7. International Immersive Projection Technologies Workshop
9. Eurographics Workshop on Virtual Environments (2003)
J. Deisinger, A. Kunz (Editors)

# Search Space Reduction in Optical Tracking

Robert van Liere and Arjen van Rhijn

Center for Mathematics and Computer Science, CWI
Amsterdam, the Netherlands
{robertl | arjenvr}@cwi.nl

**Abstract**

*In this paper we present a practical method for reducing the space when searching for point patterns in optically tracked virtual reality applications. The method uses a predictor to estimate the future position of a point. The key idea is to define a metric that determines the quality of the predictor, and use this metric to construct a 3D region around the predicted position of each point. The size and shape of the 3D region is based on the kinematic properties of the predicted point. The 3D region is projected into 2D to obtain the required search window.*
*The contribution of the paper is that the search space can be reduced substantially by making use of adaptive window shapes and sizes.*
***Keywords**: optical tracking, feature detection, search spaces.*
***CR Categories and Subject Descriptors:** I.3.1 [Computer Graphics]: Input devices; I.3.6 [Computer Graphics]: Methodology and Techniques;. I.4.8 [Image Processing and Computer Vision]: Scene Analysis;*

## 1. Introduction

Optical tracking for Virtual and Augmented Reality can provide a valuable alternative over other tracking methods like magnetic, gyroscopic, and mechanical trackers. Advantages of optical tracking are that it allows for wireless 'sensors', it is less susceptible to noise, and it allows for many objects to be tracked simultaneously. To avoid complex and computationally expensive image processing, optical tracking systems often use specific markers that can easily be detected in an image. Several systems use retro-reflective material in combination with infra-red lighting. The image processing algorithm then simply comes down to finding light blobs in an otherwise dark grayscale image. These blobs however, do not contain any information for identification, i.e. in the images it is not known which blob of pixels belongs to which marker.

A common approach in optical tracking of interaction devices is to equip the devices with a pattern of retro-reflective markers in a known 3D configuration. In a previous paper, we introduced a method for tracking marker patterns using projective invariant properties [1]. Point patterns are searched in a cloud of points and invariant properties are used to compare the patterns to a description of a model pattern in a database. The pattern best representing the model pattern is chosen. The 3D pose of the interaction device can be reconstructed if a pattern is found in the images of both cameras.

Searching markers patterns, however, requires an exhaustive search over all possible point positions in the 2D image. This is a combinatorially explosive operation and cannot be performed in real-time when a large number of markers are used. For example, an exhaustive search for a K point pattern in an image of $N$ points will require $\binom{N}{K}$ combinations to be examined. For this reason, many techniques have been introduced to reduce the search space. These techniques usually place a small window around the estimated position of the pattern, and search for the pattern using only the points in the window. The number of combinations to be examined would decrease if the number of points in the search window is less than the total number of points. If the pattern cannot be found in the search window (i.e. a 'miss'), then an exhaustive search over all points is required. The problem of defining an appropriate search window is characterized by two factors: the probability the pattern is detected in the window, and the number of points in the search window. These two measures are intimately correlated; increasing the probabil-

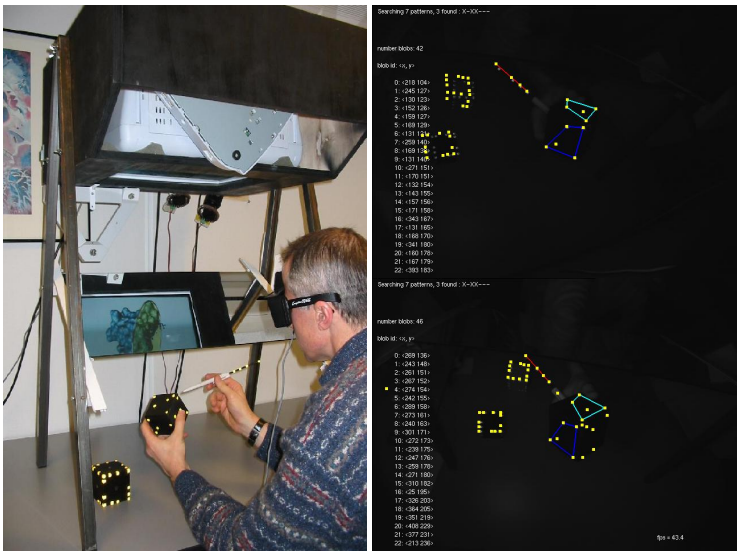

**Figure 1:** *Left: A pen and cube device in the Personal Space Station. A 4 point collinear pattern is pasted on a 8 cm rod that is mounted on a plastic pen. Six 5 point coplanar patterns are pasted on each side of a wooden cube (held in hand). A second cube rests on the table top. Right: The captured images from the left (top image) and right (bottom) cameras contain 42 and 46 blobs respectively. Lines in different colors are used to highlight found patterns. The frame rate for such point clouds is approximately 43 frames per second.*

ity of detection (by increasing window sizes) will inherently increase the number of points in the search window.

