Qualitative vision-based navigation based on sloped funnel lane concept

Mohamad Mahdi Kassir 1 · Maziar Palhang 1 · Mohammad Reza Ahmadzadeh 2

Received: 9 July 2019 / Accepted: 28 November 2019 / Published online: 17 December 2019 © Springer-Verlag GmbH Germany, part of Springer Nature 2019

Abstract
Funnel lane is a map-less visual navigation technique that tries qualitatively to follow a path that has been recorded before by a camera. Unlike some other methods, funnel lane does not require any calculation to relate world coordinates to image coordinates. However, the funnel lane has some shortcomings. First, it does not provide any information about the radius of rotation of the robot. This reduces the robot maneuverability of the robots and, on some occasions, does not let the robot to correct its path if a deviation occurs. Second, funnel lane constraints sometimes do not distinguish between forward or turning movement of the robot while the robot is in the funnel lane, and command the robot to go forward. This prevents the robot to follow the desired path and leads to failure of the robot’s mission. This paper introduces the sloped funnel lane technique to address these shortcomings. It sets the rotation radius based on the observed frames. Moreover, it reduces translation and rotation ambiguity. Therefore, the robot can follow any desired path leading to more robust and accurate navigation. Experimental results on challenging scenarios on a real ground robot demonstrate the effectiveness of the sloped funnel lane technique.

Keywords Visual path · Qualitative visual navigation · Funnel lane · Sloped funnel lane · Robot navigation

1 Introduction
The process of determining and following a safe and appropriate path from a starting point to a goal point is called navigation. There are various methods that use different sensors to perform the task. Recently, visual navigation methods have been considered by the researchers due to the development of powerful processing modules and the expansion of their applications in mobile robots. These methods are used in both ground [1–10] and aerial [11–16] autonomous robots.

Regardless of the kind of robots, the visual navigation methods can be categorized into two types [17]: map-based and map-less. Map-based visual navigation methods [7,8,18] rely on a model of the environment (map) where the robot has to find its location on it. Map-less visual navigation methods do not need such a model to navigate in the environment [19–21]. The robot relies on the elements observed in the environment to navigate.

Some navigation methods represent the environment with sequential images that characterize the desired path. They are considered as map-less visual navigation methods [22–29] that are based on visual teach and repeat technique. The main advantages of these methods are scalability, not needing global metric map construction, and simple implementation. The images can be gathered easily from an environment. These methods can have more applications especially for robots with limited memory. On the other hand, according to the lack of scale and geometric information, following such paths is not an easy task.

In this paper, our navigation system falls into the category of visual teach and repeat technique. In the teaching phase
In the teaching phase, \( N \) keyframes are extracted from \( m \) frames of the recorded video (Fig. 1), the robot is guided to follow a path while recording a video. After that, keyframes are extracted from the recorded video to make the visual path. The intervals between two consecutive keyframes are called segments.

In the repeating phase (Fig. 2), the robot has to follow the visual path autonomously. Usually, a method is used to control the robot inside a segment in the visual path and a criterion is defined to switch from the current segment to the next one until reaching the last keyframe. Visual servoing is a well-known technique that is used to control the robot inside a segment. Visual servoing approaches usually need calculations such as Jacobian [30], homography or fundamental matrix [5,11,12], which are heavy processes.

Another approach is the funnel lane that was proposed by Chen and Birchfield [2]. The robot follows the path by making qualitative comparisons between the features extracted from the images in the teaching phase and the repeating phase. The method does not make use of traditional calculations of Jacobian, homographies, fundamental matrices or the focus of expansion, and does not require any camera calibration. Funnel lane assumes that the optical axis of the attached camera is parallel to the heading direction of the robot. For each feature, a region is determined based on two constraints of that feature. These regions are called funnel lanes. Their intersection forms the combined funnel lane. The robot tries to keep itself inside the combined funnel lane to reach its destination. Funnel lane has been implemented on ground robots [2,3] and on quadrotors [14,31,32].

In [14], funnel lane was extended to two dimensions and was used for a quadrotor. After that, Nguyen et al. [31] proposed to perform feature matching instead of tracking the features inside a segment. Their reasoning was that Kanade–Lucas–Tomasi feature tracker (KLT) [33] technique is sensitive to the lighting conditions in the environment. Recently, the concept of funnel lane was used by [32]. They made use of the same concept, but they used only semantic object features in their proposed method. In this paper, we make fundamental improvements to the control module of the original funnel lane method. To the best of our knowledge, this has not been investigated before.

The original funnel lane theory has its limitations. It specifies a region for the robot so that it can follow the visual path. The robot is controlled by correcting its heading continuously. However, the funnel lane does not have the ability to provide any information about the rotation radius of the robot (turning radius). For this reason, in funnel lane, the robot’s radius of rotation is pre-set (translation and rotational speed of the robot are set beforehand [2]) and the robot uses the same radius along the whole path when turning is required. Therefore, the robot is not able to deal with all turning conditions (it is not able kinematically). This shortcoming decreases the robot maneuverability and limits the robot movements. It restricts the paths that can be followed by funnel lane. This limitation should be taken into account in the teaching phase to be able to follow the visual path in the repeating phase. In other words, the robot’s radius of rotation in the teaching phase should be set with regard to its value in the repeating phase or vice versa. As a result, the robot is not allowed to take all kinds of paths in the teaching phase as well. In addition, due to this limitation, the robot faces difficulty in correcting its path when it deviates from the desired path in the repeating phase.

Another limitation is the occasional ambiguity between translation (forward movement) and rotation (turning movement) inside the funnel lane. This ambiguity can cause the robot to deviate from the desired path as we will explain later. This issue was mentioned by the authors [2] themselves; however, they tried to reduce this shortcoming by using odometry information.

Lack of heading control inside the funnel lane is another shortcoming. The robot moves forward until it goes out of the funnel lane. This may reduce the accuracy of path following in the repeating phase.

