Learning Discriminative Representations from RGB-D Video Data

Li Liu
Department of Electronic
and Electrical Engineering
University of Sheffield
S1 3JD UK
elp11ll@sheffeild.ac.uk

Ling Shao
Department of Electronic
and Electrical Engineering
University of Sheffield
S1 3JD UK
ling.shao@sheffield.ac.uk

Abstract

Recently, the low-cost Microsoft Kinect sensor, which can capture real-time high-resolution RGB and depth visual information, has attracted increasing attentions for a wide range of applications in computer vision. Existing techniques extract hand-tuned features from the RGB and the depth data separately and heuristically fuse them, which would not fully exploit the complementarity of both data sources. In this paper, we introduce an adaptive learning methodology to automatically extract (holistic) spatio-temporal features, simultaneously fusing the RGB and depth information, from RGB-D video data for visual recognition tasks. We address this as an optimization problem using our proposed restricted graph-based genetic programming (RGGP) approach, in which a group of primitive 3D operators are first randomly assembled as graph-based combinations and then evolved generation by generation by evaluating on a set of RGB-D video samples. Finally the best-performed combination is selected as the (near-)optimal representation for a pre-defined task.

The proposed method is systematically evaluated on a new hand gesture dataset, SKIG, that we collected ourselves and the public MSRDailyActivity3D dataset, respectively. Extensive experimental results show that our approach leads to significant advantages compared with state-of-the-art handcrafted and machine-learned features.

1 Introduction

Since Kinect was invented as an RGB-D sensor successfully capturing synchronized color and depth information, it has been widely applied with a large range of applications such as: human activity recognition [Wang *et al.*, 2012], robot path planning [Paton and Kosecka, 2012], object detection [Spinello and Arras, 2012], and interactive gaming [Cruz *et al.*, 2012]. The complementary nature of the RGB and depth information enables enhanced computer vision algorithms. A recent survey on the use of Kinect for improving various vision applications can be found in [Han *et al.*, 2013] . To make full use of the Kinect sensor, extracting distinctive and discriminative features from the raw RGB-D data is an imminent and challenging research topic.

Feature extraction from RGB video data is a well-explored area. Methods such as: histogram of 3D oriented gradients (HOG3D) [Kläser *et al.*, 2008], histogram of optical flow (HOF) [Laptev *et al.*, 2008], 3D speeded up robust features (SURF3D) [Bay *et al.*, 2008], 3D scale invariant feature transforms (3D-SIFT) [Scovanner *et al.*, 2007] and volume local binary pattern (volume-LBP) [Ojala *et al.*, 2002] are used to extract the most salient features (edges, corners, orientation, and motion), the choice of which would greatly influence the performance of high-level vision tasks such as recognition.

Since depth cameras are relatively new, feature extraction from depth data is still in the early stage. Most of the current approaches apply the same or slightly adapt extraction techniques in the RGB domain. Obviously, the reliability of such techniques for depth data is questionable due to the different characteristics between RGB and depth data, *e.g.*, texture and color information on the depth data is much less than that on the RGB data.

Thus, how to exploit RGB-D data and optimally combine the sensory modalities so as to extract the discriminative information for further tasks is the topic addressed in this paper.

1.1 RGB-D information fusion

Several solutions have been introduced, at a high level, to perform information fusion of RGB -D data. An intuitive fusion scheme was used for the Kinect gaming system [Cruz *et al.*, 2012], in which different algorithmic modules are selected to extract the meaningful information from the RGB and depth channels, respectively. Specifically, depth data are always used to extract a player's skeleton, while RGB data are regarded as the inputs of facial feature based human identification algorithms. Another scheme simply feeds all available information (visual features) into an optimization algorithm. Such an example can be found in RGB-D object detection and recognition [Spinello and Arras, 2012; Lai *et al.*, 2011]. In addition, some researchers [Spinello and Arras, 2011] combined multiple modalities such as dense depth data, optical flow maps, and color images to detect people in urban environments. Ni *et al.* [Ni *et al.*, 2011] developed two color-depth fusion techniques to extract the most discriminative features for human action recognition.

Although the above fusion schemes have been successfully applied to some vision problems, they are not sophisticated and adaptive enough for different RGB-D data sources. Sometimes a crude fusion algorithm would even degrade and slow down a vision system. Therefore, it is highly desirable to develop an adaptive and simultaneous RGB-D feature extraction and fusion methodology.

1.2 Feature Learning and motivations

To overcome the deficiency of hand-designed features, recently, some learning algorithms such as the deep belief network (DBN) [Larochelle *et al.*, 2007], the convolutional deep belief network (CDBN) [Lee *et al.*, 2009] and the convolutional neural network (CNN) [Ranzato *et al.*, 2007] have been utilized to automatically learn multiple layers of non-linear features from images and videos. In these architectures, the features are extracted at different levels (*e.g.*, edges, object parts, and objects). These feature learning methods are very powerful and the learned features can be used for various high-level applications. However, they have not been applied to learn the fusion procedure of RGB-D data. In this work, we attempt to adopt another powerful learning approach, *i.e.*, genetic programming, to learn feature extraction and fusion from RGB-D data. We consider GP to be flexible and adaptive and can easily evolve features according to desired objectives.