In this paper, we introduce an adaptive method for the definition and placement of a search window in a 2D image of highly cluttered and noise corrupted data. The method uses a predictor to estimate the future position of a point. The key idea is to define a metric that determines the quality of the predictor, and use this metric to construct a 3D region around the predicted position of the point. The size and shape of the 3D region is based on the kinematic properties of the predicted point. The region is then projected into 2D to obtain the required search window.

We use our adaptive method for optical tracking in the Personal Space Station (PSS), a near-field VR/AR environment [2]. In this system, a head tracked user looks into a mirror in which stereoscopic images are reflected. Using graspable interaction devices, the user reaches under the mirror to interact directly with virtual objects. The interaction volume is approximately 50 cm x 50 cm x 50 cm and is illuminated by rings of IR LEDs mounted closely around the camera lenses (figure 1 left). Retro-reflective markers are pasted onto the interaction devices. IR light is reflected by the markers into the lens such that, after thresholding, blobs of white pixels can be found in the acquired image (figure 1 right).

The contribution of the paper is twofold. First, we show that using a search window with constant size and shape does not suffice in highly cluttered and noise corrupted data. Second, we show that the search space can be reduced substantially by making use of adaptive window shapes and sizes.

The paper is organized as follows. In the next section we discuss related work. In section 3 we give a description of the steps involved in the construction and projection of the 3D region. In section 4 we show how the method performs on a selection of interaction tasks. Finally, in section 5 we discuss the pros and cons of the method.

## 2. Related Work

Optical tracking systems using retro-reflective markers attached to an input device have been developed in other VR/AR systems; eg. Dorfmüller [3] and Ribo et al. [4]. Both systems first solve the correspondence problem between points in left and right image of the stereo vision system. Fixed 3D distances between points in a triangle structure are used to resolve ambiguities that arise after correspondence problem. A best fit method is used on all distances of 3D-points to find the sought triangle. Both systems use the notion of a search window to speed up search; a rectangular window with a constant width and height is positioned at the point's 2D position in the previous image. Our adaptive method differs from the above mentioned approaches in two ways. First, the size and shape of the search window is not of constant width and height. Instead, each search window around the point has a unique shape and size based on the kinematic properties of the point. Second, the position of the search

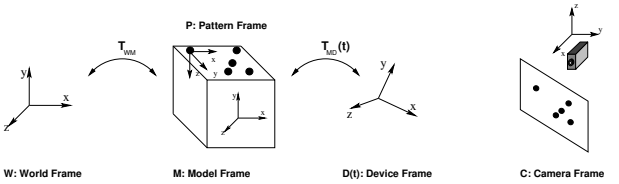


**Figure 2:** *Frames of reference.*

window is computed by prediction methods on the 3D reference frame of the input device itself. In this way, prediction techniques can make use of the device's angular velocity and acceleration. This information is not available if prediction is performed solely on 2D information.

A different approach can be taken by integrating magnetic and optical trackers [5, 6]. Magnetic trackers are fast but inaccurate while optical trackers can be slow when features are complicated to process. By combining the two trackers, a rough estimate of the position of a sensor is obtained from the magnetic trackers and the optical trackers use this position to determine a window to search for a feature in the image. Integrating magnetic trackers in the PSS is not an option for a number of reasons. First, the PSS is a compact desktop environment in which the monitor will cause magnetic interference. Second, multiple interaction devices should be tracked simultaneously. Attaching magnetic sensors to each device would result in many (intertwined) wires. Also, blob detection is a simple operation which can be done fast. No computationally expensive image processing is necessary.

