

1 Introduction

In the early days of computer vision and pattern recognition, writing a program to recognize an object meant carefully specifying the form of that object with a geometric model, a difficult, tedious, and error-prone task. A program would then search the visual world for items that matched that geometric form, to within some tolerance. For example, a handwritten “8” would be described as two circles, one directly above and tangent to the other. A quick look at a few handwritten documents will convince anyone that many “8”s in the real world do not conform to this overly simple description. These programs suffered from a lack of generality and adaptability, poor robustness to noise, and typically an inability to incorporate contextual information.

In the last 20 years, computer vision has seen tremendous progress. The advent of “learning” algorithms, or algorithms that adjusted their processing to optimize some function of the input data, addressed many of these problems. No longer did the exact form of an object need to be specified. Examples of objects could now be provided to describe in a natural fashion the form of an object and its variability. Of course, there were still decisions to be made, such as which features to use to best characterize an object, but the painstaking process of *object description* was no longer required. We had learned how to *teach the computer*.

As the excitement grew about learning methods, a new tedious task arose to replace the old tedium of careful object specification. This time, it was *data set collection*. For example, to provide “training data” for handwritten digit recognition problems, thousands of examples of each character were collected from hundreds of different subjects [16]. This was followed by databases of faces, then cars, motorcycles, airplanes, and many other objects. Unfortunately, we currently need just as many examples to teach the computer about motorcycles, irrespective of whether it has learned about cars. Collecting these data sets is difficult, time consuming and expensive. And in many cases, data collection must be retrospective, drawing from a finite set of already existing images. For example, it would be hard to gather additional pictures of Abraham Lincoln in order to better train a classifier. Thus relying on large data sets, one for each object or problem of interest is not a very scalable strategy if the goal is to produce a vision system with the breadth of a human system.

Thus the current state of computer vision might be summarized as follows. We can teach computers to do one thing at a time quite well, using sophisticated statistical methods and large amounts of training data. This includes not only object recognition, but other skills such as tracking, segmentation, and outlier detection. What we have not yet done successfully is designed computers so that they can learn by themselves, or at least with minimal human supervision.

A quick look at human beings will clarify this idea. Humans certainly benefit from teachers, but they are not nearly so needy as current computers in learning new tasks. For example, humans can

- learn to recognize a face from a single photograph,
- learn a new character from a single example,
- learn the meaning of a word from hearing it in context,
- learn to recognize a font they’ve never seen before, and
- learn to play a video game without being taught.

Each of these examples requires at most a single training example from a “teacher” or labeller. In each case, the human is leveraging other knowledge sources such as the structure of faces; the structure of handwriting; the meaning of some words, even if the person doesn’t know all words; the structure of English; and the knowledge of how to explore.

The goal of this research is to endow computers with the abilities to *benefit from their own experience* and, to a great extent, to *teach themselves*. This does not mean that we want to completely remove the human

teacher, but that we want to reduce the need for the human teacher to provide huge numbers of examples, and to have to do this for each new task, even when the task may be similar to tasks the computer has already learned. In other words, we want to make computers learn as *efficiently* as humans.

This proposal discusses five examples of research in this direction:

- building a classifier of handwritten digits from a single example of each digit, using knowledge previously acquired (by the computer) about general handwriting;
- learning a notion of color constancy for most objects under most lighting conditions, with only a single object provided as training data;
- learning to recognize any particular car or face from a single example, given other pairs of cars or faces that match and mismatch;
- learning to recognize typewritten text in a font never seen before, without *ANY* training examples of that font; and
- developing software for robots to continuously explore the visual world and the interactions between the visual and other senses.

The common thread in all of these tasks is that they *relieve the burden on the teacher of the computer*. The goal of my research is ultimately to develop computers that can be taught simply and rapidly, and that explore on their own. Perhaps if we achieve this goal, then computers will be able to learn enough skills to be generally useful, and to have the long sought after common sense (which will not come from being programmed with thousands of rules).

To become widely useful, and to exhibit general intelligence, designing special systems for each problem by providing large amounts of training data is probably not practical for several reasons.

- To collect training data for each different domain is too work intensive.
- Even if we were willing to collect such training data, it is often not available.
- If we want systems to be truly robust and adaptive, they must be able to rapidly adapt to situations they have never encountered before. In other words, they must be able to be self-taught.

