

Virtual Parties

Philipp K. Janert
(Dated: March 2006)

We develop a simple model to simulate the behavior of guests at a party. The dynamics of the system is determined by the guests' efforts to place themselves at an "ideal" distance from each of the other guests.

A VIRTUAL PARTY

If we want to give a party, we may ask ourselves how it's going to go. Given who else will be coming, is inviting John a good idea? Will a party with both Jane and Jill attending be a disaster? Do all the guests have something in common, or will the party break up into a set of separate clusters?

Having a computer by our side, we it seems natural to turn to the neutral grey box for an answer. Is it possible to simulate the outcome of the planned get-together? Of course it is — as long as we don't take the results too seriously!

THE MODEL

To model the interaction between the guests at the virtual party, we assume that each guest has a favorite distance to any other guest and will move in such a way as to be as close to his or her optimal spot as possible.

To translate this into practical terms, we define for each person the sum over all deviations from the ideal separation to all the other party guests as the "measure of unhappiness". Each guest will now move in such a way as to minimize his "unhappiness".

The distance between any two people (located at (x_i, y_i) and (x_j, y_j) , resp.) is given by the Pythagorean Theorem:

$$d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

The total unhappiness \mathcal{U} of the i th guest is then given by his unhappiness with respect to all *other* guests j :

$$\mathcal{U}_i = \sum_j |d_{i,j} - d_{i,j}^{\text{ideal}}|.$$

Here, $d_{i,j}$ is the actual distance between guest i and guest j , while $d_{i,j}^{\text{ideal}}$ is the distance that guest i would like to have to guest j . Note that in general this will *not* be the desired distance that j would like to have to i — this is what makes planning parties so difficult!

In the expression for the total unhappiness above, we have summed over the absolute value (denoted by the vertical bars: $|\cdot|$) of the deviations. The absolute value of a quantity is always positive (i.e. $|-1| = 1$), since in our model we assume our unhappiness to be equally

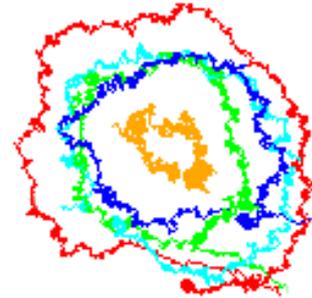


FIG. 1: Guests mingle at a Virtual Party

strong, no matter whether we are too close, or too far away from each other.

To simulate our virtual party, we need to read in the ideal distances for all of our guests. Then, at each time-step we calculate the total unhappiness for each guest, and determine the direction in which his or her unhappiness would diminish the most. Finally, all guests take a step in their optimal direction and the process starts over.

Finding the optimal direction may seem daunting at first. A naive approach would have us evaluate the unhappiness on several points which are "one step" away from each person's current position to find the next optimal spot. However, it turns out that all we need is the unhappiness at two additional points, to find the optimal direction.

We can measure the rate of change of the unhappiness function, by evaluating the unhappiness where we are, and at another point, a little bit away from our current position:

$$\mathcal{U}'(r) = \frac{\mathcal{U}(r + \epsilon) - \mathcal{U}(r)}{\epsilon}.$$

Here, \mathcal{U}' is the rate of change and ϵ is a small distance.

We can evaluate the rate of change along the x-axis and along the y-axis independently by taking a little step in x- or y-direction, respectively, denoting them \mathcal{U}_x and \mathcal{U}_y . The quantity $(\mathcal{U}_x, \mathcal{U}_y)$ (also known as the *gradient*) then points *uphill* in the direction of the greatest overall change. Since we want to go downhill, we need to make

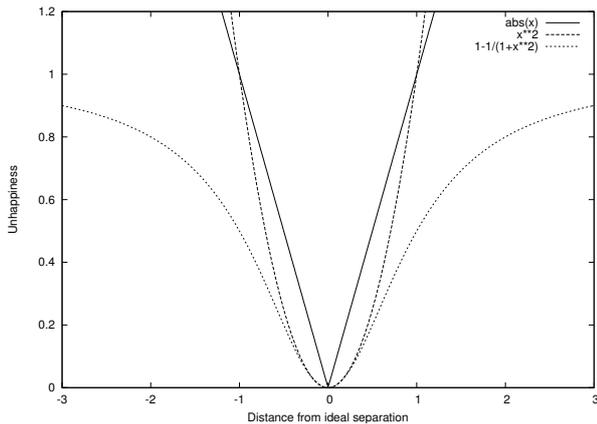


FIG. 2: Different unhappiness functions that can be used (Unhappiness vs. Distance from ideal separation)

a step in opposite direction: $(-U_x, -U_y)$ (note the minus signs). Moreover, the gradient does not only give the optimal direction, but by itself is also a measure for the *size* of the step to take. To find the new position, we can simply subtract the gradient from our current position. (To take smaller or larger steps, multiply the gradient by a number smaller or larger than one — creating more of a ballroom or a disco atmosphere in the room!)

THE MODEL, IMPROVED

The model as described above basically works as expected. However, when running it, we will quickly notice a number of minor unrealistic features.

The most disturbing is a tendency for guests to perform “jerky” movements when close to their ideal distance. The reason for this is the “kink” of the absolute value function at the origin. If guests are a little too far from each other (i.e. to the right of the kink), the gradient points to the left. Once they take this step and now are a little too close, the gradient points to the left *with the same magnitude*, and so guests may perform a jittery movement, back and forth. What we need is another function which is smooth close to the ideal distance, so that guests *slow down* when they are close to their optimal separation.