Considerable research has been done in filtering and estimation using an Extended Kalman Filter and maximum likelihood estimation. For example, Extended Kalman Filter have been used to predict head positions in augmented reality applications [7, 8]. The power of Kalman filtering is that it can estimate future events even when the precise nature of the modeled system is unknown. Maximum likelihood formulations have been used to describe robust measures and efficient search strategies for edge template matching [9]. An advantage of uncertainty estimation techniques is that they allow pattern selection to be performed by choosing points that minimize the localization uncertainty. It is beyond the scope of this paper to give a detailed comparison of these techniques. However, we observe that the speed of input devices are more erratic and can far exceed those of head movements, and it is not clear if models can be defined that perform will on cluttered images. As such, it is not clear how well these techniques perform in the PSS environment. In section 5 we will discuss the effects of using a better predictor and maximum likelihood estimators.

## 3. Method

We give a description of the construction and placement of a search window in highly cluttered and noise corrupted data. † The method is described in five steps: predict the position and orientation of each device, predict the position of all points in each device, define a metric that determines the 'quality' of the prediction, construct a 3D region around each point, project the region into 2D.

### Device and Point Kinematics

The frames of reference in a PSS environment are shown in figure 2. The world frame of reference $W$ and the two camera frames of reference $C$ are at an arbitrary but fixed location in the working volume. The model frame defines the positions of all patterns on the device, which is placed in the origin and aligned with the axes of the model frame. The model frame and the world frame are related by a transformation matrix $T_{WM}$, that, for simplicity and without loss of generality, we set to the identity matrix $\mathbf{I}$. The pattern frame of reference $P$ is defined with respect to the model frame $M$. Each point is defined with respect to the point pattern frame of reference. Point patterns can consist of 4 collinear points or 5 coplanar points. A collinear pattern is used to reconstruct the position and direction of an interaction device whereas a coplanar pattern is used to reconstruct the complete 6D pose.

In a previous paper, we have shown how to use projective invariants to efficiently identify point patterns using two calibrated stereo 2D images [1]. Given an identified point pattern, the pose of the device frame $D(t)$ can be reconstructed by using epi-polar geometry to compute the 3D position and orientation of the pattern frame with respect to $W$ and applying a pattern to device frame transformation to the found pattern frame using the model frame and the transformation matrix $T_{MD}(t)$.

The computed position and orientation of the device and the computed positions of the points are called measured positions and orientations and are all relative to $W$. We denote measured device and point positions as $\mathbf{X}^D(t)$ and $\mathbf{X}(t)$. Measured device orientations are denoted as $\theta^D(t)$.

The kinematics model of a device is captured through the equations:

$$\mathbf{X}^D(t+\Delta t) \;=\; \mathbf{X}^D(t) \;+\; \dot{\mathbf{X}}^D(t)\Delta t \;+\; \frac{1}{2}\ddot{\mathbf{X}}^D(t)\Delta t^2$$

† The appendix gives a table of symbols used in the equations.

and

$$\theta^D(t+\Delta t) \;=\; \theta^D(t) \;+\; \dot{\theta}^D(t)\Delta t \;+\; \frac{1}{2}\ddot{\theta}^D(t)\Delta t^2$$

Kinematics of a point is captured through the equation:

$$\mathbf{X}(t+\Delta t) \;=\; \mathbf{X}(t) \;+\; T_{MD}(t+\Delta t)\mathbf{X}_{model}$$

where $T_{MD}(t+\Delta t)$ is the 4x4 transformation matrix describing the transformation from model frame to the device frame $D(t+\Delta t)$ and is defined by $\mathbf{X}^D(t+\Delta t)$ and $\theta^D(t+\Delta t)$, and where $\mathbf{X}_{model}$ is the position of the given point in the model frame.

The kinematics equations are used to predict the position of a point at time $t+\Delta t$ given the measured position and orientation of the device at time $t$. The predicted position of a point is denoted as $\hat{\mathbf{X}}(t+\Delta t)$.

**Predictor quality**

The predictor quality is a metric that provides an indication of how well aspects of the predictor perform at a certain moment in time. Predictor qualities can be defined for position, velocity, acceleration, angular velocity and angular acceleration.

