution of birefringence color pattern, i.e., the color changes



(a)

(b)

Figure 13: (a) Camera location accuracy with different number of visible anchor points. (b) Camera direction accuracy with different number of visible anchor points.



(a)

(b)

Figure 14: (a) Camera localization accuracy for different localization distance to the tag. (b) CDF of camera localization error when the camera moves on a known plane.

more slowly around the center of the pattern. We can see evidence in Figure 3 and RainbowLight [8]. The similarity for the color distribution in the center area introduces errors in detecting the real direction from each anchor point.

6.1.3 Number of visible anchor points. According to our observation, not all of the anchor points on a tag can provide good color information all the time. An anchor point maybe not bright enough with some observing directions. The reflection of the polarizer surface increases with the incident angle, leading to an incident light intensity decrease. We say that an anchor point is visible when the color of the anchor point is bright enough. We explore the influence of the visible anchor points number on the camera localization accuracy. The box plot in Figure 13a and Figure 13b shows the location error and direction error with different number of visible anchor points, respectively. The number of anchor points varies from four to eight. We can see that the location error and the direction error both decrease with the number of visible anchors increase. We can also see that four anchor points are enough to achieve acceptable localization accuracy.

6.1.4 Impact of distance. We also explore the impact of distance between the camera and the tag. Figure 14a shows the location error and direction error with different distances between the camera and the tag using one photo. In this experiment, we fix the relative posture of the camera and the tag, and move the camera to change the distance. Note that the moving average improvement is not suitable for this case due to the fast distance change. We can see that the location error and direction error is acceptable when the distance is within 2.8 m, but both increase quickly when the distance is above 2.8 m. This is because we fix the camera's exposure in this experiment and the reflected light intensity becomes lower for a longer distance. When the distance is above 2.8 m, the anchor points on a tag are difficult to be identified. In order to test the performance for a longer distance, we change the exposure to fit the lightness, and move the camera to the distance of 4 m. We measure the location error and direction error on a known plane at 4 m. The result is shown in Figure 14b. The median direction error is about 1.34°, and the median location error is 20 cm. We can see that LiTag can still provide a high direction and location accuracy for such a distance as long as the brightness is enough.

6.2 Camera posture estimation accuracy

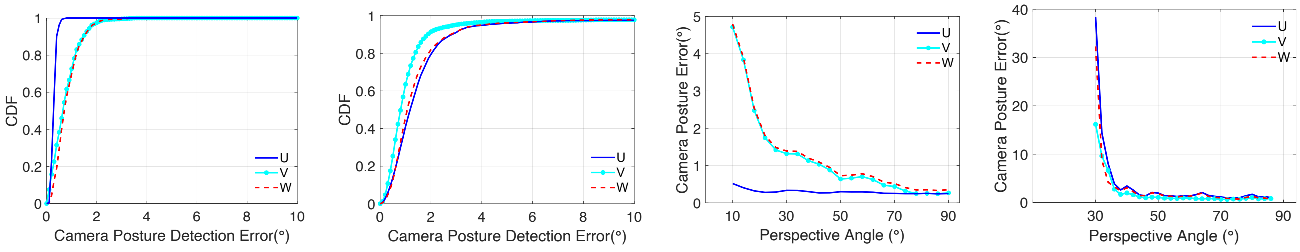
In this section, we evaluate the effectiveness of camera posture estimation. We use **axis error** to represent the camera posture error. The axis error is the corresponding angle difference between the camera's real axes $\overrightarrow{U}, \overrightarrow{V}, \overrightarrow{W}$ and the estimated axes $\overrightarrow{U}', \overrightarrow{V}', \overrightarrow{W}'$.

6.2.1 Basic performance. As shown in Section 4.3, the estimation of camera posture is mainly supported by projection transformation result (coordinate of A and B) and the camera position P_{cam}. We firstly use the ground truth of camera location as the input of our posture detection algorithm to measure the influence of projection transformation. Theoretically, as discussed in Section 4.3, the detection of $\overrightarrow{U}$ axis only relies on the projection transformation result of point A, while the detection of axes $\overrightarrow{V}$ and $\overrightarrow{W}$ require the result of both point A and B. Because of the error existed in both A and B, the error on $\overrightarrow{U}$ should be lower than that on $\overrightarrow{V}$ and $\overrightarrow{W}$. This can be verified in Figure 15a which shows the angle error of three axes of camera. We can see that the error on $\overrightarrow{U}$ axis is lower than 1° and more than 97% of errors on $\overrightarrow{V}$ and $\overrightarrow{W}$ axis are lower than 2°.