In this paper, we introduce sloped funnel lane which does not have these limitations. Sloped funnel lane looks at all features together, whereas the original funnel lane looks at each feature individually. More information is extracted by looking at all features together. This information is used to solve
the shortcoming of the original funnel lane. In our method, the robot sets the radius of rotation according to the situations it faces. Therefore, the robot is free to take any path with different turning conditions in the teaching phase. The ambiguity is resolved without using any other sensors. As we explained, instead of creating a funnel lane for each feature and intersecting them to form the combined funnel lane, one funnel lane is created by looking at all features simultaneously. Also, two slopes based on the whole features are considered: Enough light exists in the environment, the scene is often static, the environment contains enough texture to extract enough features, there is sufficient overlap between consecutive keyframes, and the change of the conditions in the teaching phase and repeating phase does not affect the feature matching process in the repeating phase very much.

Some notations are used in this paper as follows:

- $c$ is the current image of the robot.
- $V_i$ is the video taken from path $i$.
- $\text{KF}_{i,j}$ is the keyframe number $j$ in path $i$.
- $\text{KF}_{Si}$ is all keyframes in path $i$.
- $S_{i,j}$ is the segment $j$ in path $i$, $S_{i,j} : j \in \{1, 2, \ldots, n - 1\}$.
- $F_a$ features of image $a$.
- $RF_a$ right features of image $a$.
- $LF_a$ left features of image $a$.
- $\text{MF}(a, b)$ matched features of image $a$ with image $b$ (in image $a$).
- $\text{MF}(b, a)$ matched features of image $b$ with image $a$ (in image $b$). Note that $M F(b, a)$ is different with $M F(a, b)$ because the coordinates of the matched features in image $a$ are not necessarily similar to the coordinates of the matched features in image $b$.
- $\text{NMF}(a, b)$ is the number of matched features of image $a$ with image $b$.
- $\sigma_{\text{MF}(a, b)}$ is the standard deviation of $x$ coordinates of $\text{MF}(a, b)$.
- $\text{StdRatio}(a, b)$ is the ratio of standard deviation of $x$ coordinates of $\text{MF}(a, b)$ to the standard deviation of $x$ coordinates of $\text{MF}(b, a)$.
- $\text{ED}(a, b)$ is the Euclidean distance between the median of $x$ coordinates of $\text{MF}(a, b)$ and the median of $x$ coordinates of $\text{MF}(b, a)$.

### Table 1: A comparison of original funnel lane and sloped funnel lane

|                     | Original funnel lane                                                                 | Sloped funnel lane                                                   |
|---------------------|--------------------------------------------------------------------------------------|---------------------------------------------------------------------|
| **Definition**      | Based on one feature (intersection to obtain the combined funnel lane)               | Based on all the features (no intersection)                        |
| **Paths can follow**| Restricted (constant radius of rotation)                                              | Unrestricted (it is free to take any path including rotation in place) |
| **Ability of correcting its direction** | Harder (constant radius of rotation)                                                  | Easier (adaptive radius of rotation)                                |
| **Ambiguity inside the funnel lane** | Exist                                                                                 | Resolved                                                           |
| **Control inside the funnel lane**  | No heading control (moving forward)                                                   | Control in a balanced way                                           |
3 Qualitative vision-based navigation based on funnel lane

In this section, the teaching phase and the repeating phase which exist in qualitative vision-based navigation methods are explained.

3.1 Teaching phase

A robot is controlled to follow a path manually while it is recording a video. Some keyframes are selected from the video. The selected keyframes are called visual path. To select these keyframes, features of the first frame are detected and tracked by Kanade–Lucas–Tomasi feature tracker (KLT) [33] in the video. A keyframe is selected when the percentage of successfully tracked features falls below 50% [2]. The process is repeated until reaching the end of the video. The remaining successfully tracked features in each segment are stored with their coordinates because they are used in the repeating phase. In Fig. 4, the flowchart of keyframes selection is presented.

3.2 Repeating phase

In the repeating phase, the robot is controlled according to its current image and to the first destination keyframe (inside the segment) of the visual path. A keyframe switching criterion is used to determine when the robot reaches the destination keyframe and will be controlled according to the next keyframe. This will be repeated until reaching the last keyframe in the visual path. The original funnel lane and the sloped funnel lane are used to control the robot inside the segments. In the next sections, both methods (original and sloped funnel lane) will be discussed, and then, the keyframe switching criterion will be presented.

4 Original funnel lane

Original funnel lane concept was introduced by Chen and Birchfield [2]. The robot is controlled such that it is able to reach a destination image according to the image it receives from its attached camera. The camera optical axis is parallel to the robot heading, and its optical axis passes through the axis of rotation of the robot. In the following, we explain the original funnel lane. Then, the motion control based on it will be described.
Suppose that the robot wants to move from the current location to location $K_{Fi,j}$. There are some fixed landmarks that are seen in the camera of the robot in both locations as shown in Fig. 5. Suppose we have both the current image and the destination keyframe image and the origin of the feature’s coordinates is at the intersection of the optical axis and the image plane. If the robot goes forward in a straight line with the same heading direction as that of $K_{Fi,j}$, the point $u^c$ will move away from the origin of the feature’s coordinates toward $u^f$. When the robot reaches the destination, point $u^c$ will reach $u^f$. Therefore, the funnel lane is defined as follows:

**Definition 1** A funnel lane of a fixed landmark $L$ and a robot location $K_{Fi,j}$ is the set of locations $FL_{L,K_{Fi,j}}$ such that, for each $C \in FL_{L,K_{Fi,j}}$, the two funnel constraints are satisfied [2]:

\[
|u^c| < |u^f| \quad \text{(constraint 1)}
\]

\[
\text{sign}(u^c) = \text{sign}(u^f) \quad \text{(constraint 2)}
\]

where $u^c$ and $u^f$ are the horizontal coordinates of the image projection of $L$ at locations $C$ and $K_{Fi,j}$, respectively.

