Automatic retrieval of visual continuity errors in movies

Lyndsey Pickup, Andrew Zisserman
Department of Engineering Science, Parks Road, Oxford, OX1 3PJ
{elle,az}@robots.ox.ac.uk

ABSTRACT

Continuity errors occur in many movies and television series, in spite of careful checking by those employed to minimise them. In this work we develop a scheme for automatically detecting these errors and producing a ranked list of the most likely inconsistencies, working from a commercial DVD release. We use the editing structure of the movie to detect pairs of shots within a scene that might have arisen from different takes, and thus are possible candidates for continuity errors. These pairs are then registered and examined for differences that lack an obvious cause – suppressing changes arising from humans and other moving objects by using upper body detectors and trackers. The result is a ranked list of possible continuity errors for the movie.

We show discovered errors for a number of feature length movies including relatively recent releases, such as 'Love Actually' (Curtis, 2003) and classics, such as 'Pretty Woman' (Marshall, 1990). We discover mistakes that have previously been missed in the listings for these movies on such websites as `moviemistakes.com`.

Categories and Subject Descriptors

I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding

General Terms

Algorithms

Keywords

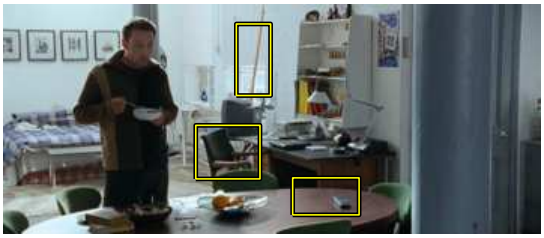
movie, continuity

1. INTRODUCTION

A continuity error, in terms of movies and television, is a lapse in the self-consistency of the scene or story being portrayed. They can be introduced by writers, for example the revisions of the eponymous character's birthdate in the television series 'Buffy the Vampire Slayer', or can come about unintentionally during the filming process.

CIVR '09, July 8-10, 2009 Santorini, GR

(a) End of first candidate shot



(b) Beginning of second candidate shot

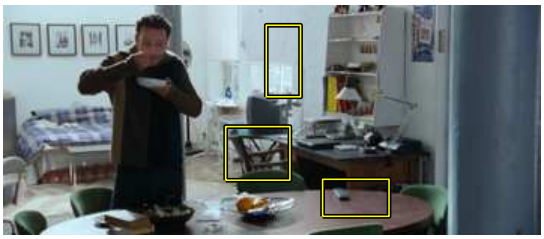


Figure 1: An example of automatically detected continuity errors. (a) The final frame in one shot from the movie 'Love Actually' (frame 74085). (b) The first frame in the next shot that is taken from a similar camera angle (frame 74165). The boxes drawn onto the frames indicate significant unexplained differences in the scene.

We are interested here in continuity errors that arise when shots which are edited together to appear to be continuous in time in fact come from different *takes*, such as those illustrated in Figure 1. Between takes props can move inconsistently, shadows lengthen or shorten, or the level of drink in a glass can vary. On a working movie set, it is the job of the *script supervisor* to maintain notes and comprehensive images of the set in order to preserve continuity. However, actors can still cause difficulties, especially with props they are required to interact with over the course of the scene being filmed, as is the case with the TV remote control on the table in Figure 1 which is picked up and used over the course of the second shot.

Because such errors exist in films, humans become interested in tracking them down, giving rise to websites and forums where dedicated fans can exchange observations, and putting more pressure onto the film makers to achieve higher standards of continuity throughout their productions. Some websites exist solely to comment on errors in movies, such as `moviemistakes.com` [11]. Others like the *Internet Movie Database* (IMDb) [1] cater for the wider interests of the movie-going public, but also contain user-contributed lists of continuity and other errors; such lists appear on IMDb in

the "goofs" section. There are also many websites dedicated to particular films or TV series which list such errors – including those in entirely computed generated films such as 'Finding Nemo'.

Our goal in this paper is to recover such visual continuity errors automatically. In essence we treat the problem as that of a "spot-the-difference" challenge: the puzzle in which a person is given a pair of pictures and asked to identify small discrepancies between them. For a human this challenge is difficult when the pictures are presented side by side, but much simpler when presented as a temporal sequence or when toggling between the two images on a computer screen, as the human perceptual system can then use motion to establish correspondence (or otherwise) between the pictures. We provide the necessary correspondence in this work using automated registration methods from computer vision.

