
Peer reviewed version

Link to published version (if available):
10.1049/ic.2016.0060

Link to publication record in Explore Bristol Research

This is the author accepted manuscript (AAM). The final published version (version of record) is available online via IEEE at http://ieeexplore.ieee.org/document/7801344/?arnumber=7801344. Please refer to any applicable terms of use of the publisher.

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
http://www.bristol.ac.uk/pure/about/ebr-terms.html
Real-time Estimation of Physical Activity Intensity for Daily Living

L. Tao, T. Burghardt, M. Mirmehdi, D. Damen, A. Cooper, M. Camplani, S. Hannuna, A. Paiement, I. Craddock

SPHERE, Faculty of Engineering, University of Bristol, Bristol, UK, BS8 1UB

Keywords: Assistive monitoring, computer vision, energy expenditure, activities of daily living

Abstract

Estimating a person’s energy expenditure and activity intensity over time is an important component in managing various health conditions or tracking lifestyle choices. To implement an automatic estimation, most current systems ultimately require users to wear sensor devices. In contrast, this paper presents a framework for the contact-free, real-time estimation of energy expenditure, applicable to daily living scenarios. This is a new application in real-time computer vision. We demonstrate the effectiveness and the benefits of utilising a basic set of features and evaluate the resulting framework on the challenging SPHERE-calorie dataset. To ensure accurate evaluation, automated estimates are compared against a simultaneously taken indirect calorimetry ground truth based on per breath gas exchange. Following detailed experiments, we conclude that the proposed real-time vision pipeline is suitable for monitoring physical activity levels in a controlled environment with higher accuracy than the commonly used manual estimation via metabolic lookup tables (METs), whilst being significantly faster than existing automated methods.

1 Introduction

Physical activity is an important determinant in understanding the development of chronic diseases. Current evidence-based guidelines indicate that people who are regularly physically active have a 20% to 40% lower risk of developing conditions such as cardiovascular disease and type 2 diabetes than those who are inactive, and suggest that adults should accumulate at least 150 minutes of moderate intensity physical activity each week or 75 minutes of vigorous activity, or a combination of the two.

Energy expenditure, also referred to as ‘calorific expenditure’, is one commonly used single metric to quantify physical activity levels over time. It can be accurately measured using a calorimeter which operates based on the respiratory differences of oxygen and carbon dioxide in the inhaled and exhaled air. Measurements can either be direct via a sealed respiratory chamber or indirect which requires carrying gas sensors and wearing a breathing mask. However, these devices are impractical to use routinely in daily life due to their high cost, lack of portability and cumbersomeness. On the other hand, wearable devices have become a popular choice to measure coarse categorisation of activity intensity levels. Among these, tri-axial accelerometers are the most broadly used inertial sensors.

Computer vision techniques that help with the diagnosis and management of health and wellbeing conditions have also started to draw some research attention recently. Yet, although there exists a significant body of literature describing the inference of activities from 2D colour intensity imagery, RGB-D data, and skeleton-based data, studies on energy expenditure using visual sensors have been relatively limited. RGB only video has recently been used by Edgcomb and Vahid to coarsely estimate daily energy expenditure where a subject was segmented from the scene background and changes in height and width of the subject’s motion bounding box, together with vertical and horizontal velocities and accelerations, were then used to estimate calorific uptake. However, we note that their regression models were trained based on the ground truth readings reported by wearable accelerometer, which may provide only an approximate benchmark. Tsou and Wu took this idea further and estimated calorie consumption using full 3D joint movements tracked as skeleton models using a Microsoft Kinect. In this setting skeleton data is commonly noisy and currently only operates reliably when the subject faces the camera, thus the method has difficulties to generalise to more unconstrained scenarios.

The above examples exemplify that calorific uptake and linked activity levels can often be directly related to body motion. Motion information could also be recovered directly using the optical flow derived from two adjacent RGB images or 4D surface normals and more recently, dense 3D flow from depth images. These approaches, however, often suffer from unaffordable run time, for example with a reported computation time of up to 9 minutes per frames.