Genetic programming (GP), as a prevailing evolutionary computation method, has been gradually adopted in computer vision applications [Trujillo and Olague, 2006; Torres *et al.*, 2009; Poli, 1996]. Usually a group of processing operators are first randomly assembled into a variety of programs as the initialized population and then GP evolves (hopefully) better-performing individuals in the next generation. GP search is an NP-hard problem, which can be solved by evolutionary methods in a tractable amount of computer time compared to the exhaustive enumerative search. After the training of GP finishes, one best-so-far individual can be selected as the final solution (*i.e.*, the near-optimal solution).

Later, to drive GP to be a more natural procedure, Poli *et al*. [Poli and others, 1997] introduced graph-based GP (GGP), a form of GP that is suitable for the evolution of highly parallel programs which effectively reuse partial results. Programs are represented in GGP as graphs with nodes representing functions and terminals. Edges represent both control flow and data flow. Recently, some researchers have also attempted to use GP for 3D vision tasks. For example, GP has been applied to extract discriminative features for action recognition [Liu *et al.*, 2012] and gesture recognition [Liu and Shao, 2013].

In this paper, inspired by previous successful applications, we aim to develop an effective learning methodology to learn spatio-temporal features (representations), which simultaneously fuse the RGB and depth information, for high-level recognition tasks. We address this issue as an optimization problem using our proposed restricted graph-based genetic programming (RGGP), in which some grammatical constraints are added to restrict the graph-based programs into a relatively fixed structure according to the problem domain knowledge. The features learned through the proposed RGGP are proved to yield significantly superior recognition accuracies compared with state-of-the-art hand-crafted and machine-learned features.

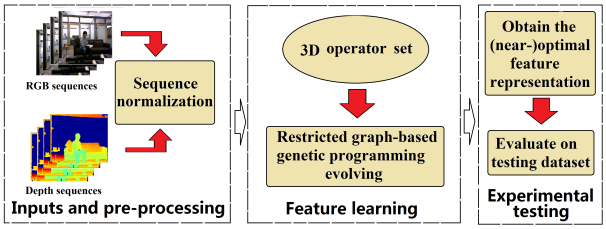

Figure 1: The main flowchart for our proposed method.

1.3 Contributions

The main contributions of this paper lie in the following three aspects:

(1) To the best of our knowledge, this is the first application of GP to learn discriminative spatio-temporal features from RGB-D data for high-level tasks.

(2) Our proposed RGGP leaning mechanism provides an effective way to simultaneously extract and fuse the color and depth information into one feature representation.

(3) We introduce a new dataset - **S**heffield **KI**nect **G**esture (SKIG) dataset for hand gesture recognition. The details of this new dataset are described in Section 3.2.

The remainder of the paper is organized as follows: The architecture of our methodology is detailed in Section 2. Experiments and results are described in Section 3. In Section 4, we conclude this paper.

2 Spatio-Temporal Feature Learning

In this paper, a domain-adaptive machine learning method has been developed by applying the restricted graph-based genetic programming (RGGP) to automatically extract discriminative spatio-temporal features from videos for high-level vision tasks. In our architecture, the Kinect captured RGB and depth sequences are the inputs, and a group of 3D operators are adopted to evolve individual programs, which can successfully fuse the RGB and depth information, and finally a discriminative feature representation is obtained. We learn our proposed system over a training set, in which computed features are evolved by minimizing the error rate of recognition through the defined fitness function, and then the selected best-performing program (feature) is evaluated on a testing set to demonstrate the effectiveness of our method. This section describes the proposed program structure along with the required terminal set, function set and fitness function. The outline of our method is illustrated in Fig. 1.

2.1 Building Multi-layer Program Structure

To efficiently learn discriminative and meaningful features from RGB-D sequences, here we propose the restricted graph-based genetic programming (RGGP), in which certain syntactic constraints are defined according to the domain knowledge of corresponding tasks. In our architecture, we

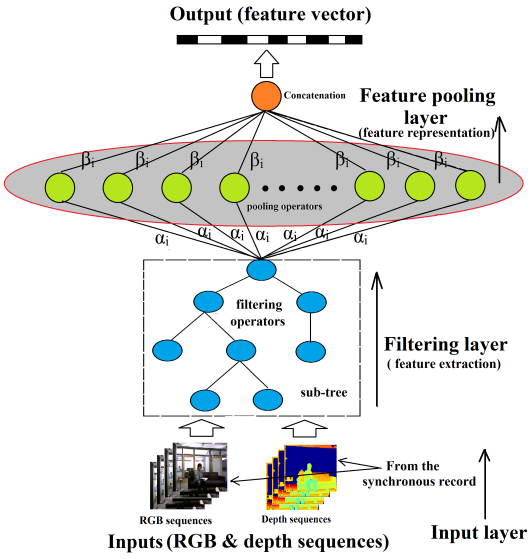

Figure 2: The multi-layer program structure of an RGGP individual. This proposed structure can effectively extract and fuse the RGB and depth information into one feature representation through the filtering layer and the feature pooling layer.

restrict our RGGP feature extraction programs into three layers: input layer, filtering layer and feature pooling layer. The order of the layers is always fixed: the input layer is at the bottom, the filtering layer is in the middle and the feature pooling layer is on the top (including the root of the program). The proposed architecture is consistent with the physical structure of the human visual cortex [Lee and Mumford, 2003] illustrated in Fig. 2.