However, there is an additional degree of difficulty in films: while children's cartoon-style spot-the-difference puzzles often contain people, they usually stay in the same pose between the two images, unless small changes in their pose are part of the set of puzzle answers. The main difference in the movie spot-the-difference challenge is that certain objects can be expected to move, and these are *not* continuity errors. Consequently, it is also necessary to define (and detect) a set of allowable changes – with humans and other moving objects being the principal causes.

We demonstrate here a completely automatic method for detecting and ranking continuity errors. It is a plug and play system: movie in, ranked list out. The approach has four components: first, the editing structure of films is used to identify likely candidate shot pairs (section 2); second, the pairs are registered and areas containing humans identified (section 3); third, a decision on change detection is made for the remaining image areas taking account of moving objects (such as humans and vehicles) (section 4); finally, the frames deemed to differ are ranked, with the errors delineated. We use the film 'Love Actually' as the running example to illustrate this process, from DVD to ranked errors.

The ability to locate such discrepancies within a much larger body of data such as a feature film has several useful applications. In movie post-production, an automatic retrieval tool like this could be used to highlight areas where digital touching-up to repair the continuity may be desirable. Away from the movie industry, the detection and localisation of continuity errors could be useful for those evaluating the validity of video evidence such as footage from a closed circuit TV system. Such an approach could also be applied to data collected in geographical information systems. For example, Google StreetView is currently doing a first pass on major cities, but on subsequent sweeps the types of methods developed here can be used to automatically detect and quantify changes. A similar need arises in satellite surveillance of the same areas over time.

2. USING THE MOVIE STRUCTURE

Movies are generally filmed and edited using a well defined set of rules [10, 13]: establishment shot, medium shot, close up; ABAB shot structuring for two character dialogue exchanges; the 180 degree rule, etc., as in Figure 2. We use this editing structure to target pairs of frames that should have strong continuity, and therefore are likely candidates for continuity errors. In particular we concentrate on $A_1, B_1, A_2, B_2, \ldots$ shot sequences which alternate between two camera viewpoints covering a scene. In this scenario each camera has an approximately fixed viewpoint, with the A shots originating from one camera and the B from the other. The problem (for the continuity editor) is that there are multiple takes of a scene, so that both the scene and actors can inadvertently change between the take used for A_1 and that used for A_2.

Figure 2: An example of a wide scene establishment shot followed by four shots in the "A_1, B_1, A_2, B_2" pattern, taken from 'Love Actually'.

In principle, this editing structure can be undone to place the shots from the same camera within a scene, $A_1, A_2, \ldots$, into a continuous *thread* [3]. Once consecutive shots in a single thread are identified, we expect to be able to identify most continuity errors by examining the final frame in the first shot and the first frame in the following shot for visual discrepancies. To this end we first partition the movie into shots, and then identify frame pairs that should match unless a continuity error is present. For a typical movie with 100K–200K frames, around 200–550 candidate pairs are identified. These steps are described next.

2.1 Shot detection

A comprehensive review of shot-detection techniques is given by Lienhart in [9]. For our implementation, we describe each frame using a 64-bin histogram of its R, G and B channels, then find differences between consecutive frames using a thresholded L_1 norm difference.

We choose the threshold to *over-segment*, rather than under-segment, since over-segmentation is not a problem here – contiguous frames from the same shot should not produce continuity errors, and should be removed by the later processing stages. There are typically 1000 or more shots in a film, and we typically find 1000–2000 shots with our choice of threshold.

2.2 Shot matching

We now mine the list of shots to find sets-of-shots taken from very similar camera viewpoints. The RGB histograms computed for shot detection are used for a second time, now to establish whether there is a link between the final frame of each shot to the first frames of *all* chronologically later shots in the movie, by considering the L_1 histogram difference again.

A small penalty is added to the difference so that shots

Figure 3: Each row shows the first, middle and last frames in a thread of shots from 'Love Actually'. The colour histogram difference between the last frame of shot 535 (top) and the starting frame of shot 537 (middle) is greater than that to the starting frame of shot 545 (bottom), because of small lighting changes in the background which change the histogram (though this is hard to perceive by eye). However, in order to find continuity errors, it is important that the middle shot is matched, as this is the shot which comes from a different take to the majority of other shots in this thread.

close to each other in the space of movie frame numbers are slightly more likely to be associated together, *e.g.* for frame numbers f_1 and f_2, with corresponding frames $I(f_1)$ and $I(f_2)$, then the score, s, is taken to be

$$s(f_1, f_2) = \|\text{hist}\,(I(f_1)) - \text{hist}\,(I(f_2))\|_1 + w\,(f_2 - f_1)\,, \quad (1)$$

where w is a scalar. For example, if shots 1, 3 and 5 all form part of a thread, but 1 and 5 are more similar (perhaps due to an out-of-plane rotation of an actor's head between 1 and 3), we want to return shots 1 and 3 as one pair, and shots 3 and 5 as another. The DVD frame rate is 25 frames per second, and the majority of shots are a few seconds in length, making a typical shot around 50–250 frames in length. Since frame histogram differences are of the order of 10^5, the penalty coefficient, w, is set around 10^4.

