Task and Context aware Performance Evaluation of Computer Vision Algorithms

Wolfgang Ponweiser and Markus Vincze

Vienna University of Technology
Automation and Control Institute
Gusshausstr. 27-29/E376, A-1040 Vienna, Austria
{ponweiser,vincze}@acin.tuwien.ac.at
http://www.acin.tuwien.ac.at *

Abstract. Developing a robust computer vision algorithm is very difficult because of the enormous variation of visual conditions. A systems technology solution to this challenge is an automatic selection and configuration of different existing algorithms according to the task and context of arbitrary applications. This paper presents a first attempt to generate the required mapping between the task/context to the optimal algorithm and algorithm configuration. This mapping is based on an extensive performance evaluation. To practically handle the exhaustive search for optimal solutions a new optimization challenge the Multiple-Multi Objective Optimization (M-MOP) and an according solution based on genetic algorithms is developed and evaluated. The results show the robustness of the approach and guide further development towards an automatic vision system generation.

Key words: automatic system generation, performance evaluation, task and context dependency, multi objective optimization

1 Introduction

The evolution of computer vision algorithms is enormous. As an example the detection of objects has emerged from specific object instances with black background (e.g. the COIL database [1]) to object categories of outdoor images (e.g. the PASCAL challenge [1]). However if classical vision difficulties as different lighting, background or motion will be relaxed the performance of even well known algorithms drops down. The classical approach of computer vision science is a further analysis of the image data to *generate novel* advanced performing algorithms. The concept presented in this paper is inspired by the idea to analyze image data to *select and configure existing* algorithms according to the present task and context.

This analyze is carried out by an extensive evaluation on the performance of vision algorithms. More precisely the Pareto front, the set of optimal configurations, has to be filtered out. Optimality in this respect has to be carried out according to all possible context dimensions. The algorithms themselves

* This work is supported by projects S9101, S9103 and S9106 of the Austrian Science Foundation.

[1] http://www1.cs.columbia.edu/CAVE/

are treated as black boxes. Only their configuration as input and their context depending performance as output are used to represent the algorithms. The concept presented in this paper generates this data to enable dynamical selection of the optimal algorithm and algorithm configuration according to the application task and context. The task specifies the constraints and preferences between the different performance metrics. The application context specifies a subset of the evaluation ground truth which leads to context specific performance values and therewith the optimal configuration for this context.

The following Section presents the related state of the art. Section 3 describes the incorporation and application of task knowledge into the metering of computer vision algorithms and Section 4 incorporates context information. The results of the behavior of the evaluation methodology are summarized in Section 5.

2 State of the Art

As already Foerstner [2] pointed out performance evaluation of computer vision algorithms is a challenging task. There are two main approaches. One is to split the overall algorithm into several sub-tasks which can be evaluated analytically. A typical example is Courtney [3]. The second approach is to perform an exhaustive evaluation regarding some ground truth with the challenge to manage the required evaluation effort. Exemplary Appenzeller and Crowley [4] focused on the parameter control for fair evaluation. Vogel and Schiele [5] present a method for the optimal adaptation using performance prediction on simulated data. From a technological point of view Everingham, Muller and Thomas [6] developed the most similar approach. They also use a genetic algorithm to enable a comprehensive evaluation. The main difference is that we additionally integrate the analyze of context dependencies.

The term of context is widely used. Some definitions can be found in Dey [7]. Winograd [8] enhances these definitions focusing on the 'use' of context. Of course there are approaches that directly use image context such as Torralba [9]. Paletta [10] already separates between internal and external context for improving a specific object detection algorithm. Finally Braun et. al. [11] applies context out of the processing history as an additional cue at a voting stage.

Sharing our main objective of incorporating the context into the composition and configuration Strat et al. [12] uses context as an ontology in a production system to solve this task. Shekhar and colleagues [13] use a knowledge base to search the configuration space for optimal configurations. Lombardi and Zavidovique [14] defined context states that where related by a Markoff model and Thonnat and colleagues analyze the knowledge based adaptation to scene [15] and present selection techniques for algorithms based on a neural network and a selection technique for parameter selection based on image feature similarity [16]. Crowley et. al. [17] map user context represented by roles and relations to system configuration using an ontology. Our approach differs to all of these approaches by replacing the knowledge bases or ontologies by specifying the evaluated configuration/performance mapping according to a predefined set of contexts.