To show the influence of the camera position in camera posture estimation, we use the camera's location result P_{cam_r} with moving average obtained in section 6.1 as P_{cam} in Eq. 4. Figure 15b shows the angle error of three axes of the camera. From Figure 15b, we can see that more than 91% of errors are lower than 3° on all three axes. Comparing with Figure 15a, we can see that although the detected location P_{cam_r} of camera may deviate from P_{cam_g}, LiTag system still has a high accuracy on camera posture detection. It means that when the estimated direction is accurate, LiTag could get an accurate posture estimation no matter what the location error is.

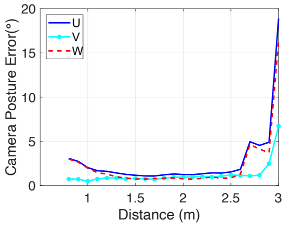
As the localization of camera only works well for the perspective angle $\psi <= 35°$, we only involve data samples with perspective angle $\psi > 35°$ in Figure 15a and Figure 15b.

6.2.2 Perspective angle. We illustrate the impact of perspective angle ψ on the camera posture based on the camera location ground truth P_{cam_g} and the camera localization result P_{cam_r} in Figure 15c and Figure 15d, respectively. From Figure 15c, we can see the error on all three axes decreases with the increase of angle ψ. This is because when ψ is smaller, the image of the tag deforms more severe in the photo and the projection transformation will be less accurate.



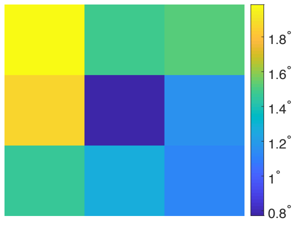
(a) (b) (c) (d)

Figure 15: (a) CDF of camera posture detection error on camera's 'U' (forward), 'V' (up) and 'W' (side) axis with the ground truth of camera location as P_{cam}. (b) CDF of camera posture detection error with the camera localization result as P_{cam}. (c) Camera posture estimation error with different perspective angles using ground truth of the camera location. (d) Camera posture estimation error with different perspective angles using the camera localization result.



(a)

Figure 16: Camera posture error with different localization distance.



(a)

Figure 17: Camera posture error with tag's image in different areas of the photo.

In Figure 15d, we can see that LiTag system can provide accurate camera posture detection when $\psi > 35°$.

6.2.3 Impact of distance. We also measure the impact of the localization distance. We keep the perspective angle ψ unchanged, and vary the distance between the tag and camera in this experiment. In Section 6.1.4, we show that the localization distance can impact camera localization accuracy. Therefore, we use the localization result P_{cam_r} as P_{cam} in this experiment. Figure 16a shows the relation between the localization distance and camera posture estimation accuracy. We can see that camera posture detection error increases when the localization distance exceeds 2.8 *m*. This is because the anchor point becomes darker for a longer distance. However, posture detection error remains relatively small, and does not increase with distance within 2.8 *m*. LiTag can provide high posture detection accuracy within its working distance.

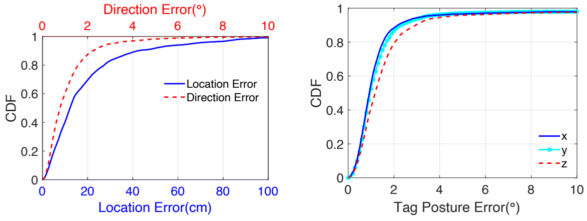
6.2.4 Impact of camera distortion. Camera distortion may degrade the performance of projection transformation. We conduct an experiment to examine the impact of camera distortion. In order to get rid of the effect of camera localization accuracy, we use P_{cam_g} as P_{cam} in this experiment. Figure 17a shows the summation of the mean axis error of $\overrightarrow{U}$, $\overrightarrow{V}$ and $\overrightarrow{W}$ with the tag's image in the corresponding area of the captured photo. From Figure 17a, we can see that although a higher total axis error appears when the tag is shown at the edge of a photo than the tag is shown at the center, all of the errors are lower than 2°. It means that the camera distortion has little impact on our work.