If the robot is on the path toward the destination keyframe $K_{Fi,j}$ with the same heading direction, the funnel lane will be as shown in Fig. 6a. Note that the region is specified by two lines which represent the constraints of the funnel lane. The two constraints are satisfied when the robot is inside the funnel lane. For a right side feature ($u^f > 0$), the first constraint ($|u^c| < |u^f|$) is violated when the robot exits from the right side and the second constraint ($\text{sign}(u^c) = \text{sign}(u^f)$) is violated when it exists from the left side. For a left side feature ($u^f < 0$), the opposite is true.

If the heading direction of the robot is not the same direction of the destination keyframe $K_{Fi,j}$, the lines of the funnel lane are rotated by an angle depending on the angle that the robot has with destination keyframe $K_{Fi,j}$ as shown in Fig. 6b.

For each landmark, a funnel lane region is created. By intersecting all funnel lanes, a combined funnel lane is obtained in which the constraints of all features are satisfied. Figure 7 shows an example of how the combined funnel lane will be if we have two features.

### 4.1 Motion control based on original funnel lane

First, the features of the current image are matched with the features of the beginning $K_{Fi,j-1}$ in the segment. Then, the matched features are tracked and their horizontal coordinates are compared with the horizontal coordinates of their correspondence features in the destination $K_{Fi,j}$. If no constraint for each feature is violated, the robot continually moves forward because it is assumed to be inside the combined funnel.
lane. Whenever constraint 1 of a right side keyframe feature \( u_j \geq 0 \) is violated, it means that the robot has gone outside the funnel lane from the left side so it has to get a right turning command, and whenever constraint 2 of a right side feature is violated, it means that the robot has gone outside the funnel lane from the right side so it has to get a left-turning command to get it back to the funnel lane. If the keyframe feature is left side \( u_j < 0 \), the directions are reversed. The constraints are checked for each feature. The final command will be the majority command gets by all features.

### 4.2 Funnel lane limitations

Motion control based on original funnel lane has some limitations which are:

1. **Constant radius of rotation**
   
   In funnel lane, the robot is moving forward and it turns by an amount to the right or to the left depending on the command it gets [2,3]. Note that the translational and rotational speeds are set beforehand. In other word, the radius of rotation of the robot is set beforehand. This reduces the maneuverability of the robot. The robot cannot take any path in the teaching phase. Moreover, in the repeating phase according to this reduction of maneuverability, the robot cannot correct its direction easily when it deviates from the desired path especially in turnings.

2. **The ambiguity of translation and rotation**

   An ambiguity exists between translation (going straight) and rotation (turning) inside the funnel lane itself [2]. Falling inside the funnel lane does not necessarily mean a translation command to the robot. To make it more clear, consider Fig. 8 where there are features just in the right side and the \( x \) coordinates of the destination features lay on the right side of the current features. In the first case, a turning causes the destination features to lay on the right side of the current features (Fig. 8a). In the second case, the path is straight forward and therefore the destination features lay on the right side of the current features (Fig. 8b). In the original funnel lane, the two constraints \( (|u^c| < |u^l| \text{ and } \text{sign}(u^c) = \text{sign}(u^l)) \) are satisfied for all features and the robot falls on the combined funnel lane, which means it will get a straight forward command for both cases. This causes the robot to deviate from the desired path in case of Fig. 8a.

   Existing destination features on both sides of the image help to narrowly constrain the path of the robot. This explains why existing features on both sides in the original funnel lane is necessary [2]. But unfortunately, the ambiguity will remain inside the funnel lane. Moreover, it is not guaranteed that the destination matched features lay on both sides. In turning conditions, the tracked features come out from the frame and the remaining common features between two consecutive keyframes will be shifted to the right or to the left side of the image. In other words, the common features in the destination keyframe will be shifted. To make it clear, consider Fig. 9 which shows two consecutive keyframes that are selected to create the visual path in turning condition. As it is seen, the remaining features are shifted to the right because a turning to the left has occurred. In addition, in the repeating phase at the feature matching process, not all features are matched due to changes of view, light, etc. Moreover, some features are lost due to tracking failure (inside the segment) or due to moving objects. As a result, especially in turning conditions the destination matched features are not guaranteed to be on both sides.

3. **No control inside the funnel lane**

   The robot is moving forward until it gets out of the funnel lane. After getting out, it receives a command to return it back to the funnel lane.

### 5 Sloped funnel lane

Sloped funnel lane is a method that overcomes the shortcomings of the original funnel lane. First, we will explain the sloped funnel lane. Then, the motion control based on it will be described. After that, we will show how the sloped funnel lane can overcome the limitations of the original funnel lane.

The original funnel lane gives no information about the radius of rotation, and there is an ambiguity between trans-
Fig. 9 Two consecutive keyframes selected in left-turning condition, a the first keyframe and b the next keyframe

...lation and rotation as explained. The original funnel lane is created according to the fact that the features will move away from the center of the feature’s coordinates toward the edge of the images when the robot moves in a straight line toward the destination image.

Actually, in the original funnel lane for each feature, a funnel lane is created and later they are combined. However, more information can be extracted by looking at all features together. In straight movements, as seen from the robot’s camera, features move away from the center, in addition, will move away from each other as the robot moves forward. So, we can conclude that the ratio of the standard deviation of $x$ coordinates of all matched features in the current image to the standard deviation of $x$ coordinates of their corresponding features in destination image will become greater as the robot moves forward toward the destination.

To take this fact into account, we add slopes to the original funnel lane. The idea is inspired by the movement of a ball on a sloped surface. If the surface has a slope toward front, the ball moves forward. If the surface has a slope toward left or right sides, the ball will roll to the left or right. Moreover, if the surface has a slope toward front and left/right side at the same time, the ball will roll forward and tend to the left/right.

Depending on the amount of the slope toward forward and toward left/right, the ball will roll in different trajectories.

