Real Time Gait Analysis Using RGB Camera

Mario Nieto-Hidalgo(✉) and Juan Manuel García-Chamizo

Department of Computing Technology, University of Alicante,
Campus San Vicente del Raspeig, Alicante, Spain
{mnieto,juanma}@dtic.ua.es
http://www.dtic.ua.es

Abstract. In this paper we propose a vision based gait analysis approach that work under real time constraints. We propose the use of a multiresolution pyramid image representation that allows to provide suboptimal responses if the deadline is reached. The impact of each suboptimal response is analysed showing that although there is an impact in the quality of the output, the gait analysis algorithm still provides satisfactory results. In addition, the adjustment to time constraints of the proposed approach is also analysed showing suitability for real time constraints.

Keywords: Real time · Gait analysis · RGB camera · Background subtraction

1 Introduction

Vision based gait analysis is the focus of our project. We aim at developing a low cost, non-invasive system to obtain the parameters of human gait. We are focusing in the gait of elderly people with the objective of early detection of frailty and dementia syndromes. This is supported by several studies that link the degeneration of gait with those syndromes [3,12–14].

Although we focus in the gait of elderly people, our approach could be applied to other population sectors (children, athletes...) for rehabilitation or identification of deviations from standard patterns.

The use of computer vision to analyse gait provides an objective external measure of kinematic parameters unlinking the results from the dynamic approach where there are physiological and emotional influences that could compromise the objectivity of the output.

Computer vision demands a high computational power that makes it difficult to work in low cost devices under time constraints, however, it is convenient to advance in this matter so every geriatrics could make use of these kind of solutions. In this paper, we propose a real-time approach for our previous work [6,7] in which a method for vision based gait analysis that uses only RGB camera is presented.

This method performs background subtraction to obtain the silhouette of the subject and then obtains the heel and toe of each foot. A processing based

C.R. García et al. (Eds.): UCAmI 2016, Part I, LNCS 10069, pp. 111–120, 2016.
DOI: 10.1007/978-3-319-48746-5_12

on gradient analysis extracts the heel strike (HS) and toe off (TO), which are the spatio-temporal parameters used in gait analysis. Finally, the obtained features are processed by a classifier that determines whether the subject presents abnormal gait patterns or not.

Now that the extraction of spatio-temporal parameters and classification has been sufficiently resolved, we focus in providing real time capabilities that could provide a greater variety of applications. With real time capabilities, not only the method could be used to analyse gait, but also to provide feedback to the subject which is precisely the evolution of our project.

1.1 Related Works

Most of the vision based systems proposed can work at real time only for specific computational power, i.e., if the device that perform the computation is not sufficiently faster then the system cannot work under time constraints. These approaches are sufficient when they are executed in specific hardware and do not require scalability. However they cannot be executed in lower computational performance hardware due to their inability to reach the deadline in time. A proper real time application should adapt to the available computational power, providing a suboptimal response if the deadline is reached before finishing the processing. A way to obtain a suboptimal response in our case is the multiresolution pyramid.

The pyramid approach has been widely used in computer vision and computer graphics. The simplest pyramid approach consists in iteratively computing copies of lower resolution of the original image. An example of this is the mipmap used in computer graphics for texture filtering. A multiresolution texture is computed at loading time, then the appropriate resolution is selected depending on the distance at which the texture has to be rendered [15].

According to Lindeberg [4,5], pyramids, wavelets and multi-grid methods were the precursors of the scale space theory developed in image processing, computer vision and signal analysis. Space scale theory deals with multi-scale representation of images (or signals). This states that when building a multiscale representation image, each subsequent downsampling should constitute a simplification of previous level and should not add artefacts due to the smoothing method used. Therefore, the requirement for a kernel is linearity and spatial shift invariance. A kernel that fulfil these premises is the Gaussian kernel and its derivatives.

There are two main approaches for pyramid representation: Gaussian Pyramid and Laplacian Pyramid. In the former, each level of the pyramid is first processed with a gaussian smoothing and then downsampled to 1/2 size, the process is repeated for the number of levels required. In the latter, in each level, a difference of the blurred image of previous level is stored, except the smallest level in which the image is only blurred. The objective is to have different scaled convolution versions of the same image in an efficient way. The orientation invariant approach is called Steerable Pyramid [2].

There are many applications of these approaches. Some of these applications are image compression, detail manipulation, computer graphics... [1,8].

