# ABC: Using Object Tracking to Automate Behavioural Coding

**Aitor Apaolaza**
**Robert Haines**
**Amaia Aizpurua**
University of Manchester, UK
[aitor.apaolaza, robert.haines,
amaia.aizpurua]@manchester.ac.uk

**Andy Brown**
**Michael Evans**
**Stephen Jolly**
BBC R&D, Salford
[andy.brown01, michael.evans,
stephen.jolly]@bbc.co.uk

**Simon Harper**
**Caroline Jay**
University of Manchester, UK
[simon.harper,
caroline.jay]@manchester.ac.uk

*CHI'16 Extended Abstracts*, May 07-12, 2016, San Jose, CA, USA.
ACM 978-1-4503-4082-3/16/05.
http://dx.doi.org/10.1145/2851581.2892483

## Abstract

Video data of people interacting with devices contains rich information about human behaviour that can be used to design or improve user experience. As a first step, it must be interpreted – or coded – into a form that can be analyzed systematically. The coding process is currently performed manually, and it can be slow and difficult, and biased by subjectivity. This is particularly problematic when trying to obtain data that should be objective, such as the movements of a user in relation to a device. We describe Automated Behavioural Coding (ABC), an open source object tracking technique designed to log user and device movements, and then output positional data that can be used to model interaction. We validate the technique in a study of dual screen TV viewing, and show that the ABC tool is able to correctly classify the direction of gaze to the TV or tablet up to 95% of the time, in a fraction of the time it takes to capture this data manually.

## Author Keywords

Behavioural coding; object tracking; visual attention; television; reproducible methods.

## ACM Classification Keywords

H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous

## Introduction

Video is a great means of gathering rich data about interaction processes. It enables the capture of movements and expressions, which provide valuable information about the user experience, and how to improve it [1, 9]. To extract meaning from the data, it must be segmented into specified units, which can then be analyzed systematically [3].

Although tools exist to help with video coding [17] and expression analysis [22], this segmentation process – commonly referred to as coding – remains time consuming and difficult. Investigators must watch the video through, often more than once, and at a slow playback speed, whilst making annotations about their observations. It is also a subjective process, and it is therefore necessary for at least two people to participate, and check for observer agreement [18].

Crowdsourcing the coding process has been shown to significantly speed it up, whilst also providing a means of flagging and resolving ambiguities in the scheme or results [19, 7]. Limitations of manual coding remain however – in particular its inability to provide precise, quantitative accounts of movements.

As HCI research moves away from studying desktop interaction, and focuses more on the use of multiple devices in complex environments (and particularly in the wild), videoing behaviour, and being able to generate models from it, is becoming increasingly important. Here we describe our difficulties using a traditional approach to behavioural coding in a media-focused 'living room' environment, and explain how we addressed them using an object-tracking technique.

## Understanding dual-screen TV viewing

This work was motivated by the desire to model how people interact with mobile devices while watching TV. The goal of the work was to understand and categorise the movements and gestures people made in this situation, to support the design of interaction experiences that would anticipate the content people required next on their mobile device, and provide it to them in a streamlined fashion.

There are many items of interest in this situation, including the user's head, hands and possibly other limbs, as well as the devices with which they are interacting. Our initial approach was to develop a complex code book, covering what we believed to be the most important movements made by the participants, and the potential positions of the devices. Examples of interesting movements might be 'looking at TV', 'looking at tablet', 'hand moving towards tablet', 'hand on tablet', 'tablet on table'. We discovered three problems with this approach. Firstly, identifying discrete movements was difficult. If a participant put his/her hand on the side of the tablet, and then moved it to the top of the tablet half a second later, is this one movement or two? Secondly, we were limited in the data we could obtain. For example, the speed of movements could be important, but it was not possible to capture this accurately purely through human observation. Thirdly, movements could occur in parallel. To ensure data are not missed, it is necessary to watch the video for one movement category at a time, increasing the length of the procedure by orders of magnitude [18].

Here we describe a first step towards automating the capture of behavioural data from video so that we can ultimately use it to model interaction in complex scenarios. We adapt an object tracking algorithm to gather positional and rotational data of a user's face, and evaluate this against ground truth data obtained using a manual coding approach. The study shows that our prototype application is able to classify the direction of a person's attention (to the TV, or to a mobile device), with over 90% accuracy. We outline the benefits and limitations of the current approach, with regard to the extent it could replace behavioural coding, and discuss the future applications of the technique, in particular its potential to considerably extend current behavioural coding methods, by automating the capture of data that it is not possible to access through human observation alone.