In our case, the ball is the robot and the surface is the sloped funnel lane. The different trajectories are considered to be turnings with different radii of rotations. In Fig. 10, different trajectories with different radii of rotation when the robot turns to the left are shown. To simplify things, the radius of rotation is specified through the forward slope. The sharper slope means the larger radius of rotation. The right and left slopes are only used to determine the direction of the turn or whether the robot should turn or not. In a nutshell, if there is a right or left slope, the robot will turn right or left according to the radius of turning specified by the forward slope; otherwise, the robot will not turn.

To define such a surface, we define the slope around $y$ axis inversely proportional to the ratio of the standard deviation and the slope around $x$ axis is defined proportional to the difference of the current and destination feature coordinates.

The farther the current image is from the destination keyframe, the slope of the funnel lane around $y$ axis should be larger, and it is reduced when we go toward the destination keyframe. Thus, we define this slope inversely proportional to the ratio of $\sigma_{\text{MF}(c,KF_{i,j})}$ to $\sigma_{\text{MF}(KF_{i,j},c)}$:

$$S_y = 1 - \frac{\sigma_{\text{MF}(c,KF_{i,j})}}{\sigma_{\text{MF}(KF_{i,j},c)}} \quad (1)$$

In addition, the slope around the $x$ axis depends on the distance of the current features with the destination features. The more difference causes the more slope. This slope is used to control the robot inside the funnel lane. We calculate two slopes according to the right and left features. The features...
of the current image are considered as right or left features according to being on the right or left side of the destination keyframe. Two features that represent right and left features are chosen. The feature that represents the right features is the median of the right features ($\mu_l$), and the other one that represents the left features is the median of left features ($\mu_r$). In the case of existing just one feature at each side, the only existing feature is chosen to represent the side. In the absence of the right or left features, the slope is created just by one feature that represents the other ones. The right features create a negative slope around the $x$ axis, while the left features create a positive slope. The final slope is the sum of both slopes. It is noteworthy that the slopes should be normalized before summing their values to balance between the left and right features. So, we define the slope around $x$:

$$S_x = \frac{\mu_c^r - \mu_l^r}{|\mu_l^r|} + \frac{\mu_c^l - \mu_l^l}{|\mu_l^l|}$$ (2)

where $\mu_l^l$ and $\mu_c^l$ are the median coordinates of the left features at the location $KFi, j$ and the median coordinates of their correspondences at location $c$, respectively, $\mu_l^r$ and $\mu_c^r$ are the median coordinates of the right features at the location $KFi, j$ and the median coordinates of their correspondences at location $c$, respectively.

Figure 11 shows an example of summing these two slopes. The sum of two slopes in Fig. 11a will be positive and in Fig. 11b will be negative.

This slope is used to control the robot inside the funnel lane itself. Instead of waiting for the robot to get out of the funnel lane, this slope helps to keep the robot inside it. These two slopes are added to the funnel lane, and as we mentioned in sloped funnel lane, just one funnel lane is created by all features together. Therefore, the definition of the sloped funnel lane will be as the following:

**Definition 2** A sloped funnel lane (SFL) of a set of fixed landmarks $L$, where some of them are left landmarks $L_1 : 1$ to $m$ (projected on the left side of the destination keyframe) and the others are right landmarks $L_2 : m$ to $n$ (projected on the right side of the destination keyframe) at a robot location ($KFi, j$), is the set of locations $SFL_{L, KFi, j}$ such that, for each $C \in SFL_{L, KFi, j}$, the following four funnel constraints are satisfied:

$$|\mu_c^r| < |\mu_l^l|$$ (constraint 1)

$$|\mu_c^l| < |\mu_l^r|$$ (constraint 2)

$$\text{sign}(\mu_c^r) = \text{sign}(\mu_l^l)$$ (constraint 3)

$$\text{sign}(\mu_c^l) = \text{sign}(\mu_l^r)$$ (constraint 4)

and the funnel lane slope around $y$ axis (pitch) is:

$$S_y = 1 - \frac{\sigma_{MF(c, KFi, j)}}{\sigma_{MF(KFi, j, c)}}$$

and the slope around $x$ axis (roll) is:

$$S_x = \frac{\mu_c^r - \mu_l^r}{|\mu_l^r|} + \frac{\mu_c^l - \mu_l^l}{|\mu_l^l|}$$

where $\mu_l^l$ and $\mu_c^l$ are the median coordinates of the image projection of $L_1 : 1 - m$ at the location $KFi, j$ and the median coordinates of their correspondences at location $c$, respectively, $\mu_l^r$ and $\mu_c^r$ are the median coordinates of the image projection of $L_2 : m - n$ at the location $KFi, j$ and the median coordinates of their correspondences at location $c$, respectively. $\sigma_{MF(c, KFi, j)}$ and $\sigma_{MF(KFi, j, c)}$ are the standard deviation of the coordinates of the matched features of the current image with the destination keyframe $KFi, j$ at locations $c$ and $KFi, j$, respectively.

Figure 12b shows the obtained sloped funnel lane when the robot heading angle is the same as the destination keyframe with a slope around the $y$ axis and no slope around the $x$ axis ($S_y > 0$ and $S_x = 0$ which means a forward movement should happen). Figure 12a demonstrates with the same conditions but with just a negative slope around the $x$ axis ($S_y = 0$ and $S_x$ is negative which means a left-turning in place should happen). Figure 12c shows a sloped funnel lane with a slope around the $y$ axis (pitch) and Fig. 12d shows a sloped funnel lane with a slope around the $x$ axis (roll) in case of the absence of the left features.

In the sloped funnel lane similar to the original funnel lane, if the heading direction of the robot is not in the same
Fig. 12  a Sloped funnel lane created with a negative slope around x axis (roll), b sloped funnel lane created with a slope around y axis (pitch). The obtained sloped funnel lane in case of the absence of the left features is shown in c with a negative slope around x axis and in d with a slope around y axis (pitch).

direction of the destination keyframe $j$, the lines of the funnel lane are rotated by an angle which is equal to the angle that the robot has with the destination keyframe $j$.

