Semi-interactive construction of 3D event logs for scene investigation
Dang, T.K.

Citation for published version (APA):
A Theoretical Analysis of The Perspective Error in Blob Detectors

Many modern feature detectors such as SIFT, SURF, and Hessian-Affine detect blob-like structures. Thanks to their robustness, those blob detectors have become very popular in many applications. We focus on 3D reconstruction, an application where next to robustness accuracy is an important factor.

We identify three factors influencing the accuracy namely lens distortion errors, stochastic errors, and perspective errors. The first two are well studied in literature, we consider the perspective errors. Our analysis is based on the observation that blob detectors detect the centroids of the projections of representative blobs of structures as features; but because of perspective distortion, centroids of projections are not mapped to the centroids of representative blobs. We analyze the resulting erroneous localization using a simplified theoretical model, and show that the effect is small, but systematic and measurable for six modern detectors. In addition, we predict two effects in 3D reconstruction, which we confirm in a typical experimental setup. In practical settings, the random errors and lens distortion errors have a higher impact. We conclude that in such practical cases priority should be given to reducing the other kind of errors before finding ways to correct for the perspective errors.
3.1 Introduction

Feature detection is a crucial step in many computer vision applications. Over the last decade, many new feature detectors have been invented [93, 13, 103, 128, 99]. Those modern detectors are significantly more robust than traditional detectors such as the Harris corner detector [59]. They have empowered and realized many attractive applications such as image retrieval [92] and 3D reconstruction from Internet photo collections [142]. SIFT [93], has become the de facto standard, and is being used in many applications [142, 20, 76].

While for example in image retrieval accuracy is not crucial, it is deemed “especially important” [157] in 3D reconstruction. The reason is that 2D coordinates are back projected to 3D locations, so feature location errors are magnified and might greatly affect the final result. Many modern detectors such as SIFT, but also SURF, Hessian-Laplace and Hessian-Affine, detect blobs [93, 13, 103]. Intuitively, the localization of a blob feature would seem to be inherently more inaccurate than the localization of the more traditional corner features [59, 140].

There are three kinds of location errors known so far. The lens distortion error, which is typically a radial function, is well studied [35, 60, 80]. In order to correct it, many methods have been developed to calibrate the camera lens with or without calibration objects [169, 94, 35]. The stochastic errors are well known, and stems from various sources such as intensity noise or quantization noise. Researchers commonly model it as a normal distribution with varying standard deviation [65, 115, 122, 23]. The third kind of error, which we call perspective drift, is studied in this work.

We are encouraged to study this perspective drift because of a well known problem in 3D reconstruction, namely the projective drift. As described in [29], reconstruction results from long sequences are worse than from short sequences, and may even fail. The cause is accumulation of errors; and the errors may be random or systematic. For random errors (e.g. in the feature locations), solutions to prevent failure exist: they are either local (breaking the sequence down and correcting it [29, 125]), or global (applying bundle adjustment [91]). For systematic errors (e.g. due to an incomplete camera model [28]), we need to study their theoretical basis as well as their practical impact.

The perspective drift is a systematic error that applies to all blob detectors. It is based on the perspective effect: blob detectors detect centroids of the projections of representative blobs of structures as features, but because of perspective distortion, centroids of projections are not mapped to the centroids of the representative blobs.

Our analysis of the perspective drift is structured as follows. We first provide some background on blob detectors in the next section. Then we provide a simplified model for the perspective drift of a blob. From this model, we derive the characteristics of the error and its effects on 3D reconstruction. We perform experiments to confirm those theoretical results. Finally, we give suggestions on how to deal with perspective drift in practice.

3.2 Background on Blob Detectors

In the extensive survey on detectors by Tuytelaars and Mikolajczyk [157], four out of the five blob detectors identified are Hessian-based blob detectors, namely SIFT [93], SURF [13],
Hessian-Laplace and Hessian-Affine [103]. It is this group of detectors we consider here, and we will compactly refer to them as “blob detectors”. We briefly explain how those blob detectors are related, and then summarize their presently known characteristics.