We only use the velocity and accelerator predictor quality. The velocity predictor quality at time $t_i$ is defined as the average difference between the predicted and measured point's velocity within a given history of measured points; i.e.

$$\eta_v(t_i,H) = \frac{\sum_{j=i-H+1}^{i} \|\dot{\mathbf{X}}(t_j) - \hat{\dot{\mathbf{X}}}(t_j))\|}{H}$$

where $H$ is a constant denoting the number of samples in the history and

$$\hat{\dot{\mathbf{X}}}(t_j) = \frac{\hat{\mathbf{X}}(t_j) - \mathbf{X}(t_{j-1})}{t_j - t_{j-1}}$$

The accelerator predictor quality is defined as

$$\eta_a(t_i,H) = \frac{\sum_{j=i-H+1}^{j} \|\ddot{\mathbf{X}}(t_j) - \hat{\ddot{\mathbf{X}}}(t_j))\|}{H}$$

with $\hat{\ddot{\mathbf{X}}}(t_j)$ defined analogous to $\hat{\dot{\mathbf{X}}}(t_j)$. In the current implementation $H$ is set to 5.

Note that $\eta_v$ and $\eta_a$ approach zero when the predictor behaves well, while higher values of $\eta_v$ and $\eta_a$ imply bad predictions.

**3D Region Construction**

A 3D region is defined by constructing an ellipsoid around each predicted point. Geometrically, the ellipsoid is centered at the predicted point position $\hat{\mathbf{X}}(t+\Delta t)$, with its axes defined by the direction of the velocity $\hat{\dot{\mathbf{X}}}$ and the component of the acceleration $\hat{\ddot{\mathbf{X}}}_n$ normal to the velocity, see figure 3. The length of the axis in the direction of $\hat{\dot{\mathbf{X}}}$ is proportional to $\eta_v + \eta_a \frac{\|\hat{\dot{\mathbf{X}}}_t\|}{\|\hat{\dot{\mathbf{X}}}\|}$, i.e. the part of the acceleration tangent to $\hat{\dot{\mathbf{X}}}$ also contributes to the length of the region. Similarly, the length of the axis in the direction of $\hat{\ddot{\mathbf{X}}}_n$ is proportional to $\eta_a \frac{\|\hat{\ddot{\mathbf{X}}}_n\|}{\|\hat{\dot{\mathbf{X}}}\|}$.

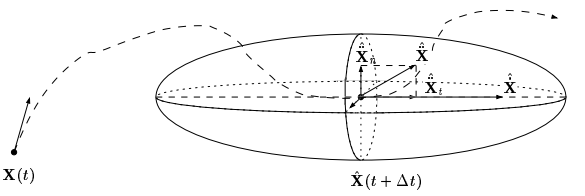


**Figure 3:** *The 3D region is defined as an ellipsoid around the predicted point position $\hat{\mathbf{X}}(t+\Delta t)$. Its axes are in the direction of the predicted velocity and the component of the acceleration normal to the velocity. The length of the axes are proportional to $\eta_v$ and $\eta_a$.*

Analytically, the 3D region spanned by the ellipsoid is defined as

$$\frac{x^2}{a^2} \;+\; \frac{y^2}{b^2} \;+\; \frac{z^2}{c^2} \;<\; 1$$

$$\begin{aligned} a &= \tfrac{1}{2}c(\eta_v + \eta_a \tfrac{\|\hat{\dot{\mathbf{X}}}_t\|}{\|\hat{\dot{\mathbf{X}}}\|}) + \varepsilon \\ b &= \tfrac{1}{2}c(\eta_a \tfrac{\|\hat{\ddot{\mathbf{X}}}_n\|}{\|\hat{\dot{\mathbf{X}}}\|}) + \varepsilon \\ c &= \varepsilon \end{aligned}$$

where $c$ is a scaling parameter and $\varepsilon$ denotes the noise at the predicted point position.

The constant $\varepsilon$ is used to estimate the noise. More accurate would be to model the static and dynamic aspects of the noise as a function over the working volume (for example, see [7]). Note that when the predictor quality is high (i.e. $\eta_v$ and $\eta_a$ approach 0) the shape of the region approaches a sphere defined by the amount of noise in the region.

Our implementation uses only the constructed 3D region based on the translation position predictor. In general, the region would also depend on the orientation predictor. This region could be parameterized with predictor quality metrics (i.e. $\eta_{\dot{\theta}}$ and $\eta_{\ddot{\theta}}$). The shape of the constructed region based on an orientation predictor would resemble a patch of a sphere.

**Region Projection**

Although the constructed region can be projected onto a 2D ellipse, the implementation fits a bounding box around the ellipsoid and projects the bounding box onto 2D (see figure 4).