Below, in Sections 2, 3, and 4, I describe five projects aimed at making computers either easier to teach or making them fully autonomous learners. We start with two completed projects in Section 2, discuss ongoing work in Section 3, and two future projects in Section 4. In Section 5, I present an integrated plan for outreach and education that is symbiotic with this research.

2 Recent Work

In this section, I describe two completed projects that typify the thrust of my proposed career focus. In the first, I show how it is possible to dramatically improve the accuracy of *handwritten digit* classifiers by getting the computer to leverage previously learned knowledge about *handwritten letters*. The classifier developed uses only a *single training example* of each digit, dramatically reducing the load on the “teacher”.

In the second, I show how the classical problem of “color constancy” can be solved using statistical methods and no labelled information of any kind. This gives the computer a basic capability which humans take for granted in their visual system, with almost no training burden.

Learning from One Example. This project was motivated by the ability of humans to learn a useful model of a class from a small number of examples, often just a single example [30]. Consider the symbol for the new European currency, the “euro,” shown in Figure 1. This symbol was only recently conceived,

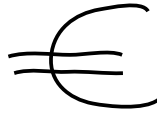


Figure 1: A handwritten version of the symbol for the new European currency, the euro.

and many people saw their first example of it during the last few years. Fortunately, after seeing a single example of this character, humans can recognize it in a wide variety of contexts, styles, and positions. It would certainly be inconvenient if people needed 1000 examples of such a new character in order to develop a good model of its appearance.

Since human beings are bound by the laws of probability and estimation, it would appear that we achieve this sparse-data learning by using prior¹ knowledge. Some of the most fundamental questions in machine learning and artificial intelligence concern prior knowledge, namely:

- What is its form and how is it obtained?
- How is it applied in new learning scenarios to improve the efficiency of learning?²
- Can prior knowledge be used to build a good model of a class from a single example?

In the following work [26, 21], we provide one set of answers to these questions. In particular, we show how a system that has access to a large training set of *handwritten letters* can use the general knowledge of variability derived from these letters to learn about *handwritten digits*.

The first step in this process is to model the variability of certain handwritten characters. This is done through a technique I developed for the joint alignment of a set of images, and which I refer to as *congealing*. An example of the congealing process is given in Figure 2. On the left of the figure is a set of 36 handwritten zeroes from the NIST database of handwritten characters. These digits were written by different people in different styles and exhibit some of the typical variations one sees in handwritten characters. On the right of the figure are the same zeroes after they have been “congealed”. Notice that the algorithm has transformed each digit to be as similar as possible to some notion of central tendency. It did this without any prespecified notion of what zeroes should look like. Rather, it simply makes each character look as much like the others as possible using a simple set of (affine) transformations. It does this in an iterative fashion until no more change is detected, at which point it outputs the resulting images.

The variability of the resulting images is much easier to model than the original images. But there is another, perhaps more important benefit. In addition to producing a set of aligned characters, the congealing algorithm also produces a *set of transformations* describing the typical variability in the set of characters. That is, by collecting information about the transformation that each character underwent in the congealing process, we have a simple way to model the natural variability of the character. In probabilistic terms, we can develop a *probability distribution over transformations* on the original characters [29], enabling us to answer questions like, “How common is a rotation of 30 degrees relative to the notion of central tendency?”

If we run the congealing algorithm on many sets of characters, say As, Bs, Cs, ..., Zs, and develop a distribution over the transformations in each case, we encounter a remarkable fact: **The distributions over transformations are approximately the same for different types of characters.** That is, the amount of rotation, shear, and size-change experienced by characters in the congealing process appears to be about

¹By *prior* knowledge we mean knowledge obtained prior to a particular task.

²*Efficiency* can be thought of, informally, as the number of training examples required to achieve a certain test performance in a particular task.