3.2.1 The blob detector family

Blob detectors use the Hessian, i.e. the complete local second derivative of image intensity, to detect features. In the trace of the Hessian, more often referred to as the Laplacian of Gaussians (LoG), blob-like structures are exposed as local extrema. The detector using this detection scheme is called the Hessian-Laplace detector [103]. There are two approximations of the Hessian-Laplace detector, SIFT [93] and SURF [13]. SIFT approximates the LoG using Difference of Gaussians; and SURF approximates the Hessian using box filters.

An improvement of the Hessian-Laplace detector is the Hessian-Affine detector [103]. As the name suggests, it handles affine transformations better than Hessian-Laplace. From an initial location, detected by the Hessian-Laplace scheme, the Hessian-Affine detector iteratively estimates an elliptical region approximating the structure, normalizes it to a circular one, and then re-detects the location of the structure. This makes the detected region covariant to affine transformations, thus improving the re-detection rate under affine transformations.

As shown in [102], the LoG detection scheme is very stable over difference scales. Some corner detectors, namely the Harris-Laplace and the Harris-Affine [103], also try to exploit that advantage by using the LoG to find the optimal scale of features. This makes “the difference between corner and blob detectors less outspoken” [157]; and we thus expect that those detectors will behave similarly to Hessian-based blob detectors. That is why we decided to include them in our treatment.

In brief, the Hessian-Laplace, the Hessian-Affine [103], SIFT [93], and SURF [13] form a group of detectors that extract blob-like structures from images. Modern corner detectors, namely Harris-Laplace and Harris-Affine [103], may have similar behavior to those blob detectors, because they are also based on LoG.

3.2.2 Characteristics

Blob detectors are robust and efficient. In [105], it is shown that the Harris-Affine blob detector is one of the best affine covariant detectors, in terms of robustness against various effects like viewpoint or scale change. In [13], the Hessian-Laplace is shown to be more robust to viewpoint change than its sibling corner detector, the Harris-Laplace [103]. SIFT is shown to be robust against lighting change, viewpoint change, and image compression [93]. In another intensive evaluation of interest points for geometric applications [2], SIFT and Hessian-Affine are among the most robust detectors. Two out of the three efficient implementations discussed in [157] are blob detectors. Both SIFT and SURF run in real-time [68, 13]. The above characteristics explain the omnipresence of these blob detectors in modern computer vision.
3.3 Perspective Drift

In this section, we present a theoretical model to understand the feature localization error of blob detectors caused by the perspective effect. Since our purpose is to understand the effect of the error in 3D reconstruction, we derive equations for the error on the object surface. The analysis is geometrical, i.e. abstracted from image processing procedures, to make it applicable for all blob detectors we considered.

Blob detectors trigger on blob-like signals by finding scale-space extrema of the LoG. To study the effect analytically, we assume that the detected structure was originally a circular blob on a locally flat surface. Our experimental results in Section 3.4, showing the effect is as predicted, support the validity of this assumption.

Since we assume that the blob is circular and on a tilted plane, it will appear as an ellipse in the camera image. Whether the detector is scale invariant or affine invariant, it will always try to find the center of the blob as the location of the structure. In this case, it is the center of the ellipse. However, as we will show below, under a perspective projection, the projected center of an ellipse is not the center of the projected ellipse. Therefore, all blob detectors will make an error in localization. This error we have coined in the term perspective drift, and we analyze it in this section.

![Figure 3.1](image-url): A visual analogy of the perspective drift effect: the image of a circular wheel (a) is an ellipse (b), but the wheel's center in the image is not the center of the ellipse.

We first derive the perspective drift for one camera, to get the basic equations. Then we analyze the relative effect in two (or more) camera positions, which is of interest in practice.

3.3.1 Notation

We use the following notations throughout the chapter. Lower case letters denote scalars, e.g. $x_1$. Bold lower-case letters denote Euclidean vectors, e.g. $\mathbf{x}$. Barred lower case letters denote homogeneous coordinates, e.g. $\bar{c}$. Upper case letters denote matrices, e.g. $H$.