The input layer is a single-depth layer serving as the entry of the programs. All the RGB and depth sequences are first normalized into the identical size along the spatial and temporal dimensions and then fed to the RGGP programs. The filtering layer is located above the input layer. In this layer, a set of 3D operators are automatically assembled into a dynamic-depth tree-structured chain to extract the features from the input RGB-D data. To ensure the closure criterion [Poli *et al.*, 2008], inputs and outputs of the nodes (operators) in the filtering layer must have the same size. After feature extraction in the filtering layer, all the data are processed by the third layer: feature pooling layer, in which two kinds of popular pooling techniques, *i.e.*, max pooling and average pooling, are selected by RGGP to get the most robust and distinctive response on different data resolutions. To generate a reasonable and task-adaptive structure for feature representation, in this layer, we further set a series of grammatical constraints (details in Section 2.3) to guarantee that the evolutionary search process will better suit our task specifications. Thus, the designed programs can be described by a restricted graph-based structure, where only the feature pooling layer is allowed to compose a graph map with a full-connection between the output of the filtering layer and the pooling operators shown in Fig. 2. The final output of a program (the value of the root node) is the learned feature vector which is concatenated by the representations computed from all the pooling operators in the RGGP architecture. Our proposed RGGP architecture can simultaneously learn useful features from the input data and fuse the RGB and depth information for further applications.

Table 1: Statement of terminal nodes

Terminal	Type	Description
V_{rgb}	Sequence	RGB information of the data
V_{depth}	Sequence	Depth information of the data
α_i	double	Always equal to 1
β_i	double	Returns a random value between 0 and 1

2.2 Terminal Set

To simulate the human visual cognition system, which makes decisions relying on both color and distance information of the surfaces of scene objects from a viewpoint, we expect to extract informative spatio-temporal features by fusing RGB and depth information. Therefore, in our task, we consider extracting features from the RGB-D data (*i.e.*, RGB channel and depth channel) captured by Kinect synchronously. In addition, the terminal of the RGGP structure is data independent, which means that each video sequence V_i from the training set has a corresponding T_i defined by: T_i=$\{V_{rgb}$ and $V_{depth}\}$. All the sequences in the terminal set are located as the bottom leaves of the entire restricted graph structure and connected with the higher function nodes directly.

Additionally, we define a group of graph link parameters C_i=$\{\alpha_i$ and $\beta_i\}$ in our terminal set for constructing a more flexible feature representation. In our RGGP architecture, α_i is always set to be 1, while values of parameter $\beta_i \in (0, 1)$ are randomly distributed through RGGP evolving. Consequently, a weighted linear concatenation of pooling outputs is computed as the final feature representation. We term our complete terminal set: $T_i \bigcup C_i$ as listed in Table 1.

2.3 Function Set

The function set plays a key role in the GP structure, which constitutes the internal nodes of the tree and is typically driven by the nature of the problem. In our approach, we divide the function set into two layers: the filtering layer (bottom part) and the feature pooling layer (top part). The order of the structure is always fixed. Note that not all the operators listed in the function set have to be used in a given structure and the same operator can be used more than once. The topology of our RGGP structure is essentially unrestricted.

Functions for the Filtering Layer

In this layer, we adopt 15 unary operators and 3 binary operators to extract the meaningful features from RGB-D data including: 3D Gaussian filter series, 3D Laplacian filter, 3D wavelet filter, 3D Gabor filters and other effective filtering operators. Besides, some basic arithmetic functions are also chosen here. Table 2 lists the corresponding operators used in this layer. We adopt 3D Gaussian and Gaussian derivative filters due to their remarkable ability of reducing image noise and enhancing details. They have been widely used for denoising and smoothing target image information and

Table 2: Functions for the Filtering Layer

Function Name	Input	Function Description	Operator Type
Gau1	1 sequence	3D Gaussian smooth filter with $\sigma = 1$	Filter
Gau2	1 sequence	3D Gaussian smooth filter with $\sigma = 2$	Filter
LoG1	1 sequence	3D Laplacian of Gaussian filter with $\sigma = 1$	Filter
LoG2	1 sequence	3D Laplacian of Gaussian filter with $\sigma = 2$	Filter
Lap	1 sequence	3D Laplacian filter	Filter
GBO-0	1 sequence	3D Multi-scale-max Gabor filter with orientation of 0 degree	Filter
GBO-45	1 sequence	3D Multi-scale-max Gabor filter with orientation of 45 degrees	Filter
GBO-90	1 sequence	3D Multi-scale-max Gabor filter with orientation of 90 degrees	Filter
GBO-135	1 sequence	3D Multi-scale-max Gabor filter with orientation of 135 degrees	Filter
Wavelet	1 sequence	3D CDF '9/7' wavelet filter	Filter
EHIS	1 sequence	Histogram equalization	Enhancement
Aver	1 sequence	3D Averaging filter with $5 \times 5 \times 5$ sampling window	Filter
Med	1 sequence	3D Median filter with $5 \times 5 \times 5$ sampling window	Filter
ABS	1 sequence	Take the absolute value pixel by pixel	Arithmetic
DoF	1 sequence	Subtract between adjacent frames of the input sequence	Arithmetic
ADD	2 sequences	Add two input sequences pixel by pixel	Arithmetic
SUB	2 sequences	Subtract two input sequences pixel by pixel	Arithmetic
ABSsub	2 sequences	Absolute subtract two input sequences pixel by pixel	Arithmetic