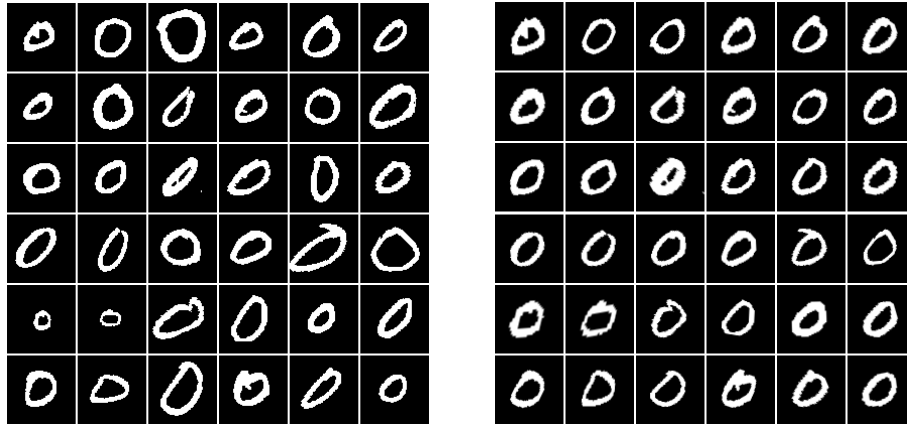


Figure 2: **Left:** Samples of handwritten zeroes from the NIST database. **Right:** The zeroes after congealing.

the same across the different letter classes (e.g. As, Bs, Cs, etc.).³ If the computer sees a character from a completely new class, like a handwritten euro symbol, it is reasonable to assume, given this evidence, that the *distribution over transformations will be the same for the new class*. Hence, one can trivially form a model for the (affine) variability of this new class.

A One Example Classifier. Using this idea, I built a classifier for handwritten digits using just one example of each digit. After developing a transformation distribution using large sets of *handwritten letters*, these distributions were combined with the single example of each digit to produce a model of variability for each digit class. This process is illustrated in Figure 3. Using models created from one example, a classifier is built using a simple and standard maximum likelihood type of classifier [26].

Results. The best handwritten digit classifiers currently get over 99% accuracy. However, these classifiers are trained on *thousands* of examples of each class. The goal of this work has been to improve performance on a very different problem, when one is given just a single example of a class. Many traditional classifiers, like neural networks, cannot reasonably trained at all using just a single example. Many others get accuracy rates below 20%, and none are better than 40% when given only a single training example. Using the process described above, we were able to increase accuracy of classification given only a single training example from only 29.7% (using our default classifier) to 89.3%, far exceeding the accuracy of any other classifier on this problem. This is a tremendous leap forward in accuracy for a problem that is much more realistic than being provided with thousands of examples of each class.

The key point is that once a computer has been given a large number of examples of a few classes, it can learn about new classes much more rapidly. This satisfies the general goal of this research, which is to lighten the training data requirement. In the next project is another example of how structure of appearance changes in one object can help us learn about how other objects change appearance. In this case, the variability is not spatial distortion as with the digits, but change in the illumination of real objects and scenes.

Learned Color Constancy. Color constancy refers to the ability of people to perceive fixed “colors” for objects even under lighting conditions that may make the true light coming off an object quite different. For example, an apple seen under a slightly yellowish light may actually be brownish, or under a slightly blue light it might be purple, and yet in each case people to see it as “red”. The stable perception of color under various lighting conditions helps people use perceived color as a stable feature for object recognition.

³The hypothesis that transformation distributions are approximately equivalent across classes is tested using statistical methods in my Ph.D. work [28].

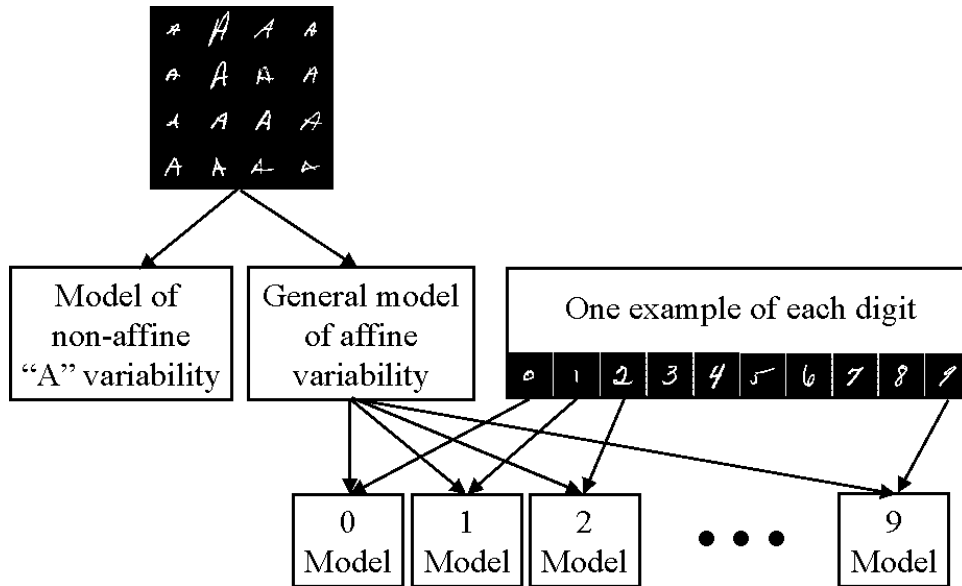


Figure 3: Diagram of our method for sharing models of affine variability. A support set (shown as a set of “A”s) is given to the learner. From this support set, a general model of affine variability is derived. Combining the model of affine variability with a single example of a handwritten digit, a model is made for the digit that incorporates the spatial variability.