6.3 Localization accuracy of the tag

After obtaining the location and posture of camera in the reference frame of the tag, we can derive the location and posture of the tag in the reference frame of the camera as stated in section 4.4. In this section, we show the performance of LiTag system by the localization and posture detection accuracy: (1) localization accuracy represented by **location error** between the estimated p_t' and the real p_t in coordinate system *uvw*, and **direction error** between vector $\overrightarrow{p_{cam}p_t{}'}$ and vector $\overrightarrow{p_{cam}p_t}$. (2) posture accuracy of the tag: **axis error** is the angle between the detected axes $\overrightarrow{x}'$, $\overrightarrow{y}'$ and $\overrightarrow{z}'$ and the ground truth of axes of the tag $\overrightarrow{x}$, $\overrightarrow{y}$ and $\overrightarrow{z}$ in the coordinate system *uvw*.

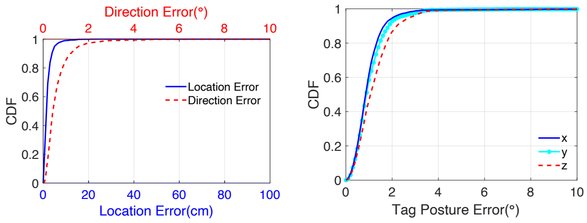
6.3.1 Basic performance. Figure 18a shows the localization accuracy of the tag. We can see that the median location error is about 12 *cm* and 80% of the location errors are lower than 27 *cm*. The median direction error is about 1° and 90% of the direction errors are lower than 2.7°. Figure 18b shows the posture accuracy of the tag. We can see that the median error on all three axes are about 1°, and more than 80% of errors on each axis are lower than 2°. Although the distance measurement still has some room for improvement, the accuracy of direction detection and posture detection is very high. This enables high precision posture based services in many application environments.

6.3.2 Tag on 2D plane. In real localization scenarios, it is often to see that objects move on a specific plane, e.g., a sweeping robot on the floor, transport vehicle on the roof of a warehouse, and an object on a conveyor belt. This feature of the objects can help to further improve the performance. Our method can detect the direction of the tag quite precisely. Thus, if the working plane is known, we can easily derive the distance between the tag and camera using the precise direction, and obtain a more accurate localization result. We conduct experiments to explore how much improvement can be achieved given a target in 2D plane. Figure 19a shows the tag's localization error. We can see that the median location error is about 1.6 *cm*, and 90% of location errors are lower than 4 *cm*. The



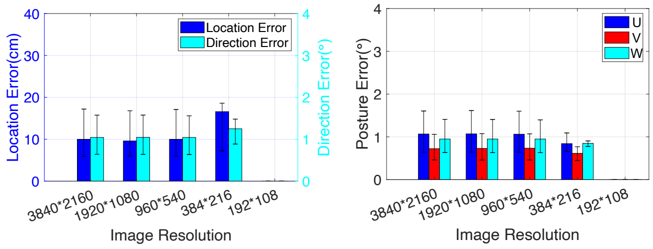
(a) (b)

Figure 18: (a) CDF of the location accuracy of the tag. (b) CDF of the posture detection error of the tag.



(a) (b)

Figure 19: (a) CDF of the location error of the tag in 2D. (b) CDF of the posture detection error of the tag in 2D.



(a) (b) (c)

Figure 20: (a) Camera localization error with different image resolutions. (b) Camera posture detection error with different image resolutions. (c)Tag successful detection rate with different image resolutions.

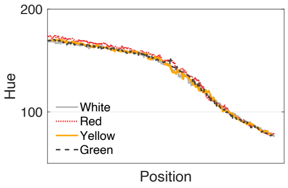


Figure 21: Hue value of the tag's anchor point varies for different positions. It is, however, stable under different color of light.

direction detection accuracy is also improved to 90% of direction errors lower than 1.2°. Figure 19b shows that the posture detection accuracy is also improved. This experiment shows that LiTag system can provide high accuracy localization and posture estimation for objects moving on 2D plane.