A method for hallucinating high resolution faces from low resolution ones using Steerable Pyramid is proposed in [11]. Strengert et al. proposed the construction of pyramid based filters using the built in bilineal filtering capabilities of Graphics Processing Units (GPU) [10]. An edge aware image processing method using Laplacian Pyramid is proposed by Paris et al. [9]. Yadav et al. [16] present a method based on Gaussian Pyramid to classify microscopic images of hardwood species.

2 Real Time Background Subtraction

In most vision based applications, background subtraction is the initial step followed by noise reduction techniques like morphological operations. Those techniques often reduce the resolution of the silhouette. This resolution downgrade is accepted because the benefits of a de-noised silhouette are greater than those of a high resolution one. Most of the applications do not even require high resolution silhouettes. Assuming this, the first step in order to reduce the computation time of a background subtraction algorithm would be reducing the input image resolution. However, even if the silhouette resolution is not critical, a higher resolution could be beneficial in some cases and usually provides better results. We performed some experiments to test the impact of different resolutions in our gait analysis system. Results of this experiment are shown in Sect. 3.

In our approach the silhouette extraction phase using Mixture of Gaussians [17] is the most computationally expensive task. With Full HD resolution, 90 % of the time is consumed there. This task takes around 90 ms to complete using an Intel Core i7 2630QM CPU at 2.00 GHz and 8 GB RAM DDR3.

For our approach to work under time constraints, we need to redesign this critical phase so it can provide a fast suboptimal response that can be iteratively improved depending on the available time.

2.1 Iterative Multiresolution Processing in Real Time

What we propose is an Iterative Multiresolution Processing in Real Time (IMPReTi) approach that provides increasingly higher resolution silhouettes as long as there is time remaining.

IMPReTi First Approach. The first step is to create a multiresolution image of the original, to do so we proceed creating a mipmap of the input image. The mipmap is organized following the order shown in Fig. 1 using the Algorithm 1. This organization ensures that the first obtained silhouette will be the lower resolution one providing also some steps of some of the next resolutions. However, this also increases the memory and time needed by 50 % (considering a linear algorithm) to complete the maximum resolution. The usable space required is only 1/3 bigger than the original image since the sum $1/4 + 1/16 + 1/64 + \cdots +$

$1/2^{2n}$ converges to 1/3. The mipmap approach [15] achieve this 1/3 increase in space by splitting the RGB channels of the image as shown in Fig. 2. This however will require a modification in the background subtraction algorithm to work with this specific layout.

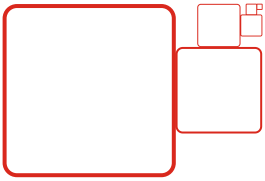

Fig. 1. Mipmaps generated. Overall image size increased by 1/2.

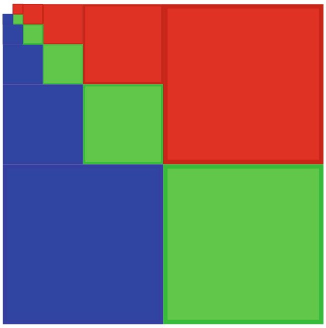

Fig. 2. RGB optimized mipmap. Overall image size increased by 1/3.

Algorithm 1. MipMap organization

$lastRect \leftarrow Rect(0, 0, src.width, src.height)$
$lastImage \leftarrow src$
$dst(lastRect) \leftarrow src$
for $i < numMipMaps$ **do**
  $Resize(lastImage, 0.5)$
  if $imod2 = 0$ **then**
    $lastRect.x \leftarrow lastRect.x + lastRect.width0$
    if i = numMipMaps -1 **then** ▷ The smallest resolution must be put on top so it is processed first
      $lastRect.y \leftarrow 0$
    else
      $lastRect.y \leftarrow lastRect.y + lastRect.height * 0.25$
    end if
  else
    $lastRect.x \leftarrow lastRect.x + lastRect.width * 0.25$
    $lastRect.y \leftarrow 0$
  end if
  $dst(lastRect) \leftarrow lastImage$
end for

Algorithm 2 shows the procedure to work in real time using the mipmap image.

Algorithm 2. Real-time iterative background subtraction first approach

for $i < m$ **do**
  $ProcessLine(i)$
  if $timeexceeded$ **then**
    $break$
  end if
end for
$Result \leftarrow GetMaxResolutionCompleted(i)$ ▷ Depending on the number of lines processed obtain the maximum resolution completed

This approach could provide good results with linear or less complexity algorithms. However, as the size of the image is increased a 50 % (being a 19 % empty) it will significantly increase the processing time required to even compute the lower resolution.