3.3.2 Perspective drift for one camera

We choose parameters as illustrated in Figure 3.2. The circular blob lies in a plane, let us call it the local object plane. Coordinates are chosen such that the camera points at $(0,0)$ in
3.3. Perspective Drift

This plane, from a distance $d$, and at an angle $\varphi$ relative to the vertical axis. The circular blob is at $(u, v)$ in the object plane, and has radius $\rho$. Since we are interested in using the blob detectors for 3D reconstruction, we wish to compute the error in the object plane, i.e., the difference between the actual center of the blob in the object plane, and the back projection of the detected center.

![Figure 3.2: Parameterization for computing perspective drift, one camera case.](image)

Euclidean points $(x, y)$ on the circular blob with radius $\rho$ and center $(u, v)$ are characterized by $(x - u)^2 + (y - v)^2 = \rho^2$, which can be rewritten in homogeneous coordinates as $\bar{x} \cdot \bar{C} \bar{x} = 0$, with $\bar{x} = (x, y, 1)^T$ and

$$C = \begin{bmatrix} 1 & 0 & -u \\ 0 & 1 & -v \\ -u & -v & (u^2 + v^2 - \rho^2) \end{bmatrix}.$$ (3.1)

The projective transformation for the camera in the setup of Figure 3.2 is

$$P_c = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c & -s & 0 \\ 0 & -s & -c & d \end{bmatrix}.$$ (3.2)

where we introduced the shorthand $c = \cos \varphi$, $s = \sin \varphi$, and we took its focal length to be 1 without loss of generality (or equivalently, we measure distances in terms of the focal length). The projection results in a homography $H$ from the object plane to the image plane of the camera, obtained simply by ignoring the third column of the projection matrix, see e.g. [61, chap.12]:

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c & 0 \\ 0 & -s & d \end{bmatrix}.$$ (3.3)

By this homography, the circle is now observed as the conic section

$$S = H^{-T} CH^{-1},$$ (3.4)

which typically is an ellipse (in the degenerated case, it is a line segment).
Lemma 3.3.1  *The homogeneous coordinate vector of the center of a nondegenerate conic homogeneously represented by the symmetric matrix $S$, is the last column of $S^{-1}$.*

**Proof**  (For 3D, but easily adapted to $n$-D.) Let $\mathbf{c} = [c_1, c_2, 1]^T$ denote the center of the conic homogeneously represented by the symmetric matrix $S$. Consider a point $\mathbf{x}$ on the conic, and write it as $\mathbf{x} = \mathbf{c} + \mathbf{y}$, so that $\mathbf{y} = [y_1, y_2, 0]^T$ is a direction vector. Since $\mathbf{c}$ is the center of the conic, the ‘opposite’ point $\mathbf{c} - \mathbf{y}$ should also be on the conic. That gives two equations:

\[
(\mathbf{c} + \mathbf{y})^T S (\mathbf{c} + \mathbf{y}) = 0 \\
(\mathbf{c} - \mathbf{y})^T S (\mathbf{c} - \mathbf{y}) = 0.
\]

The sum of these equations gives information on the length of $\mathbf{y}$. Their difference gives the condition we are looking for. We subtract and expand, and manipulate using the symmetries of the resulting expression, to find that for all $\mathbf{y}$ reaching a point on the conic we should have the equivalent conditions:

\[
\mathbf{y}^T S \mathbf{c} + \mathbf{c}^T S \mathbf{y} = 0 \iff \mathbf{y}^T S \mathbf{c} = - (\mathbf{y}^T S \mathbf{c})^T \iff \mathbf{y}^T S \mathbf{c} = 0. \tag{3.5}
\]