6.4 Impact of image resolution

The measurement of colors on the tag and the projection transformation are both related to image resolution. We conduct experiments to evaluate the performance for images of different resolutions including 3840 × 2160, 1920 × 1080, 950 × 540, 384 × 216 and 192 × 108. Figure 20a and Figure 20b show camera localization error and camera posture error under different image resolutions. There is no obvious performance difference among resolutions 3840 × 2160, 1920 × 1080 and 950 × 540. The localization error for the image resolution 384 × 216 is slightly higher. Note that it is difficult to extract usable data with the resolution 192 × 108. Figure 20c shows the tag detection rate under different image resolutions. We denote the detection rate as the normalized number of successfully detected photos, i.e., dividing the number of successfully detected tags under other resolutions by the number under 3840 × 2160 resolution. We can see that the detection rate are almost 100% for both 1920 × 1080 and 950 × 540. This shows that a typical resolution on common cameras is enough for a 15 × 15 *cm* tag. The detection rate is close to 0 for lower resolution, e.g., only 10 tags are detected under the resolution of 384 × 216. These photos, however, are too blurred to detect the tag and extract usable information, so it means other vision-based methods may also fail at such a low resolution.

6.5 Influence of ambient light

We conduct an experiment to evaluate the robustness of LiTag under different kinds of ambient light. We fix the camera and move the tag on the same path, and change the color of light from a bulb. Figure 21 shows the hue of one anchor changes along the moving path, but the hue value on the same position is very similar under different color of ambient light. The tag uses reflective birefringent chips which can reflect light to its source. For example, the light from the light bulb on the ceiling will be reflected to the source position, and thus will not be received by the camera. Moreover, the light captured by the camera is the light reflected by the retroreflective layer, which is emitted by the camera. Thus, this reduces the impact of ambient light to the optical pattern of the tag.

6.6 Computation overhead

To enable LiTag to provide the localization and posture estimation service, we need to enable typical surveillance cameras to detect the tag and extract the features, i.e., tag colors and pixel positions, from images captured. Then, the features are sent to the backend server for further calculation. Therefore, we need to characterize the computational overhead for edge devices and network. We implement the tag detection function using OpenCV on a low-cost Raspberry Pi 4B and test the running time for the detection. The result shows that it takes 0.3 s to successfully detect a tag in a

950 × 540 image. It means that Raspberry Pi supports processing 3 images per second. For each anchor point, we need to transmit 12 bytes (4 bytes for the hue value, and 8 bytes for the pixel coordinates) for the extracted features. Thus, for a tag with 6 anchor points, the total transmission overhead can be 72 bytes. According to our experiment result, the localization and posture estimation on an Intel i7-10700K CPU-based desktop costs 0.47 s. Such a cost is acceptable for real system implementation and can support real-time localization and posture estimation.

7 RELATED WORK

7.1 Geometry-based methods

The camera-based approaches leverage camera intrinsic parameters and known geometry relationship among landmarks. These approaches can localize camera with known landmarks or localize landmarks with known camera position. There are many localization methods on the geometry information [9–12, 22, 23]. High precision camera parameters are essential to these methods. In computer vision, many marker-SLAM methods based on geometry method are proposed [24–30]. These methods use specially designed markers as the known points to localize the camera, which is widely used in robot navigation, virtual reality and augmented reality. Some Geometry-based camera localization approaches identify lamps with known position leveraging VLC (visible light communication) methods [31, 32] and localize the camera using the geometry constraints. Luxapose [2] is a smart application which uses flashing frequency for lamp identification. PIXEL [1] also uses geometry to localize camera, and uses optical rotatory dispersion for lamp identification.