Figure 3 illustrates an example of one case where the extra weighting term is required in order to ensure the shots are threaded together in the correct order. The histogram difference between the upper shot to the middle one is 50% higher than the difference between the upper shot to the third one shown. We are able to identify correctly that 537 follows on from 535, so the final frame of 535 (top right) falls next to the first frame of 537 in this thread. Given the short time spanned by the intervening shot (3.2 seconds, or 80 frames), these two frames *ought* to be entirely visually consistent.

We take each shot in the movie in turn and pair it with the best-matching "following" shot that occurs later in the movie. If there are n shots in the movie then $O(n(n-1)/2)$ pairs are considered. Pairs whose score s is below 1.5 times the threshold used for shot detection are then assumed to be valid matches, and are passed on as candidate shot pairs to be evaluated.

3. SPOTTING DIFFERENCES

From a pair of candidate frames, the goal is to obtain some measure of how well they agree, and to localise inconsistencies if they do exist. To this end, we would like a point-to-point registration between the frame pairs found in the shot-matching section above. We compute this using a

Figure 4: Upper-body detections in two frames from a thread of 'Love Actually'. The regions are extended down to the bottom of the frame to account for lower body regions, and points within these extended regions are not used in the homography computation, leading to a more accurate registration of the scene background.

homography (planar projective transformation), which gives the correct registration if the objects within the scene are distant enough from the camera that they can be assumed to be lying in a plane, *or* the camera itself only pans or zooms, so that no parallax effects are introduced by motion of its optical centre relative to the scene [6]. This transformation is represented as:

$$\mathbf{x}' \quad = \quad \mathbf{H}\mathbf{x} \quad (2)$$

where $\mathbf{x} = [x, y, 1]^T$ is the location of any point in the first image in *homogeneous coordinates*. Then $\mathbf{H}$ is the 3×3, eight-degrees-or-freedom homography matrix, and the point corresponding to (x, y) in the second image is $(x'/c', y'/c')$, where $\mathbf{x}' = [x', y', c']^T$.

The homography $\mathbf{H}$ is estimated for a pair of frames using interest point matches and RANSAC, as described in Section 3.4.

3.1 People-detection

An obvious source of movement between frames in most movies is people; these motions do not constitute continuity errors generally, because the time between shots is usually long enough for a character to change their pose believably.

The area of the image occupied by humans can be detected using a human upper-body detector – this is a similar problem to face detection [14] or pedestrian detection [4, 8], but the detector is specifically tuned to track the upper body only, because this is often all that is visible of an actor in a television show or movie. Once people are located, point matches are removed from those parts of the images, so that the estimated homography relates the backgrounds of the two shots (and not the independently moving people).

We use the publicly-available upper-body detection software of [5], based on *Histograms of Oriented Gradients* [4], on the two frames independently. Detections are accepted if a similar detection box appears at a similar location in both frames such that the overlap area is at least 50% of

Figure 5: The steps of the spot-the-difference process. Top row: two initial frames. Middle row: Upper body detections in each frame. Bottom left: Thresholded differences found between the two frames after registration. Bottom right: differences classed as either potential error (red) or as a discrepancy due to a person (green).

the total area of either detection; otherwise the detection is assumed to be a false positive. Running on an entire shot would make these detections more robust, but would also significantly slow down our overall processing pipeline. We show an example of the typical detections found for a pair of frames from 'Love Actually' in Figure 4. Once these matches are found, all boxes are extended to the base of the frame to account for the lower body.

