

GENERALIZED PATTERN MATCHING USING ORBIT DECOMPOSITION

Yacov Hel-Or and Hagit Hel-Or

ABSTRACT

Image Processing and Computer Vision applications often require finding a particular pattern in a set of images. The task involves finding appearances of a given pattern in an image under various transformations and at various locations. This process is of very high time complexity since a search must be implemented both in the transformation domain and in the spatial domain. Contributing to this complexity is the chosen distance metric that measures the similarity between patterns. The Euclidean distance, for example, may change drastically when a small transformation is applied to the pattern. Applying a different metric distance might be advantageous, though at the expense of losing the norm structure of the Euclidean space. In this work we present a new method for fast search in the transformation domain which can also be applied in metric spaces. The method is based on recursive decomposition of the transformation domain, and a rejection scheme which enables the process to quickly reject as irrelevant large percentages of this decomposition.

1. INTRODUCTION

Pattern Matching is the task of finding patterns in images. This problem may appear in various forms. In the context of this paper we consider the Pattern Matching problem where a pattern may appear under different transformations (e.g. object recognition, motion tracking etc). To demonstrate the complexity of this problem, we represent a pattern as a point in a high dimensional *pattern space* (e.g. a 10×10 pattern is represented by a point in \mathcal{R}^{100}). Consider a given 2D pattern P of size $n = k \times k$ and consider a set of transformations $T(\alpha)$ that may be applied to P , where α is the transformation parameter. Denote by $T(\alpha)P$, the transformation $T(\alpha)$ applied to pattern P . The pattern P and the transformed pattern $T(\alpha)P$ are points in the n -dimensional pattern space. If the set of transformations $T(\alpha)$ is a group $T(\alpha)P$ for all α forms an *orbit* in \mathcal{R}^n . In general, the number of transformation parameters d is less than n , hence the orbit forms a d -dimensional manifold in n -dimensional space.

Current Address: Stanford University, Department of Statistics, Sequoia Hall, Stanford, CA, U.S.A. 94305-4065. Email: toky@idc.ac.il, hagit@cs.haifa.ac.il. This work was supported by The Israeli Ministry of Science

We reformulate the generalized pattern matching problem as follows: let W be an image window of size $k \times k = n$ and $T(\alpha)P$ the orbit of patterns to be matched. In order to evaluate the match between the image window and any pattern in the orbit, the distance between W and the orbit $T(\alpha)P$ must be calculated. Let $d(P, Q)$ be a distance measure between any 2 points in \mathcal{R}^n , then the *orbit distance* to be evaluated is:

$$\Delta_\alpha(W, P) = \min_\alpha \{d(W, T(\alpha)P)\} \quad (1.1)$$

If the orbit distance is below a given threshold, the window is considered as matching the pattern, otherwise it is a non-match.

Unfortunately, actually calculating the orbit distance is complicated and expensive; In most cases, the orbit is highly complex in the sense that two patterns that are close in the transformation domain may be distant in pattern space. This complex behavior of the orbit, is in addition to its being non-convex and embedded in a high-dimensional space, making the calculation of the orbit distance time consuming.

2. PREVIOUS APPROACHES

Several approaches were suggested to deal with such problems. In general, the complexity of search within a complex manifold can be reduced if the manifold has a simpler structure or is of smaller dimension. There are several common techniques implementing this strategy:

Orbit simplification: It is possible to nullify some of the transformation parameters by making the problem invariant to these parameters. Pre-processing the input data and representing it in a canonical form, is a common strategy in this approach [1]. Nullifying some of the transformation parameters, subsequently, reduces the dimensionality of the manifold and thus simplifies the search problem. Another approach is to find a function defined over the pattern space, that is constant over some transformation parameters and thus invariant to these parameters [6, 13, 8, 5]. For example, applying a dot product between a pattern and a kernel is rotation invariant if the kernel is rotationally symmetric.

Dimensionality reduction: Another approach attempts to reduce the dimensionality of the pattern space. A popular example is the Wavelets (or DCT) representations where

the energy of natural patterns is concentrated in few coefficients, thus, reducing the dimensionality of the pattern space [4, 9]. Another effective approach is to find a reduced linear basis for the pattern manifold using principal component analysis (PCA), and searching the manifold in this reduced space [2, 12].

Fast search: The two previous approaches try to simplify the geometry of the problem. A different strategy is to apply an exhaustive search in some of the transformation domains. This may be possible only if a fast search technique is available. For example, it is relatively simple to apply an exhaustive-search in the translation and scale parameters exploited the efficiency of the pyramidal representation and the fast implementation of convolutions.

Although some the above methods try to reduce the orbit complexity or its dimensionality, their performance is limited due to intrinsic complications such as using “normed” spaces for representing 2D patterns. Typically, a pattern is represented by a point in a linear space and the distance between two different patterns is defined by the Euclidean distance of their corresponding points.