Since $\mathbf{y}$ is of the form $[y_1, y_2, 0]^T$, it follows that $S \mathbf{c}$ must be proportional to $[0, 0, 1]^T$ (plus possibly elements of the nullspace of $S$, but we assume $S$ is non-degenerate). Therefore $\mathbf{c}$ is:

\[
\mathbf{c} = S^{-1} [0, 0, 1]^T, \tag{3.6}
\]

i.e., $\mathbf{c}$ is the last column of $S^{-1}$. \hfill \Box

According to Lemma 3.3.1 the affine center $\mathbf{c}_A$ of the imaged ellipse can be computed as the last column of $S^{-1} = HC^{-1}HT$:

\[
\mathbf{c}_A = S^{-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = HC^{-1}HT \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} u(d - vs) \\ \rho^2 cs + vc(d - vs) \\ -\rho^2 s^2 + (d - vs)^2 \end{bmatrix}.
\]

We project $\mathbf{c}_A$ back from the image plane to the original plane to obtain $\mathbf{c}_A'$ as:

\[
\mathbf{c}_A' = H^{-1} \mathbf{c}_A = C^{-1}HT \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} u(d - vs) \\ v(d - vs) + \rho^2 s \\ (d - vs) \end{bmatrix}. \tag{3.7}
\]

Now the difference of the corresponding Euclidean location with the original circle center $\mathbf{c} = [u, v]^T$ gives the interpreted perspective drift in the object plane simply as:

\[
\text{Perspective drift in object plane} : \delta = \mathbf{c}_A' - \mathbf{c} = \begin{bmatrix} 0 \\ \rho^2 \\ \frac{d}{\sin \phi - v} \end{bmatrix}. \tag{3.8}
\]

Lemma 3.3.2  *The perspective drift is:

- Purely in the direction of the camera tilt, having the same sign as the camera tilt angle.
3.3. Perspective Drift

- Proportional to the area of the blob.

**Proof** From (3.8), we see that the drift in the direction perpendicular to the camera tilt plane is zero. The observable condition is that the object must be in front of the camera:

\[ d - vs > 0. \]

Since \( \phi \) is in the range of \((-\pi/2, \pi/2)\), \( \sin \phi \) has the same sign as \( \phi \). Thus we can rewrite the condition as

\[ \text{sgn}(\phi)(d/s - v) > 0, \]

where \( \text{sgn}(\cdot) \) is the sign of a value. This proves that the condition has the same sign as tilt angle \( \phi \).

Considering perspective drift as a function of the area of the blob, \( \pi \rho^2 \), we see from (3.8) that the second point holds. \( \square \)

The first point of the lemma implies that the detected points on the object surface “drift” along the camera tilt direction. Hence the name “perspective drift”. The dependency of perspective drift on the blob’s area means that the effect differs for features at different scales. We will study this in the experimental section 3.4.2.

3.3.3 Relative perspective drift for two cameras

When we have two cameras, they will in general be rotated and translated relative to each other. In our considered coordinate system relative to the object plane, this can be modeled by a 2D rotation around the origin followed by a translation, see Figure 3.3.

\[ R \delta_2 - \delta_1, \quad (3.9) \]

with the \( \delta_i \) for the two cameras given by (3.8), and \( R \) the rotation of the \((x_2,y_2)\)-axes relative to the \((x_1,y_1)\)-axes. It is clear that the effect is still proportional to the area of the blob feature, and in the direction of camera movement.

**Figure 3.3:** Parameterization for computing perspective drift, two camera case.
3.3.4 Effects on 3D reconstruction

Deriving general equations of the effect of perspective drift on 3D reconstruction is difficult, because in practice not every parameter in (3.8) is known. Fortunately, many reconstruction problems are about man made objects, in simple setups.

In a typical 3D reconstruction setup, a camera may move around an object while fixating on it. We can approximate this setup by equation (3.8), with angle $\varphi$ varying to model the changing aspect of locally planar object patches. Lemma 3.3.2 shows that perspective drifts have the same sign as the camera tilt angle. This will lead to a systematic error in the reconstruction result.

