ARDO: Automatic Removal of Dynamic Objects

ARDO: AUTOMATIC REMOVAL OF DYNAMIC OBJECTS

Exclude Moving Objects Automatically from a Collection of Images using a Consumer Camera

Patrik Goorts, Philippe Bekaert

Hasselt University - Expertise Centre for Digital Media, Wetenschapspark 2, 3590 Diepenbeek, Belgium

patrik.goorts@uhasselt.be, philippe.bekaert@uhasselt.be

Keywords: Object Removal:Image Synthesis:Feature Detection:Image Registration:Median:Linear System

Abstract: We present a novel method to remove tourists, cars and other moving objects from photos of monuments, buildings and other attractions. Typically, this is a manual process. Nevertheless, our method is fully automatic and requires neither image stabilization or advanced hardware during acquisition. The method requires a set of images taken from roughly the same location using any consumer photo or video camera and synthesizes the decluttered result using two phases. In the first phase, these images are aligned on to each other using image features and affine transformations. The second phase merges the images to the final result using a median-based per pixel filtering. The results are pleasant looking, even with moving clouds and trees, and outperforms other techniques considering quality and manual intervention.

1 INTRODUCTION

When making holiday pictures, these photos are typically cluttered with tourists, cars and other moving objects. This can be undesirable, especially when trying to photograph monuments, buildings and other attractions. It is, most of the time, not possible to ask people to leave the scene to take a picture. To solve this problem, we introduce an algorithm to create a decluttered photo of the scene, without tourists and cars, using a number of images taken with a standard handheld camera. It is sufficient to stay roughly at the same location; no tripods or other stabilization devices are needed. The algorithm is fully automatic; only the images are needed as input and the method requires no manual intervention whatsoever.

Alternatively, a video sequence can be used. This will result in a larger number of images, thus more data for a correct result.

The method requires a few seconds between every input image to give the best result. This is to ensure that the difference between the images is large enough to detect motion. When the motion is limited, it might be necessarily to increase the time between the acquisition of the images. If video input is used, this requirement can be met by using only one frame or less per second, and dropping unnecessary frames. Typically, only 5 to 10 images are required if the motion of the moving objects is large enough and the total amount of objects is limited. However, when using a crowded scene, it might be necessarily to acquire more images. Using video input, 30 seconds to a few minutes should suffice to obtain good results.

The algorithm uses two phases: aligning (registering) the images and merging the images in the final result. First, all images are aligned such that it looks like every photo is taken from exactly the same position. The alignment step is done using spatial image features. Secondly, these aligned images are merged in the final result using a median-based per pixel filtering. This will result in the removal of moving objects in the scene; only the static scene itself is kept.

2 PREVIOUS WORK

There are some other techniques to remove tourists from photos, however each with their drawbacks. In the age of analog photography, moving objects were removed using long exposure times,which was too long to register walking persons. In digital photos, objects can be removed from a sequence of images using Photoshop (or a similar application) (Hoffman, 2010). However, the technique requires a lot of manual intervention. All the images should be taken from the same location using a tripod, or the alignment should be done using manual transforma-