IMPReTi Second Approach. Another solution is to compute separately each resolution, starting with the lowest and iteratively increasing it as long as there is time remaining. We proceed creating different resolution images like in the previous method, however we do not combine them. Then the real time algorithm will be as shown in Algorithm 3. The advantage of this approach is that it only increases the total time and space in a 1/3 rather than a 1/2 like the previous approach (assuming the initialization time of the background subtraction algorithm is minimum and linear complexity) achieving the same spatial increase as the RGB mipmap. As downside, in this approach, each resolution is processed independently so a previously computed resolution do not provide anything to next resolution computation.

Algorithm 3. Real-time iterative background subtraction second approach

$time \leftarrow 0$
for $i < numResolutions$ **do**
  $lastTime \leftarrow CalcTime(Process(resolution[i]))$
  $time \leftarrow time + lastTime$
  if $time + PredictNextResolutionTime(lastTime) > frameTime$ **then**
    $break$
  end if
end for

We propose the use of lower resolution computed to remove noise. To do so we average all the computed resolutions and keep as silhouette only the pixels with value greater than a threshold. This threshold can be empirically adjusted to reduce more or less noise.

Being independent each resolution, multiprocessing techniques could be added to process each in parallel, in which case the algorithm will be as shown in Algorithm 4.

Algorithm 4. Real-time iterative background subtraction second approach in parallel

```
for i < numResolutions do
    StartThread(Process(resolution[i]))
end for
Wait(frameTime)
CancelUnfinishedThreads() ▷ This function should keep in mind that at least one
    resolution has to be completed.
```

Another alternative would be to use the times from the previous frames computed to estimate the maximum resolution that can be computed in the required time and compute only that. The first frames are processed exactly like shown in Algorithms 3 or 4 and then, when a sufficient amount of frames computed could provide an accurate estimation of the required time for each resolution, use the estimator to only compute the maximum resolution possible.

3 Results and Discussion

The first experiment performed was aimed at determining how the resolution of the silhouette affect the final output of the heel strike and toe off detection. We tested four different resolutions (1920×1080, 960×540, 480×270, 240×135 pixels respectively). The silhouette output of each resolution is shown in Fig. 3. We compared the output of the second filter method described in [7].

Fig. 3. Silhouettes computed in each resolution. From left to right: 1080, 540, 270, 135 pixels height.

Figure 4 shows the Root Mean Square Error (RMSE) of the heel strike and toe off detection compared to a manual marking, as described in [6,7], for each resolution. There is practically no difference between 1080 and 540 resolutions in terms of accuracy.

Figure 5 shows the amount of undetected cases for each resolution. In this case, 1080 shows the best results closely follow by 540.

Finally, Fig. 6 shows the processing time required for each resolution as well as the frames per second (FPS) allowed with that amount of time. The processing time increases linearly along with the resolution size, this was as expected since the complexity of the background subtraction algorithm used was linear (number of pixels in the image). This chart shows the greatest differences between each resolution. Therefore it is precisely time the parameter that establishes which resolution to use.

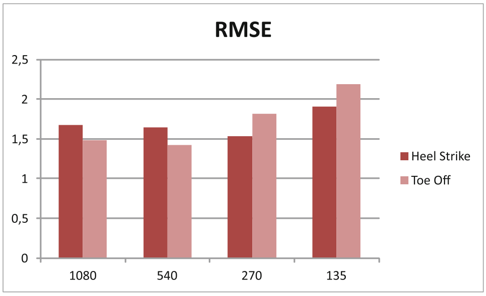

Fig. 4. RMSE result of the heel strike and toe off detection algorithm for each resolution.

