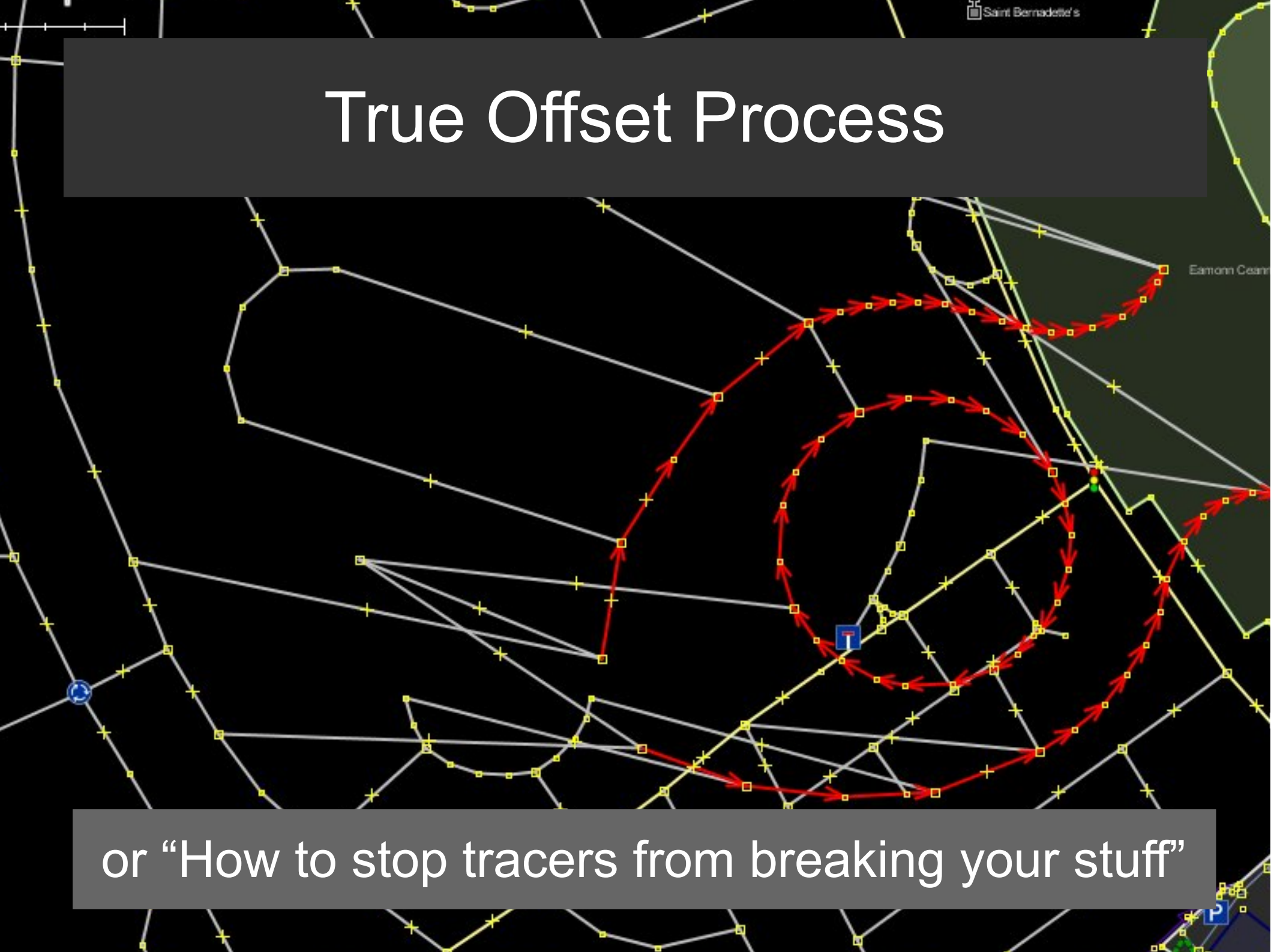


True Offset Process

or “How to stop tracers from breaking your stuff”



Servus!

- Dermot McNally
- Irish mapper
- Co-founder Directski.com
- Co-founder Dobini



Tracing from Aerial Imagery

- Simple
- Can be more accurate than GPS surveys
- Lends itself to large scale mapping sessions
- Popular
- You don't have to go and survey it yourself

Snags of Tracing from Imagery

- You don't have to go and survey it yourself
- Calibration can be off (up to 30m)
- Mappers often (mostly?) don't know editors can correct offsets. So the existing editor functionality isn't being used
- Those who do may not know that there is an offset
- Those who understand can't always quantify the offset
- Those who correct the offset do so manually each time
- The result: inconsistent alignment (surveyed V traced)
- Even a prospect of edit wars

Offset? What Offset?

- Is your local imagery well-aligned? Lucky you!
- On Bing, much imagery at least as accurate as GPS...
- ...but some is off by as much as 30m
- Errors on that scale are really detrimental to the map
- Ideal solution would be to fix back at imagery provider
- But no back channel for that and non-critical for providers
- So checking and correcting offset essential when tracing

The Right Thing (today)

- Realise that imagery can be inaccurate
- Realise that your editor can correct the offset
- Be able to quantify the offset (GPS trace? Existing data?)
- Notice when editing moves to an area of different offset
- Use editor function to adjust imagery (manual, bookmark)
- Editor supports bookmarking? Better consistency
- Watch for clues that offset has changed (new imagery...)

Only one of these steps must fail to break the map

The Right Thing (with True Offset)

- Realise that imagery can be inaccurate (or remain ignorant)
- Realise that your editor can correct the offset (or remain ignorant)
- With luck, somebody else has quantified the offset
- Editor sees when you move to an area of different offset
- Editor looks up offset from True Offset Web-service and adjusts imagery (a better, shared bookmark)
- All mappers use same offsets. Even better consistency
- Users must still watch for clues that offsets have changed – but once one mapper updates True Offset, everybody benefits.

The vision: The editor just fixes the offset without user involvement – overriding this becomes the exception, not the rule

The Science Bit

- Offsets for defined areas are stored “somewhere” (OSM)
- Offset tagging identifies data provider, applicable zoom range and simple x/y error expressed in degrees
- Editors look up a web service querying on a test lat/long (normally centre of edit window)
- Avoid too many lookups by returning a bbox of applicability with all responses – fresh lookups required only when user moves outside this bbox
- Deployed and working on dev server
- http://wiki.openstreetmap.org/wiki/True_Offset_Process

Boo! Get a Haircut!

- It doesn't handle rotation, non-uniform scaling, the effect of big hills etc. (then again, neither do the editors)
- Offsets can change and invalidate our stored bookmark – but that's true of the personal bookmarks we have today and it needs only one mapper to fix True Offset
- Area-based setting of offsets likely to be less flexible than control points in the long run. True Offset web service is agnostic on this point and could support either system or a mix.
- It can never work so there's no point trying. But hey, this is OSM and we tend not to let that problem worry us too much.

A note on Editor Workflow

- In an ideal world, editors would just fix the error silently
- In practice, it may be important to tell users an offset has been applied (or ought to be applied, and request consent)
- Having True Offset support that only works if you know it's there offers very low value – the big gain is that less advanced users will no longer trace unadjusted imagery.

Questions?

dermot@dobini.com