3.2 Matches and comparisons

Given the two frames and the homography, a discrepancy score is computed for each pixel in the first image, indicating how similar the corresponding region of the second image is. Simple pixel-wise differences between (warped) frames are noisy, and susceptible to sub-pixel changes in camera angle and small parallax effects. Instead, we look for discrepancies $d(\mathbf{x})$ using a small patch centred on each pixel location to describe its local neighbourhood, *i.e.*

$$d(\mathbf{x}) \quad = \quad d\left(\mathcal{N}_1(\mathbf{x}), \mathcal{N}_2(\mathbf{H}\mathbf{x})\right), \tag{3}$$

where $\mathbf{H}$ is the homography as in equation (2), $\mathcal{N}_i(\mathbf{x})$ is a smoothed neighbourhood around the location of point $\mathbf{x}$ in image i. Details of the smoothing operation and $d(\cdot,\cdot)$ are given in Section 3.4.

If nothing moves in the scene, and $\mathbf{H}$ accurately describes any change in viewing angle between the frames, then we expect the the discrepancy d to be small for all pixel locations. However, noise, interpolation effects (neighbourhoods in the the second image are centred on $\mathbf{H}\mathbf{x}$, which is not necessarily an integer pixel location) and small instantaneous lighting changes mean it is generally not zero.

Thresholding d allows us to classify each pixel as either "different" or "similar", with the vast majority of pixels being expected to fall into the latter category. Additionally, these "different" pixels can be labelled as "explained" or "unexplained", depending on whether or not they intersect the people-detection boxes. Pixels belonging to the same 4-connected neighbourhood in the d map as other "explained-different" pixels are also labelled "explained-different', even if they fall outside the initial person-detection box, since they are likely also to be part of the detected person.

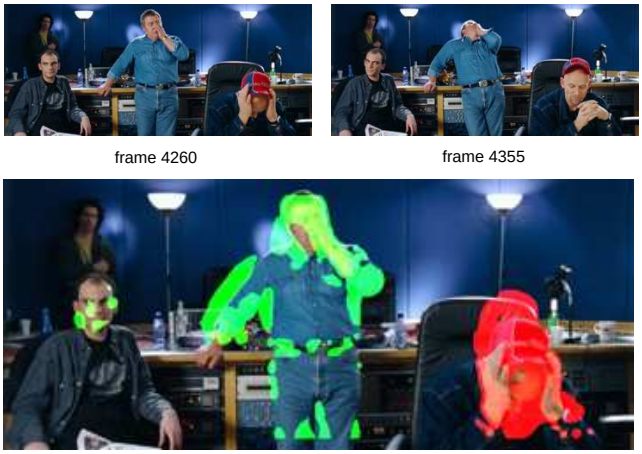


Figure 6: The discrepancies between two matched frames. The colour coding is: green, for pixels in regions which overlap the upper-body detection bounding boxes; red, for pixels which do not initially have an explanation. In this case the upper-body detection has missed the right-most person, causing him to be highlighted in red. Further refinement of the results and motion detection subsequently (correctly) remove this area, as described in Section 4.

An example of these steps of difference detection and labelling is shown in Figure 5, where green represents differences which can be explained by person-detections, and red represents the unexplained differences. A second example in Figure 6 shows the labelled regions overlaid on the first of the original images. In this case the person at the right of the shot was missed by the detector, so is coloured red. More ways to detect independently moving objects in order to explain away plausible motions like these are discussed in Section 4.

3.3 Removing poor matches

The shot detection and shot-matching of Section 2 explicitly uses a high threshold on histogram differences which still constitute a match, so as to include shots with some variation, but at the same time this encourages the inclusion of a number of incorrect pairs which do not belong to the same scene, but whose histograms may coincidentally be similar. Such pairs can be removed by examining the number of inliers found in RANSAC, and if the number is too low the pair can be thrown out of the list of shot matches. We can also remove pairs when the estimated homography contains too much foreshortening or too large a camera translation, or where there is too large a rotation in the image plane, since these cases do not tend to occur in movie threads, and are likely instead to have come from poor registration estimates. These cases can all be found simply by examining $\mathbf{H}$ scaled as:

$$\mathbf{H} \quad = \quad \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix}. \tag{4}$$

Immediately, h_{31} and h_{32} give us information about the fore-

shortening, and h_{13} and h_{23} give the overall translation. Taking these into account, the upper-left 2×2 block can then be decomposed to yield scaling, shear, and rotation [6].

3.4 Implementation details

To compute a homography, we first extract interest points using Harris corners, and compute candidate matches based on each interest point's local neighbourhood (*e.g.* using [7]). Points lying inside our extended people-detection boxes are not considered for these matches because there is a higher probability that they will move independently from the rest of the scene. We find the transform which minimises the error in these matched points' locations when projected between the two images. Outliers due to bad initial matches can cause poor estimates of homographies, so robust fitting is carried out using the RANSAC algorithm, which ignores outliers and finds a homography that fits as much of the data as possible [6].