enhancing image structures. 3D Laplacian filters are often used for separating signals into different spectral sub-bands and capturing intensity features for high-level classification tasks. Furthermore, wavelet filters can capture relatively precise contour information of the targets in sequences, such as gestures and actions, and also be utilized for multi-resolution analysis. Gabor filters can provide an effective mechanism to extract features from targets with different frequencies and orientations. In this paper, we present a comprehensive Gabor operator in our function set named: 3D multi-scale-max Gabor filter. Following Riesenhuber and Poggio's work [Riesenhuber and Poggio, 1999], we first convolve an input data sequence with 3D Gabor filters at six different scales (*i.e.*, $7 \times 7 \times 7$, $9 \times 9 \times 9$, $11 \times 11 \times 11$, $13 \times 13 \times 13$, $15 \times 15 \times 15$ and $17 \times 17 \times 17$) with a certain orientation (*i.e.*, 0, 45, 90, or 135 degrees), and then apply the max operation to pick the maximum value across all six convolved sequences for that particular orientation. The mathematical definition of our multi-scale-max Gabor filter is as follows:

$$GBO\text{-}\theta_s = \max_{(x,y,z)} [V_{7\times7\times7}(x,y,z,\theta_s), V_{9\times9\times9}(x,y,z,\theta_s), ..., V_{15\times15\times15}(x,y,z,\theta_s), V_{17\times17\times17}(x,y,z,\theta_s)] \quad (1)$$

where GBO-θ_s is the output of the multi-scale-max Gabor filter and $V_{i\times i\times i}(x,y,z,\theta_s)$ denotes the convolved sequences with the scale $i \times i \times i$ and the orientation θ_s.

In addition, the other 3D operators we choose in this layer also show their merit for data enhancement and transformation and extract effective features in different ways. Furthermore, we use basic arithmetic functions to increase the variety of the selection for composing individuals during the RGGP running and make the whole evolution procedure more natural.

Functions for the Feature Pooling Layer

We build this layer using two popular feature pooling techniques: max pooling and average pooling. Max pooling is regarded as a key mechanism for object recognition in the visual cortex system. It provides a robust response, tolerating shift and scaling, in the case of recognition in clutter or with multiple stimuli in the receptive field [Riesenhuber and Poggio, 1999]. And, average pooling, essentially a low-pass filter, is widely used as an effective way to reduce noise in target data, decreasing the amount of intensity variation between the central pixel and the surrounding ones and presenting general features of the inputs. In this layer, we define 20 pooling functions which have been divided into two groups with pooling grids (resolutions) varying from $1\times1\times1$ to $10\times10\times10$ as listed in Table 3. For each pooling function, an input sequence is first divided by the pooling grid, the local max/average pixel values from all sub-blocks are obtained and then flattened into a 1D representation as the pooling function's output. The two pooling functions are defined in Eq. (2) and Eq. (3):

Table 3: Functions for the Feature Pooling Layer

Function Name	Input	Description
$f^{Aver}_{1\times1\times1}$ $f^{Aver}_{2\times2\times2}$ . . . $f^{Aver}_{10\times10\times10}$	1 sequence	10 average pooling functions with various pooling grids from $1 \times 1 \times 1$ to $10 \times 10 \times 10$
$f^{Max}_{1\times1\times1}$ $f^{Max}_{2\times2\times2}$ . . . $f^{Max}_{10\times10\times10}$	1 sequence	10 max-pooling functions with various pooling grids from $1 \times 1 \times 1$ to $10 \times 10 \times 10$

$$f^{Aver}_{n\times n\times n}(V) = [\frac{1}{m}\Sigma_{i=1}^{i=m} p_{1\,i}, \frac{1}{m}\Sigma_{i=1}^{i=m} p_{2\,i}, \ldots, \frac{1}{m}\Sigma_{i=1}^{i=m} p_{j\,i}, \ldots] \quad (2)$$

$$f^{Max}_{n\times n\times n}(V) = [max_{i=1}^{i=m} p_{1\,i}, max_{i=1}^{i=m} p_{2\,i}, \ldots, max_{i=1}^{i=m} p_{j\,i}, \ldots] \quad (3)$$

where *V* is the input sequence for pooling functions. The right side of both equations is represented as a $n \times n \times n$ dimensional vector. $p_{j\,i}$ indicates the i-*th* pixel in the j-*th* sub-block and $\frac{1}{m}\Sigma_{i=1}^{i=m} p_{j\,i}$/$max_{i=1}^{i=m} p_{j\,i}$ denotes the elements in vectors, respectively. Note that which pooling functions will be used and their order in the final learned feature representation are automatically selected in the GP evolution.

An implicit assumption underlying this approach is that all combinations of operators are equally likely to be useful. In many cases, however, we know in advance that there are constraints on the structure of the solution, or we have a strong belief about the likely form that the solutions will take [Poli *et al.*, 2008].