Many approaches to the problem of imparting color constancy to machines have started by trying to estimate the light impinging upon a scene. This is frequently done by placing an object with known properties in the scene. For example, if a truly white and opaque sheet of paper is put in a room, and it appears pink, I can deduce that the lighting in the room is a particular shade of red. I can then use this information to discover the appropriate perceived color of other objects in the room. With respect to the goals of this grant, we ask the question, “For each new situation a computer encounters, must it be provided with a known reference object in order to estimate the lighting before it can understand the true colors of the objects present?” That is, must a teacher be present to give the computer an appropriate reference each time? Some algorithms for color constancy [23, 7] try to circumvent this requirement by assuming that the brightest object in a scene would be white under neutral lighting or that the average color of objects would be gray under neutral lighting. However, these assumptions are wrong so frequently that they cannot be relied upon.

Our approach to the color constancy problem is based upon the following idea. Figure 4 shows two pairs of pictures, each taken under different simulated lighting conditions. It is easy for a person to confirm that these pictures represent the same scenes, partly because the colors, though different in each image, have changed in a way that matches our experience regarding how colors change under different lighting conditions. Thus, rather than necessarily identifying the “true color” of each object, we simply recognize that the objects have changed colors in a plausible fashion.

In previous work [27, 34], we completely avoid the issue of estimating the lighting or the surface properties. To avoid estimating lighting, we build a statistical model of *how objects change color under common lighting changes*. Then two images can be inferred to represent the same object if there is a statistically common mapping between the colors of the two images.

Notice that this addresses a slightly different problem than traditional color constancy approaches. It

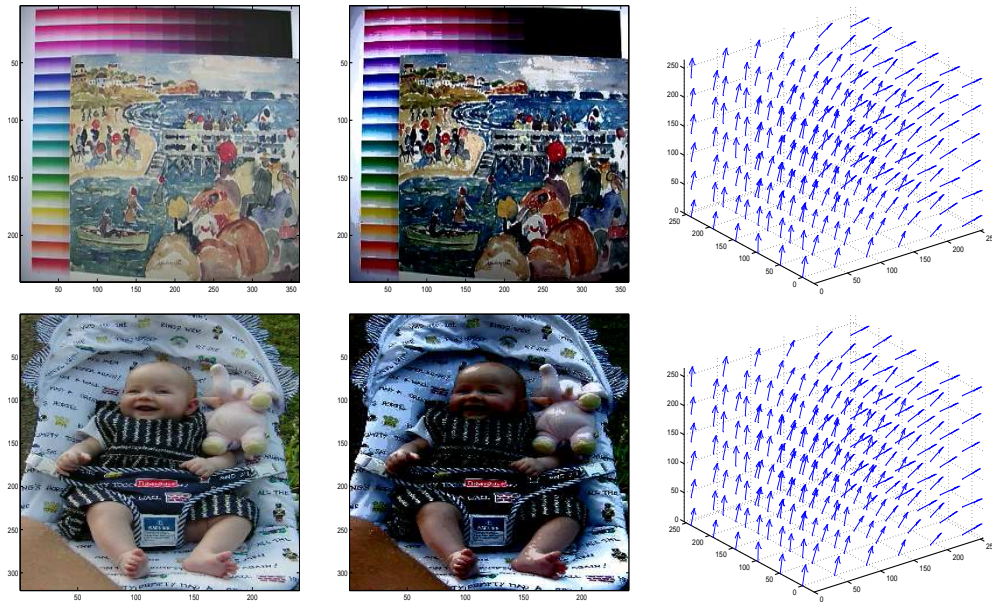


Figure 4: Lighting change, color change, and color flows. Each image pair (in the first two columns) represents the same object but under different simulated lighting conditions. Color flow fields (far right) are representations of the color changes across images. Each vector in the color flow field shows how one color moved to a different color as the lighting was changed.

implies that the important thing is not whether we can identify lighting, or the precise properties of objects, but rather that we can recognize that a certain distribution of colors is the same as another distribution of colors, up to some common lighting change. If we can do this, then we can use color to recognize objects, which is the primary goal of color constancy algorithms in the first place.

