CS-BIGS 6(1):10-16

http://www.csbigs.fr

Crossing the street in Saigon

Xavier Bressaud
Université Paul Sabatier - Toulouse III
Institut de Mathématiques de Toulouse

We present a simple agent-based model that aims to reflect the strategies of pedestrians trying to cross the street in Ho Chi Minh City.

Keywords : *Agent-based model. Probabilistic model*

The density of motorbikes in Ho Chi Minh city is striking. Crossing the street as a pedestrian is an experience shared by most visitors. It might be dangerous when there is a traffic light, because motorbikes turning right do not always stop. But crossing the amazing stream of motorbikes on a section with no traffic light, for instance on Ton Duc Thang street, along the Sai Gon River, seems really out of reach. Although there are crosswalks, there are no traffic lights and the flux of motorbikes seems to be —and indeed is— continuous. However, the average Joe Tourist will learn, from a Vietnamese friend or from observation, that it is possible to reach the other side safely (in finite time). To make it simple, the strategy is: start carefully and keep constant speed! The speed should not be too high (do not run) and, in any case, do not panic: do not go backward, do not change speed; be predictable. It makes sense and it works; but the experience is impressive: the flux of motorbikes seems to be alive and surrounds you fluently until you reach the other side, the river bank.

Nothing magic. The drivers see you, evaluate your speed and anticipate your future position. If they believe there is a risk of collision, they adapt their direction to pass before or behind you. We propose here a basic agent-based model accounting for this phenomenon.

The street is represented by a strip $S = \mathbb{R} \times [0, 1]$. We consider that it is one-way: say the traffic flows with positive speed. Our pedestrian wants to cross vertically at the origin, i.e. go from point $(0, 0)$ to $(0, 1)$.

We assume that the flux of motorbikes is Poissonian in the following sense: at time 0, we choose the positions and speeds of all motorbikes according to a Poisson measure on $S \times \mathbb{R}_+$ of intensity $I \times$ Lebesgue (on S)$\times$ the speed distribution (say Lebesgue measure on $[0, v_{max}]$). In fact we are interested only in those motorbikes with negative x coordinate (those which may hurt our pedestrian). We index them starting from the rightmost: Let (x_0, y_0, v_0) be the position and speed of the rightmost negative mark of the Poisson measure and define recursively (x_n, y_n, v_n) the positions/speeds of the other marks in such way that for all integers n, $x_{n+1} < x_n$. From time 0 on, the dynamics of the motorbikes is deterministic.

1. Free Flow

In a first version —free flow— each motorbike flows according to its speed. The dynamics is driven by the equations: $x_n(t) = x_n(0) + tv_n, y_n = y_0$.
In these conditions, as we will see, the pedestrian who tries to cross at constant (vertical) speed $V > 0$ takes some risks. Its position is $x(t) = 0$, $y(t) = tV$ at time t.
Assume there is a collision if a motorbikes passes at distance less than some fixed scale parameter $s > 0$ from the pedestrian, i.e. if for some motorbike n, at time when it crosses the pedestrian line $x = 0$, we have $|y_n(t) - y(t)| < s$.
Then, we can compute the probability of a collision. The time at which the motorbike indexed by n reaches the pedestrian line is obviously $t = -x_n(0)/v_n$ (recall $x_n(0) < 0$). So that there may be a collision with motorbike n only on the event

$$A_n = \left| y_n(0) + V\frac{x_n(0)}{v_n} \right| < s.$$

There is a collision if one of the motorbikes produces it (the collision will really occur only with the first one, i.e. that with the smallest $y_n(0)$), hence, on the event

$$A = \cup_{n \geq 0} A_n.$$

In other words, there is no collision if there is no mark in the subset of $\mathbb{R}^3$ defined by

$$D = \left\{ (x, y, v) \in S \times [0, v_{max}] : |y + V\frac{x}{v}| < s \right\}.$$

The probability that there is no mark in D is given by the Poisson intensity of D; hence the probability of a collision is given by:

$$P = 1 - e^{-I.L},$$

where L is the Lebesgue measure of the domain D.
This dangerous situation is illustrated by movie 1.