Apart from the currently performed activity, energy values are also highly dependent on the previous energy expenditure, as adaptations of the human body cause an exponential increase/decrease to a plateau in oxygen consumption until a steady state corresponding to the current activity is attained. Therefore, the motion information needs to be recovered from a sequence of data over some time window to infer calorific uptake levels. Concatenating per-frame descriptors is straightforward, but it often suffers from the curse of dimensionality and related high computational cost. Compacting data within a temporal window may be achieved to some degree by abstracting large feature arrays, but remains a challenge. Thus, in essence, any system will require capturing...
visual aspects relevant to calorific expenditure, whilst limiting
the dimensionality of the descriptor.

Here, we propose a real-time framework for estimating
calorific expenditure levels from bounding box features only.
We evaluate the proposed system over daily activities per-
formed in a living room environment. Figure 1 shows in bold a
flowchart of our proposed approach – extracting features from
the bounding box, mapping the features directly to calorie esti-
mates via a monolithic classifier, and adding a cascaded and re-
current classifier as the last step to capture temporal dependen-
cies (RM in short). The proposed method is compared against
a ground truth as-exchange measurements (GT in short) and
two alternative methods also shown Figure 1: (1) direct map-
ning to calorie estimates without recurrent approach (DM in
short), and (2) manual mapping from activity classes to calo-
rie estimates via the Metabolic Equivalent Task lookup tables
[4] (MET in short). We also compare the processing time
of feature extraction and estimation accuracy of the proposed
method against that of a fully fledged vision system [21], which
uses detailed flow and depth features at the cost of sacrificing
real-time capabilities. We will show that the proposed system
can operate under real-time constraints whilst achieving accu-
rate activity intensity level prediction outperforming the widely
used MET method.

2 Proposed Method

2.1 Feature Representation

We first extract base features from bounding boxes, and then
form higher level motion features by a set of temporal filters.
We use the bounding box returned by the OpenNI SDK [13]
person detector and tracker using an Asus Xtion for capture.
Our per-frame descriptor describes the velocity vector and the
ratio of height and width of the bounding box.

To represent both short and long term temporal changes in
a video, one may model how the local/global information is
changing over time. Pooled motion features were first pre-
sented in [18], designed for egocentric video analysis. We
modify this pooling operator to make it more suitable for our
data.

Figure 2 illustrates the overall process of feature assem-
bly covering the initial time series representation, followed by
temporal pyramid alignment, and the final, serialised represen-
tation of the descriptor vector.

Let S be a set of time series data, such that $S = \{S_1, \ldots, S_N\}$, $S \in \mathbb{R}^{N \times T}$ for a video in matrix form, where
$N$ is the length of the per-frame feature vector, and $T$ is
the number of frames. A time series $S_n = [s_n(1), \ldots, s_n(T)]$ is
the $n^{th}$ feature across $1, \ldots, T$ frames, where $s_n(t)$ denotes
$n^{th}$ feature at frame $t$. The time series data $S$ consists of a set
of time segments as $S = [S_1^i, \ldots, S_T^i]$ at level $i$. A set of tem-
poral filters with multiple pooling operators is applied to each
time segment $[t_{min}, t_{max}]$ and produces a single feature vector
for each segment via concatenation.

As Figure 2 illustrates, we use three pooling operators, that
is max pooling, sum pooling, and spectral pooling. The first
two are defined respectively as

$$O_{\text{max}}(S_n) = \max_{t=t_{\text{min}}}^{t_{\text{max}}} s_n(t)$$  \hspace{1cm} (1)

$$O_{\text{sum}}(S_n) = \sum_{t=t_{\text{min}}}^{t_{\text{max}}} s_n(t).$$  \hspace{1cm} (2)

Spectral pooling is used to perform dimensionality reduction
of the time series $S_n$ in the frequency domain by the discrete
cosine transform and then truncating the representation. The
pooling operator takes the absolute value of the $j$ lowest fre-
cquency components of the frequency coefficients $D$, in order
to help remove high frequency noise

\[ O_{dct}(S_n) = |M_{1:n}S_n| , \] (3)

where \( M \) is the discrete cosine transformation matrix.

2.2 Recurrency

We pose the energy expenditure estimation problem as a sequential and supervised regression task. We train a support vector regressor to predict calorie values from the given features over a training set. The sliding window method naturally converts the sequential supervised learning problem into the classical supervised learning problem, which maps each input window of width \( w \) to an individual output value \( y_t \). The window contains the current and the previous \( w - 1 \) observations. The window features are assembled by temporal pooling from the time series \( S = \{S_{t-w+1}, \ldots, S_t\} \).

