Dynamic Footprints of Dots

Max Dupenois

13/11/2009

With regard to the dynamics of dot patterns there a few important considerations. This document intends to provide a summary of these considerations.

Firstly we need to ascertain what we mean by dynamics. Commonly this term is used to describe the process of addition and deletion of dots. Movement of dots would be denoted by the term kinetics. However, as any algorithm that can deal with dynamics can deal with movement by treating it as a special case of addition and deletion, we can subsume kinetics into dynamics and use dynamics to cover both terms. Although preferably we should be able to treat them differently if doing so is more efficient.

Any algorithm which allows for dynamics focuses on one or more of the following aspects:

- **Approximation:** Allowing some margin of error before recomputing the footprint. This is particularly useful for dynamic footprints as they are vaguely defined. As a result we are not so much giving a margin of error as stating that the footprint is still valid within a given threshold.
- **Data Structures:** Some of the algorithms for dynamic convex hulls have concerned themselves with organising the dots into tree structures and focusing on methods for updating the structure. This seems to be easier when only focusing on additions as with additions only dots on the hull need to be stored.
- **Partial Updating:** This can be seen to apply both to tree updates and geometry based updates. The idea is that when the footprint is no longer valid can you update only the part of it which is invalid as opposed to recomputing it in its entirety.
- **Efficiency of update:** Obviously connected with the idea of partial updates this is simply a general approach to using the data already obtained to reduce the cost of updates. For example maintaining data such as distances between static dots.
- **Validity Checking:** This is the essence of any dynamic algorithm, at what point is it decided that a footprint is no longer valid. With the convex hull there are some mathematical certainties that can be used as 'certificates', if these fail then the hull is no longer valid. For footprints in general there are no such certainties so any validity threshold is arbitrary.

Looking at data structures appeared to be a sensible first step. It seemed that a sensible approach to maintaining the dot locations would allow for fast

data checks, such as distances between dots and extremes in any direction. However the problem with such structures is that they tend to only be useful for one or two checks, proving to be a hindrance in others. [Comment:Needs to be explained in greater detail]

Validity checking and approximation has yielded the best results in terms of direction for producing an algorithm.

1. Using an algorithm A compute the footprint of dot pattern D.
2. Record data on information relevant to D regardless of the footprint. This data will be values for descriptors, some that are modified versions of 2D shape descriptors like elongation, convexity, rectangularity, rectilinearity, sigmoidality, circularity etc. and some focusing on the dot pattern values such as standard deviation from the centroid.
3. While the algorithm is running check these descriptors.
4. If total change goes past a threshold value rerun A.

What's important here is we are no longer checking the relation between the footprint and the dot pattern but the total change of the dot pattern itself. This is something we can have as definite quantitative values and thereby have strong definable thresholds on which to base updates. It even allows for internal optimisation; if one descriptor is changing infrequently, or is having little effect on total change we can reduce the regularity with which it's checked. The difficulty comes in choosing which descriptors to use and how to set an appropriate threshold change, the latter will probably become clearer after having run some test cases, the former is a little more complex. It is clearly important that the descriptor can actually be converted to run over the dot pattern instead of over a shape, but just as important is the efficiency loss in checking each descriptor. For example should the descriptor requires you to produce the convex hull each time it's checked it may not be worth running often or at all.

So the dot pattern descriptors we wish to use should probably not be based on any notion of using boundary points as to find those we'd simply be redrawing the footprint anyway. Ellipticity can be worked out from the invariant moments, however I do not know how this applies to dot patterns. The same applies to rectangularity and triangularity.
[Comment:http://users.cs.cf.ac.uk/Paul.Rosin/resources/papers/icpr2.pdf
http://en.wikipedia.org/wiki/Image_moment]

Rectilinearity is judged on the interior angles belonging to the set $\{\pi/2, 3\pi/2\}$, as such it makes little sense to apply it to footprints in generality.

The definition of convexity is give as

Definition 1. *For a given planar shape S its convexity measure $C(S)$ is defined to be the probability that for randomly chosen points A and B from S all points from the line segment $[AB]$ also belong to S, under the assumption that A and B are chosen uniformly.*

While this obviously makes most sense when applied to points on the boundary there is possibly a way to apply this to dot patterns, just as a suggestion:

Theorem 1. *For a given dot pattern D its convexity measure $C(D)$ is defined to be the probability that for randomly chosen points A and B from D the line segment $[AB]$ coincides with a dot also belonging to D, under the assumption that A and B are chosen uniformly.*

While not a particularly accurate measure it shows that there is a possibility for convexity to be ascertained without finding the boundary first.

There does seem to be something in adapting shape descriptors, however, the majority of them seem to require the boundary (unsurprisingly). As a result of this requirement it is difficult to see how to easily modify them.

The centroid is a far easier to work with descriptor, as is the standard distribution from the centroid. Other operations on dot patterns are:

- Median point on the each axis.
- As well as the centroid, the mean of the axes individually.
- Population density – Although choosing the area over which to do this may be difficult.
- Distances
  - Longest
  - Shortest
  - Average

References