2. Intelligent drivers

Now we introduce the safe strategy of the agents to avoid collision. We assume that when they are at a distance say d from the pedestrian line, they estimate the position of the pedestrian at the moment (still $-x_n(0)/v_n$) they will reach the pedestrian line; for this estimation, they use the observed present speed of the pedestrian. If ever they observe that there is a prospect of a collision (that is if $|y_n(t) + Vx_n(0)/v_n| < s$) they give an impulsion to their "vertical" speed: at such times, their dynamics becomes:

$$x_n(t) = x_n(0) + tv_n, \; \frac{dy_n}{dt} = \pm\epsilon,$$

where $\pm$ is the sign of $y_n(t) + Vx_n(0)/v_n$ (it has probability 0 to be exactly 0).
It is clear that if distance d and speed ϵ are large enough compared to scale s, the motorbike will not stay in the dangerous zone — provided the pedestrian speed remains constant. Let t_d be the time when the bike is at distance d. The computation is $|y_n(t) + Vx_n(0)/v_n| = |y_n(t_d) + Vx_n(0)/v_n| + \epsilon(t - t_d)$ which becomes greater than s before $t - t_d = s/\epsilon$. It is fine if this happens before the time needed to run distance d, i.e. $t_d + d/v_n$.
Now we only need to ask the pedestrian to choose correctly its starting time : he must ensure that no motorbike is actually in such a position that it will not have enough time to avoid him. To be sure, it suffices to ask him to start at a time when there is no vehicle in the (small) domain:

$$\Sigma = [-s.v_{max}/\epsilon, 0] \times [0, s].$$

It appears that, provided v_{max} is given, we can set the parameters so that collision occurs with probability 0. A safe world.
This situation is illustrated by movie 2.

3. A dangerous tourist

It is clear that this strategy is elaborated for pedestrians who *know* and *trust*: hence they

keep constant speed and no collision should occur.
Imagine a recently arrived tourist wishing to see the riverfront coming out from the Majestic Hotel. If he is adventurous enough to start to cross but tries to adapt his speed to his intuition —we archly decided to model this by random changes in the speed— he might be in a dangerous situation as shown by movie 3.
The risk of collision certainly highly depends on the bad strategy —the worst would be to accept to go backwards— adopted by the tourist and we do not presume to perform any computation for this case.

4. Reality

As you may have observed from the movies, to make the simulations more realistic, we decided to allocate some space to the motorbikes, too, and hence they should not overlap (this would yield another type of accident which, indeed, happens). Mathematically, this changes the initial measure because we condition by *no overlap*. Moreover, we had to introduce in the strategies a constraint that forbids overlapping. When the density is high, this changes the behavior of the model because some motorbikes cannot follow the *safe* strategy and finally hurt the pedestrian.
In real Saigon, we observe that there are also cars. But rather few compared to the number of cyclos. They usually do not care about pedestrians and our tourist should take this fact into account.
When I tried to apply the strategy with closed eyes, I was reminded by my Vietnamese friend that, although they are rare, there exist crazy or inattentive drivers and that I should be more careful. The worst case is that of drivers going the wrong way for whom it is dangerous to deviate trajectory!
Other exceptional events may happen and popular wisdom says that the good thing to do if you really feel that things turn the wrong way is to stop (do not go backward): after a short while your position becomes easily (and precisely) predictable and the motorbikes will avoid you. Choose carefully your moment to start again.
Usually tourists move by groups. This observation makes things even more complicated because, then, a good synchronization is required. Groups may also turn to be an advantage : it is common to see tourists looking for the *protection* of a local crosser.
An element is definitely missing in this model attempt: the noise made by horns of all these vehicles which is a part of the experience.
Movies 4 and 5 show up real situations on Ton Duc Thang street, as seen from the Majectic Hotel terrace and from the pedestrian (and author) point of view, respectively.
Let us conclude by a remark that expresses the limit of mathematic modeling : any serious economist knows that the best strategy is to buy a motorbike !

Majestic Hotel, August 1st, 2014.