This paper suggests a new technique that performs a fast search within a pattern orbit. In addition to the fast performance of this method, it can be applied in metric spaces as well, opening the scope to a large variety of new metric distances that can be designed to simplify the orbit complexity [3, 10, 11].

Figure 1 shows two examples of such orbits. A 16×16 image was used as a pattern. The 2D rotation transformation group was sampled at equal rotation angles and applied to the pattern. Figure 1a shows the pattern orbit in Euclidean space. For visualization, the orbit was projected onto the 3 dominant directions (Eigen vectors associated with the three largest Eigen values). The segments connect between orbit points that are associated with consecutive sampled parameter values. It can be seen that the orbit is highly irregular, thus a search within the orbit is not easily simplified and sped up. Figure 1b shows the pattern orbit in a metric space, using the following metric distance: $d(P, Q) = \delta(P, Q) + \delta(Q, P)$ where $\delta(P, Q) \doteq \sum_{x,y} \min_{i,j \in \{-1,0,1\}} [P(x-i, y-j) - Q(x, y)]^2$. In this case, the three dominant directions were calculated using multidimensional scaling [7]. The simple and regular behavior of the metric orbit is self evident.

3. FAST SEARCH IN METRIC SPACE

As above, assume $d(Q, S)$ is a distance metric defined between any 2 points in \mathcal{R}^n . This measure of similarity between two patterns may be of any form, linear or non-linear, closed form or algorithmic. The only requirement is that $d(\cdot, \cdot)$ is a metric. In order to determine whether an image

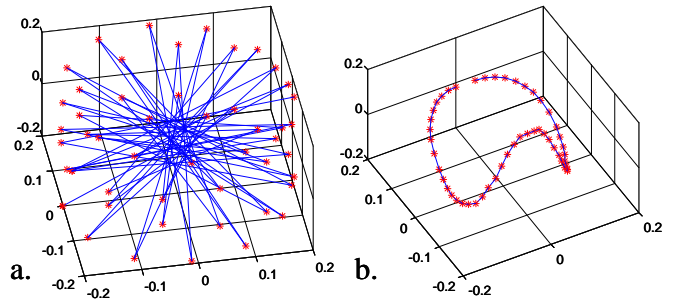


Fig. 1. Examples of pattern manifolds (see text). a) Pattern orbit in Euclidean space. b) the same orbit in a metric space.

window W is similar to $T(\alpha)P$ for any α , one must estimate the orbit distance $\Delta_\alpha(W, P)$ as defined above (Equation 1.1). In the general case, an orbit distance Δ_α is not a metric, as it does not satisfy the triangular inequality. However, for a large class of distances d , the orbit distance Δ_α is a metric:

Theorem 1 *If the distance measure $d(Q, S)$ is transformation invariant, i.e.*

$$d(Q, S) = d(T(\alpha)Q, T(\alpha)S)$$

then $\Delta_\alpha(W, P)$ is a metric.

(Proof is omitted due to space limitation.) Moreover, in such a case where $d(Q, S)$ is transformation invariant, it is easy to show that the point-to-orbit distance is equivalent to the orbit-to-orbit distance, namely:

$$\min_\alpha d(Q, T(\alpha)S) = \min_{\alpha, \beta} d(T(\beta)Q, T(\alpha)S)$$

which is an even stronger result, with respect to pattern matching applications.

In this paper we restrict our approach to distances that are transformation invariant. The metric property of the orbit distance is used to apply fast search within the pattern orbit, by exploiting the triangular inequality.

3.1. The Orbit Tree

The transformation group $T(\alpha)$ is a continuous group since the parameter space α forms a continuous domain. In practice, however, the transformation group is approximated by using a discrete group generated by uniformly sampling the parameter space α . For simplicity, and w.l.o.g, assume $T(\alpha)$ is a one parameter continuous group, and let $\{T(\epsilon i)\}$ be the discrete group. Using the discrete group, an approximation of $\Delta_\alpha(W, P)$ is then given by

$$\Delta_\epsilon(W, P) = \min_i \{d(W, T(\epsilon i)P)\}$$

$\Delta_\epsilon(W, P)$ can be calculated naively, by computing $d(W, T(\epsilon i)P)$ for all i . However, since distance computation may be time-consuming, run time can be improved, given that Δ_ϵ is a metric:

Consider, again, the orbit $T(i\epsilon)P$. It may be divided into 2 sub-orbits: $T(2\epsilon i)P$ and $T(2\epsilon i)P'$ where $P' = T(\epsilon)P$. The distance $\Delta_\epsilon(W, P)$ can then be rewritten:

$$\Delta_{T(\epsilon)}(W, P) = \min(\Delta_{2\epsilon}(W, P), \Delta_{2\epsilon}(W, P')) \quad (3.2)$$

However using the fact that $\Delta_{2\epsilon}$ is a metric, the triangular inequality gives:

$$|\Delta_{2\epsilon}(W, P) - \Delta_{2\epsilon}(P, P')| \leq \Delta_{2\epsilon}(W, P')$$

Note, that $\Delta_{2\epsilon}(P, P')$ can be calculated in advance, prior to the actual search. Thus, if the distance $\Delta_{2\epsilon}(W, P)$ is found to be large and the distance $\Delta_{2\epsilon}(P, P')$ is small, we may deduce that $\Delta_{2\epsilon}(W, P')$ is large as well without any actual distance calculations.