3 Task Awareness

Task descriptions for computer vision systems can be separated into two different parts. The first part describes the type of execution requested, the task category or services. Typical examples are object recognition, classification, tracking and segmentation. The second part of the description details the conditions and preferences according to the application at hand. These properties can be expressed using required performances. Examples are processing time, spatial accuracy and success rate. These contradicting criteria are usually related using a certain decision function.

Consider the example of an inspection or surveillance task. Most important for these applications is to detect all positives (objects to identify). As the system is not perfect the price for that is a reasonable number of false positives. For example also some shadows might be reported as a material defect. On the other hand using the evaluation for the selection of tools for manipulation, some missed objects not being grasped (false negatives) are acceptable compared to grasping for example into a wall (false positives). It is the task that specifies the relation between different performance metrics.

Lets further specify the approach based on the service of object recognition. All success metrics of the ROC [18] are reused as performance criteria (number of true positives TP, number of false positives FP, number of true negatives TN and the number of false negatives FN). It is assumed that the object recognizer provide a confidence value for every object reported. The average of these confidence values for true positives (confidenceTP) and false positives (confidenceFP) are used as performance criteria, too. Further criteria are the average of the processing time for a single image and the spatial accuracy of the center of gravity of a single object.

Considering the configurability of algorithms and the contradicting criteria described above, performance evaluation ends up in a search for optimal solutions. Since the decision function is not known at evaluation time the optimal configurations for all possible decision functions have to be found. Respectively all combinations of configuration parameter values that are not optimal for any decision function have to be filtered out.

Mathematically this can be expressed as function between the vector of performance metrics and the configuration parameters which is a typical Multiobjective Optimization Problem (MOP) formulation (see Eq.: 1).

$$PerformanceMetrics = f(ConfigurationParameters) \tag{1}$$

A mathematical definition of the MOP can be found in [19]. Additionally all related terms like dominance, Pareto set and Pareto front are well defined in this book. Figure 1 presents a graphical introduction.

The semantic background of this optimization framework is based on the only constraint for the decision function that it has a monotonic trade-off between performance parameters. If the decision function rates more optimal values as better (which is a very natural assumption) the Pareto set is exactly the set of all optimal algorithm configurations (solutions). For example, this is not the case if the Pareto front is replaced by the envelop of solutions in the performance space. The envelop only guarantees, that for every linear weighting of performance

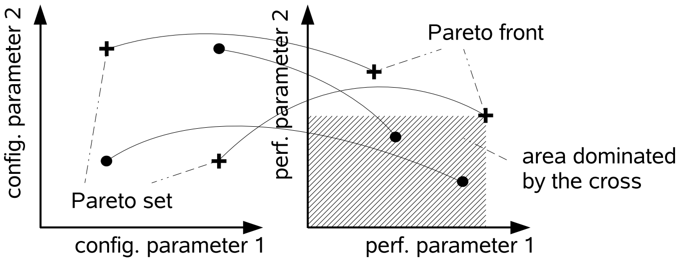
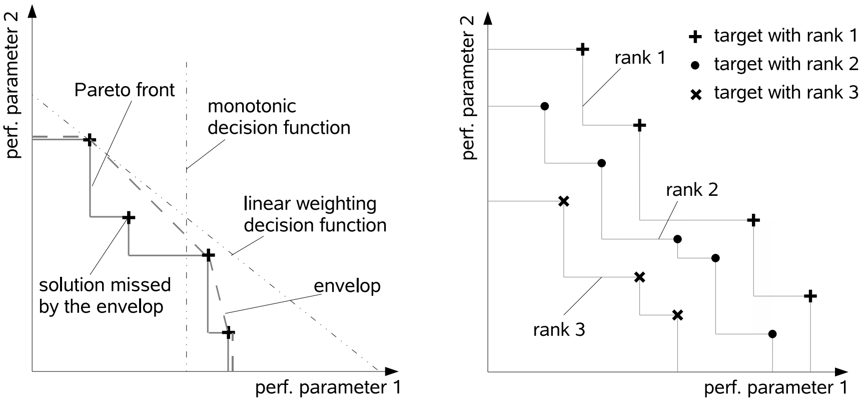


Fig. 1. The performance of different configuration settings. The dominating area filters out non-optimal solutions. All non dominated solutions constitute the Pareto front in the performance space and the Pareto set in the configuration space.

metrics all optimal solutions are selected. In contrast, e.g., a soft real time system which is expressed as a maximal processing time, the envelop could miss certain optimal solutions. This behavior is demonstrated in Figure 2(a).