7.2 Light feature based methods

Some works use specially designed LED light to generate identifiable features on the localization space [3, 4, 8, 33–39], and use light intensity or color change to measure the location change. [39] develops a light cover with polarizer and birefringent film to cast light polarization patterns in the space. CELLI [4] generates two sweeping lines with different polarization direction and uses sweeping lines for localization. SmartLight [3] modulates lights with different frequency by a LED array, then it use a convex lens to project the light into the 3D space. The target is usually a photodiode, which should have a battery and a MCU. Pulsar [5] uses inherent features of photodiode, and finds the relationship between light incident angle and RSS. Those works usually need to modify the hardware of lamp to modulate information into the illumination, or require sampling of the light inherent features. RainbowLight [8] uses birefringence to generate different color pattern in different locations, then it derive the position of the camera related to the anchor. However, RainbowLight uses active light and can only provide localization for cameras. Many methods such as [2, 6, 7, 40–42] identify lights with known position. LiTell [6] and iLAMP [7] use inherent features of fluorescent such as frequency and color spectrum to identify each light. Those two approaches do not need any extra modifications on the lamp.

8 DISCUSSION

Non-Line-of-Sight Scenarios: Typically, light propagates through a straight line in the air. Thus, VLP systems usually cannot work in the Non-Line-of-Sight (NLOS) scenario. However, in practical environments such as the supermarket and the warehouse, there are typically multiple surveillance cameras. The surveillance cameras can be arranged to reduce the blind spot and support coverage from different directions to the tag. Therefore, the NLOS problem can be alleviated as long as a target with a tag can be captured by at least one camera.

Working distance and tag size: The tag size is related to the working distance, i.e., the larger the tag is, the longer distance the system can work. In our prototype implementation, the tag size is about 15 ×15 cm. The tag size can be changed according to the application scenario, working distance, or camera resolution. For example, for small objects, small tags should be used. Accordingly, the camera should be deployed closer to the objects or cameras with higher resolution should be used.

Comparison with RFID: LiTag is aiming to be a novel and supplementary technique to RFID and visual-based markers. Comparing with RFID, LiTag may have different advantages and disadvantages. LiTag should require LOS scenario to work while RF based approaches do not require LOS. LiTag system can provide localization and posture estimation with a single camera, whether the tags are static or moving. In RFID, it is still not easy to achieve accurate localization and posture estimation with a single tag. RFID-based applications typically suffer from multi-path effects, while LiTag is less impacted by multi-path. The working range of RFID is related to the RF range while the working range of LiTag is related to camera resolution and tag size. LiTag still has practical limitations which should be addressed before it can be widely applied in practice.

9 CONCLUSION

We present LiTag system, a low-cost and easy-to-deploy solution for object localization and posture estimation. LiTag system uses uncalibrated camera to achieve localization and posture estimation for a special designed *chip-less* and *battery-less* optical tag. The key idea is combing the spatial information extracted form the tag's visible light properties and the geometry information from camera imaging. We implement LiTag system and the evaluation results show that it can achieve posture estimation with a median error of 1°, 2D localization with a median error of 1.6 *cm* and 3D localization with a median error of 12 *cm*. We believe LiTag shows the potential to enable localization and posture estimation for everyday objects with widely deployed uncalibrated cameras for a wide range of scenarios, such as warehouse, supermarket, industrial environment, etc.

ACKNOWLEDGEMENT

We thank our anonymous shepherd and reviewers for their insightful comments to improve the quality of our work. This work is in part supported by NSFC No. 61722210, No. 61932013, No. 61872081. Jiliang Wang is the corresponding author.

REFERENCES

[1] Zhice Yang, Zeyu Wang, Jiansong Zhang, Chenyu Huang, and Qian Zhang. Wearables can afford: Light-weight indoor positioning with visible light. In *Proceedings of ACM MobiSys, May 18-22, 2015, Florence, Italy.*

[2] Ye-Sheng Kuo, Pat Pannuto, Ko-Jen Hsiao, and Prabal Dutta. Luxapose: Indoor positioning with mobile phones and visible light. In *Proceedings of ACM MobiCom, Sept 7-11, 2014, Maui, Hawaii.*

[3] Song Liu and Tian He. Smartlight: Light-weight 3d indoor localization using a single led lamp. In *Proceedings of ACM SenSys, Nov 5-8, 2017, Delft, The Netherlands.*

[4] Yu-Lin Wei, Chang-Jung Huang, Hsin-Mu Tsai, and Kate Ching-Ju Lin. Celli: Indoor positioning using polarized sweeping light beams. In *Proceedings of ACM MobiSys, June 19-23, 2017, Niagara Falls, NY, USA.*