The energy values for a particular time are highly dependent on the energy expenditure history, thus the sliding window methods can be extended by including recurrent information. In our system, these are most directly expressed by previous calorific predictions during operation. Thus, employing recurrent sliding windows offers an option to not only use the features within a window, but also take the most recent \( d \) predictions \( \{\hat{y}_{t-d}, \ldots, \hat{y}_{t-1}\} \) into consideration to help predict \( y_t \). During learning, as suggested in [9], the ground truth labels in the training set are used in place of recurrent values.

3 Experimental Results

3.1 Dataset and Parameter Settings

In order to quantify performance of the proposed approach, we conducted experiments on the SPHERE-calorie dataset[1][21]. It is a very challenging dataset for calorific expenditure estimation collected within a home environment covering daily living activities. The dataset consists of an RGB-D video sequences captured by a Asus Xtion camera mounted at the corner of a living room and ground truth readings from a COSMED K4b2 portable metabolic measurement system. The dataset was generated over 20 sessions by 10 subjects with varying anthropometric measurements containing up to 11 common household activity categories per session. Each session lasts around 30 minutes, and totalling around 10 hours recording time. To reflect variations in transitions between activity levels, we consider 9 different combinations of three activity intensities in each session.

Colour and depth images were acquired at a rate of 30Hz. The calorimeter gives readings per breath, which occur approximately every 3 seconds. Figure[3] shows a detailed example of calorimeter readings and associated sample RGB images from the dataset, together with activity intensity levels. The raw breath data is noisy (in red), and so we apply an average filter with a span of approximately 20 breaths (in blue). The participants were asked to perform the scripted activities based on their own living habits without any extra instructions.

The categories and their associated MET values (in brackets) are: sit still (1.3), stand still (1.3), lying down (1.3), reading (1.5), walking (2.0), wiping table (2.3), cleaning floor stain (3.0), vacuuming (3.3), sweeping floor (3.3), squatting (5.0), upper body exercise (4.0).

We compare the proposed method RM to the direct mapping method DM and the Metabolic Equivalent Task method MET. DM is formalised as \( Y_t = f(X_t) \), where \( Y_t \) is the target calorie value regardless of activity at time \( t \), and \( X_t \) contains the associated feature vector over a window. The goal is to find a function \( f(\cdot) \) that best predicts \( Y_t \) from training data \( X_t \). MET, one of the widely used methods for recording of the intensity of a physical activity by clinicians and physiotherapists, assumes that the clusters of activity are known. A MET value is assigned to each cluster, together with anthropometric characteristics of individuals. The amount of activity-specific energy expended can then be estimated as \( \text{energy} = 0.0175(\text{cal/kg/min}) \times \text{weight(kg)} \times \text{MET values} \).

According to [4], activity can be categorized into three different intensity levels based on either MET values for each activity or the average energy consumed per minute. Table[1] outlines the activity intensity levels and their associated energy expenditure ranges. In our experiments, activity intensity levels...
are quantised based on the ground truth readings (see Figure 3) instead of using MET values. This is because a fixed number is assigned to each activity which may overlook the drift during activity and transition between activities.

We use a linear support vector regressor for predicting calorie values from training data. The libsvm [7] implementation was used in the experiments. For testing, we apply leave-one-subject-out cross validation on the dataset.

This process iterates through all subjects, and the average testing errors of all iterations are reported. We use the root-mean-squared error (RMSE) as a standard evaluation metric for the deviation of estimated calories from the ground truth.

Table 1. Physical Activity Intensity Levels. The table shows intensity levels and their associated energy expenditure ranges.

<table>
<thead>
<tr>
<th>Intensity</th>
<th>MET values</th>
<th>kcal/min</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light</td>
<td>&lt; 3.0</td>
<td>&lt; 3.5</td>
</tr>
<tr>
<td>Moderate</td>
<td>3.0 - 6.0</td>
<td>3.5 - 7</td>
</tr>
<tr>
<td>Vigorous</td>
<td>&gt; 6.0</td>
<td>&gt; 7</td>
</tr>
</tbody>
</table>

Table 3 shows the effect of using recurrent information. The best results for each activity are highlighted. In general, RM1 outperforms the other approaches for most activities. As expected, a recurrent method captures information that was not only being captured by the current sliding window. However, the full recurrency, RM2, suffers from significant drift and produces the worst results for half of the activities and also overall. We select RM1 as the “the proposed method RM” in the following sections.