Finally the term of Pareto rank can be assigned to a solution [19]. This is carried out by calculating the Pareto set. Assigning the associated solutions a rank of 1. Removing the set with rank 1 from the solution set the next Pareto set is calculated and a rank of 2 is assigned. All further ranks are assigned accordingly (see Figure 2(b)).



(a) The difference between the Pareto front and the envelop. For more than only linear weighting decision functions some solutions may be lost using the envelop.

(b) The Pareto rank assignment according to the iteration of Pareto front calculations.

Fig. 2. The Pareto front and the related concepts of the envelop and the Pareto rank.

3.1 Application to Task Awareness

As already motivated the configuration of an algorithm can be adjusted to the current demands of an application. As explained above, these demands are expressed by a decision function relating the different performance values to each other. Even if these demands change over time the configuration and therewith the optimal algorithm behavior can be adapted. Practically this can be performed using a lookup table that relates the application demands, the performance metrics, to optimal algorithm configurations (as found by evaluation).

The same procedure can be used to select the algorithm itself. Consider a set of algorithms that provide the same type of task (the services) but with different performance behaviors. The task of selecting the optimal algorithm is only an extension of the configuration selection presented before. Simply, the algorithm instance has to be added to the configuration space. Practically this is carried out by extending the lookup table to a yellow page like database.

Furthermore there is another option to use the evaluation data. Using Bayes decision rules for interpreting the output of the algorithms an optimal decision border can be calculated. The confidence value at the decision border is defined by Equation 2 [20]. Using a task specific configuration, the decision border is made task specific, too. Simply the distribution of the confidence values (confidenceTP, confidenceFP) have to be recorded during the evaluation to enable the calculation of $p(conf \mid TP)$ and $p(conf \mid FP)$. Therewith a task specific decision mechanism is implemented.

$$\frac{p(conf \mid TP)}{p(conf \mid FP)} = \frac{P(FP)}{P(TP)} \tag{2}$$

4 Context Awareness

The aim is to incorporate all types of information available into a system for optimal application of existing algorithms. For the approach described in this paper all influences on the algorithm performance behavior that are anywhere represented in the systems knowledge can be interpreted as context.

From this context specification some general properties can be derived. Because the algorithms are treated as black boxes the context has to be applicable *from outside* of the algorithm. Another important property is the *impact on visual processing*. Although there might be data available that can be essential for the entire system, it can be irrelevant for the vision algorithm. Such type of data is not appropriate as vision context. Another property a context dimension has to fulfill is *generality*. E.g. if a book is opened or closed can be delivered from outside and has a strong influence on the appearance of the book in an image. However this context is extremely object specific and is therefore not generic.

For the tests performed the *camera ego motion* as an image specific context is used. *Appearance, geometry, reflectance, solidness* and *topology* are evaluated as object specific contexts. Further more *object motion* and *occlusion* are used as image and object specific contexts. All of these contexts fulfill the criteria specified above. They can be potentially delivered from outside the algorithm and will hopefully influence the vision algorithm behavior.

Most of these context dimensions are difficult to measure in a generic fashion. On the other hand this is not essential for the evaluation task. E.g. it will be unlikely that the occlusion of an object to recognize will be exactly known in advance. But a system that performs activity analyzes will have some knowledge about the scene and can therefore derive a potential of occlusion. As a result all the context dimensions are quantized using only two to four values. E.g. the values for the object topology are 'convex', 'concave' and 'with holes'. Furthermore the decision for context labels of ground truth images is carried out by hand. Accordingly these decisions will have some spread. However for achieving reasonable conclusions about the influence of context and the optimal algorithm configurations for specific contexts these approximations are sufficient. Additionally consider that a reduced set of possible context values reduces the evaluation effort.