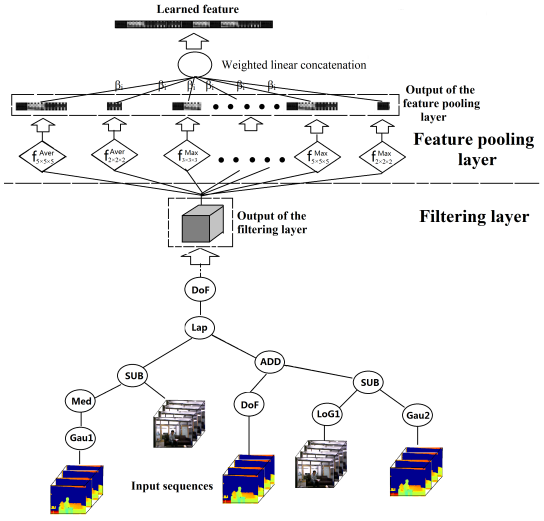

Figure 3: An illustration of a possible learned structure. All the 3D operators and pooling functions used in structure are randomly selected through the RGGP evolving.

Therefore, to make our solution more task-adaptive, we define some grammatical constraints for this layer as follows:

$$\begin{cases} Output = Concatenate(\beta_1 P_1, \beta_2 P_2, ..., \beta_i P_i) & (4) \\ P_i = f_{n\times n\times n}(x_i) \in \{pooling functions\} & (5) \\ x_i = \alpha \times Subtree_{out}. & (6) \end{cases}$$

where Output is the final learned feature vector, $f_{n\times n\times n}$ is the pooling function with the resolution of $n \times n \times n$ selected from the pooling layer, α_i and β_i indicate the graph connection parameters (α_i=1, $\beta_i \in (0, 1)$) and $Subtree_{out}$ denotes the output of the filtering layer.

Consequently, in the structure of our evolved programs, each selected pooling function works as an internal node sharing the same output of the filtering layer (see in Fig. 3). We consider this as a restricted graph-base program, which restricts the GP evolving in a specific form of the graph-based architecture.

2.4 Fitness Function

As a significant part in GP evolving, fitness function evaluates how well a program is able to solve the problem. Since the basic principle of the evolutionary method is to maximize the performance of individual solutions, we design our fitness function by using the classification error rate computed by a linear support-vector-machine (SVM) classifier on the training set. In the evaluation procedure, features learned from the candidate graph-based structures are first reduced in dimensionality via principle component analysis (PCA) keeping the 98% principal components, and then fed as the inputs of the SVM classifier. To obtain a more reliable and fair fitness evaluation, for each candidate RGGP program, a five-fold cross-validation is applied to estimate the classification error rate. Specifically, we randomly divide the training set into five sub-sets with the identical size and repeatedly train the SVM on 4/5-ths of the set and test on the remaining fifth. We further calculate the average error rate of the five-fold cross-validation as the final fitness value. The corresponding fitness function is defined as follows:

$$E_r = (1 - (\sum_{i=1}^{n} (SVM[acu_i])/n)) \times 100\% \quad (7)$$

where $SVM[acu_i]$ denotes the classification accuracy of fold *i* by the SVM and *n* indicates the total number of folds executed with cross-validation. Here *n* is equal to 5.

Table 4: Parameter settings for RGGP running

Population Size	800
Generation Size	70
Crossover Rate	85%
Mutation Rate	8%
Elitism Rate	2%
Selection for Reproduction	'lexictour'
Stop Condition	$E_r \leq 2\%$

3 Experiments and Results

In this section, we describe the implementation details of RGGP, the datasets used and the relevant experimental results.

3.1 RGGP Implementation

We implemented our proposed system using Matlab 2011a on a server with a 12-core processor and 54GB of RAM running the Linux operating system. Table 4 shows some significant parameters used in this implementation. A high population size is used to prevent early convergence, and a lexicographic parsimony pressure selection [Luke *et al.*, 2002], in which the best-performing individual with the least structure complexity (number of nodes) will be chosen, is adopted to maintain population diversity and control the bloat [Poli *et al.*, 2008]. In addition, we set the GP termination as $E_r \leq 2\%$, which means if the value calculated by the fitness function is equal to or lower than 2%, our GP running will be stopped and return the best-so-far individual to users.

3.2 Datasets

In this paper, we systematically evaluate our proposed RGGP approach on two RGB-D datasets: the SKIG dataset[1], which is collected by us, and the MSRDailyActivity3D dataset [Wang *et al.*, 2012]. Fig. 4 shows some example frames of these two datasets. Details of the datasets are provided below.

SKIG: This new dataset contains 2160 hand gesture sequences (1080 RGB sequences and 1080 depth sequences) collected from 6 subjects. All these sequences are synchronously captured with a Kinect sensor (including a RGB camera and a depth camera). This dataset collects 10 categories of hand gestures in total: *circle (clockwise), triangle (anti-clockwise), up-down, right-left, wave, "Z", cross, comehere, turnaround and pat*. In the collection process, all these ten categories are performed with three hand postures: fist, index and flat. To increase the diversity, we recorded the sequences under

[1]This new dataset can be downloaded from our website: http://lshao.staff.shef.ac.uk/data/SheffieldKinectGesture.htm

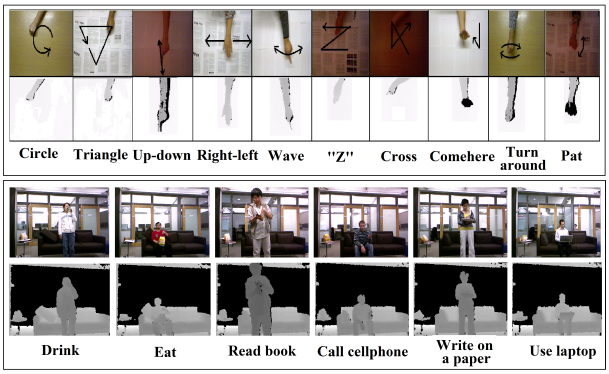

Figure 4: Some example frames of two datasets. Samples in the top black-box are from the SKIG dataset and samples in the bottom black-box are from the MSRDailyActivity3D dataset.