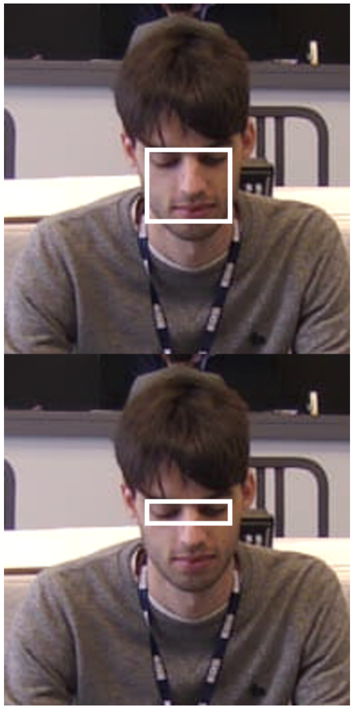

**Figure 1:** Marking out the object of interest with a bounding box: the centre of the face, top; tight to the eyes, bottom.

## Object Tracking

### *Algorithm selection*

The goal of the automated behavioural coding (ABC) process is to gather participant and object movement data from video stream. There are a number of different methods that might be suitable for collecting such data, including specialized models such as facial recognition and kinematics, and more general object tracking systems. If the analysis relies on subtle facial cues then facial recognition software would likely be a good choice; if whole-body movements are of interest, then kinematics might be more suitable. For our prototype application we chose a generic object tracking algorithm as we needed to capture both a participants' movements and their interaction with devices—in this case a TV and a tablet.

### *Tool selection and modification*

We selected `CppMT` [20] as our object tracking tool for the following reasons: it is not specialized for tracking a particular type of object (e.g. faces); the underlying algorithm is robust when tracking deformable objects [21]; it can run in a non-interactive "batch" mode that does not require constant user supervision; it is Open Source and actively developed. We modified `CppMT` to fit into our workflow as follows [16]: we added functionality to output detailed information about the tracked object's position, rotation and size for each frame, and we improved the accuracy by which the object within the scene to be tracked is selected. These modifications have been passed back to the `CppMT` developers for inclusion in the original tool. Additionally, we developed a tool to help us validate the results of the object tracking process—`CppMT-replay` [10]. This combines the original video with its associated object tracking data output from `CppMT`, compositing the tracked object bounding box outline and centre point into the video. Watching the resulting video allows investigators to review the performance of the object tracking step.

### *Performing the object tracking*

The process of tracking an object in a video file is simple: locate the beginning of the experiment in the video file and mark out a bounding box around the object of interest at that point (figure 1); run the object tracking tool (`CppMT`) using the experiment start time and bounding box as inputs; optionally run the resultant tracking data through the validation tool (`CppMT-replay`), to validate the results by eye.

Our modified version of the `CppMT` tool outputs the following data, per frame, in CSV format: *frame number* (from the start of the video file), *frame timestamp* (from the start of the video file, in milliseconds), *bounding box centre point* (in pixels, from the top left corner of the frame), *bounding box width and height* (in pixels), *bounding box rotation* (in degrees, where the rotation of the original bounding box is established at zero degrees), *bounding box vertices* (in pixels, from the top left corner of the frame).

## Attention Classification Study

To evaluate the object tracking accuracy, we used data from a study designed to investigate how people divide their visual attention between the TV and a tablet device, when watching a programme with 'companion content' – additional information, supplied to the mobile device, about what is happening onscreen [4]. Participants watch a 15-minute clip of the BBC nature programme "Autumnwatch" for which companion content – simple quizzes, slides and diagrams – had already been created [5]. Full details of the experiment are described in Brown *et al.* (2014) [6]. The goal of the study was to understand what caused people to direct their attention to either the TV or the tablet, so a first step was to determine, for a given point in time, where the participant was looking. The location of attention was determined by observing the video, and logging when it shifted to TV, or to the tablet. The TV content was presented on 47 inch LG television from a computer via VGA at 1024x720 resolution; this computer also served HTML content to an iPad, updating the content at set times during the clip.

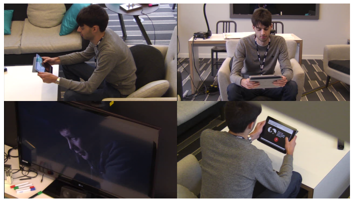

**Figure 2:** Screen capture from the laboratory scene cameras, taken during the pilot. In this analysis, only video from the front facing camera (top right) was used.