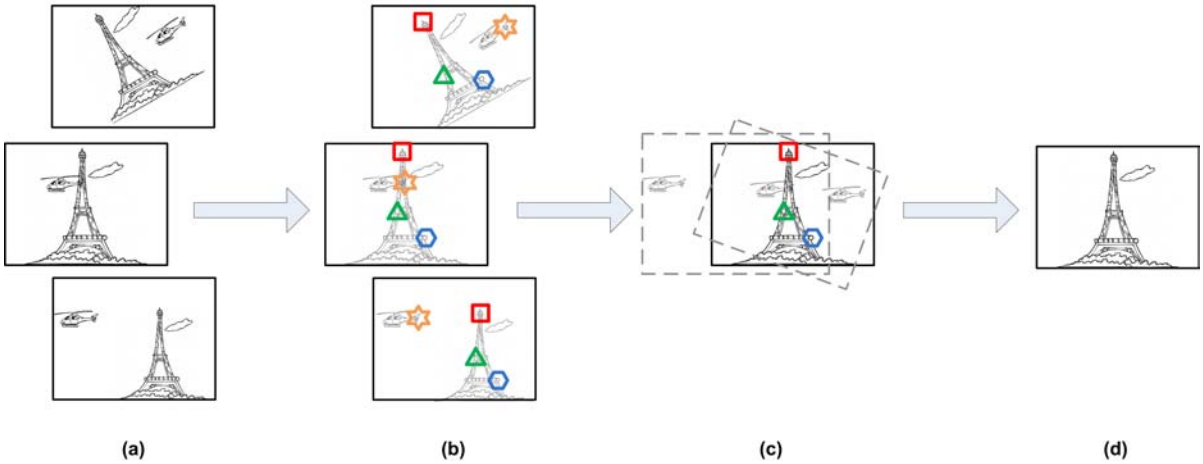

Figure 1: Overview of the algorithm. In the first phase, matching features are detected between the reference image (middle image) and the other images (b). These images are then registered using these features by calculating a homography between the reference image and the other images (c). In the second phase (d), these images are synthesized using a median-based per pixel filtering.

tions. This process is time intensive and error-prone.

Inpainting can be used to remove objects from a single image (Bertalmio et al., 2000). However, this will not give a correct image of the desired scene, because the lack of information. Texture and fine detail is lost and must be reconstructed from visible patches in the scene. This method also requires a lot of manual work to generate plausible results, nonetheless, only one image is required.

The tourist remover application of Snapmania (Snapmania, 2011) provides an automatic way to remove tourists from a set of photos. No information is provided about the method. We tested the application using our data sets (see section 4).

Agarwala et al. (Agarwala et al., 2004) describe an interactive technique to remove elements from a set of photos using optimization. The user paints strokes where the tourists are and the algorithm constructs a new image using this input. A cost function is minimized using graph cuts to construct the final image. This algorithm also has applications in image stitching, portrait compositing and creating of image mosaics. This method produces good quality results even when using a small number of images, but requires user input to detect the tourists.

3 METHOD

The algorithm consists of two parts. The first part registers the images using feature descriptors. The second part merges the registered images in a final image. An overview of the method is given in Figure 1.

3.1 Registration

Registering images is the process of calculating a transformation from one coordinate system to another, such that every image shares the same coordinate system i.e. all image are aligned to one another (Szeliski, 2006). For this application, we consider only affine transforms of the image, such as translation, rotation and scaling. The transforms of the images are only affine, because we use the same camera for a static scene during the whole acquisition. This linear transformation of one image to another is represented by a homography using a 3-by-3 matrix H (Hartley and Zisserman, 2000).

$$p_r = H_c p_c = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} \quad (1)$$

where p_r is an image point in the reference image and p_c an image point in the image being processed. The matrix is calculated with a linear system using image features as input.

For every image, we calculate a set of scale and rotational invariant, spatial image features using SIFT (Lowe, 1999). Using these features, we calculate the matches between an arbitrary chosen reference image and every other image. Because SIFT features are invariant to translation, rotation and scaling, the matching will be robust for this application.

Using these feature matches, we calculate a homography between the chosen reference image and every other image. This homography is acquired by solving the linear system defined by the homography

and the matched features. For every homogeneous feature position $f_r = [x_r, y_r, 1]^T$ in the reference image and the matching homogeneous feature position $f_c = [x_c, y_c, 1]^T$ in the image being processed, we acquire two linear equations from equation 1:

$$\begin{cases} \frac{h_{11}x_c+h_{12}y_c+h_{13}}{h_{31}x_c+h_{32}y_c+h_{33}} \approx x_r \\ \\ \frac{h_{21}x_c+h_{22}y_c+h_{23}}{h_{31}x_c+h_{32}y_c+h_{33}} \approx y_r \end{cases} \quad (2)$$