Table 5: Comparison results of classification accuracies (%) on the SKIG dataset.

Method	RGB channel	Depth channel	RGB-D concatenation	RGB-D fusion
RGGP	**84.6**	**76.1**	**86.2**	**88.7**
HOG3D	83.8	75.4	85.2	-
HOG/HOF	81.7	72.1	83.4	-
HOF	81.5	67.9	82.0	-
HMHI	79.4	66.3	80.5	-
SURF3D	77.8	55.1	79.3	-
3D-SIFT	76.4	61.3	77.6	-
3D-Gabor-bank	74.0	50.3	75.2	-
Volume-LBP	64.2	44.5	67.1	-
CNN	81.7	72.6	82.9	83.8
DBN	83.1	73.8	84.7	85.9

3 different backgrounds (*i.e.*, wooden board, white plain paper and paper with characters) and 2 illumination conditions (*i.e.*, strong light and poor light). Consequently, for each subject, we recorded $10(categories) \times 3(poses) \times 3(backgrounds) \times 2(illumination) \times 2(RGBanddepth) = 360$ gesture sequences.

In our experiments, each sequence in the SKIG dataset is first normalized to $96 \times 72 \times 50$ pixels by linear interpolation. We further use all the sequences collected from the first four subjects as the training set for our RGGP evolving and then evaluate the learned features by adopting three-fold cross-validation with linear SVM on the remaining data (six subjects).

MSRDailyActivity3D: This is an action dataset captured with the RGB channel and the depth channel using the Kinect sensor. The total sequence number is 640 (*i.e.*, 320 sequences for each channel) with 16 activities: *drink, eat, read book, call cellphone, write on a paper, use laptop, use vacuum cleaner, cheer up, sit still, toss paper, play game, lie down on sofa, walk, play guitar, stand up, sit down*. There are 10 subjects in the dataset and each subject performs each activity twice, once in standing position, and once in sitting position.

In the pre-processing stage, each sequence is resized to $60 \times 80 \times 40$ pixels. Later, the sequences performed by subject 1 to subject 6 are adopted as the training set to evolve our RGGP programs and the rest of the dataset is used for evaluation. Here, the same three-fold cross-validation scheme is applied.

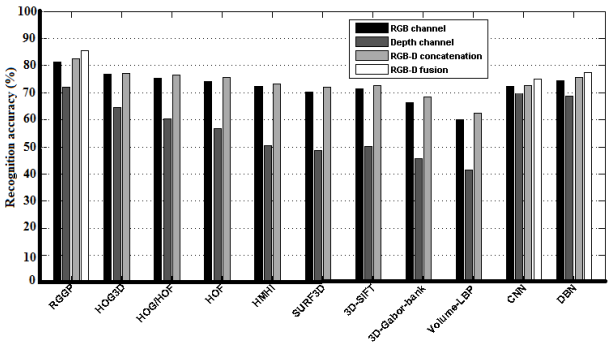

Figure 5: Comparison of action recognition accuracies (%) on the MSRDailyActivity3D dataset.

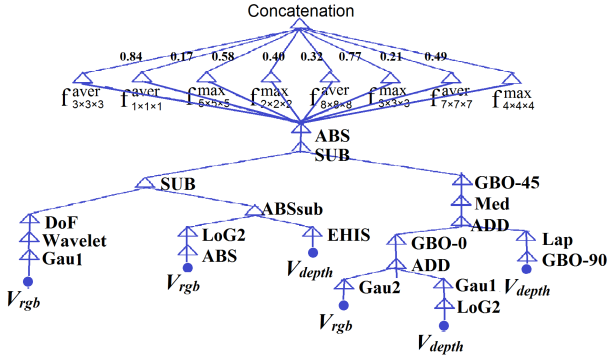

Figure 6: The final graph-based program for RGB-D feature extraction and fusion via RGGP on the SKIG dataset.

3.3 Results

For the SKIG dataset, we train our proposed RGGP architecture on RGB data, depth data and the combined RGB-D data, respectively, to extract the discriminative features for gesture recognition. The corresponding results can be seen in Table 5. It is obvious that only using RGB sequences (84.6%) to learn features can lead to 8.5% higher classification rate than just adopting depth sequences (76.1%), and the accuracy obtained with the direct concatenation (86.2%) of features separately learned from RGB and depth data is better than that of both individual data channels. In addition, the result (88.7%) is further improved when we use our RGGP approach to extract and fuse the RGB and depth features simultaneously. The corresponding RGGP program is visualized in Fig. 6, in which 3D Gaussian, 3D Laplacian and 3D Gabor operators can be automatically selected by GP at the filtering layer to extract the orientation and intensity features, and several scales of pooling operators can get the most robust and distinctive responses to different data resolutions on the top layer. The whole learned architecture is indeed consistent with the physical structure of the human visual cortex. Fig. 7 shows the decreasing error rates of the best-so-far programs during the genetic evolution.