5.1 Motion control based on sloped funnel lane

The robot moves forward until it is inside a funnel lane with no slope around the x axis. The robot is inside the funnel lane when the four constraints are satisfied. Whenever constraint 1 or constraint 4 is violated, it means that the robot has gone outside the funnel lane from the left side so it gets a right turning command, and whenever constraint 2 or constraint 3 is violated, it means that the robot has gone outside the funnel lane from the right side so it gets a left-turning command to keep it in the funnel lane. While the robot is inside the funnel lane but the funnel lane has a positive slope around the x axis, the robot gets a right command, and when it has a negative slope, it gets a left command. Note that the radius of rotation is determined according to the slope around the y axis in all turning commands. The less the y slope, the sharper the robot turns and vice versa. As the slope around y axis gets near zero, the radius of rotation in turning command will also be near zero and the turning will be more like rotation in place.

The motion control based on the sloped funnel lane is presented in Algorithm 1.

5.2 Dealing of sloped funnel lane with original funnel lane limitations

The sloped funnel lane can deal with the limitations that are mentioned in Sect. 4.2. We will demonstrate the limitations and explain how the sloped funnel lane can handle them.

1. Constant radius of rotation

The radius of rotation is defined in the sloped funnel lane. As we explained, the slope around the y axis determines...
Algorithm 1 Motion control based on sloped funnel lane

1: \( \text{Radius of rotation} = f(S_y) \) 
2: if four constraints are satisfied then \( \triangleright \) inside SFL 
3: \( S_x = 0 \) then \( \triangleright \) zero roll 
4: Move forward 
5: else if \( S_x < 0 \) then \( \triangleright \) roll is negative 
6: Turn left 
7: else if \( S_x > 0 \) then \( \triangleright \) roll is positive 
8: Turn right 
9: end if 
10: else 
11: if constraint 1 or constraint 4 are violated then 
12: Turn right 
13: else if constraint 2 or constraint 3 are violated then 
14: Turn left 
15: end if 
16: end if 

the radius of rotation, which means that the robot has more maneuverability. It is free to take any path in the teaching phase with different turning conditions including rotation in place. In the repeating phase, the robot will set its radius of rotation adaptively, depending on the situation it faces. In addition, if the robot deviates from the path especially in turnings, it can correct its direction more easily by changing its radius of rotation. For example, as shown in Fig. 13, suppose that the robot starts to follow the desired path from A. The robot in position B gets the turning command. In Fig. 13a, the robot faces a problem to correct its direction due to its constant radius of rotation, while in Fig. 13b, the robot corrects its direction easily.

2. The ambiguity of translation and rotation

In the sloped funnel lane, a slope around the \( y \) axis is added which looks at all features together. This slope helps to resolve the ambiguity of rotation and translation. A small slope means a small radius of rotation which means a small translation the robot has to do and vice versa. For example, in Fig. 8, the original funnel lane does not distinguish between both keyframes as we have shown before, but the slope around the \( y \) axis in the sloped funnel lane helps to distinguish between them.

The reason is that the slope around the \( y \) axis is inversely proportional to the standard deviation ratio which in the first case is closer to 1 than the second case. In both cases, a left command is sent. Therefore, no features exist on the left side and slope around the \( x \) axis will be negative. But in the first case, the robot will turn sharper near to rotation in place (less translation), and in the second case, a turning near to moving straight forward occurs (less rotation).

As a result, the sloped funnel lane by resolving this ambiguity prevents the robot from deviating and from getting out of the desired path.

3. No control inside the funnel lane

In the sloped funnel lane, the slope around the \( x \) axis is added. This slope is used to control the robot inside the sloped funnel lane. The slope helps the robot to move in a balanced way through the funnel lane. This helps to keep the robot inside the funnel lane instead of waiting for leaving out of it.

6 Keyframe switching criterion

Funnel lane is a method to control a robot between two keyframes and how to move inside a segment. An important issue is how to define the criterion to switch to another keyframe. Mean square error between the coordinates of the current features and features in the destination keyframe (\( \text{MSE}_{c,j} \)) can be used as a criterion. Chen and Birchfield [3] proposed a method based on MSE. They supposed that the MSE error will become smaller as the robot moves toward the destination image, and the error is decreasing until reaching it. In practice, in our experiments, we noticed that this error was not decreasing uniformly due to losing features and
insensitive steering. This criterion is related to the movement of the robot which makes it so sensitive. Figure 14 shows a sample of this error in a real experiment. As it is shown, the error was oscillating and a lot of switching happens because the criterion needs very sensitive steering. So, steering a little more than necessary or even losing some features causes the MSE not to decrease.

Another method uses mean square error with odometry information to define a probability for switching [2]. We prefer to define a criterion just based on the features themselves, and not using odometry. In another approach [31], feature matching is performed with two in front successive keyframes. The features of the current image are matched with the features in the destination keyframe and with the features in the keyframe next to the destination keyframe. A switching happens whenever the number of matched features with the destination keyframe becomes less than the number of matched features of the keyframe next to the destination keyframe. Therefore, two matchings are required for every cycle to know when to switch.

In our work, a simple method based on the slope around $y$ defined in the sloped funnel lane is used. When StdRatio (current image, destination keyframe) becomes greater than 1 and the Euclidean distance of the median of both coordinates $ED$ (current image, destination keyframe) becomes less than a threshold, a switching happens.

7 Experimental results

Real experiments were conducted on a robot with a VEX platform [34]. The robot uses an IP camera and sends the images $320 \times 200$ using WiFi to a laptop. Blob features are used in this paper. A well-known blob detection technique is SIFT [35] that uses the difference of Gaussian operator to detect features. SURF [36] is a speeded up version of SIFT. It approximates the Gaussian with a box filter, and the convolution with a box filter can be calculated simultaneously for different scales. In our experiments, we choose SURF detectors to speed up the navigation algorithm and its length is chosen to be 64. Larger length gives more accuracy, but it decreases the speed of features matching. For feature tracking, Kanade–Lucas–Tomasi (KLT) algorithm with default block size [31 31] is used. The algorithm is executed on a laptop, and the commands are sent to the robot for path following. The algorithm is implemented in MATLAB 2016 on a VAIO laptop (core i7 1.73GHz RAM 4GB). The robot is shown in Fig. 15. First, the robot is controlled manually from the laptop while recording a video from the traversed path. After that, the visual path is constructed as explained in the previous sections. Then, the robot is placed on the same initial point and is controlled by the algorithm running on the laptop to follow the recorded visual path.