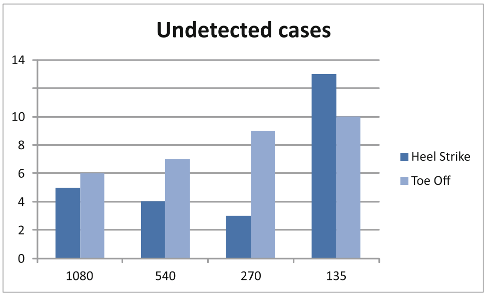

Fig. 5. Amount of undetected cases of heel strike and toe off for each resolution

The second experiment checked the real-time suitability of the Algorithm 3. The IMPReTi algorithm was executed with four different FPS constraints: 15,

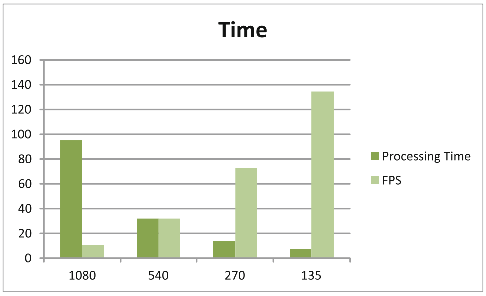

Fig. 6. Processing time in milliseconds and FPS required for silhouette extraction phase for each resolution.

30, 60 and 120; that means the deadline was 66.67 ms, 33.33 ms, 16.67 ms and 8.33 ms respectively. The time estimator was computed as shown in Eq. 1. The time of the next resolution was estimated to be four times bigger than the last since the next resolution have four times more pixels than the previous one. The objective of our project is to develop a low-cost system with a minimal infrastructure needed to execute the algorithm. For that reason we used a general purpose operating system (Windows 7×64) without proper real time capabilities where is difficult to estimate the amount of time certain operation will take to complete. If we add multicore capabilities it becomes even difficult. Therefore we decided to use the theoretical time even if in reality it might be different. A 4 level

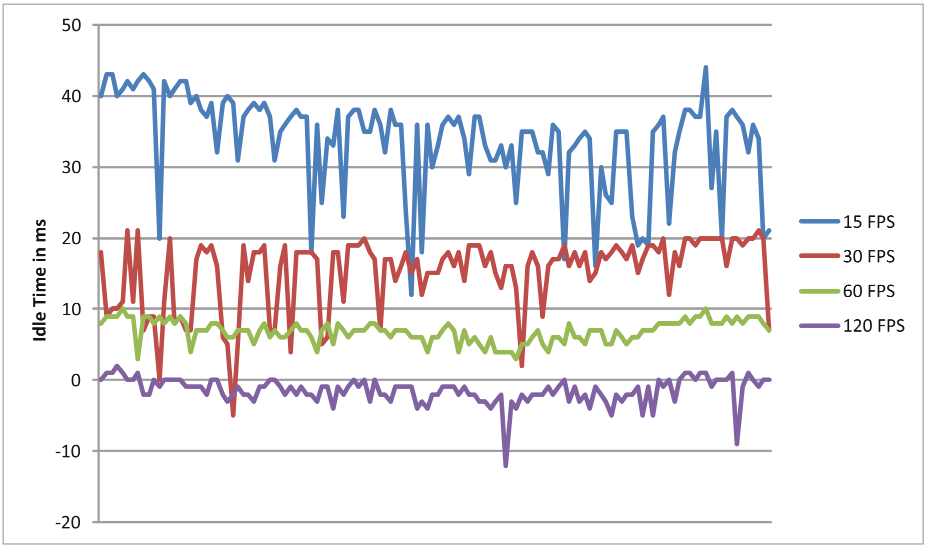

Fig. 7. Graph showing the idle time for each frame with different time constraints. A negative value means that the time constraint is exceeded.

pyramid (1920×1080, 960×540, 480×270, 240×135 pixels) was used and we considered the time between the task was completed (the algorithm determines that the next pyramid level cannot be computed) and the deadline as a measure of adjustment to time constraint. If that measure is zero or positive, that means the algorithm adjusted successfully to the time constraint, otherwise it did not.

$$elapsedTime + lastTime \times 4 < deadline \tag{1}$$

Figure 7 shows the adjustment to the deadline performed by the algorithm with each FPS constraint. The graph represents the idle time for each frame processed, i.e., the remaining time between completing the task and the deadline is reached. As shown in the graph, 120 FPS is the limit at which the algorithm can work with the tested hardware. The reason is that the minimum time required for computing the smaller resolution is greater or equal to the time constraint. Improvements can be made by adding more pyramid levels but smaller resolutions could compromise the accuracy of the results. The graph also shows a great variability in processing time, that is due to the use of a general purpose operating system without proper real time capabilities.