To demonstrate generalizability, we further evaluate the proposed method on the MSRDailyActivity3D dataset, for which the results we obtained are a little lower due to its

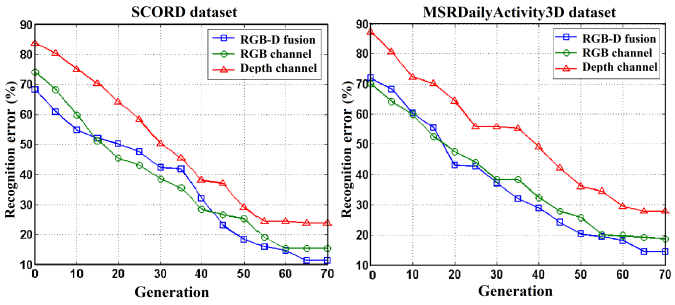

Figure 7: Evolved best-so-far values of fitness on two datasets by using RGB channel only, depth channel only and RGB-D data fusion, repectively.

$$\text{Concatenation}[0.37\times f^{aver}_{5\times5\times5}, 0.72\times f^{max}_{10\times10\times10}, 0.20\times f^{max}_{3\times3\times3}, 0.53\times f^{aver}_{7\times7\times7}, 0.83\times f^{max}_{5\times5\times5}, 0.61\times f^{max}_{2\times2\times2}](\text{ABSsub}(\text{ABSsub}(\text{Gau1}(\text{ADD}(\text{Med}(V_{depth}), \text{GBO-90}(V_{rgb}))), \text{Lap}(\text{Wavelet}(V_{depth}))), \text{SUB}(\text{Aver}(\text{LoG2}(V_{rgb})), \text{DoF}(\text{GBO-135}(\text{Gau2}(V_{rgb}))))))$$

Figure 8: The LISP format of the RGGP-generated program for RGB-D fusion on the MSRDailyActivity3D dataset.

more complex backgrounds and large intra-class variations as shown in Fig. 5. Consequently, the final (near-)optimal program selected by RGGP achieves the classification accuracy of 85.6% through RGB and depth data fusion. We have also included the recognition rates respectively computed by adopting only RGB (81.3%) and depth (72.1%) sequences for our RGGP evolving. The concatenated feature from separately-evolved programs can lead to an accuracy of 82.4%. Fig. 7 shows the evolved best-so-far values of the fitness, and the LISP expression of the (near-)optimal program obtained by RGGP evolving for RGB-D fusion is visualized in Fig. 8.

Since our proposed RGGP approach can effectively fuse the RGB-D data and extract discriminative features, it achieves outstanding results for high-level recognition tasks.

For comparison, we also list the recognition rates calculated on both datasets by some prevalent hand-crafted 3D descriptors including: HOG3D, HOG/HOF [Laptev *et al.*, 2008], HOF, HMHI [Davis, 2001], SURF3D, 3D-SIFT, 3D Gabor bank and volume-LBP. We use the hierarchical motion history image (HMHI) as a holistic 3D descriptor to extract the motion information for later recognition. 3D-Gabor-bank, which is considered as an effective and efficient way to obtain the orientation information, simulates the biological mechanism of the human visual cortex by applying 3D Gabor filtering with 4 orientations at 6 different scales. The output of each filter is then averaged on a $10 \times 10 \times 10$ grid to form a vector. As the other six 3D descriptors (*i.e.*, HOG3D, HOG/HOF, HOF, SURF3D, 3D-SIFT and volume-LBP) are usually used as local descriptors, dense sampling is first applied on each sequence in a dense grid with the block size of $10 \times 10 \times 10$ pixels and an overlap of 5 pixels in each dimension, and the final representation vector is the concatenation of the descriptor calculated on all blocks. Following the same experimental setting, we first adopt PCA for dimension reduction and then apply the linear SVM with 'three -fold' cross-validation to compute the recognition accuracies.

In addition, we have also used two popular deep learning methods: DBN and CNN for feature learning. For DBN, we train a hierarchical architecture on the training sets with neuron numbers in the hidden layers: 500-500-2000 with back-propagation fine-tuning and then utilize the learned architecture (with associated parameters) to extract features on the test sets combined with the linear SVM classifier for recognition. Similarly, a 5-layer feature extraction structure has been trained using the CNN and further adopt the same classification mechanism to compute the final accuracy.

To make the comparisons clear and fair, Table 5 and Fig. 5 show the final results computed by using all the above techniques on RGB data only, depth data only and RGB-D feature concatenation, respectively. For DBN and CNN, we can also use the combined RGB and depth data as the architecture inputs to learn features for RGB-D fusion. From the listed results, it is observed that our RGGP method significantly outperforms the state-of-the-art hand-crafted and deep learning techniques on both datasets, thanks to the superior performance of the simultaneous description and fusion of RGB and depth channels through the proposed methodology. The implicit supervised nature of the feature learning mechanism also contributes to the discriminative power of the RGGP-built features.