Model Comparison - Table 6 provides the results for each sequence of the dataset individually. We report estimation accuracy and also the correlation between the ground truth and the observed values. The proposed RM achieves higher accuracy and correlation in more sequences than DM and MET based methods, and obtains better rates on average. In addition, we compare performance to a system using complex visual features (VF in short) [21] instead of bounding box features only. VF produces, as expected, better prediction results in most cases, however, RM operates more than 400 times faster than VF as detailed in the next section.

Looking at coarse categories of calorific expenditure, Table 4 lists overall results for the accuracy of predicting activity intensity levels only. For this task it is worth noticing that the proposed RM is able to produce comparable results to VF.

Processing Time - To analyse the efficiency of the proposed method further, we compare the processing time of feature construction procedures in RM and in VF where the average runtime of each frame though all subjects is reported. All the re-
Table 2. Temporal Window Size and Calorific Expenditure Prediction. Calorific expenditure prediction error (RMSE) with different window length. The best results in each activity are in bold.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Window Size (ms)</th>
<th>RM</th>
<th>DM</th>
<th>MET</th>
<th>VF</th>
</tr>
</thead>
<tbody>
<tr>
<td>standing</td>
<td>450</td>
<td>0.68</td>
<td>0.76</td>
<td>0.92</td>
<td>0.84</td>
</tr>
<tr>
<td>sitting</td>
<td>900</td>
<td>0.65</td>
<td>0.72</td>
<td>0.92</td>
<td>0.83</td>
</tr>
<tr>
<td>walking</td>
<td>1800</td>
<td>0.60</td>
<td>0.68</td>
<td>0.90</td>
<td>0.80</td>
</tr>
</tbody>
</table>

Table 3. Activities and Calorific Expenditure Prediction. Average calorific expenditure prediction errors (RMSE) for each activity with different learning approaches. The best results in each activity are in bold.

<table>
<thead>
<tr>
<th>Feature Extraction (ms)</th>
<th>Temporal Pooling (ms)</th>
<th>Overall (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RM</td>
<td>7.21E-04</td>
<td>1.4</td>
</tr>
<tr>
<td>VF</td>
<td>446.6</td>
<td>449.9</td>
</tr>
</tbody>
</table>

Table 4. Estimation of Activity Intensity Levels. Recognition accuracy (%) of activity intensity levels. The table shows the performance of the proposed real-time method RM, compared to other approaches.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Light</th>
<th>Moderate</th>
<th>Vigorous</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>RM</td>
<td>0.87</td>
<td>0.60</td>
<td>0.80</td>
<td>1.31</td>
</tr>
<tr>
<td>DM</td>
<td>0.57</td>
<td>0.68</td>
<td>0.90</td>
<td>1.40</td>
</tr>
<tr>
<td>MET</td>
<td>0.57</td>
<td>0.68</td>
<td>0.90</td>
<td>1.40</td>
</tr>
<tr>
<td>VF</td>
<td>0.82</td>
<td>0.69</td>
<td>0.82</td>
<td>1.31</td>
</tr>
</tbody>
</table>

Table 5. Runtime Performance Results. Average computational costs (in milliseconds) for each frame processed by the VF and DM methods.

4 Conclusion

This paper presented a real-time vision system for a contact-free estimation of calorific expenditure estimation in daily living scenarios. The proposed method used pooled temporal pyramids of bounding box features, and subsequently built a recurrent sliding window approach upon it. We demonstrated the effectiveness and efficiency of the proposed method via detailed experiments on accuracy and runtime performance in a comparative study.

The proposed method shows its ability to outperform the widely used METs estimation approach in estimating calorie expenditure, and to provide results in the same region of accuracy of an approach using complex visual features in estimating activity intensity levels, at a fraction of the computational cost. Future work will include investigating fusion approaches for improving prediction results based on both visual and inertial sensors.

Acknowledgements

This work was performed in the SPHERE IRC project funded by the UK Engineering and Physical Sciences Research Council (EPSRC), Grant EP/K031910/1.

References

Table 6. Estimating Caloric Expenditure. The table shows ground truth and predicted calorie values in total per sequence and its accuracy and correlation. The best results for each sequence are in bold. (Note that the total calorie value for sequence 5, 8, 11, 15 and 16 are relatively low due to shorter sequences caused by camera errors.)