h_{ab} is the element in H_c at row a and column b. This results in a linear system of $2n$ equations and 9 unknowns, where n is the number of matched features. This system will not have a solution, due to small rounding errors in the feature positions, errors in the matching process, et cetera. Therefore, to solve the equation, we will minimize the sum of the reprojection errors:

$$\begin{aligned} \sum_f & \left(x_r - \frac{h_{11}x_c+h_{12}y_c+h_{13}}{h_{31}x_c+h_{32}y_c+h_{33}} \right)^2 \\ & + \left(y_r - \frac{h_{21}x_c+h_{22}y_c+h_{23}}{h_{31}x_c+h_{32}y_c+h_{33}} \right)^2 \end{aligned} \quad (3)$$

Due to the motion of the objects, not all feature pairs will satisfy the homography. These feature pairs will be outliers and must be filtered out to acquire a correct homography. The system is solved using the *Least Median of Squares* (LMeDS) method (Rousseeuw et al., 1987), which incorporates outliers such that only 50% of the features must be inliers and no threshold should be defined beforehand. LMeDS selects a number of random features and calculates a homography using least-squares. Then we calculate the median of the reprojection error of every feature, including the features not selected for the calculation of the homography. We select the homography for which this median is minimal.

Typically, RANSAC is used to remove outliers (Hartley and Zisserman, 2000). However, this algorithm is depending on a threshold and can categorize too many inliers as outliers. This will result in an insufficient set of data points, thus in an incorrect and blurred result. The LMeDS method, however, will even work when there are only a few matches.

Feature tracking is used before in image stabilization algorithms (Censi et al., 1999; Hu et al., 2007; Battiato et al., 2007). Alternatively, more advanced video stabilization algorithms can be used to register the images. Typical methods use a kalman filter to model intentional motion and handle unintentional motion as noise, using features (Pinto and Anurenjan, 2011) or pixel intensities (Litvin et al., 2003) as input. Other methods use optical flow (Chang et al., 2002) or the frequency domain (Kumar et al., 2011) to detect global image motion.

These methods also incorporate intentional camera motion, such a walking around the building. The more advanced algorithms will only remove small and local movements and keep the large, intentional movements. Our method uses a reference image and does not output a video sequence where certain kind of movements must be kept. Therefore, our method will not need these kind of advanced methods.

3.2 Image synthesis

Once the homographies between the images and an arbitrary chosen reference image are calculated, we can map every pixel of the images to the reference image. The registered images are then merged in the final result using the median for every mapped pixel. If the desired scene is visible in the majority of the images, calculating the median will provide the correct result. A more cluttered foreground hence requires more images.

4 RESULTS

To demonstrate the technique, we acquired a set of photos and videos from monuments.

When using videos, not every frame is used. The difference between consecutive frames is too small to provide extra useful information. Using every frame will not provide better results, but will only increase the running time of the method. Therefore, we will only use one frame every 2 seconds.

The amount of clutter determines the amount of photos needed. The desired scene should be visible in the majority of the images. If many persons are located on the same location in most of the photos, the required scene is not visible, and the method won't be able to generate a plausible result.

As shown in Figure 2, moving backgrounds (like clouds and trees) won't necessarily result in bad images. The result is not the same as in any image, but still plausible and pleasant looking. This is also valid for changes in lighting, as can be clearly seen in Figure 3.

The method works best when the distance to the desired scene is large and the movement of the camera is limited. This will reduce the effects of parallax caused by movements of the camera. Because the registration uses only linear transformations from one image to another, no parallax effects can be incorporated. If the parallax is large, the registration will be inaccurate, resulting in a blurred image. Incorporating parallax remains an open problem until today (Szeliski, 2006).

Figure 2: Calculating the median of clouds will not result in an image that is correct with respect to the input, but is still plausible and sufficient for the final image.

We compared our method with other methods. As can be seen in Figure 6, our method generates more plausible results compared with the tourist remover application of Snapmania (Snapmania, 2011). Their method cannot handle relighting very well.