To compute the d map for a given image pair, we first blur the images using a Gaussian kernel with a standard deviation of 2 pixels, which helps avoid edge aliasing effects and suppress some noise and MPEG2 artifacts from the frames themselves. We take $\mathcal{N}(\mathbf{x})$ to be a 9×9-pixel patch centred on $\mathbf{x}$, and this is extracted for each of three colour channels. The discrepancy $d(\mathbf{x})$ is taken to be the *maximum* of the L2 patch differences across the three channels. Dissimilarities for pixels which are visible in only one of the images, or those near the image borders, are set to zero. A threshold of 3/20 of the possible image intensity range is used to determine whether the pixel location should be flagged as "different" or not.

A map of possible error locations computed in this way tends to be comprised of a set of smooth blobs, because of the Gaussian blur and the window-based approach. Each distinct 4-connected set of pixels is treated as a single *region*, and a region of "different" pixels is flagged "acceptable" (coloured green) if part of that region lies within a person detection, and "possible error" (coloured red) otherwise.

4. RANKING AND REFINING RESULTS

The output from the spot-the-difference algorithm of Section 3 is a dense labelling of each pixel in the first of the frame pair, indicating whether or not each pixel's neighbourhood is different to the corresponding region in the other frame, and whether or not that difference can be attributed to a nearby human. Our objective in this section is first to remove areas in the difference image that are due to legitimate moving objects, and then to rank the list of shot-pairs to give an at-a-glance summary of the most likely continuity errors in the movie.

By this stage in the pipeline, around 100–400 shot pairs may be left under consideration, so on this small subset we can afford the time to run some more comprehensive motion detection without increasing the running time of the overall system too severely. We therefore consider several more frames along the thread in each direction from the cut boundary (*i.e.* before the first frame and after the second) to check for more independent motion in the scene; this might be from undetected humans, animals, cars, airplanes, swinging doors or many other objects.

4.1 Motion suppression details

To check for independent motion in a region that is initially flagged as being a potential discrepancy, we take each of the matched frames in turn and look back or forward (depending on whether it is a first or last frame in the shot) to pull out another five frames in a small window of time. For each of these extra frames, a homography is found as before, and is used to warp pixels from the extra frame back into the flagged region of the original, discarding any images for which too many outliers are found in the homography estimate. We can then use the variance of the pixel values in each colour channel to indicate whether something is moving in this region. Pixels with a strong edge response (found using a standard Canny edge detector) are zeroed out in this motion map, to mitigate the effects of tiny motions causing large variances due for example to very slight image misregistrations, and the motion assessment for each region is based on all the remaining pixels. Figure 7 shows an example where this motion detection is used to suppress the image differences after an initial detection.

Figure 7: Motion detection: the actor is moving both before and after the cuts, so the motion is detected, and what started as a large area of discrepancies is instead given a consistent explanation as motion in the scene (and coloured blue to indicate this). Because of the unusual pose of the actor in these images, the upper-body-detector could not be expected to identify this as a human.

4.2 Ranking

We could return the list of 50-100 images which still have unexplained discrepancies remaining after the removal of moving objects, but it is far more useful to rank the list based on the probability of the flagged region actually being caused by a continuity error like a moved prop. One possibility is simply to rank the images in order of how many discrepancy pixels they contain, but there are still a number of factors that can cause erroneous discrepancies (*e.g.* due to missing person/motion detections) and consequently true continuity errors would not appear at the top of this list.

A few cases which cause erroneous discrepancies are:

1. **Images with too many pixels identified as possible discrepancies.** These tend to be from shots of crowds or similar, where almost everything is in motion. Discrepancy regions in such shots, where there are many independently moving agents in the scene, are are more likely to be from missed motion detec-