[5] Chi Zhang and Xinyu Zhang. Pulsar: Towards ubiquitous visible light localization. In *Proceedings of ACM MobiCom, Oct 16-20, 2017, Snowbird, Utah, USA.*

[6] Chi Zhang and Xinyu Zhang. Litell: robust indoor localization using unmodified light fixtures. In *Proceedings of ACM MobiCom, Oct 3-7, 2016, New York, USA.*

[7] Shilin Zhu and Xinyu Zhang. Enabling high-precision visible light localization in today's buildings. In *Proceedings of ACM MobiSys, June 19-23, 2017, Niagara Falls, NY, USA.*

[8] Lingkun Li, Pengjin Xie, and Jiliang Wang. Rainbowlight: Towards low cost ambient light positioning with mobile phones. In *Proceedings of ACM MobiCom, Oct 29- Nov 2, 2018, New Delhi, India.*

[9] Changchang Wu. P3. 5p: Pose estimation with unknown focal length. In *Proceedings of IEEE CVPR, June 8-10, 2015, Boston, Massachusetts.*

[10] Yinqiang Zheng, Yubin Kuang, Shigeki Sugimoto, Kalle Astrom, and Masatoshi Okutomi. Revisiting the pnp problem: A fast, general and optimal solution. In *Proceedings of the IEEE ICCV, Dec 3-6, 2013, Darling Harbour, Sydney.*

[11] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Real-time solution to the absolute pose problem with unknown radial distortion and focal length. In *Proceedings of IEEE ICCV, Dec 3-6, 2013, Darling Harbour, Sydney.*

[12] Yinqiang Zheng, Shigeki Sugimoto, Imari Sato, and Masatoshi Okutomi. A general and simple method for camera pose and focal length determination. In *Proceedings of IEEE CVPR, June 24-27, 2014, Columbus, Ohio.*

[13] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *Proceedings of Springer International workshop on vision algorithms, September 21-22, 1999 Corfu, Greece.*

[14] Stephen Boyd and Lieven Vandenberghe. *Convex optimization.* Cambridge university press, 2004.

[15] Klas Josephson and Martin Byrod. Pose estimation with radial distortion and unknown focal length. In *Proceedings of IEEE CVPR, June 20-25, 2009, Miami, Florida.*

[16] Mongi A. Abidi and T Chandra. A new efficient and direct solution for pose estimation using quadrangular targets: Algorithm and evaluation. *IEEE transactions on pattern analysis and machine intelligence*, 17(5):534–538, 1995.

[17] Yubin Kuang and Kalle Astrom. Pose estimation with unknown focal length using points, directions and lines. In *Proceedings of IEEE ICCV, Dec 3-6, 2013, Darling Harbour, Sydney.*

[18] Microscopy. Principles of birefringence. https://www.microscopyu.com/techniques/polarized-light/principles-of-birefringence.

[19] wikipedia. Retroflector. https://en.wikipedia.org/wiki/Retroreflector.

[20] OptiTrack. Optitrack camera. https://optitrack.com/systems/#movement.

[21] Zhengxiong Li, Baicheng Chen, Zhuolin Yang, Huining Li, Chenhan Xu, Xingyu Chen, Kun Wang, and Wenyao Xu. Ferrotag: A paper-based mmwave-scannable tagging infrastructure. In *Proceedings of ACM SenSys, Nov 10-13, 2019, New York, NY, USA.*

[22] Adnan Ansar and Konstantinos Daniilidis. Linear pose estimation from points or lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):578–589, 2003.

[23] Michael Vynnycky and Kamen Kanev. Mathematical analysis of the multisolution phenomenon in the p3p problem. *Springer Journal of Mathematical Imaging and Vision*, 51(2):326–337, 2015.

[24] Fakhr-eddine Ababsa and Malik Mallem. Robust camera pose estimation using 2d fiducials tracking for real-time augmented reality systems. In *Proceedings of ACM SIGGRAPH, August 8-12, 2004, Los Angeles, California, USA.*

[25] Filippo Bergamasco, Andrea Albarelli, Luca Cosmo, Emanuele Rodola, and Andrea Torsello. An accurate and robust artificial marker based on cyclic codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(12):2359–2373, 2016.