The simplest function which is positive both to the left and to the right of the origin, and which is smooth everywhere is the square: x^2 . However, using this has problems of its own: the square changes very strongly away from the origin, but only very little close to zero, with the effect that guests which are far from their ideal distance are seen “rushing” to improve their position, but then slowing down to a crawl once they approach it. Not realistic either!

What we need, therefore, is a function which has a minimum near zero, is smooth everywhere, but becomes

level further away from the origin. The most elementary such function is

$$f(x) = 1 - \frac{1}{1 + x^2},$$

but many other examples exist, for instance $1 - \exp(-x^2)$, or, most slickly, $(\tanh x)^2$.

The other improvement we may want to make is to introduce a little bit of randomization into the movement of our guests: at each step, guests move in their optimal distance, but also perform a little step in a random direction as well. This helps to avoid situations where guests are stuck, for instance in a corner of the room. The random step helps break the deadlock that sometimes arises out of some artificial symmetry.

With these improvements in place, the model performs quite satisfactorily.

IMPLEMENTATION NOTES

The model implementation uses PyQt to visualize the movement of the guests at the party. First, we read a file with the ideal distances in matrix format (tab-separated) from standard input. The last entry on each line gives the color to use for the guest.

Then, we do a little trickery of transforming the matrix of ideal distances (keyed by the integer index for each guest) into a hash-map (keyed directly by the object of the other guest — matrix notation is great in mathematical formulas, but a pain to get right in a program. Luckily, in this day and age, richer data structures are readily available and we should make use of them when we can!)

Not that I have chosen to let the `Person` class inherit from the `QCanvasItem` which is used to represent it graphically (instead of embedding it). In a larger program than this, I would opt for a more flexible implementation with weaker coupling between “view” and “model”.

One important, but subtle wrinkle is the use of the `scale` factor in the `Person` class. It provides a measure for the length scale of the room: we measure the position of each person in the room by their (pixel) position on the screen. On the other hand, the Unhappiness-function changes significantly over a range from $[-1 \dots 1]$, so we divide each person’s pixel position by the total size of the room (also in pixels) to *scale* it to the effective range of the Unhappiness-function. Note that this only sets the “scale” of the room (if we choose a bigger display window, the behavior will be the same), but may require fine-tuning depending on “scale” over which the function changes: for instance, when choosing $\tanh(x^2)$ instead of $1 - 1/(1 + x^2)$, we will need to adjust the scale by an additional factor of $2 \dots 5$, since \tanh varies much more rapidly (exponentially, in fact) than the rational function (which varies only according to a power-law).

UNDERSTANDING THE MODEL

When trying out different party scenarios with this model, we will quickly find that people in this model behave differently than physical particles would. This should not be too surprising, since we built the model specifically to capture a different kind of system, but it is instructive to look at some of the differences.

One simple scenario we may study is the “I love you - I hate you” encounter, which one might also call the “Blind date from hell”. Its interaction matrix looks like this:

- 40 red
10 - blue

In other words, “blue” really, really wants to be close to “red”, but “red” does not share this affection. The result: after approaching each other initially, “blue” ends up chasing “red” all over the screen.

Physical particles typically do not behave that way: either they both attract or repel each other, a fact which is summarized by Newton’s Third Law “action equals reaction”. This law expresses the conservation of overall momentum, which itself is rooted in a deep symmetry of our world, namely “translation invariance” — the physical laws are the same here, on the other side of the street, and half-across the solar system.

The important insight here is that for a specific problem, we can set up an ad-hoc model easily, and it may even be relevant for the system we want to study. However, a model derived in this way will most likely not have the kind of inherent consistency which we can find in the laws describing our physical world.

QUESTIONS AND ALTERNATIVE

There are many ways that one might want to improve this model. First of all, the current set-up does not allow people to be “neutral” with respect to one another — everyone *must* have a preferred distance to each of the other guests. One might want to relax this constraint, skipping those guests for which no preferred distance has been set when calculating the Unhappiness.

One can also introduce more complicated interactions into the calculation of the Unhappiness. For instance,

when I am deeply engaged (at my preferred distance) in a conversation with another guest — will I still be looking out for other guests I would want to chat with? In the current model, this is the case, and occasionally you see an entire clump of people following a lone stranger hogging the wall, just because one of the guests would like engage this person in a conversation. One could think of a “saturation” mechanism, which reduces a guest’s desire for contact with other guests, when at a preferred distance to at least one of the guests. The possibilities are endless. (Technically, this amounts to the introduction of “many-body” interactions, since the interaction between A and C, say, now depends on the interaction between A and B.)

Lastly, A. K. Dewdney points out that if you plug in the parameters from a system which is known to have a stable equilibrium, this model will recover the optimal configuration. The example he gives consists of the largest cities in the US. Enter their *actual* distance as their *preferred* distance, and they will arrange themselves in a configuration resembling a map of the United States (possibly upside down or otherwise distorted).

FURTHER READING

The idea for this problem was taken from A. K. Dewdney’s Computer Recreations column in *Scientific American* (Sept. 87). This column, together with many others has been reprinted in book form: **The Magic Machine** by A. K. Dewdney (Freeman and Company, 1990).

Although the example presented here is (intentionally) whimsical, modeling and simulating the behavior of people can be serious business — for instance, to design emergency escape routes from crowded structures, such as airplanes. An example of such work (together with additional references) can be found in the contribution by Michael Schreckengerger et al. in **Computational Statistical Physics**, edited by K. H. Hoffmann and M. Schreiber (Springer, 2002).

Finally, there is a branch of psychology and sociology called *proxemics* that deals with the inter-personal distances and the cultural differences of what is “personal space”. A search on the web will turn up numerous related pointers.

© 2006 by Philipp K. Janert. All rights reserved.