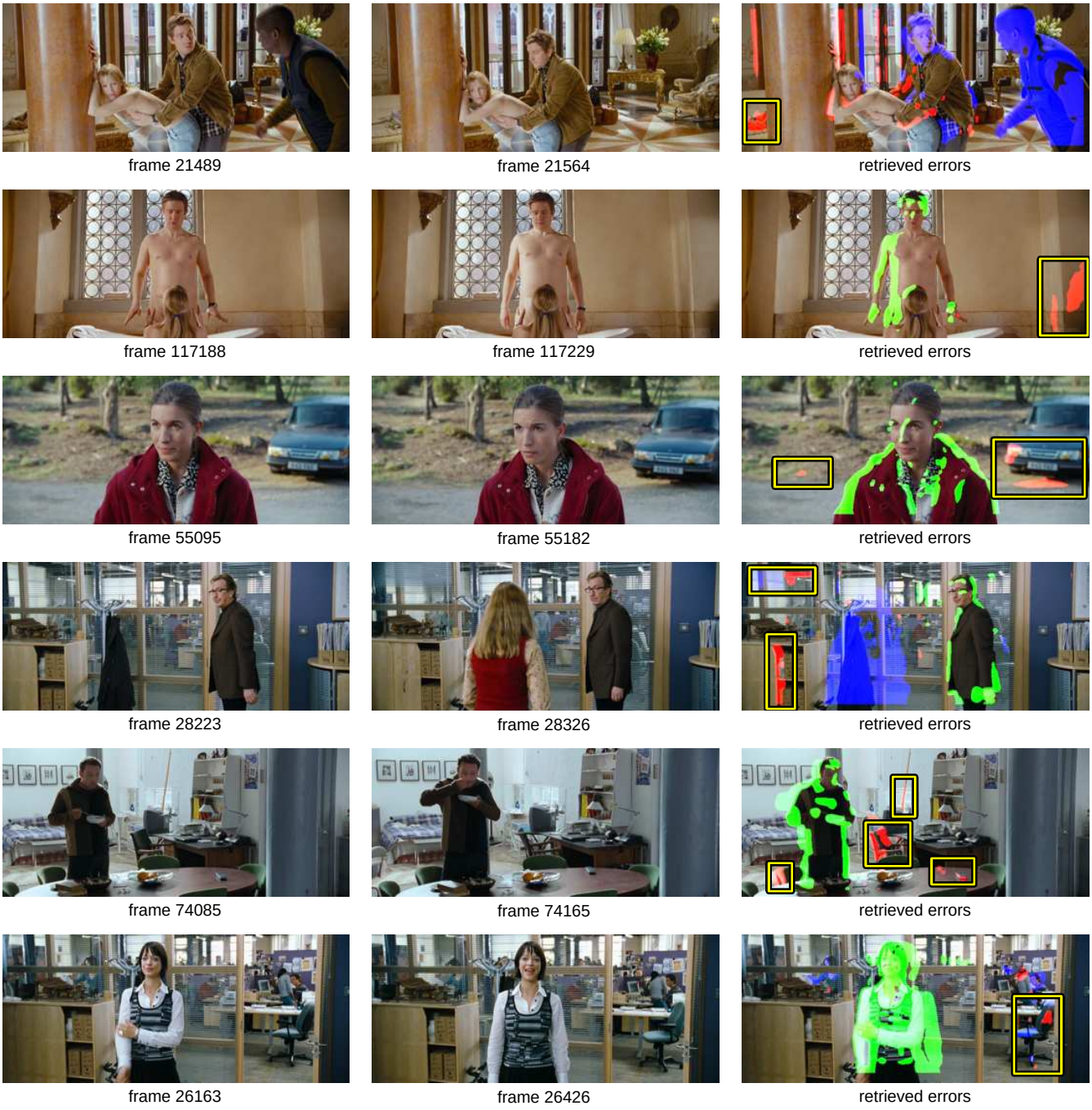


Figure 8: **Continuity errors retrieved from 'Love Actually'. Colouring of the frames on the right is done automatically; we delineate the principle discontinuities with boxes within each frame. While the first and fifth rows include some errors which are listed on moviemistakes, the others represent errors which have not yet been reported. Row 1) The clothing at the left moves. Row 2) The lighting at the right of the set changes considerably. Row 3) The dirt bank in the background changes in several locations. Row 4) The cupboard doors change configuration and the blinds change angle. Row 5) The TV remote, pole, chairs and other items all move slightly. Row 6) The chair in the background rotates, and its base moves relative to the floor.**

tions, and are therefore not continuity errors.

2. **Potential discrepancies located very near the left and right edges of the frame.** It is common to find small changes at the left or right edge during "over-the-shoulder" shots (see *e.g.* [13]) where two characters participate in a dialogue, and the speaking actor is filmed over the other's shoulder. The side or back of the listener's head is visible at the very edge of the shot. Motion of such listeners over a small number of frames is hard to detect because they are usually not in focus, and nor is there enough of the head present in the shot for them to be recognised as humans.