Even though the above comparison is sensible and fair, to further demonstrate the superiority of our method, we can deliberately compare with the whole system described in [Wang *et al.*, 2012] on the MSRDailyActivity3D dataset. We apply our RGGP-learned descriptor to the whole dataset and train the SVM classifier on half of the data and test on the rest using exactly the same setting as in [Wang *et al.*, 2012]. Our simple system can achieve 90.4% recognition accuracy, which is 2.2% higher than the result reported in [Wang *et al.*, 2012]. Considering that their system is much more complex and employs the sophisticated body joints and skeleton model, the performance gain obtained by our method simply on raw video data is significant.

4 Conclusion

In this paper, we have developed an adaptive learning methodology using our proposed restricted graph-based genetic programming (RGGP) to evolve discriminative spatio-temporal representations, which simultaneously fuse the RGB and depth information, for high-level recognition tasks. Our method addresses feature learning as an optimization problem, and allows a computer to automatically assemble holistic feature extraction by using a pool of primitive operators, which are devised according to the general knowledge of feature extraction. We have systematically evaluated our method on our new SKIG dataset and the public MSRDailyActivity3D dataset with final recognition accuracies of 88.7% and 85.6% for RGB-D fusion, respectively. In both datasets, experimental results manifest that our RGGP feature learning approach achieves significantly higher recognition accuracies compared with state-of-the-art hand-crafted and machine-learned techniques.

References

[Bay *et al.*, 2008] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[Cruz *et al.*, 2012] L. Cruz, D. Lucio, and L. Velho. Kinect and rgbd images: Challenges and applications. *SIBGRAPI Tutorial*, 2012.

[Davis, 2001] J.W. Davis. Hierarchical motion history images for recognizing human motion. In *IEEE Workshop on Detection and Recognition of Events in Video*, pages 39–46, 2001.

[Han *et al.*, 2013] J. Han, L. Shao, D. Xu, and J. Shotton. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE Transactions on Cybernetics*, 2013.

[Kläser *et al.*, 2008] A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *British Machine Vision Conference (BMVC)*, 2008.

[Lai *et al.*, 2011] K. Lai, L. Bo, X. Ren, and D. Fox. Sparse distance learning for object recognition combining rgb and depth information. In *International Conference on Robotics and Automation*, pages 4007–4013, 2011.

[Laptev *et al.*, 2008] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[Larochelle *et al.*, 2007] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *International Conference on Machine Learning*, pages 473–480, 2007.

[Lee and Mumford, 2003] T.S. Lee and D. Mumford. Hierarchical bayesian inference in the visual cortex. *JOSA A*, 20(7):1434–1448, 2003.

[Lee *et al.*, 2009] H. Lee, R. Grosse, R. Ranganath, and A.Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *International Conference on Machine Learning*, pages 609–616, 2009.

[Liu and Shao, 2013] L. Liu and L. Shao. Synthesis of spatio-temporal descriptors for dynamic hand gesture recognition using genetic programming. In *IEEE International Conference on Automatic Face and Gesture Recognition*, Shanghai, China, 2013.

[Liu *et al.*, 2012] L. Liu, L. Shao, and P. Rockett. Genetic programming-evolved spatio-temporal descriptor for human action recognition. In *British Machine Vision Conference (BMVC)*, Surrey, UK, 2012.

[Luke *et al.*, 2002] S. Luke, L. Panait, et al. Lexicographic parsimony pressure. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 829–836, 2002.

[Ni *et al.*, 2011] B. Ni, G. Wang, and P. Moulin. Rgbd-hudaact: A color-depth video database for human daily activity recognition. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1147–1153, 2011.

[Ojala *et al.*, 2002] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[Paton and Kosecka, 2012] M. Paton and J. Kosecka. Adaptive rgb-d localization. In *Conference on Computer and Robot Vision*, pages 24–31, 2012.

[Poli and others, 1997] R. Poli et al. Evolution of graph-like programs with parallel distributed genetic programming. In *Genetic Algorithms: Proceedings of the Seventh International Conference*, pages 346–353, 1997.

[Poli *et al.*, 2008] R. Poli, W.B. Langdon, and N.F. McPhee. *A field guide to genetic programming*. 2008.

[Poli, 1996] R. Poli. Genetic programming for image analysis. In *Proceedings of the First Annual Conference on Genetic Programming*, pages 363–368, 1996.

[Ranzato *et al.*, 2007] M.A. Ranzato, F.J. Huang, Y.L. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[Riesenhuber and Poggio, 1999] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature neuroscience*, 2:1019–1025, 1999.

[Scovanner *et al.*, 2007] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *International Conference on Multimedia*, pages 357–360, 2007.

[Spinello and Arras, 2011] L. Spinello and K.O. Arras. People detection in rgb-d data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3838–3843, 2011.

[Spinello and Arras, 2012] L. Spinello and K.O. Arras. Leveraging rgb-d data: Adaptive fusion and domain adaptation for object detection. In *International Conference on Robotics and Automation*, pages 4469–4474, 2012.

[Torres *et al.*, 2009] R.S. Torres, A.X. Falcão, M.A. Gonçalves, J.P. Papa, B. Zhang, W. Fan, and E.A. Fox. A genetic programming framework for content-based image retrieval. *Pattern Recognition*, 42(2):283–292, 2009.

[Trujillo and Olague, 2006] L. Trujillo and G. Olague. Synthesis of interest point detectors through genetic programming. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pages 887–894, 2006.

[Wang *et al.*, 2012] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1290–1297, 2012.