To have a clear observation of the perspective drift effect and simplify the analysis, we consider two separate problems: structure recovery given images of a fully calibrated camera, and motion recovery given the camera intrinsic parameters and the structure. In the former, the back projection rays intersect behind the real structure. Perspective drift pushes the estimated structure away from the views, creating a structure push effect. In the motion recovery problem, if we wrongly assume that the features are correct, putting different points on the object surface together pulls views closer to each other. We call it the view pull effect. The structure push and the view pull effects are illustrated in Figure 3.4. We studied both effects in our experiments.

Figure 3.4: Manifestations of the perspective drift effect in 3D reconstruction. If the camera's motions are known, back projection rays through the presumed feature positions intersect behind the real structure. Vice versa, if features are wrongly assumed to be correct, i.e. $K_1$ and $K_2$ are identical, the second view $C'_2$ is pulled closer to the reference view $C_1$.

From Equation (3.8) we see that the drift is limited by the radius $\rho$ of the blob, and strongly depends on the distance to the camera $d$ as well as the observation angle $\varphi$. In practice, since $d$ is much greater than the object radius $\rho$, and the structure is much smaller than the overall size of the scene, we expect that the perspective drifts are pretty small. To
confirm the existence of the perspective drift, its properties and effect in practice, we have
done experiments presented in the next Section.

\section*{3.4 Experiments}

Confirming the existence of perspective drift and its effects in reconstruction is not a trivial
task. In practice, structures are hardly circular thus it is impossible to pinpoint the structures’
centers. A feasible measurement is the two-frame error between images, i.e. the projections
of relative drifts given in (3.9). Utilizing the prior knowledge about scene setup, we can then
deduce characteristics of projected relative drifts.

\subsection*{3.4.1 Experimental setup}

We choose our setup conforming to a special case of (3.9), no rotation between two coordinate
systems. This makes relative perspective drifts parallel to the camera movement plane. Input
is captured in a typical object reconstruction setup, similar to the method to evaluate affine
invariant detectors in [105]. Pictures, of size 420 by 297 mm, are glued on a flat surface, and
captured using a 35 mm lens, at a distance of 1500 mm (Figure 3.5). We use a turntable to
accurately capture the objects at different angles. The angle step is set at 5 degrees. For each
picture, images are captured in the range of 60 degrees at the resolution of \(2268 \times 1512\).

The six pictures used (Figure 3.6) are graffiti, wall, building, stair photo, office photo, and
books photo, representing outdoor and indoor man made scenes. The two first pictures are
from [105].

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{image3.5}
\caption{The experimental object. Drifts are measured on the planar part}
\end{figure}

To compute the projected relative drift, we first compute the ground truth homography
between a pair of images using manually selected and matched points. Transferring coordi-
nates of a feature in the first image to the second image using the ground truth homography
gives us the coordinates of its ideal match. Two criteria are used to decide if a feature in the
second frame is the match. The first criterion is the distance to the ideal match, we require it
to be smaller than 8 pixels in our experiment. The second criterion is that the SIFT descriptor
distance ratio [93] must be smaller than 0.8. This threshold alone “eliminates 90 percent of
the false matches” [93]. The difference between the coordinates of the ideal match and the coordinates of the match is ideally the *projected relative perspective drift*, i.e. the relative perspective drift (3.9) projected to the image plane of the second camera.

### 3.4.2 Projected relative perspective drift

As explained in the experimental setup, we expect that relative perspective drifts are parallel to the camera movement plane. In the images, this means that the drift is mainly in the horizontal component of the average of the projected drifts. In other words, we expect that the $x$-component of the projected drift vectors shows perspective drift while the $y$-component is close to zero.

For the SIFT detector [93], Figure 3.7 shows that, for all data sets, the projected drift is indeed mainly in the $x$-direction and it increases as the viewpoint change increases. Note that this is even present for the small scale features in the wall data set. This confirms that the effect exists in practice. We also observe that the drift is small compared to the standard deviation of localization error, which represents the random error.