<table>
<thead>
<tr>
<th>sequence</th>
<th>Prediction (Calories)</th>
<th>Accuracy %</th>
<th>Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GT RM DM MET VF</td>
<td>RM DM MET VF</td>
<td>RM DM MET VF</td>
</tr>
<tr>
<td>1</td>
<td>59 75 78 76 71</td>
<td>73.1 68.4 71.3 80.2</td>
<td>0.46 0.70 0.66 0.83</td>
</tr>
<tr>
<td>2</td>
<td>89 87 86 78 80</td>
<td>97.7 96.5 88.2 90.3</td>
<td>0.74 0.69 0.57 0.85</td>
</tr>
<tr>
<td>3</td>
<td>74 74 93 69 81</td>
<td>74.9 81.5 92.7 90.1</td>
<td>0.90 0.87 0.63 0.84</td>
</tr>
<tr>
<td>4</td>
<td>79 52 51 43 48</td>
<td>66.1 65.2 55.0 60.4</td>
<td>0.84 0.82 0.78 0.87</td>
</tr>
<tr>
<td>5</td>
<td>37 46 46 28 39</td>
<td>75.1 73.9 77.6 98.6</td>
<td>0.87 0.84 0.77 0.90</td>
</tr>
<tr>
<td>6</td>
<td>89 86 84 107 86</td>
<td>96.7 95.2 79.9 94.3</td>
<td>0.83 0.74 0.63 0.82</td>
</tr>
<tr>
<td>7</td>
<td>101 97 96 114 96</td>
<td>96.1 95.1 87.6 95.3</td>
<td>0.58 0.39 0.61 0.61</td>
</tr>
<tr>
<td>8</td>
<td>39 40 42 35 42</td>
<td>96.8 92.8 84.4 91.9</td>
<td>0.72 0.67 0.57 0.93</td>
</tr>
<tr>
<td>9</td>
<td>82 79 80 94 76</td>
<td>96.7 98.8 85.3 92.8</td>
<td>0.81 0.73 0.71 0.86</td>
</tr>
<tr>
<td>10</td>
<td>49 79 80 68 76</td>
<td>38.9 37.7 45.2 61.5</td>
<td>0.58 0.56 0.42 0.54</td>
</tr>
<tr>
<td>11</td>
<td>28 44 45 38 38</td>
<td>43.5 40.0 66.8 65.5</td>
<td>0.56 0.45 0.56 0.64</td>
</tr>
<tr>
<td>12</td>
<td>98 93 91 79 88</td>
<td>95.3 92.5 80.8 90.3</td>
<td>0.84 0.73 0.66 0.56</td>
</tr>
<tr>
<td>13</td>
<td>36 84 82 77 77</td>
<td>49.4 52.6 62.7 82.3</td>
<td>0.80 0.75 0.62 0.78</td>
</tr>
<tr>
<td>14</td>
<td>141 87 87 74 84</td>
<td>61.8 61.8 52.8 57.4</td>
<td>0.81 0.71 0.60 0.86</td>
</tr>
<tr>
<td>15</td>
<td>40 49 48 30 41</td>
<td>77.6 78.7 74.5 98.9</td>
<td>0.84 0.86 0.94 0.94</td>
</tr>
<tr>
<td>16</td>
<td>29 32 33 38 31</td>
<td>89.7 83.9 69.1 97.3</td>
<td>0.71 0.74 0.81 0.88</td>
</tr>
<tr>
<td>17</td>
<td>81 81 82 100 85</td>
<td>99.3 97.9 76.0 94.2</td>
<td>0.38 0.30 0.70 0.74</td>
</tr>
<tr>
<td>18</td>
<td>65 82 83 94 86</td>
<td>73.7 71.7 54.7 69.3</td>
<td>0.25 0.14 0.48 0.83</td>
</tr>
<tr>
<td>19</td>
<td>92 86 85 101 89</td>
<td>93.2 91.8 90.6 94.2</td>
<td>0.88 0.75 0.72 0.75</td>
</tr>
<tr>
<td>20</td>
<td>63 83 84 86 83</td>
<td>67.5 66.0 64.4 66.9</td>
<td>0.53 0.56 0.41 0.81</td>
</tr>
<tr>
<td>Average</td>
<td>- - - - -</td>
<td>78.2 77.1 73.7 82.9</td>
<td>0.60 0.56 0.64 0.79</td>
</tr>
</tbody>
</table>