The group shot application of Agarwala et al. (Agarwala et al., 2004) provides good results compared with our method, but requires a lot of user input, which makes it more difficult to use. However, their method works better when using only 2 or 3 images.

5 CONCLUSION

We presented an automatic method to remove tourists, cars and other moving objects from a set of photos, while keeping the static scene. Typically, 5 to 10 photos are needed, with a few seconds between each photo. Alternatively, a video sequence can also be used. The algorithm uses features to calculate a transformation between the different images, so that no tripods or other advanced stabilization devices are needed during the acquisition of the photos. As can be seen from the results, the final result is plausible and pleasant looking. However, the image is not necessarily correct when there are specific background elements, like clouds or other slow moving objects. This is not a problem, because the focus of the application is on the static scene and the generated clouds and trees are still plausible. The method outperforms existing techniques when using sufficient input images, considering the quality of the result and the manual input required.

ACKNOWLEDGEMENTS

Patrik Goorts would like to thank the IWT for its PhD specialization bursary.

REFERENCES

Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D., and Cohen, M. (2004). Interactive digital photomontage. In *SIGGraph-04*, pages 294–302.

Battiato, S., Gallo, G., Puglisi, G., and Scellato, S. (2007). Sift features tracking for video stabilization. In *Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on*, pages 825–830. IEEE.

Bertalmio, M., Sapiro, G., Caselles, V., and Ballester, C. (2000). Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424. ACM Press/Addison-Wesley Publishing Co.

Censi, A., Fusiello, A., and Roberto, V. (1999). Image stabilization by features tracking. In *Image Analysis and Processing, 1999. Proceedings. International Conference on*, pages 665–667. IEEE.

Chang, J., Hu, W., Cheng, M., and Chang, B. (2002). Digital image translational and rotational motion stabilization using optical flow technique. *Consumer Electronics, IEEE Transactions on*, 48(1):108–115.

Hartley, R. and Zisserman, A. (2000). *Multiple view geometry*, volume 6. Cambridge university press.

Hoffman, M. (2010). Automatic tourist remover. http://www.tipsquirrel.com/index.php/2010/07/automatic-tourist-remover/.

Hu, R., Shi, R., Shen, I., and Chen, W. (2007). Video stabilization using scale-invariant features. In *Information Visualization, 2007. IV'07. 11th International Conference*, pages 871–877. IEEE.

Kumar, S., Azartash, H., Biswas, M., and Nguyen, T. (2011). Real-time affine global motion estimation using phase correlation and its application for digital image stabilization. *Image Processing, IEEE Transactions on*, (99):1–1.

Litvin, A., Konrad, J., and Karl, W. (2003). Probabilistic video stabilization using kalman filtering and mosaicking. In *Proceedings of SPIE Conference on Electronic Imaging*, pages 663–674. Citeseer.

Lowe, D. (1999). Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157. Ieee.

Pinto, B. and Anurenjan, P. (2011). Surf based robust video stabilization using kalman filter. In *ICTT Electronics and Communication*.

Rousseeuw, P., Leroy, A., and Wiley, J. (1987). *Robust regression and outlier detection*, volume 3. Wiley Online Library.

Snapmania (2011). Snapmania tourist remover. http://www.snapmania.com/info/en/trm/.

Szeliski, R. (2006). Image alignment and stitching: A tutorial. *Foundations and Trends in Computer Graphics and Vision*, 2(1):1–104.

Figure 3: Result from a video input using 40 frames. The camera, the cars and the tourists before the Colosseum are successfully filtered out. There is a difference in lighting in the different input images, but this is not noticeable in the final result.

Figure 4: Result from a video using 40 frames. This sequence contains a lot of large objects (cars, buses). These are all filtered out successfully.

Figure 5: Result from 8 pictures using a low budget camera. This set contains a lot of translations and rotations. The result is sharp, which demonstrates the correct registration of the images. Notice that the clouds are washed out, but are still plausible.

Figure 6: Comparison of the method of Snapmania (Snapmania, 2011) (left figure) and our method (right figure). Our method results in higher quality images.