Based on our theoretical considerations, we expect a dependency of perspective drift on the area of the circular blob representing the structure (Equation (3.8), Section 3.3.3). In practice, structures have various local shapes and geometry and it is impossible to estimate the area of a detected structure. Thus in our experiment we tried to observe that aspect through the dependency of perspective drift on a parameter reflecting the structure size, the average feature scale, which is the root of the sum of squares of feature scales (Figure 3.8). We do not see the hypothesized dependency clearly in all data sets. For instance, the office photo data set shows the dependency quite clearly, but the wall data set shows less drift for larger scale features. This is probably because of the smaller number of large scale features.

We quantitatively measure the mean projected drift’s dependency on viewpoint change for six detectors: SIFT [93], SURF [13] and Hessian-Laplace, Hessian-Affine, Harris-Laplace and Harris-Affine [103]. Harris-Laplace and Harris-Affine detectors are considered in this
3.4. Experiments

- **Figure 3.7**: Visualization of location error versus viewpoint angle of the SIFT detector [93]. The ellipse center is at the average location error, the size of its axes correspond to one standard deviation parallel and perpendicular to the average location error vector. For all data sets, the location errors are in the direction of camera movement.

- Experiment because, as mentioned in Section 3.2, they are maybe affected by the perspective drift effect as well.

- Figure 3.9 shows our results. As expected, the mean projected drift mostly increases monotonically in the $x$-direction as the viewpoint changes, while remaining about zero in the $y$-direction. The only exception is SURF where the mean error in the $y$-direction also increases, but it is still smaller than in the $x$-direction. The result to some extent follows the common knowledge that Harris-based detectors are most accurate, followed by the Hessian-based detector. Efficient implementations of the Hessian-Laplace detector, namely SIFT and SURF, are most inaccurate. In terms of perspective drift, the Harris-based detectors are not significantly more accurate than the Hessian-based detector. Indeed, with less than 30-degree viewpoint change they are almost indistinguishable. A rather counterintuitive observation is that, in terms of perspective drift, affine invariant detectors are slightly less accurate than their scale invariant associates. For example, the Hessian-Affine is slightly less accurate then the Hessian-Laplace. The error caused by perspective drift is small, less than 0.35 pixels at 60 degrees viewpoint change. This means perspective drift is unlikely to be a serious problem in practice since those those detectors quickly decrease their repeatability, and might even break down, at 50 degrees viewpoint change [13].
A Theoretical Analysis of The Perspective Error in Blob Detectors

Figure 3.8: Visualization of location error versus feature scale of SIFT detector [93] at 15-degree viewpoint change.

Figure 3.9: Projected location error versus viewpoint change of six detectors: SIFT [93], SURF [13] and Hessian-Laplace, Hessian-Affine, Harris-Laplace and Harris-Affine [103]. The error increases in the viewpoint change direction (a), except for SURF, is almost constant in the perpendicular direction (b).

3.4.3 Manifestations in reconstruction results

We have shown that perspective drift exists in many detectors. In this section we take the SIFT detector [93], the most popular blob detector, for studying the effects of perspective drift in
3.4. Experiments

Table 3.1: Average structure push of SIFT detector at original, half, and one forth size. The effect exists and does not show a strong dependency on resolution. Compared to the distance of 1500 mm from the camera to the experimental object, it is extremely small, less than 0.3 per mill in all data sets.

<table>
<thead>
<tr>
<th>Resolution (pixels)</th>
<th>2268*1512</th>
<th>1134*756</th>
<th>567*378</th>
</tr>
</thead>
<tbody>
<tr>
<td>graffiti</td>
<td>0.17</td>
<td>0.21</td>
<td>0.26</td>
</tr>
<tr>
<td>wall</td>
<td>0.19</td>
<td>0.15</td>
<td>0.13</td>
</tr>
<tr>
<td>building</td>
<td>0.07</td>
<td>0.17</td>
<td>0.29</td>
</tr>
<tr>
<td>stair photo</td>
<td>0.03</td>
<td>0.11</td>
<td>0.19</td>
</tr>
<tr>
<td>office photo</td>
<td>0.22</td>
<td>0.25</td>
<td>0.29</td>
</tr>
<tr>
<td>books photo</td>
<td>0.29</td>
<td>0.28</td>
<td>0.24</td>
</tr>
</tbody>
</table>

3D reconstruction. In 3.3.4 we predicted that the perspective drift will cause structure push and view pull effects when doing object reconstruction. We now confirm those manifestations of perspective drift using the same data sets as in the previous experiment.