In terms of pattern matching, this implies that $|\Delta_{2\epsilon}(W, P) - \Delta_{T(2\epsilon)}(P, P')|$ forms a lower bound on all possible values of $d(W, T(2\epsilon i)P')$. If this lower bound is greater than the pre-defined threshold, these distance measures need not be computed and the patterns associated with this subgroup may be rejected from further computation. Thus, a speed up is obtained by evaluating only half of the distance computations and possibly rejecting 1/2 of the transformation parameters.

This process can be further applied recursively: in order to compute $\Delta_{2\epsilon}(W, P)$, the orbit $\{T(2\epsilon i)P\}$ can be divided into 2 sub-sub-orbits: $\{T(4\epsilon i)P\}$ and $\{T(4\epsilon i)P''\}$ where $P'' = T(2\epsilon)P$. These orbits can be further sub divided and the process repeated until an orbit is obtained containing a single point. These subdivisions of the original orbit can be described in a tree structure as shown in Figure 2 for the case of a transformation group $T(\epsilon i)$ with 32 elements ($i = 0 \dots 31$).

The Pattern Matching process traverses the tree bottom-up, computing lower and upper bounds on the true distances between image window and sub-orbits of transformed patterns. Branches of the tree are pruned based on the computed bounds. The pruned branches represent distance computations which need not be computed.

An example of the matching process is shown in Figures 2. The pattern matching was applied to estimate the distance between a 20×20 image window and a 20×20 pattern under any 2D rotation about the center by an angle equal to a multiple of $360/32$ degrees. For practical purposes, the distance metric between two images was chosen as the L_2 norm. The threshold was set to 100. Figure 2 shows a portion of the traversed orbit tree. Values on the tree nodes are the computed lower and upper bounds. The encircled value denotes a distance actually computed, all other values were deduced by propagating the bounds along

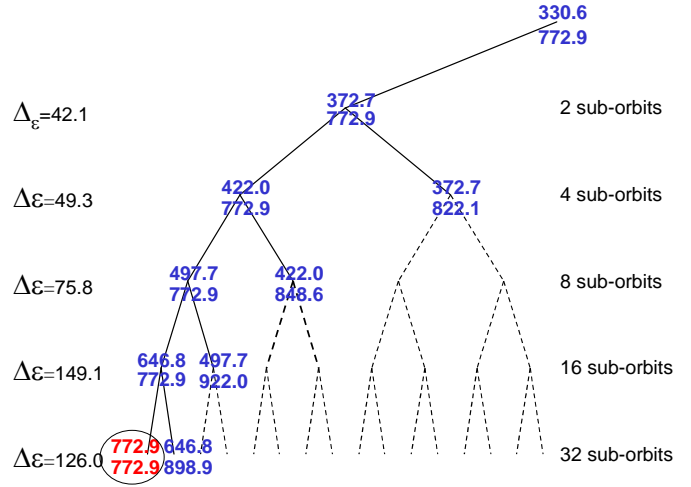


Fig. 2. Example of the pattern matching process using the orbit tree (see text).

the tree branches. Δ_ϵ values represent the orbit-to-orbit distances between nodes at a given tree level. The number of sub-orbits at each level is shown on the right. In this example a single distance evaluation was required to determine that the image window is not similar to the pattern under any of the possible transformations. Note that the lower bound values, although decreasing when ascending the tree, are always above the threshold 100.

4. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed algorithm, the pattern matching scheme was used to search in a large image for a pattern under any 2D rotation.