The experiment was performed in the BBC usability lab in Salford. This is equipped with several cameras to record activity. In this analysis, only video data from a camera situated immediately above the TV screen, pointing at the participant, was used. The video output from the cameras is shown in Figure 2.

*Manual coding of locus of attention*
Video annotation was used to log the participant's focus of attention at any point during the experiment. The length of time it took to review and annotate the videos varied between participants, with factors such as frequency of attention shifts, and lack of clarity – e.g. users wearing glasses – increasing the time it took. For this study, annotation of 10 minutes of video took a single coder around an hour.

*Automated coding of locus attention*
The manual coding results, which specified the time that attention switched to the TV or tablet (and therefore could be used to determine where a participant's attention was during any given frame) provided a ground truth against which to evaluate the ABC approach.

Object tracking was employed twice for each video. Initially participants' *full face* was designated the object of interest, with the bounding box drawn tightly around the eyes, nose and mouth. Replays of the video using `CppMT-replay` showed that obstructions – e.g. touching the face – could cause problems, so the technique was also tested with a bounding box drawn tightly to the *eyes*. There were also occasional occurrences of tracking problems that occurred over short periods. To determine the effect these tracking problems had, 3 subsets of participants' videos were used for the analysis according to the stability of the tracking: *robust*, when the object tracking showed no visible problem; *noticeable glitches*, where the object tracking is considered to be erratic; and *all users*. The result of the tracking is considered to be robust if the object of interest remains within the bounding box throughout the video, with the centre of the box located over it – i.e. the box covers the object, and is small enough to retain the accuracy of the tracking. Following these criteria, 4 videos were classified as robust, and 5 as containing noticeable glitches.

We used Rpart [24], a library for R to build classification models with recursive partitioning trees. *Bounding box centre point*, *bounding box width and height*, and *bounding box rotation* were selected as features providing information about the participant's face position, orientation, and movement. An additional feature, *vertical position*, is normalised with respect to the lowest and highest position for each participant, so it is not dependent on participant's height, or variations in the camera orientation. The features were synchronised with the manual coding results to train a model to classify participants' focus of attention to the iPad or TV.

*Results*
The classifier was trained using the manual coding data from the first 1, 2, 5 and 10 minutes from the 15-minute long experiment. Accuracy is calculated as the percentage

of correctly predicted values. Tables 1 and 2 present the accuracy results for the “full face" and “eyes" object tracking modes respectively. Where the object tracking worked well (the bounding box remained appropriately positioned throughout) accuracy is high even with just one minute of training from the manually annotated videos. Where the object tracking had periods of instability (there were ‘glitches’, such as the participant touching his/her face, which caused the bounding box to move off the face or change radically in size), prediction accuracy is lower.

Participants’ attention was directed towards the TV for twice as long as it as directed towards the iPad. The precision of predicting attention towards the iPad is calculated as the percentage of valid predictions of iPad attention (true positives) out of all iPad attention predictions (true positives plus false positives). Tables 3 and 4 report these results.Tracking just the eyes improves the results for the videos where the object tracking was not robust. 5 minutes of manually annotated videos are enough to obtain an 80% precision. In the case of videos where the object tracking was found to be robust, 2 minutes of manually annotated video was found to be enough to obtain a high iPad attention prediction rate for both tracking modes.

Results show that the use of this method allows researchers to annotate short periods of time, shortening the workload that annotation of long videos requires. The accuracy of the prediction increases in the cases where the object tracking has been considered to be robust. Validation of the robustness of the tracking is possible through the `CppMT-replay tool`, providing a cost-effective solution to ease manual annotation of participant’s attention focus. All individual results for each video are publicly available, as well as the code employed for the analysis of the tracking output [2].

| **Accuracy** | **1min** | **2min** | **5min** | **10min** |
|---|---|---|---|---|
| **Robust** | 88.909 | 89.071 | 87.109 | 95.310 |
| **Glitches** | 67.139 | 63.877 | 72.223 | 68.128 |
| **All** | 76.815 | 75.074 | 78.839 | 80.209 |

**Table 1:** Full face object tracking. Results of accuracy for attention prediction for TV and iPad.

| **Accuracy** | **1min** | **2min** | **5min** | **10min** |
|---|---|---|---|---|
| **Robust** | 84.852 | 87.511 | 91.042 | 89.132 |
| **Glitches** | 62.608 | 64.099 | 67.833 | 68.906 |
| **All** | 72.494 | 74.504 | 78.148 | 77.895 |