We set the ground truth marks at the world xy plane, hence their z-coordinates are zero. Using those marks, with known intrinsic camera parameters, we computed the ground truth pose of the cameras. Together with the feature matches, this information is enough to compute the two manifestations of perspective drift.

Given the ground truth poses of the cameras, we triangulate the SIFT features to recover their 3D location. Since features are on the same plane as the ground truth marks, ideally the average of their z-coordinates should be zero. The actual average gives a measure of the structure push effect predicted in Section 3.3.4. The result is given in Figure 3.10 for the graffiti data set. We see that for every triangulated angle the structure is pushed back. A computation using the parameters of the experimental setup, varying only the possible blob size of the features, indicates that on average the structure push is strongest at 30 degrees, which is confirmed in the graph. The dependency of the push back error on the triangulated angle is not strong though. This suggests we might reasonably approximate the correction for structure push by disregarding the triangulated angle.

The same effects are observed for other data sets. Table 3.1 shows the structure push of all data sets averaged over angles between viewpoints. The error is consistently present, but extremely small, less than 0.03 percent of the distance from the camera to the experimental object.

In practice if lower resolution is used, and thus only larger structures with larger drifts are detected, the structure push can be larger. To check, we virtually downsample the images by filtering out the low scale features. The error increases for some data sets, such as the graffiti, but overall it is not significant. The structure push simulations in Figure 3.10 suggest that the average size of detected structures in the graffiti data set is approximately 20 mm, which is still about 20 pixels in the lowest examined resolution. It explains why the downsampled results are not much different. This agrees with what we have seen in Figure 3.8 that, perspective drift weakly depends on magnitude in our datasets.
Figure 3.10: The *structure push* effect of the SIFT detector in the graffiti data set in Figure 3.6. The effect is seen at every triangulated angle. The dotted lines are theoretical view-pull effects corresponding to different circular structure radius (Equation (3.8)). They suggest that the average radii of representative blobs is about 20 mm (which is about $1/20$ the size of the graffiti image), and the peak error is around 30 degrees.

Given the camera’s intrinsic parameters we can estimate the motion from either ground truth points or features. Figure 3.11 shows results of recovered motions. The cameras’ positions estimated using features, are closer to the reference view compared to the ground truth cameras. This confirms the expected view pull effect of Section 3.3.4.

Figure 3.11: The *view pull* effect of SIFT detector in the graffiti data set (best viewed in color). The ground truth views are in blue. Estimated views, in red, are pulled closer to the reference view, the 0-view. The effect is best observed at views 6 and 7, which are 30 and 35 degrees different.
3.5. Conclusion

We have identified and studied a specific kind of location error, called perspective drift that applies to all blob detectors. When the viewpoint changes, blob detectors detect different points on the object surface. Assuming that the original structure is circular and flat, we have found that this error is in the same direction as the viewpoint move, and proportional to the area of the structure.

We have shown that, in all four popular blob detectors considered, the perspective drift effect exists in practice. It is even seen in some modern corner detectors, namely Harris-Laplace and Harris-Affine. Since it is systematic, we can predict the effects on reconstruction results. Our results confirm the manifestations of perspective drift in both experimental reconstruction problems we considered.

Fortunately, through the experiments we also confirm that in practice the location errors caused by perspective drift are small. Consequently, the errors in 3D reconstruction are small. Thus, we conclude that in most 3D reconstruction applications, priority should be given to handle the lens distortion and stochastic errors.