The method used for visual navigation after creating the visual path is presented in Algorithm 2.

In Sect. 5.2, we show how the sloped funnel lane outperforms the original funnel lane. In the sloped funnel lane, unlike the original method, the robot is free to take any path (with different radii of rotations) in the teaching phase proving the dominance of our method over the original one. However, to be fair, in our experiments we only consider paths that the original method can handle as well. Our experiments show that even under these relaxed conditions, the original funnel lane in some cases fails to follow the desired path.

Therefore, these experiments are conducted to show the impact of the original funnel lane limitations on following the paths in the repeating phases even when the robot takes a path with a similar constant radius of rotation in the teaching phase.
Algorithm 2 visual navigation

1: assumed: The visual path i consists from n keyframes, robot starts from segment 1
2: C=capture the current image
3: j=1
4: Detect surf features of C
5: Match features of C with \( KF_{i,j} \)
6: switch = false
7: \( Nof\ F = NMF(C,KF_{i,j}) \)
8: lost=false
9: while \( j < n \) or lost=false do
10: if \( \text{StdRatio}(C,KF_{i,j+1}) > 1 \) and \( ED(C,KF_{i,j+1}) < \text{Threshold1} \) or switch=true then \( \triangleright \) A switching to the next segment is happens
11: \( j = j + 1 \)
12: C=capture the current image
13: Detect surf features of C
14: Match features of C with \( KF_{i,j} \)
15: \( Nof\ F = NMF(C,KF_{i,j}) \)
16: else \( \triangleright \) Control inside a segment
17: if \( Nof\ F > \text{Threshold2} \) then \( \triangleright \) Sufficient features remained
18: Track the matched features with KLT
19: \( Nof\ F = Nof\ F - lost\ features \)
20: Control the robot with the sloped funnel lane
21: else
22: \( time = 0 \)
23: while \( Nof\ F < \text{Threshold2} \) do \( \triangleright \) Robot deviates or features lost
24: C=capture the current image
25: Detect surf features of C
26: Match features of C with \( KF_{i,j} \)
27: \( Nof\ F = NMF(C,KF_{i,j}) \)
28: Stop the robot
29: \( time = time + 1 \)
30: if \( time > \text{Threshold3} \) then \( \triangleright \) lost=true
31: return
32: end if
33: end while
34: end if
35: end if
36: end if
37: end while
38: Stop the robot

Six practical scenarios are considered to show that. Moreover, two paths are chosen to compare the accuracy and the repeatability of our method with the original funnel lane.

First, the visual path is created. Then, the robot is placed at the initial point and it tries to follow the visual path once with the sloped funnel lane and again with the original funnel lane. Figure 16 shows the features in the current image and their correspondence features in the destination keyframe. Also StdRatio\((c,KF_{i,1})\) is shown at the top of the figure.

### 7.1 Six practical scenarios

The goal is to evaluate the path following the ability of both algorithms in six challenging scenarios. Three scenarios are indoors and the rest are outdoors. Two of the three chosen indoor scenarios are short and challenging, while the other one is almost a straight path. The first one is a 9-m path inside a room with a narrow space. First, the robot is controlled to follow the path after that the robot is placed at the same initial point. In the first trial, the robot follows the path with the original funnel lane and in the second trial, it follows the path with a sloped funnel lane. Figure 17a shows the teaching path and both paths followed by the robot with the standard and sloped funnel lane. The robot was not able to follow the path by the original funnel lane, and it hits the chair. The reason is that the radius of rotation is set beforehand. So, although the robot was able to take the path in the teaching phase, it fails in the repeating phase. This is because the original funnel lane does not care about the rotation radius required to compensate for incurred path deviation. So, a small deviation from the desired path or delayed switching (that are normal to happen) makes it impossible to correct or compensate its direction especially in scenarios with narrow areas. On the other hand, our proposed method is able to compensate for path deviations due to the adaptive rotation radius. In other words, our method chooses the appropriate rotation radius according to the destination keyframe. This shows how the sloped funnel lane is more reliable in such situations.

The second scenario is another 6-m path with one turning to the left and with wide space. The robot in the repeating phase is placed 2 m in front of the initial point in the teaching phase. This experiment is chosen to clarify the impact of the fixed rotation radius in correcting its direction as we explain in Sect. 5.2. Figure 17b shows the followed paths with both methods. Even though in the original funnel lane the robot constantly gets left commands, it is not able to follow the path because of fixed rotation radius. The sloped funnel lane was able to correct its direction because it decreases its radius of rotation to take a sharper turn to get back on the desired
path. This scenario demonstrates the ability of the proposed method in correcting its direction and getting back to the desired path. This makes it more robust than the original funnel lane.

The third indoor path is almost straight 25 m in a corridor as shown in 17c. The results were very close and both methods followed the path successfully.

We have also chosen three outdoor scenarios. The first one is a parking lot. The robot is controlled to park between two cars near each other as shown in Fig. 18a. Both methods get to perform equally well. But in the original funnel lane, the robot corrects its direction hardly and it gets closer to the side of the car which increases failure risk. Again this issue is due to fixed rotation enforced by the original funnel lane. One more time, this scenario shows that the sloped funnel lane is more trustable than the original one. Another outdoor scenario is a circular path with a dynamic situation. In the teaching phase, the robot is controlled to follow a circular path, and in the repeating phase two of the parked cars are left and the ability to follow the path with both methods is evaluated. Figure 18b shows the results of both methods. The gray cars are the ones left in the repeating phase. The robot failed to follow the path by the original funnel lane because a lot of features of one side were lost and the ambiguity causes the robot to deviate from the desired visual path. As we explained in Sect. 4.2, the robot moves forward although it is supposed to turn. This causes the robot to defer path correction until it is too late to take any suitable action. Relying on the sloped funnel lane prevents the robot from getting into this problem. Last outdoor scenario is a path with wide turning, and as shown in Fig. 18c, both methods follow the path successfully.