**Table 2:** Eyes object tracking. Results of accuracy for attention prediction for TV and iPad.

| **Precision** | **1min** | **2min** | **5min** | **10min** |
|---|---|---|---|---|
| **Robust** | 89.225 | 88.033 | 92.416 | 93.225 |
| **Glitches** | 51.113 | 79.430 | 63.109 | 75.674 |
| **All** | 68.052 | 83.253 | 76.134 | 83.474 |

**Table 3:** Full face object tracking. Results of precision for attention prediction to the iPad.

| **Precision** | **1min** | **2min** | **5min** | **10min** |
|---|---|---|---|---|
| **Robust** | 77.350 | 89.960 | 92.855 | 93.740 |
| **Glitches** | 69.948 | 76.411 | 81.806 | 82.737 |
| **All** | 73.238 | 82.433 | 86.716 | 87.628 |

**Table 4:** Eyes object tracking. Results of precision for attention prediction to the iPad.

## A Reproducible Method for ABC

One of the highest barriers to reproducibility is replicating the software environments used in the original studies [23]. Making sure that the correct versions of the right software packages are installed on compatible operating systems can be a daunting, maybe even impossible, task, and any differences in experimental setup can have varied and unexpected consequences.

To mitigate against these problems, and to make it easier for ourselves and other investigators to use our automated pipelines, we have used containerization to encapsulate our entire process. We use software containers to wrap up each stage of our method with everything needed for it to run: applications, system tools, system libraries—anything you would normally need to install by hand. This guarantees that our tools will always run the same way, regardless of the environment they are running in, and reduces the number of dependencies required to run them to one—`docker` [8].

The containers that make up our pipeline are as follows: *Video processing*, for any preprocessing required on the input videos; *CppMT*, for the object tracking tools `CppMT` and `CppMT-replay`; *Object tracking*, for initializing and performing the object tracking process; *Analysis*, for the subsequent statistical analysis of the object tracking outputs [15, 11, 13, 12]. We have also written some commandline tools to further simplify the process of running these tools within the containers [14].

## Discussion

We propose the ABC technique, a means of automating certain forms of behavioural coding, thus saving coding time, and allowing the capture of movement data that is not possible using purely observational techniques.

### *Replacing Behavioural Coding*

We evaluate the technique by using the extracted positional data to train a classifier to determine whether a person is looking at the TV or the tablet, and comparing the results with ground truth data obtained through manual coding. Where the object tracking is robust, the model is able to correctly identify the direction of attention over 90% of the time, even with just 5 minutes of training. Nevertheless, depending on the application, this level of error may not be acceptable. We consider these results promising, but future work will focus on identifying where the object tracking breaks down, and finding ways to address this. This may involve improving the algorithm, or flagging when tracking is unreliable, and excluding this data from analysis.

### *Extending Behavioural Coding*

To use ABC to replace ‘traditional’ behavioural coding [3], as we describe in this paper, still requires a degree of manual coding, to provide data with which to train the classifier. We used this evaluation approach to get an idea of the accuracy of the object tracking output data. If it could be used to identify a given behaviour as well as a human, this indicates that is able to provide data that is useful for understanding interaction. A key motivation for ABC, however, is to provide a means of modelling behaviour that is not possible purely through observation. It can potentially be used to quantify the direction and velocity of movements, and the spatial relationships between devices and people, where observational techniques can only place them in coarse-grained categories. It cannot, of course, provide qualitative description, but we anticipate that using ABC to replace some forms of coding, and to extend others, will make it a powerful extension to our toolbox for understanding behaviour in complex scenarios.

## Acknowledgements

We would like to thank the EPSRC and the BBC for their funding contribution to this work via grants EP/K503782/1, EP/H500154/1, and EP/M017133/1.

## References

[1] Nalini Ambady and Robert Rosenthal. 1992. Thin Slices of Expressive behavior as Predictors of Interpersonal Consequences : a Meta-Analysis. *Psychological Bulletin* 111, 2 (1992), 256–274. http://www.tufts.edu/~{}nambad01/Thin%20slices%20of%20expressive%20behavior%20as%20pred

[2] Aitor Apaolaza. 2016. IDInteraction Object Tracking Analysis. (Jan. 2016). DOI:http://dx.doi.org/10.5281/zenodo.44679

[3] Roger Bakeman. 2000. *Handbook of research methods in social and personality psychology*. Cambridge University Press, New York, NY, Chapter Behavioral observation and coding, 138–159.