A 2D region is constructed by projecting all 8 points of the bounding box and computing the convex hull of the projected points. It can easily be seen that the convex hull is a polygon with 4 or 6 sides. Finding points in the region is then easily done by a point-in-polygon test.

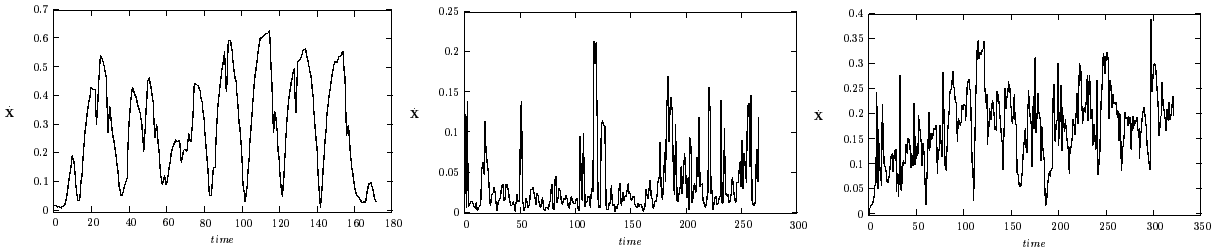


**Figure 5:** *Point velocity over time for three interaction tasks.*

| | Set 1 | | Set 2 | | Set 3 | |
|---|---|---|---|---|---|---|
| | $\mu$hits | %miss | $\mu$hits | %miss | $\mu$hits | %miss |
| adaptive | 3.54 | 4 | 3.37 | 8 | 3.47 | 13 |
| 10x10 | 0.62 | 80 | 1.12 | 60 | 1.39 | 73 |
| 20x20 | 1.64 | 66 | 2.79 | 24 | 4.05 | 29 |
| 40x40 | 4.39 | 31 | 4.82 | 4 | 7.41 | 10 |

**Table 1:** *The adaptive method is compared with using a constant window size of 10x10, 20x20 and 40x40 pixels. Average number of points and percentage of misses tabulated for each dataset.*

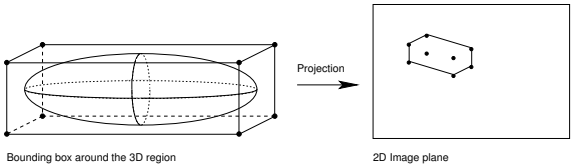


**Figure 4:** *A bounding box of the ellipsoid and its projection onto the 2D plane with the convex hull.*

### 3.1. Algorithm Summary

The complete algorithm is summarized in the following steps:

**foreach** device :

1. predict the position and orientation
2. determine predictor quality measures $\eta_v$ and $\eta_a$
3. **foreach** point $\in$ device :
   a. construct 3D region
   b. project region onto a region of the 2D image
   c. use the projected region as the 2D search window

It should be noted that many optimizations can be made to this algorithm. For example, instead of constructing the predicted regions of each marker, only those markers that are predicted to be visible by the cameras should be taken into account. Visibility is easily computed since the predicted pose of each device is known.

## 4. Results

The adaptive method was run on three recorded motion datasets that are considered representative of interaction tasks in the PSS. The first dataset consists of mostly 3D translations, resulting from simple positioning tasks. The second dataset consists of mostly 3D rotations which result from orientation tasks. The third dataset consists of a combination of translations and rotations. Figure 5 plots the velocity of a point for each recorded dataset. From the plots it can be seen that the velocity profile is very capricious.

Table 1 compares the adaptive method with three constant window sizes for each dataset. A window size of 10x10, 20x20, and 40x40 pixels was used. The average number of points in the search window and the percentage of misses is tabulated. The table illustrates the trade-off in finding an appropriate search window: when the window size is small (10x10 pixels), the number of misses is high. On the other hand when the window size is large (40x40 pixels), the number of additional points is high.

With respect to the percentage misses, the adaptive method is competitive with window sizes of 40x40. The adaptive method is superior to window sizes of 10x10 and 20x20. With respect to the number of points in the window, the adaptive method is superior to window sizes of 40x40, but usually performs less than window sizes of 10x10 and 20x20.