7.2 Accuracy and repeatability comparison

The six practical scenarios showed the ability of both methods to follow some challenging paths. In this section, we compare the accuracy and the repeatability of both methods. The comparison method is proposed by the authors of funnel lane itself [2]. Two indoor paths are chosen and the experience was repeated ten times by both algorithms. The first one is a 10-m path with one sharp turn to the left and low texture indoor environment. Figure 19 shows the selected keyframes that create the visual path of the route. The second one is a 10-m indoor almost straight route. The distance between the final point reached by the robot and the desired final point is calculated. The average RMS Euclidean distance and the standard deviation which expresses the accuracy and the repeatability of the algorithms are calculated by the following equations:
Fig. 18  a The first outdoor parking scenario, b the second outdoor scenario is a closed loop that two cars are left in the repeating phase (funnel lane failed) and c the third outdoor scenario with wide turning

![Image](image1.png)

(a) keyframe 1  (b) keyframe 2

![Image](image2.png)

(c) keyframe 3  (d) keyframe 4

![Image](image3.png)

(e) keyframe 5  (f) keyframe 6

Fig. 19  The keyframes selected to create the visual path which original funnel lane fails to follow and sloped funnel lane follows successfully

![Image](image4.png)

(a) keyframe 1  (b) keyframe 2

![Image](image5.png)

(c) keyframe 3  (d) keyframe 4

![Image](image6.png)

(e) keyframe 5  (f) keyframe 6

The first outdoor parking scenario, b the second outdoor scenario is a closed loop that two cars are left in the repeating phase (funnel lane failed) and c the third outdoor scenario with wide turning

\[
\text{accuracy} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \| x_i - x_g \|^2} \quad (3)
\]

\[
\text{repeatability} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \| x_i - \mu \|^2} \quad (4)
\]

where \( x_g \in \mathbb{R}^2 \) is the desired final point and \( x_i \in \mathbb{R}^2 \) is the final reached point and \( \mu \) is:

\[
\mu = \frac{1}{n} \sum_{i=1}^{n} x_i \quad (5)
\]

The results are shown in Table 2.

Actually, the robot fails to follow the path in the sharp turn with the original funnel lane, while the sloped funnel lane was able to follow the path successfully in most cases. The reason is that the robot in such cases is facing difficulties in correcting its direction due to its constant radius of rotation. This is compounded by the impact of the ambiguity which causes the robot to deviate from the desired path.

It is noteworthy that the sloped funnel lane is as good as the original funnel lane and the experiments performed to show the deficiencies of the original funnel lane have been solved successfully.
The comparison of the accuracy and the repeatability of both original funnel lane and sloped funnel lane

|                      | Original funnel lane acc./rep. | Sloped funnel lane acc./rep. |
|----------------------|--------------------------------|----------------------------|
| Sharp turn           | 3.45/0.55                      | 1.31/0.51                   |
| Almost straight      | 1.19/0.62                      | 1.0/0.46                    |

In general, the shortcomings of the original funnel lane show their effects clearly in some conditions. One of them is in the paths with almost sharp turning. Another one is in narrow spaces where the robot does not have a lot of space to get back to the desired path. We explained these limitations and their effects in Sect. 4.2 and the experiments confirm them. Briefly, the limitations of the funnel lane cause the robot to completely fail to perform its mission, whereas the sloped funnel lane successfully performs it.

Do not forget that in the sloped funnel lane the robot’s radius of rotation is assigned adaptively, depending on the situation it faces. Therefore, the robot can deal with different turning condition including rotation in place. The robot, unlike the original funnel lane, is free to take any path (turnings with any radius) in the teaching phase. To obviate the situations for original funnel lane, in these experiments the robot’s radius of rotation was considered almost similar and constant in both phases; however, in some cases, the original funnel lane failed to follow them.

Two additional experiments are conducted to demonstrate the effectiveness of the approach. The first one is a 30-m indoor path inside the department, and the second one is a 70-m outdoor path inside IUT campus. Figure 20a, b shows the results. The most important thing in experiments is to consider the assumptions mentioned in Sect. 2.

8 Conclusion

In this paper, qualitative visual navigation based on the sloped funnel lane concept was proposed. In the teaching phase, the robot is controlled manually to follow a path. In the repeating phase, the robot has to follow the desired path autonomously. First, a visual path was created by selecting some keyframes from the video taken by the robot in the teaching phase. After that in the repeating phase, the concept of the sloped funnel lane which overcomes some limitations of the original funnel lane was introduced. The rotation radius, unlike the original funnel lane, is set adaptively based on the observed conditions leading to versatile robot maneuverability. Our method also reduces the ambiguity of translation and rotation which exists in the original funnel lane. As a result, a more robust and reliable method than the original funnel lane has been proposed. The limitations of the original funnel lane were explained in detail, and we demonstrated how the proposed sloped funnel lane overcomes them. Moreover, some experiments were conducted on a real robot, and the results showed that our proposed method outperforms the original funnel lane.

In general, the sloped funnel lane focuses on how to make the robot enjoying more maneuverability and more robustness. However, like many other visual navigation methods, it relies on features. So, the enhancement in each of feature matching and tracking processes will enhance the navigation method overall. These enhancements could be the subject of future work. Another interesting direction for future work is to present an environment by multiple connected visual paths.
Then, a graph of them can be created and the robot should perform visual path planning to reach its destination. Moreover, a way to deal with the obstacles and moving objects which appear in front of the robot in the repeating phase can be considered.

Acknowledgements The authors would like to thank Artificial Intelligence laboratory members for their support.