3. **Potential discrepancies within a few pixels of already-explained discrepancies.** If most discrepancies in a region are explained by a person or moving object, it is likely that spatially close discrepancies have the same explanation, even if the detection wasn't quite over the threshold. This case also covers circumstances like the case when a prop lies near an actor's hand whose motion can therefore be attributed to the actor even though it is not part of the person itself.

4. **Images containing neither people-detections nor motion detections.** These tend not to contain visual continuity errors; the discrepancies are much more likely to be the result of failures to detect the actual cause of the difference – such as a moving person or other moving object.

To handle the first point above, shot pairs are removed from the working set if more than one third of the pixels are flagged as having a discrepancy of any kind (even if attributed to motion or people), or if more than 12K (out of around 200K) pixels have unexplained discrepancies.

The remaining list is ranked based on the total number of discrepancy pixels, except that the totals for some regions – those indicated by points two and three above – are down-weighted by a factor of two, so that regions with discrepancy pixels close to (within 10 pixels of) the left/right image edge or to an explained region (such as a detected human or motion) are less likely to appear high up in the ranked list. Additionally, only images with discrepancy regions greater than 50 connected pixels are considered for the ranking.

To address point four above, as a final step, shot pairs with *no* explained discrepancies are moved to the end of the ranked shot list, since they are likely to contain failed person or motion detections.

5. RESULTS

For the movie 'Love Actually', there are 184871 frames, from which 1859 shots are extracted. Of these, 537 candidate shot pairs are identified out of the possible 1.7M pairs (*i.e.* $1858 \times 1857/2$ pairs). A subset of 123 of the shot pairs are returned in the ranked list.

Six of the errors found are shown in Figure 8, arranged in the order in which they are ranked. Of these errors, only the first and fifth rows show errors which had been listed on the `moviemistakes.com` website when this work was carried out. The other four are definitely instances of continuity errors – in the sense that there is no plausible explanation for the changes in the set – and indeed are generally more pronounced than the error returned in the top row even though

Figure 9: An example from the movie 'The Fifth Element'. Items on the tripod are correctly flagged as red (probable errors) in the bottom image. However, the tripod legs are incorrectly classed as moving (coloured blue); this is a result of camera tracking in both shots, and the tripod being at a different depth from the rest of the scene. This is an example where full 3D modelling of the shot should increase the accuracy of our detections.

they were missing from the listing on `moviemistakes.com` until we submitted them ourselves.

Of the 123 returned shot pairs, the ranks of the six shots shown in Figure 8 are 5^{th}, 9^{th}, 20^{th}, 21^{st}, 39^{th}, and 76^{th}.

5.1 Results on other movies

We give here a selection of the errors discovered by processing other DVDs (using the same parameter settings). Figure 9 shows a continuity error found in 'The Fifth Element' (Besson, 1997), which is also confirmed as being a known continuity error on the website `moviemistakes.com`. Figure 10 shows an error found in 'Pretty Woman'; in spite of the fact that this movie is over 18 years old, the discrepancy found here is not listed on any of the 'goof'-finding websites to the best of our knowledge. Finally, Figure 11 shows the change in orientation of a background prop in the movie 'Forrest Gump'. Again, this mistake is already known to `moviemistakes.com`.

6. DISCUSSION AND FURTHER WORK

There are three main areas for future work. First, we could exploit the editing structure of the movie more thoroughly. For example, the movie could first be partitioned into scenes, and then all shots could be examined in the context of a scene. Alternatively, $A_1, B_1, A_2, B_2, A_3, B_3$ sequences could be identified and differences detected by considering a set-of-shots $\{A_i\}$, rather than just a pair. In both these cases, there is higher likelihood that the shots are supposed to be contiguous in time, so there is a higher confidence that detected differences are continuity errors. Also, at a longer range, flashbacks to earlier scenes could be detected. These are often shots quite distant from the originals, so there is less chance that humans will have detected them when watching the DVD.