From this table it can be argued that the adaptive case performs better than 10x10 and 20x20 cases since the percentage of misses is far less. Also, the adaptive case performs better than 40x40 case since the number of points in the window is clearly less, while the percentage misses is similar.

| Set 1 | $\eta_v < 0.50$ | | $\eta_v < 0.33$ | | $\eta_v < 0.25$ | |
|---|---|---|---|---|---|---|
| | $\mu$hits | %miss | $\mu$hits | %miss | $\mu$hits | %miss |
| adaptive | 3.41 | 7 | 3.20 | 6 | 2.69 | 2 |
| 10x10 | 0.81 | 88 | 0.99 | 52 | 1.35 | 16 |
| 20x20 | 2.09 | 69 | 2.69 | 37 | 3.46 | 9 |
| 40x40 | 5.42 | 28 | 6.14 | 5 | 5.69 | 0 |

**Table 2:** *The adaptive method is compared with using a constant window size of 10x10, 20x20 and 40x40 pixels. Average number of points and percentage of misses tabulated for each dataset.*

Figure 6 plots the predictor quality $\eta_v$ (top) and the area of the projected 3D region in pixels (bottom) for the translation task (left plot in figure 5). The predictor quality fluctuates greatly, which indicates that the predictor has difficulty in predicting the point velocity. The area fluctuates over time between 100 and 550 pixels. The plot profiles in figure 6 are correlated. High values for $\eta_v$ result in large 3D regions.

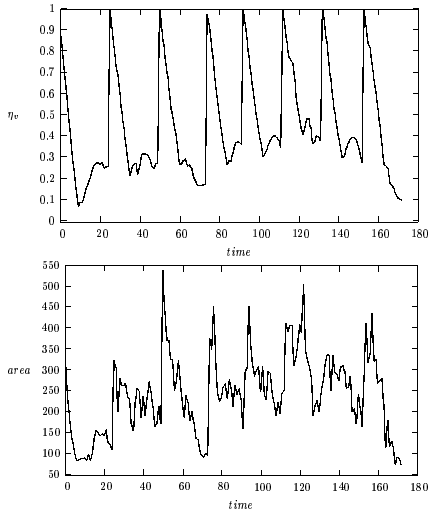


**Figure 6:** $\eta_v$ *(top) and area of the projected 3D region in pixels (bottom) over time.*

Table 2 compares the adaptive method with constant window size when $\eta_v$ is low. The average number of points in the search window and the percentage misses are computed only when the predictor quality is less than 0.50, 0.33 and 0.25. With respect to the percentage misses, the adaptive method is competitive with window sizes of 40x40. The adaptive method is superior to window sizes of 10x10 and 20x20. With respect to the number of points in the window, the adaptive method is superior to window sizes of 40x40.

From the table it can be seen that the adaptive method performs competitive even for cases when a better predictor is used. From this table it can be argued that the adaptive case performs better than 10x10 and 20x20 cases since the percentage of misses is far less. Also, the adaptive case performs better than 40x40 case since the number of points in the window is clearly less, while the percentage misses is similar.

## 5. Discussion

In the previous sections we have described a method for predicting a search window size and shape in cluttered and noisy data. The key idea is to define a quality of the predictor metric. The metric to construct a 3D region around the predicted position of each point. The required search window is constructed by projecting the 3D region onto image space.

The results show that the method performs favorably compared to search windows of constant size, even when the good predictors are used.

We now discuss some pros and cons of the method:

- Impact on performance.
  The adaptive method introduces an additional performance penalty to compute the region and its projection. To discuss the impact on the total performance, we analyze the trade-off between exhaustive search and local search for a pattern.
  Exhaustive search for a $K$ point pattern in a cloud of $N$ points will require

$$\begin{pmatrix} N \\ K \end{pmatrix} = \frac{N \cdot N-1 \cdot \ldots \cdot N-K+1}{K \cdot K-1 \cdot \ldots \cdot 1}$$

  combinations to be examined. Each combination must be checked with a model pattern in a database, and the combination best representing the model will be chosen. This is combinatorially explosive and cannot be performed in real-time when a large number of points are used.
  The number of combinations examined when using search windows is:

$$\prod_{i=1}^{K} N_i = N_1 \cdot N_2 \cdot \ldots \cdot N_K$$

  where $N_i$ denotes the number of points in each search window of the $K$ point pattern. If $N_i << N$ then the number of combinations will be significantly less in the case of search windows.

For all test cases the performance of the adaptive method is never worse than that of a constant window size and for most cases it gives superior performance. Tables 1 and 2 shown that the average number of points in a 40x40 window is at least 30 percent more than in an adaptive window.