References

1. Diosi A, Remazeilles A, šegvi´c S, Chaumette F (2007) Experimental evaluation of an urban visual path following framework. In: IFAC proceedings volumes (IFAC-PapersOnline).
2. Chen Z, Birchfield ST (2009) Qualitative vision-based path following. IEEE Trans Robot 25:749–754
3. Zhichao C, Birchfield ST (2006) Qualitative vision-based mobile robot navigation. In: Proceedings—IEEE international conference on robotics and automation
4. Guerrero JJ, Martinez-Cantin R, Sagüis C (2005) Visual map-less navigation based on homographies. J Robot Syst 22:569–581
5. Liang BLB, Pears N (2002) Visual navigation using planar homographies. In: Proceedings of 2002 IEEE international conference on robotic automation (Cat. No.02CH37292)
6. Royer E, Lhuillier M, Dhome M, Lavest J-M (2007) Monocular vision for mobile robot localization and autonomous navigation. Int J Comput Vis 74:237–260
7. Remazeilles A, Chaumette F, Gros P (2006) 3D navigation based on a visual memory. In: International conference on robotics and automation
8. Matsumoto Y, Ikeda K, Inaba M, Inoue H (1999) Visual navigation using omnidirectional view sequence. In: 1999 IEEE/RSJ international conference on intelligent robots and systems
9. Pasteau F, Narayanan VK, Babel M, Chaumette F (2016) visual servoing approach for autonomous corridor following and doorways passing in a wheelchair. Robot Auton Syst 75:28–40
10. David J, Manivannan PV (2014) Control of truck-trailer mobile robots: a survey. Intell Serv Robot 7:245–258
11. Remazeilles A, Chaumette F (2007) Image-based robot navigation from an image memory. Robot Auton Syst 55:345–356
12. šegvi´c S, Remazeilles A, Diosi A, Chaumette F (2007) Large scale vision-based navigation without an accurate global reconstruction. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition
13. Do T, Carrillo-Argce LC, Roumeliotis SI (2018) Autonomous flights through image-defined paths. In: Bicchi A, Burgard W (eds) Robotics research. Springer Proceedings in Advanced Robotics, vol 2. Springer, Cham, pp 39–55
14. Nguyen T, Mann GKI, Gosine RG (2014) Vision-based qualitative path-following control of quadrotor aerial vehicle. In: 2014 international conference on unmanned aircraft systems, ICUAS 2014—conference proceedings
15. Royer E, Bom J, Dhome M, Thuilot B, Lhuillier M, Marmoiton F (2005) Outdoor autonomous navigation using monocular vision. In: Proceedings of IEEE/RSJ international conference intelligent robotics system
16. Do T, Carrillo-Argce LC, Roumeliotis SI (2019) High-speed autonomous quadrotor navigation through visual and inertial paths. Int J Robot Res 38:486–504
17. Bonin-Font F, Ortiz A, Oliver G (2008) Visual navigation for mobile robots: a survey. J Intell Robot Syst 53:263
18. Kidono K, Miura J, Shira Y (2002) Autonomous visual navigation of a mobile robot using a human-guided experience. Robot Auton Syst 40:121–130
19. Chao H, Gu Y, Gross J (2013) A comparative study of optical flow and traditional sensors in UAV navigation. In: 2013 American control.
20. Srinivasan MV (2011) Honeybees as a model for the study of visually guided flight, navigation, and biologically inspired robotics. Physiol Rev 91:413–460
21. Chao H, Gu Y, Napolitano M (2013) A survey of optical flow techniques for UAV navigation applications. In: 2013 international conference on unmanned aircraft systems, ICUAS 2013—conference proceedings
22. King P, Vardy A, Forrest AL (2018) Teach-and-repeat path following for an autonomous underwater vehicle. J Field Robot 35:748–763
23. Furgale P, Barfoot TD (2010) Visual teach and repeat for long-range rover autonomy. J Field Robot 27:534–560
24. Ostafew CJ, Schoellig AP, Barfoot TD (2013) Visual teach and repeat, repeat, repeat: iterative learning control to improve mobile robot path tracking in challenging outdoor environments. In: IEEE international conference on intelligent robots and systems
25. Warren M, Greeff M, Patel B, Collier J, Schoellig AP, Barfoot TD (2019) There’s no place like home: visual teach and repeat for emergency return of multirotor UAVs during GPS failure. IEEE Robot Autom Lett 4:161–168
26. Clement L, Kelly J, Barfoot TD (2017) Robust monocular visual teach and repeat aided by local ground planarity and color-constant imagery. J Field Robot 34:74–97
27. Wang Z, Lambert A (2018) ICSP based visual teach and repeat for outdoor car-like robot localization. In: 2018 10th computer science and electronic engineering conference (CEEC)
28. Bista SR, Giordano PR, Chaumette F (2016) Appearance-based indoor navigation by IBVS using mutual information. In: 2016 14th international conference on control. Robotics and vision, ICARCV,自动化, p 2017
29. Krajnik T, Majer F, Halodova L, Vitrn T (2018) Navigation without localisation: reliable teach and repeat based on the convergence theorem. In: IEEE international conference on intelligent robots and systems
30. Burschka D, Hager G (2001) Vision-based control of mobile robots. In: Proceedings of IEEE international conference on robotic automation
31. Nguyen T, Mann GKI, Gosine RG, Vardy A (2016) Appearance-based visual-teach-and-repeat navigation technique for micro aerial vehicle. J Intell Robot Syst Theory Appl 84:217–240
32. Toudeshki AG, Shamshirband F, Vaughan R (2018) UAV visual teach and repeat using only semantic object features. CoRR
33. Tomasi C (1991) Detection and tracking of point features. School of Computer Science, Carnegie Mellon University, Pittsburgh
34. http://www.vexrobotics.com. Accessed 2018
35. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60:91–110
36. Bay H, Ess A, Tuytelaars T, Van Gool L (2008) Speeded-up robust features (SURF). Comput Vis Image Underst 110:346–359

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.