Second, the registration and human segmentation could be improved so that there are fewer false positive differences. Registration could move towards a full 3D scene reconstruc-

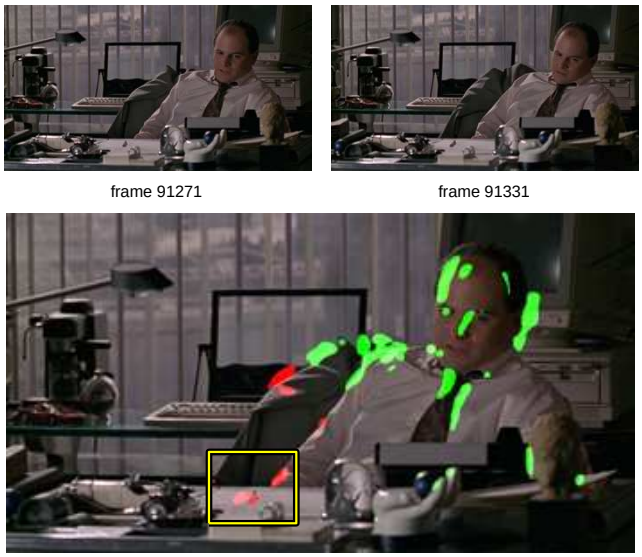


Figure 10: An example from the movie 'Pretty Woman'. The actor is correctly identified as a person, and therefore motion due to him is coloured green. However, the movement of the pen on the desk is flagged as a probably discrepancy (red), since it is stationary at the end of the first shot and at the start of the second, yet its position has changed.

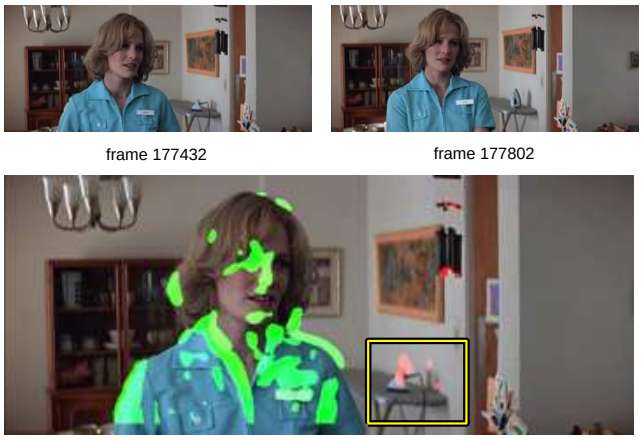


Figure 11: An example from the movie 'Forrest Gump'. The iron in the background is correctly identified as having moved (and therefore is shown in red in this figure), while the slight change in position of the actor in the foreground is explained as acceptable human motion (and so is coloured green here).

tion: scanning for patches along an epipolar line [6], would allow a wider collection of scene matches to be compared, and would be a particularly useful addition given that both a change in camera position and the movement of props in a scene can occur for different takes. There are several instances of mistakes people have spotted between close up and wide viewpoints that we haven't been able to match here because their histograms are too different. Instead of using the upper body detection boxes, humans could be segmented out (e.g. using methods like [2, 5, 12]), so that objects in the background but within the figure bounding box can also be detected.

Third, a human appearance model could be incorporated, to check for differences in clothing and hairstyle that are often reported on movie mistake sites. For example, there is a well-known "dry raincoat" error in 'Casablanca', where a character is dripping wet one moment, and dry the next.

Acknowledgments: We are grateful for the support of the EPSRC for this work through a Platform grant.

7. REFERENCES

[1] Internet Movie Database (IMDb). http://www.imdb.com/.

[2] N. E. Apostoloff and A. W. Fitzgibbon. Automatic video segmentation using spatiotemporal t-junctions. In *Proc. BMVC.*, 2006.

[3] T. Cour, C. Jordan, E. Miltsakaki, and B. Taskar. Movie/script: Alignment and parsing of video and text transcription. In *Proc. ECCV*, volume 4, pages 158–171, 2008.

[4] N. Dalal and B. Triggs. Histogram of Oriented Gradients for Human Detection. In *Proc. CVPR*, volume 2, pages 886–893, 2005.

[5] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *Proc. CVPR*, Jun 2008.

[6] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[7] P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing. The University of Western Australia. http://www.csse.uwa.edu.au/∼pk/research/matlabfns/.

[8] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *Proc. CVPR*, 2005.

[9] R. Lienhart. Reliable transition detection in videos: A survey and practitioner's guide. *International Journal of Image and Graphics*, 1(3):469–486, 2001.

[10] J. Monaco. *How to Read a Film: The World of Movies, Media, Multimedia – Language, History, Theory*. OUP USA, Apr 2000.

[11] MovieMistakes. http://www.moviemistakes.com/.

[12] D. Ramanan. Using segmentation to verify object hypotheses. In *Proc. CVPR*, 2007.

[13] T. J. Smith. *An Attentional Theory of Continuity Editing*. PhD thesis, University of Edinburgh, 2006. Unpublished Doctoral Thesis.

[14] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*, pages 511–518, 2001.