5. Simulations

We have described the mathematical model we have in mind. Let us now propose a simple implementation of a concrete simulation based on this model. We propose a discrete (time and space) version in a finite piece of space, around the pedestrian.
To simulate the Poisson measure, we use a non stationary procedure : we start with no motorbike. Then, at each discrete time step, the nth motorbike appears at the left hand (`av(n) = 3`) bound with a given probability (`densite`) ; its vertical position (`pos(n)`) and speed (`sp(n)`) are choosen randomly uniformly on a suitable interval. If, from then on, the motorbikes follow the prescribed dynamics, the steady state should be a reasonable approximation of that obtained with a Poisson measure.

```
naissance = rand;
if naissance<densite
    nbmob=nbmob+1;
    av(nbmob)=3;
    pos(nbmob)=bord + rand*largeur;
    sp(nbmob)=vmin + rand*(vmax−vmin);
    drift (nbmob)=0;
end
```

Now, while the dynamics for the motorbikes runs, we insert the pedestrian. It starts at the bottom (`py=0`). Before he reaches the road (`py < bord`) itself, he can decide to stop if safety conditions are not fulfilled, i.e. if he thinks that some motorbikes may not have time to avoid him.

```
if py<bord
    pspeed = pietonspeed;
    for mob=1:nbmob
        if av(mob) < px
        big= taille *vmax/epsilon
            if px−av(mob) < big
                if pos(mob)−py< taille
                    pspeed=0;
                end
            end
        end
    end
end
```

Next step is, for each motorbike on the left-hand side, to check if its trajectory may meet the pedestrian and give a drift (`epsilon`) to its vertical speed if needed. This part could be erased to check the behavior of the model in its free flow version.

```
for mob=1:nbmob
    drift (mob)=0;
    if av(mob)<px
        if av(mob)>px−d
            t = (px − av(mob))/sp(mob);
            pieton= py + t*pspeed ;
            choc=abs(pos(mob)−pieton);
            dd = sign(pos(mob)−pieton);
            if choc < taille
                drift (mob) = epsilon*dd;
            end
        end
    end
end
```

Then, we check for an accident, i.e. a too short distance between the pedestrian and one of the motorbikes. In which case we plot the pedestrian in red and pause the algorithm.

```
for mob=1:nbmob
    A=(px − av(mob))^2;
    B=(py−pos(mob))^2;
    if A + 4*B < .2* taille ^2
        plot(px, py,'ro' ,'MarkerSize', taille )
        hold on
        pause
    end
end
```

Before plotting, we erase motorbikes arriving out of the screen and check if the pedestrian has reached the other boundary of the street.

```
if max(av)>longueur−10
    well=av<longueur−10;
    av=av(well);
    pos=pos(well);
    drift=drift(well);
    sp = sp(well);
    nbmob = length(av);
end

if py > largeur+bord
    py=0;
end
```

Finally, we let the motorbikes and the pedestrian change their positions according to their speed. Then, we plot the road and the moving objects at their new positions. Motorbikes are made of three pieces with an effect to emphasize the vertical speed.

```
av = av + sp;
pos = pos + drift;

clf
plot([0 300],[1 1], 'b')
hold on
plot([0 300],[bord bord],'r')
hold on
plot ([0,300],[ largeur+bord,largeur+bord],'r')
hold on
plot ([0,300],[ largeur+2*bord,largeur+2*bord],'b')
hold on

plot(av−3, pos,'bs')
plot(av, pos+drift,'bo')
plot(av+3, pos+2*drift,'b>')

plot(px, py,'go','MarkerSize', taille )
hold on

drawnow
```

We can insert two other optional parts. The first one makes the simulation more realistic: it gives a drift to motorbikes that come too close to each other; however this effect may enter in conflict with the anticipation of the pedestrian position and yield accidents.

```
for i=1:nbmob
    for j=1:i-1
        dij = av(j) - av(i);
        if abs(dij) < 10
            eij = pos(j) - pos(i);
            if abs(eij) < 3
                m1=i; m2=j;
                if dij>0
                    m2=i; m1=j;
                end
                w=sign(pos(m2)-pos(m1));
                drift(mob2) = .2 * w;
            end
        end
    end
end
```

The other option is that making the pedestrian change speed at random times.

```
vpmin=.01
vpmax=1
teteenlair=rand;
if teteenlair < .1
    pspeed = vpmin + (vpmax-vpmin)*rand
end
```