There are several options to integrate these context parameters into a MOP. Using the fact that all different MOPs share the same input space (the configuration parameters) the *Multiple-MOP (M-MOP) mapping* generates optima for all contexts, is mathematically feasible and efficient (see [21]).

4.1 Efficient and Robust Pareto set calculation

Of course the evaluation of all possible configurations is inefficient and for continuous configuration parameters impossible. Therefore an optimization methodology is required. Genetic algorithms (GA) are chosen for this task because they deal with multiple solutions per default, they need no heuristics which are not available since the vision algorithms to evaluate are not known right now and finally because only robust optima instead of peaky global optima are required.

There is a large set of different GA approaches for optimizing classical MOPs. An excellent survey can be found in [19]. Recent prominent algorithms are PAES [22], NSGA-II [23] and SPEA2 [24]. The main contradicting properties these methods have to deal with are the convergence to the Pareto front and the diversity of the solutions found as well as the efficiency represented using the processing time or the number of evaluations required.

The genetic algorithm developed is inspired by Deb's NSGA-II [23]. NSGA-II was selected because it realizes elitism, it requires no sharing parameter and introduces a novel representation of the density of the population, the crowding distance.

The extensions of E-NSGA-II developed for dealing with a *multiple* MOP (M-MOP) are the extensive use of the configuration space e.g. for the density calculation as well as the fusion of solutions over all context settings required for the selection step of the GA. A detailed description of the developed algorithm can be found in [25]. A summary of the performance and robustness is presented in Section 5.

4.2 Application to Context Awareness

As already motivated at the beginning of this paper the objective is a task and context aware configuration of vision components. The context aware evaluation provides the database for an on-line adaptation of the configuration setting of an algorithm. Practically this is a simple extension of the lookup table mechanism already presented in Section 3.1. An important property in this respect is the default context value '-'. It expresses the independent or 'I do not know' entry for this context type. Therefore if only some context dimensions can be specified, all the others will be set to this default value.

Another important application of a context aware performance evaluation is the detailed insight into the strength and weaknesses of the algorithm itself. This is a very valuable contribution for the algorithm developer since he/she is informed about the performance behaviors against different contexts.

Furthermore such an evaluation enables a fair competition between different algorithms. The dependency on the detailed task and the context is made explicit. Performing the competition in a task and context aware manner generates a closer insight into the performance relationships and unveils the strength and weakness of the different processing approaches.

Finally the context awareness can be applied to the calculation of the decision border according to confidence values of the algorithm. But in contrary to the task awareness alone now also an adaptation can be performed if the configuration is not changed at all. Consider the case where the selection and configuration of an algorithm is carried out without any context consideration. In this case the optimal configuration for the default context ('-') will be employed. Using this configuration different values for TP, FP confidenceTP and confidenceFP according to different context settings are evaluated ($p(conf \mid TP) \Rightarrow p(conf \mid TP)(context)$). Therewith a context specific decision border for a single algorithm configuration can be calculated:

$$\frac{p(conf \mid TP)(context)}{p(conf \mid FP)(context)} = \frac{P(FP)(context)}{P(TP)(context)} \tag{3}$$

5 Evaluating the Evaluator

The approach of selecting and configuring algorithms according to the task and context at hand requires an evaluation of the algorithm's behavior under these conditions. During evaluation time neither the task description nor the context are known. For this reason the optimal configurations for all possible tasks and contexts need to be found. The establishment of a M-MOP solution technique by the E-NSGA-II enables the efficient calculation of the required data. The main properties of a Genetic Algorithm (GA) that need to be evaluated are the convergence and diversity. Further more the robustness and computational effort of the approach need to be analyzed.

The evaluation is carried out using the data of two different object recognition algorithms. The first one is based on General Color Histograms, further called GCH [26], where the second one uses a Support Vector Machine SVM [27] to discriminate different image patches.

The E-NSGA-II algorithm is instantiated with the following parameters: number of generations = 25, initial population size = 10, crossover probability = 0.8 and mutation rate = 0.12. To show the dependency to random values all experiments are repeated 10 times with different seed values. The evaluation data set and the image database with the according ground truth are available from the author.