- A practical solution.
  As was discussed in the related work, an Extended Kalman Filter or maximum likelihood estimation could be used to implement a predictor or to estimate 3D confidence regions based on probabilistic formulations. A better predictor would increase the predictor quality metrics (i.e. decrease $\eta_v$ and $\eta_a$) and decrease 2D search window area size. A probabilistic formulation of the 3D region would allow for searching strategies to be implemented based on confidence levels. Also, we have defined the 3D region based on the translation position predictor quality. In general, the region would also depend on the orientation predictor quality.
  In this study we have chosen for a very simple predictor, a geometric method for the construction of the 3D region, and an simple projection scheme to obtain search window. Hence, the 'practical solution'. It is not clear what the impact would be if Kalman filters or probabilistic formulations were used. For example, table 2 shows that the adaptive method still gives superior performance in case the predictor quality is high.
- Optimizations.
  We have used the adaptive method to reduce the space for searching features efficiently in highly cluttered and noise corrupted data. An interaction device is constructed from one or more marker patterns and each pattern is constructed from four or five markers.
  The adaptive method can be improved in two ways. Firstly, projective invariant properties of the point patterns can be used to determine the position of other search windows in the pattern. In this way, only one search window needs to be defined and the position and orientation of other search windows in the point pattern will be fixed. Secondly, by making use of the device, point pattern, and point hierarchy, all points belonging to the device can be removed once a point pattern on the corresponding device is found. Both of these optimizations would reduce the number of points in the image substantially.

## 6. Conclusion

In this paper we have presented a practical method for reducing search space for tracking point patterns in optical tracking. The key idea is to define a metric that determines the quality of the predictor, and use this metric to construct a 3D region around the predicted position of each point. The region is then projected into 2D to obtain the required search window.

The problem of defining an appropriate search window is characterized by two factors: the probability of point detection, and the number of points in the search window. These two measures are intimately correlated; increasing the probability of detection (by increasing window sizes) will inherently increase the number of points in the search window. For all test cases the performance of the presented method is never worse than that of using a constant search window size and for most movements it gives superior performance.

## References


1. R. van Liere and J.D. Mulder. Optical tracking using projective invariant marker pattern properties. In *IEEE Virtual Reality 2003*, 2003.
2. J.D. Mulder and R. van Liere. The personal space station: Bringing interaction within reach. In S. Richer, P. Richard, and B. Taravel, editors, *Proceedings of the Virtual Reality International Conference, VRIC 2002*, pages 73–81, 2002.
3. K. Dorfmüller. Robust tracking for augmented reality using retroreflective markers. *Computers and Graphics*, 23(6):795–800, 1999.
4. M. Ribo, A. Pinz, and A. Fuhrmann. A new optical tracking system for virtual and augmented reality applications. In *Proceedings of the IEEE Instrumentation and Measurement Technical Conference*, volume 3, pages 1932–1936, Budapest, Hungary, 2001.
5. A. State, G. Hirota, D. Chen, W. Garrett, and M. Livingston. Superior augmented reality registration by integrating landmark tracking and magnetic tracking. *Computer Graphics (Proceedings SIGGRAPH '96)*, 30:429–438, 1996.
6. T. Auer and A. Pinz. The integration of optical and magnetic tracking for multi-user augmented reality. *Computers and Graphics*, 23(6):805–808, 1999.
7. R. Azuma and G. Bishop. Improving static and dynamic registration in an optical see through HMD. *Computer Graphics (Proceedings SIGGRAPH '94)*, 28:197–204, 1994.
8. R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent advances in augmented reality. *Computer Graphics and Applications*, 22(12):34–47, 2001.
9. C. Olson. Maximum-likelihood image matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(6):853–857, 2002.

**Appendix**

| symbol | description |
|---|---|
| $\mathbf{X}$, $\theta$ | measured position and orientation |
| $\hat{\mathbf{X}}$, $\hat{\theta}$ | predicted position and orientation |
| $\dot{\mathbf{X}}$, $\ddot{\mathbf{X}}$ | velocity and acceleration |
| $\dot{\theta}$, $\ddot{\theta}$ | angular velocity and acceleration |
| $T_{WM}$ | world to model 4x4 transformation matrix |
| $T_{MD}$ | model to device 4x4 transformation matrix |
| | |
| H | length of history |
| $\eta_v$ | predictor quality for velocity |
| $\eta_a$ | predictor quality for acceleration |