[4] Andy Brown, Michael Evans, Caroline Jay, Maxine Glancy, Rhianne Jones, and Simon Harper. 2014a. HCI over Multiple Screens. In *CHI ’14 Extended Abstracts on Human Factors in Computing Systems (CHI EA ’14)*. ACM, New York, NY, USA, 665–674. DOI:http://dx.doi.org/10.1145/2559206.2578869

[5] Andy Brown, Caroline Jay, and Simon Harper. 2014b. Eye-tracking the dual-screen experience. In *Transactions of the Web Ergonomics Lab*. University of Manchester. http://dx.doi.org/10.6084/m9.figshare.993993 org / 10. 6084 / m 9. figshare. 993993.

[6] Andy Brown, Caroline Jay, and Simon Harper. 2014c. *Understanding the division of attention between TV and companion content: experiment 2, without eye-tracking.* Technical Report. University of Manchester, School of Computer Science. http://iam-data.cs.manchester.ac.uk/presentations/5

[7] R. Di Salvo, D. Giordano, and I. Kavasidis. 2013. A Crowdsourcing Approach to Support Video Annotation. In *Proceedings of the International Workshop on Video and Image Ground Truth in Computer Vision Applications (VIGTA ’13)*. ACM, New York, NY, USA, Article 8, 6 pages. DOI:http://dx.doi.org/10.1145/2501105.2501113

[8] Docker, Inc. 2016. What is Docker? (Jan. 2016). https://www.docker.com/what-docker

[9] Joshua Hailpern, Karrie Karahalios, James Halle, Laura Dethorne, and Mary-Kelsey Coletto. 2009. A3: HCI Coding Guideline for Research Using Video Annotation to Assess Behavior of Nonverbal Subjects with Computer-Based Intervention. *ACM Trans. Access. Comput.* 2, 2, Article 8 (June 2009), 29 pages. DOI:http://dx.doi.org/10.1145/1530064.1530066

[10] Robert Haines. 2015. CppMT-replay 1.0. (Dec. 2015). DOI:http://dx.doi.org/10.5281/zenodo.44672

[11] Robert Haines. 2016a. IDInteraction CppMT Docker Image 2.0. (Jan. 2016). DOI:http://dx.doi.org/10.5281/zenodo.44675

[12] Robert Haines. 2016b. IDInteraction Object Tracking Analysis Docker Image 1.0. (Feb. 2016). DOI:http://dx.doi.org/10.5281/zenodo.45983

[13] Robert Haines. 2016c. IDInteraction Object Tracking Docker Image 3.0. (Jan. 2016). DOI:http://dx.doi.org/10.5281/zenodo.44677

[14] Robert Haines. 2016d. IDInteraction Object Tracking Tools. (Jan. 2016). DOI:http://dx.doi.org/10.5281/zenodo.44670

[15] Robert Haines. 2016e. IDInteraction Video Processing Docker Image 3.0. (Jan. 2016). DOI:http://dx.doi.org/10.5281/zenodo.44676

[16] Robert Haines, Georg Nebehay, Clemens Korner, and GitHub user ‘trobro’. 2015. CppMT modified for use in IDInteraction. (Sept. 2015). DOI:http://dx.doi.org/10.5281/zenodo.44674

[17] Birgit Hellwig. 2015. ELAN - Linguistic Annotator. (2015). http://www.mpi.nl/corpus/html/elan/

[18] R. E. Heyman, M. F. Lorber, J. M. Eddy, T. West, E. H. T. Reis, and C. M Judd. 2014. *Handbook of research methods in social and personality psychology*. Cambridge University Press, New York, NY, Chapter Behavioral observation and coding, 345–372.

[19] Walter S. Lasecki, Mitchell Gordon, Danai Koutra, Malte F. Jung, Steven P. Dow, and Jeffrey P. Bigham. 2014. Glance: Rapidly Coding Behavioral Video with the Crowd. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 551–562. DOI: http://dx.doi.org/10.1145/2642918.2647367

[20] Georg Nebehay, Robert Haines, Clemens Korner, and GitHub user 'trobro'. 2015. CppMT. (Aug. 2015). https://github.com/gnebehay/CppMT

[21] Georg Nebehay and Roman Pflugfelder. 2015. Clustering of Static-Adaptive Correspondences for Deformable Object Tracking. In *Computer Vision and Pattern Recognition*. IEEE.

[22] Noldus. 2015. Viso. (2015). http://www.noldus.com/viso

[23] Roger D Peng. 2011. Reproducible research in computational science. *Science (New York, Ny)* 334, 6060 (2011), 1226.

[24] Terry Therneau, Beth Atkinson, and Brian Ripley. 2015. *rpart: Recursive Partitioning and Regression Trees*. http://CRAN.R-project.org/package=rpart R package version 4.1-10.