The complete listing.

```
%%% Definition of variables and parameters
tmax=100000 % number of iterations
bord=10 % lower boundary of the road
largeur = 80 % width of the road
longueur=300 % length of the road
densite=.15 % probability of a new motorbike.
d=100 % anticipation distance
epsilon= .15 % vertical drift to avoid pedestrian
vmin=.5 % minimum speed
vmax=.9 % maximum speed
px=200 % horizontal coordinate of pedestrian
py =1 % initial vertical coordinate of pedestrian
pietonspeed=.3 % vertical speed of pedestrian
taille = 12 % distance to avoid accident

%%% Initialisation
nbmob=0 % number of motorbikes
av=[] % horizontal coordinates of motorbikes
pos=[] % vertical positions of motorbikes
sp=[] % horizontal speed of motorbikes
drift=[] % vertical speed
pspeed = 0 % (vertical) speed of pedestrian.

% graphic stuff
figure(1)
clf
cla

%%% MAIN
for time=1:tmax

%%% New motorbikes ? Are there new
motorbikes arriving at the left hand boundary ?
Yes if random naissance smaller than densite.
Then choose random vertical position and
horizontal speed.

    naissance = rand(1);
    if naissance<densite
        nbmob=nbmob+1;
        av(nbmob)=3;
        pos(nbmob)=bord + rand*largeur;
        sp(nbmob)=vmin + rand*(vmax-vmin);
        drift(nbmob)=0;
    end

%%% Pedestrian waits for good moment to cross

    if py<bord
        pspeed = pietonspeed;
        for mob=1:nbmob
```

```
            if av(mob) < px
            big= taille *vmax/epsilon
                if px−av(mob) < big
                    if pos(mob)−py< taille
                        pspeed=0;
                    end
                end
            end
        end
    end

%%% Anticipate accidents and change trajectory

    for mob=1:nbmob
        drift (mob)=0;
        if av(mob)<px
            if av(mob)>px−d
                t = (px−av(mob))/sp(mob);
                pieton= py + t*pspeed;
                choc=abs(pos(mob)−pieton);
                dd = sign(pos(mob)−pieton);
                if choc < taille
                    drift (mob) = epsilon*dd;
                end
            end
        end
    end

%%% Accident motorbike / pedestrian.

    for mob=1:nbmob
        A=(px − av(mob))^2;
        B=(py−pos(mob))^2;
        if A + 4*B < .2* taille ^2
            plot(px, py,'ro','MarkerSize', taille )
            hold on
            pause
        end
    end


%     %%% Avoid accidents between motorbikes.
%     % If two bikes are too close,
%     % the one behind drifts a bit.
%
%     for i=1:nbmob
%         for j=1:i−1
%             dij = av(j) − av(i);
%             if abs(dij) < 10
%                 eij = pos(j) − pos(i);
%                 if abs( eij ) < 3
%                     m1=i; m2=j;
%                     if dij>0
%                         m2=i; m1=j;
%                     end
%                     w=sign(pos(m2)−pos(m1));
%                     drift (mob2) = .2 * w;
%                 end
%             end
%         end
%     end


%     %%% Variable pedestrian speed
%     % Tourist changes speed randomly
probability .1
%
%     vpmin=.01
%     vpmax=1
%
%     teteenlair =rand(1);
%     if teteenlair < .1
%         pspeed = vpmin + (vpmax−vpmin)*
rand(1)
%     end

%%% Dynamics

    av = av + sp;
    pos = pos + drift;


    py=py+pspeed;

%%% clear motorbikes coming out of screen

    if max(av)>longueur−10
        well=av<longueur−10;
        av=av(well);
        pos=pos(well);
        drift=drift(well);
        sp = sp(well);
        nbmob = length(av);
    end

%%% Mission fullfilled. Start again ...

    if py > largeur+bord
        py=0;
    end

%%% Plot the current situation :

    clf
```

```
    plot([0  300],[1  1],'b')
    hold on
    plot([0  300],[bord bord],'r')
    hold on
    plot ([0,300],[ largeur+bord,largeur+bord],'r')
    hold on
    plot ([0,300],[ largeur+2*bord,largeur+2*bord],
'b')
    hold on

    % Three pieces for the scooter with an "effect
" to emphasize the vertical speed.
    plot(av-3, pos,'bs')
    plot(av, pos+drift,'bo')
    plot(av+3, pos+2*drift,'b>')

    % One circle for the pedestrian :
    plot(px, py,'go','MarkerSize', taille )
    hold on
    drawnow
end
```

Correspondence: bressaud@math.univ-toulouse.fr.