Figure 3a shows the original 256×256 image. Figure 3b shows a scaled version of the 20×20 pattern. The 2D rotation group was sampled in 32 steps of equal rotation angle. In the original image, several rotated patterns were planted in various locations. All image windows were compared with the pattern under any of the rotation transformations using the proposed scheme. Figure 3c shows the state of the process after a single distance computation per window. For many of the windows this computation was enough to determine the final outcome of pattern matching; black pixels represent those windows for which the process terminated with a negative result, red squares represent windows for which the process terminated successfully (i.e. the pattern was found in the window) and yellow pixels represent windows that can not yet be classified on which the process must continue. Figures 3c-f, show the state of the process for every image window after 1, 2, 4 and 8 distance calculations. The percentage of windows that were rejected are

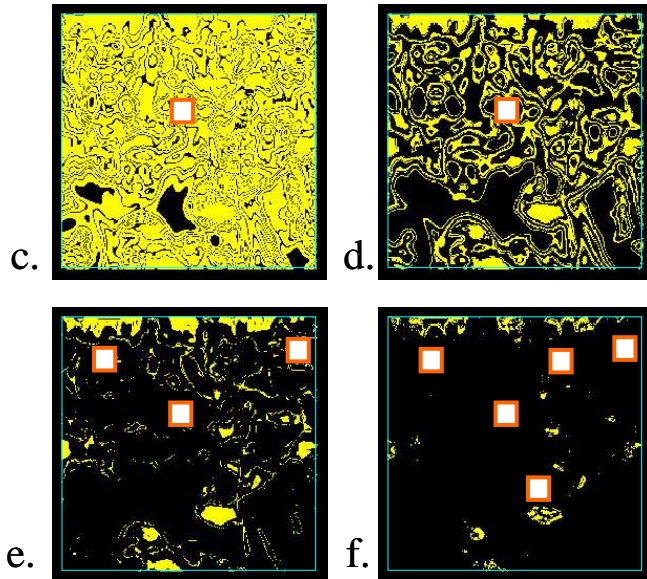


Fig. 3. Pattern Matching on a 256×256 image. a) Image. b) Scaled pattern c-f) State of the process after 1, 2, 4 and 8 distance calculations. The percentage of windows that were rejected are 20%, 68%, 91% and 97% respectively.

20%, 68%, 91% and 97% respectively. The pattern appearances in the image were found successfully after at most 6 distance calculations for every window.

Figure 4 plots the percentage of remaining windows for which the process has not yet terminated, as a function of the number of distance calculations performed. Both Figure 3 and Figure 4 show that a very large portion of image windows require very few distance computations. For this example, the average number of distance computations per pixel is 2.868 (compare with 32 computation per pixel using the naive approach).

5. CONCLUSION

A fast Pattern matching technique was presented, which can be applied when the distance measure is transformation invariant. The technique uses recursive decomposition of the pattern orbit, exploiting the fact that orbit distance is a met-

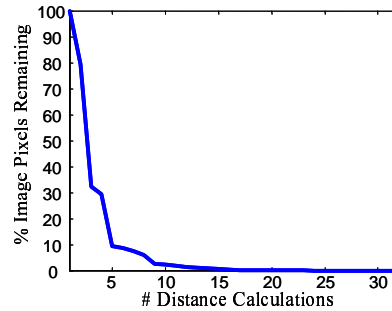


Fig. 4. The percentage of remaining windows a function of the number of distance calculations performed. The average number of distance computations per pixel is 2.868.

ric. The suggested method can be applied in metric spaces as well.

6. REFERENCES

- [1] S. Amari, *Feature spaces which admit and detect invariant signal transformations*, IJCP (Kyoto), 1978, pp. 452–456.
- [2] S. Baker, S. K. Nayar, and H. Murase, *Parametric feature detection*, IJCV **27** (1998), 27–50.
- [3] B. Girod, *Whats wrong with mean-squared error?*, Digital Images and Human Vision (A.B. Watson, ed.), MIT press, 1993, pp. 207–220.
- [4] Y. Hel-Or and H. Hel-Or, *Real time pattern matching using projection kernels*, Tech. Report CS-2002-1, The Interdisciplinary Center, 2002.
- [5] Y. Hel-Or and P. C. Teo, *A common framework for steerability, motion estimation, and invariant feature detection*, Tech. Report STAN-CS-TN-96-28, Stanford Univ., 1996.
- [6] M. Hu, *Pattern recognition by moment invariants*, Proc. of the IRE **49** (1961), 1428.
- [7] B. Kruskal and M. Wish, *Multidimensional scaling*, Sage Publications, 1978.
- [8] J. Mundy and A. Zisserman, *Geometric invariance in computer vision*, MIT Press, 1992.
- [9] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio, *Pedestrian detection using wavelet templates*, Proc. CVPR (Puerto Rico), 1997, pp. 16–20.
- [10] R. Russel and P. Sinha, *Perceptually-based comparison of image similarity metrics*, Tech. Report AI Memo 2001-14, MIT A.I. Lab., 2001.
- [11] S. Santini and R. Jain, *Similarity measures*, IEEE T-PAMI **21** (1999), no. 9, 871–883.
- [12] M. Turk and A. Pentland, *Eigenfaces for recognition*, Journal of Neuroscience **3** (1991), no. 1, 71–86.
- [13] I. Weiss, *Geometric invariants and object recognition*, IJCV **10** (1993), no. 3, 207–231.