[26] Leonid Naimark and Eric Foxlin. Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. In *Proceedings of IEEE ISMAR, Sept 30-Oct 1, 2002, Darmstadt, Germany.*

[27] Joseph DeGol, Timothy Bretl, and Derek Hoiem. Chromatag: a colored marker and fast detection algorithm. In *Proceedings of IEEE ICCV, Oct 22-29, 2017, Venice, Italy.*

[28] Mark Fiala. Designing highly reliable fiducial markers. *IEEE Transactions on Pattern analysis and machine intelligence*, 32(7):1317–1324, 2009.

[29] Rafael Muñoz-Salinas, Manuel J Marín-Jimenez, Enrique Yeguas-Bolivar, and Rafael Medina-Carnicer. Mapping and localization from planar markers. *Elsevier Pattern Recognition*, 73:158–171, 2018.

[30] Andrej Babinec, Ladislav Jurišica, Peter Hubinskỳ, and František Duchoň. Visual localization of mobile robot using artificial markers. *Elsevier Procedia Engineering*, 96:1–9, 2014.

[31] Yue Wu, Purui Wang, Kenuo Xu, Lilei Feng, and Chenren Xu. Turboboosting visible light backscatter communication. In *Proceedings of ACM SIGCOMM, August 10–14, 2020, New York City, USA.*

[32] Xieyang Xu, Yang Shen, Junrui Yang, Chenren Xu, Guobin Shen, Guojun Chen, and Yunzhe Ni. Passivevlc: Enabling practical visible light backscatter communication for battery-free iot applications. In *Proceedings of ACM MobiCom, Oct 16-20, 2017, Snowbird, Utah, USA.*

[33] Julian Randall, Oliver Amft, Jürgen Bohn, and Martin Burri. Luxtrace: indoor positioning using building illumination. *Springer Personal and ubiquitous computing*, 11(6):417–428, 2007.

[34] Nishkam Ravi and Liviu Iftode. *Fiatlux: Fingerprinting rooms using light intensity.* 2007.

[35] Jean Armstrong, Y Sekercioglu, and Adrian Neild. Visible light positioning: a roadmap for international standardization. *IEEE Communications Magazine*, 51(12):68–73, 2013.

[36] Bo Xie, Kongyang Chen, Guang Tan, Mingming Lu, Yunhuai Liu, Jie Wu, and Tian He. Lips: A light intensity–based positioning system for indoor environments. *ACM Transactions on Sensor Networks*, 12(4):1–27, 2016.

[37] Bo Xie, Guang Tan, and Tian He. Spinlight: A high accuracy and robust light positioning system for indoor applications. In *Proceedings of ACM SenSys, Nov 1-4, 2015, Seoul, South Korea.*

[38] Radu Stoleru, Tian He, John A. Stankovic, and David Luebke. A high-accuracy, low-cost localization system for wireless sensor networks. In *Proceedings of ACM SenSys, Nov 2-4, 2005, San Diego, USA.*

[39] Zhao Tian, Yu-Lin Wei, Wei-Nin Chang, Xi Xiong, Changxi Zheng, Hsin-Mu Tsai, Kate Ching-Ju Lin, and Xia Zhou. Augmenting indoor inertial tracking with polarized light. In *Proceedings of ACM MobiSys, June 10-15, 2018, Munich, Germany.*

[40] Masaki Yoshino, Shinichiro Haruyama, and Masao Nakagawa. High-accuracy positioning system using visible led lights and image sensor. In *Proceedings of IEEE Radio and Wireless Symposium, Jan 22-24, 2008, Orlando, FL, USA.*, 2008.

[41] S-H Yang, E-M Jeong, D-R Kim, H-S Kim, Y-H Son, and S-K Han. Indoor three-dimensional location estimation based on led visible light communication. *IET Electronics Letters*, 49(1):54–56, 2013.

[42] Ruipeng Gao, Yang Tian, Fan Ye, Guojie Luo, Kaigui Bian, Yizhou Wang, Tao Wang, and Xiaoming Li. Sextant: Towards ubiquitous indoor localization service by photo-taking of the environment. *IEEE Transactions on Mobile Computing*, 15(2):460–474, 2015.