A detailed analyzes of the evaluation results can be found in [25]. As a summary the two most important tables are presented. The first evaluation compares the performance of the E-NSGA-II algorithm to a purely random search. This is carried out regarding the number of evaluations, and the so called success rate.

This value measures the rate of Pareto optimal solutions found. Table 1 points out the gain of using the genetic optimization algorithm.

number of evaluations	E-NSGA-II	random
10%	0.366	0.092198
20%	0.713	0.168085
30%	0.954	0.234752
40%	0.993	0.296454

Table 1. The mean of the evaluated success rate for the GCH method of the E-NSGA-II algorithm compared to a random search at different percentages of evaluations performed regarding the number of possible evaluations.

Even though these values are not encouraging. For finding 95% of the real optimal solutions 30% of all possible solutions have to be evaluated. However the real benefit of the genetic algorithm is clarified by comparing not the pure success rate but using the so called Average Pareto Rank Difference (APRD) [2]. This metric considers that a not found Pareto optimal solutions can be replaced using a relatively well performing solution. Table 2 presents the results for the E-NSGA-II algorithm and a random search. The optimal APRD value is 1.

number of evaluations	E-NSGA-II	random
10%	2.432	2.73125
20%	1.908	2.58669
30%	1.572	2.27775
40%	1.309	2.1864

Table 2. The mean of evaluated 'Average Pareto Rank Difference' APRD for the GCH method of the E-NSGA-II algorithm compared to a random search at different percentages of evaluations performed regarding the number of possible evaluations.

By evaluating only 20% of all possible configurations by average at least the second best solutions is found, which expresses the robustness of the evaluator. Since the APRD relates the evaluation result to all real optima it measures convergence and diversity at once. To prove the GA for suitability further such analyzes with several different vision algorithms of different services have to be performed.

6 Conclusion

Applying computer vision algorithms into a system one of the open challenges is their selection and configuration. This paper presents a first concept based on the performance prediction out of a comprehensive evaluation. To enable task dependency the constraints and preferences between several contradicting

[2] The idea of the APRD is to search for Pareto optimal solutions in the population under investigation, that are probably selected instead of the real optimal solutions. For every real optimal point that dominates at least one current solution the worst Pareto rank of all these dominated solutions counts for the APRD. For a detailed definition of the APRD metric please have a look at [21]

performance metrics are taken into account. The performance evaluation becomes a Multi Objective Problem (MOP). Context dependency is incorporated by extending the MOP to a Multiple Multi Objective Problem M-MOP.

The methodology presented is a first attempt to enable automatic system generation. First results of the developed E-NSGA-II algorithm proved the use of genetic algorithms as the underlaying optimization technique. They provide robust and efficient optimizations. Even so many tests have to be performed to analyze the generic suitability for the analyzes of computer vision algorithms. Especially behavior against more recognition algorithms and several different vision tasks need to be analyzed.

The next scientific steps in advancing this systems approach are the careful analysis of the database used for evaluation. Although it is sufficient for the general evaluation task, their capabilities and statistical significance in relation to task and context has to be proven. Another open topic is the integration of the learning step of vision methods. The number of configuration parameters appearing during learning have the potential to overload the complete approach.