4 Conclusion

The gait analysis algorithm proposed in [6,7] was executed with different input image resolutions to test the impact in the final result. Results of the different resolution execution show that, although small differences, it is desirable to use a bigger resolution. The bigger the resolution the better the results, however the time also increases. Therefore there is an inverse relation between the quality of the output and the required time to obtain it. Working with time constraints, it might not be possible to compute the higher resolution, and there is where the benefits of the IMPReTi algorithm appear as it iteratively process resolutions from smaller to bigger until the deadline is reached. The IMPReTi algorithm proposed was tested with different time constraints and a 4-level pyramid. Results show that the time constraints are maintained except in the case of 120 FPS where the processing time of the smallest resolution is sometimes bigger than the time constraint.

Acknowledgements. This research is part of the FRASE MINECO project (TIN2013-47152-C3-2-R) funded by the Ministry of Economy and Competitiveness of Spain.

References

1. Adelson, E.H., Anderson, C.H., Bergen, J.R., Burt, P.J., Ogden, J.M.: Pyramid methods in image processing. RCA Eng. **29**(6), 33–41 (1984)
2. Freeman, W.T., Adelson, E.H.: The design and use of steerable filters. IEEE Trans. Pattern Anal. Mach. Intell. **13**(9), 891 (1991)

3. Fried, L.P., Tangen, C.M., Walston, J., Newman, A.B., Hirsch, C., Gottdiener, J., Seeman, T., Tracy, R., Kop, W.J., Burke, G., et al.: Frailty in older adults evidence for a phenotype. J. Gerontol. Ser. A: Bio. Sci. Med. Sci. **56**(3), M146–M157 (2001)
4. Lindeberg, T.: Scale-space theory: a basic tool for analyzing structures at different scales. J. Appl. Stat. **21**(1–2), 225–270 (1994)
5. Lindeberg, T.: Scale-Space Theory in Computer Vision, vol. 256. Springer Science & Business Media, New York (2013)
6. Nieto-Hidalgo, M., Ferrández-Pastor, F.J., Valdivieso-Sarabia, R.J., Mora-Pascual, J., García-Chamizo, J.M.: Vision based extraction of dynamic gait features focused on feet movement using RGB camera. In: Bravo, J., Hervás, R., Villarreal, V. (eds.) AmIHEALTH 2015. LNCS, vol. 9456, pp. 155–166. Springer, Heidelberg (2015). doi:10.1007/978-3-319-26508-7_16
7. Nieto-Hidalgo, M., Ferrández-Pastor, F.J., Valdivieso-Sarabia, R.J., Mora-Pascual, J., García-Chamizo, J.M.: A vision based proposal for classification of normal and abnormal gait using RGB camera. J. Biomed. Inf. **63**, 82–89 (2016)
8. Ogden, J.M., Adelson, E.H., Bergen, J.R., Burt, P.J.: Pyramid-based computer graphics. RCA Eng. **30**(5), 4–15 (1985)
9. Paris, S., Hasinoff, S.W., Kautz, J.: Local laplacian filters: edge-aware image processing with a laplacian pyramid. Commun. ACM **58**(3), 81–91 (2015)
10. Strengert, M., Kraus, M., Ertl, T.: Pyramid methods in gpu-based image processing. In: Proceedings Vision, Modeling, and Visualization 2006, pp. 169–176 (2006)
11. Su, C., Zhuang, Y., Huang, L., Wu, F.: Steerable pyramid-based face hallucination. Pattern Recogn. **38**(6), 813–824 (2005)
12. Verghese, J., Lipton, R.B., Hall, C.B., Kuslansky, G., Katz, M.J., Buschke, H.: Abnormality of gait as a predictor of non-alzheimer's dementia. N. Engl. J. Med. **347**(22), 1761–1768 (2002)
13. Waite, L., Grayson, D., Piguet, O., Creasey, H., Bennett, H., Broe, G.: Gait slowing as a predictor of incident dementia: 6-year longitudinal data from the sydney older persons study. J. Neurol. Sci. **229**, 89–93 (2005)
14. Waltson, J., Fried, L.: Frailty and the old man. Med. Clin. North Am. **83**(5), 1173–1194 (1999)
15. Williams, L.: Pyramidal parametrics. In: ACM Siggraph Computer Graphics, vol. 17, pp. 1–11. ACM (1983)
16. Yadav, A.R., Anand, R., Dewal, M., Gupta, S.: Gaussian image pyramid based texture features for classification of microscopic images of hardwood species. Optik-Int. J. Light Electron Opt. **126**(24), 5570–5578 (2015)
17. Zivkovic, Z., Heijden, F.: Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recogn. Lett. **27**(7), 773–780 (2006)