References

1. Everingham M., e.a.: The 2005 pascal visual object classes challenge. In: Proceedings of First PASCAL Challenges Workshop. Volume 1., Springer (2005)
2. Foerstner, W.: Pros and cons against performance characterization of vision algorithms. In: ECCV Workshop on Performance Characteristics of Vision Algorithms. (1996)
3. Courtney, P., Thacker, N., Clark, A.F.: Algorithmic modelling for performance evaluation. Mach. Vision Appl. **9**(5-6) (1997) 219–228
4. Appenzeller, G., Crowley, J.L.: Experimental performance characterization of adaptive filters. In: ICPR '96, Vienna (1996)
5. Vogel, J., Schiele, B.: On performance categorization and optimization for image retrieval. In: European Conference on Computer Vision ECCV. Volume IV., Copenhagen, Denmark (2002) 49–63
6. Everingham, M., Muller, H., Thomas, B.T.: Evaluating image segmentation algorithms using the pareto front. In: ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part IV, London, UK, Springer-Verlag (2002) 34–48
7. Dey, A.K.: Understanding and using context. Personal Ubiquitous Comput. **5**(1) (2001) 4–7
8. Winograd, T.: Architectures for context. Human-Computer Interaction **16** (2001) 401–419
9. Torralba, A.: Contextual priming for object detection. Int. J. Comput. Vision **53**(2) (2003) 169–191
10. Paletta, L.: Predictive visual context in object detection. In Blackburn, P., Ghidini, C., Turner, R.M., Giunchiglia, F., eds.: CONTEXT. Volume 2680 of Lecture Notes in Computer Science., Springer (2003) 245–258
11. Braun, E., Fritsch, J., Sagerer, G.: Incorporating Process Knowledge into Object Recognition for Assemblies. In: IEEE International Conference on Computer Vision, Vancouver, CA (2001) 726–732

12. Strat, T.M., Fischler, M.A.: Context-based vision: Recognizing objects using information from both 2d and 3d imagery. IEEE Trans. Pattern Anal. Mach. Intell. **13**(10) (1991) 1050–1065
13. Shekhar, C., Burlina, P., Moisan, S.: Design of selftuning iu systems. In: DARPA Image Understanding Workshop. Volume 1., New Orleans, LA (1997)
14. Lombardi, P., Zavidovique, B.: Formalization of opportunistic switching for context-adaptive vision systems. In: SMC (7), IEEE (2004) 6457–6462
15. Georis, B., Maziere, M., Bromond, F.: Evaluation and knowledge representation formalisms to improve video understanding. In: ICVS '06: Proceedings of the Fourth IEEE International Conference on Computer Vision Systems, Washington, DC, USA, IEEE Computer Society (2006) 27
16. Martin, V., Thonnat, M., Maillot, N.: A learning approach for adaptive image segmentation. In: ICVS, IEEE Computer Society (2006) 40
17. Crowley, J.L., Coutaz, J., Rey, G., Reignier, P.: Perceptual components for context aware computing. In: UbiComp '02: Proceedings of the 4th international conference on Ubiquitous Computing, London, UK, Springer-Verlag (2002) 117–134
18. Ponweiser, W., Prankl, J., Vincze, M.: Roc based evaluation for computer vision. In: Proceedings of the 30th Workshop of the Austrian Association for Pattern Recognition. Volume 1. (2006) 45–54
19. Coello, C., Veldhuizen, D., Lamont, G.: Evolutionary algorithms for solving multi-objective problems. Kluwer Academic/Plenum, New York, USA (2002)
20. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley-Interscience Publication (2000)
21. Ponweiser, W., Vincze, M.: The multiple multi objective problem - definition, solution and evaluation. In: accepted for the Fourth International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007), Matsushima/Sendai, Japan (to appear March 2007)
22. Knowles, J., Corne, D.: The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. In Angeline, P.J., Michalewicz, Z., Schoenauer, M., Yao, X., Zalzala, A., eds.: Proceedings of the Congress on Evolutionary Computation. Volume 1., Mayflower Hotel, Washington D.C., USA, IEEE Press (1999) 98–105
23. Deb, K., Agrawal, S., Pratab, A., Meyarivan, T.: A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J.J., Schwefel, H.P., eds.: Proceedings of the Parallel Problem Solving from Nature VI Conference, Paris, France, Springer. Lecture Notes in Computer Science No. 1917 (2000) 849–858
24. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland (2001)
25. Ponweiser, W., Vincze, M.: Robust handling of multiple multi-objective optimisations. In: accepted for the International Workshop on Biologically-Inspired Optimisation Methods for Parallel and Distributed Architectures: Algorithms, Systems and Applications. (to appear 2006)
26. Swain M.J., B.D.: Indexing via color histograms. In: Proceedings, Third International Conference on Computer Vision. (1990) 390–393
27. Roobaert, D., Zillich, M., Eklundh, J.O.: A pure learning approach to background-invariant object recognition using pedagogical support vector learning. In: CVPR (2), IEEE Computer Society (2001) 351–357