Joint Color Change. The model we develop in this work is of *joint color change*. It has nothing to do with how the colors of objects are distributed in the world, but only how the colors of objects change. To be effective, we must model how colors change *jointly*, not one at a time. That is, the knowledge that the color of certain pixels were red in one image and brown in another is not enough to say whether the pictures represent the same object. But if we can say that a certain group of colors were changed to another group of colors, then with high confidence we can guess whether the objects are the same.

Color Flows. Consider again Figure 4. The color changes induced by the simulated lighting change in each pair of pictures can be represented by what we refer to as a *color flow field*, as shown on the right of the figure. This is a set of vectors in three-dimensional color space which show how each color in one image mapped to the corresponding color in the second image. For example, a single vector in a color flow might show that the color red in one image (the tail of a vector) became the color purple (the tip of the vector) in a different lighting condition. It is important to note that the color flow field for each pair of images is *exactly the same*, since it represents the same lighting change, and hence the same mapping of colors. The form of the vector field has nothing to do with the composition of the images, but only how the lighting changed.

The main idea of this work is that if we observe joint color changes in enough different lighting conditions for one object, we can understand how those color changes will affect the appearance of other objects, thus achieving a sort of color constancy *without ever having to calculate the illumination itself*. To work

best, the object we observe should have as many colors as possible, so that we can see how each color in the color cube is mapped under various lighting changes.

To record common lighting changes, we created a poster on a color plotter with the full printable spectrum of colors (upper left of Figure 5). We mounted this poster on the wall of our office and started recording video of the poster. The video continued over a 24 hour period, recording changes in the ambient lighting conditions due to the sunrise and sunset, clouds passing by, lights being turned on and off in the office, computer monitors, and shadowing effects. These lighting changes were not specially designed, but were simply common light changes due to normal office activity and light coming from the office window. Figure 5 shows the poster under two different lighting conditions, one in midday and one in early morning.

Any two frames taken from this (extremely boring) video allow us to compute a single color flow, i.e. a map from the colors in one image to the colors in the other image. The collection of all possible pairs of images from the video give us a large collection of color flows. We can then ask questions like, “What single color flows represent the best linear approximation of all of the measured color flows?” These are the principal components of the color flow fields.

Having measured a large number of flows from the poster video and having computed the principal components of these color flows, we can now apply them to other images to see what sorts of changes they induce in other images. The upper right of Figure 5 shows a single picture of my face. Using the information derived from the color flows measured on the poster, we can infer what my face would look like under various changes in lighting that occur in an office setting like our own. Some of these synthetically generated lighting conditions, generated by applying the measured color flows from the poster to the picture of my face, are shown in the bottom right of the figure.

Finally, in our published work, we describe how given two images, we can determine whether one image can be mapped to the other image using the statistically common flows (also called *eigenflows*). This allows us to use color in recognizing object identity without any explicit lighting model and without any of the classic brittle assumptions about scene decomposition.

By leveraging the statistics of everyday lighting changes and how they affect a stationary multi-colored object, we were able to construct a model which is very widely applicable. Furthermore, if a robot found itself in a situation where the statistics of the lighting changes were different, it could produce a new model of the local joint color changes by repeating this procedure with any multi-colored scene or object.

Summary of recent work. These two completed applications, learning from one example and learned color constancy demonstrate my philosophy of reducing the load on the computer’s teacher. In the next section, I describe a current project that also fits in the domain of object recognition.

3 Current Work

In this section I describe an exciting ongoing project that addresses the problem of learning from a single example in a wide range of new application areas. While the first project on digit recognition addressed the problem of *classification* or *class recognition*, this work addresses the problem of identifying an instance of a specific object. This task has been referred to in the computer vision literature as *object identification*. For example, the goal of this work would be to recognize Alice’s car, rather than a car in general, or to recognize Bob’s face, rather than any face.

While object identification has received a great deal of attention recently, most of the successful work has focussed on objects that do not change appearance dramatically from photograph to photograph [22]. In contrast, we attack the problem of object identification in very challenging domains such as face identification and car identification. Faces are difficult to identify due to the fact that they can change both shape

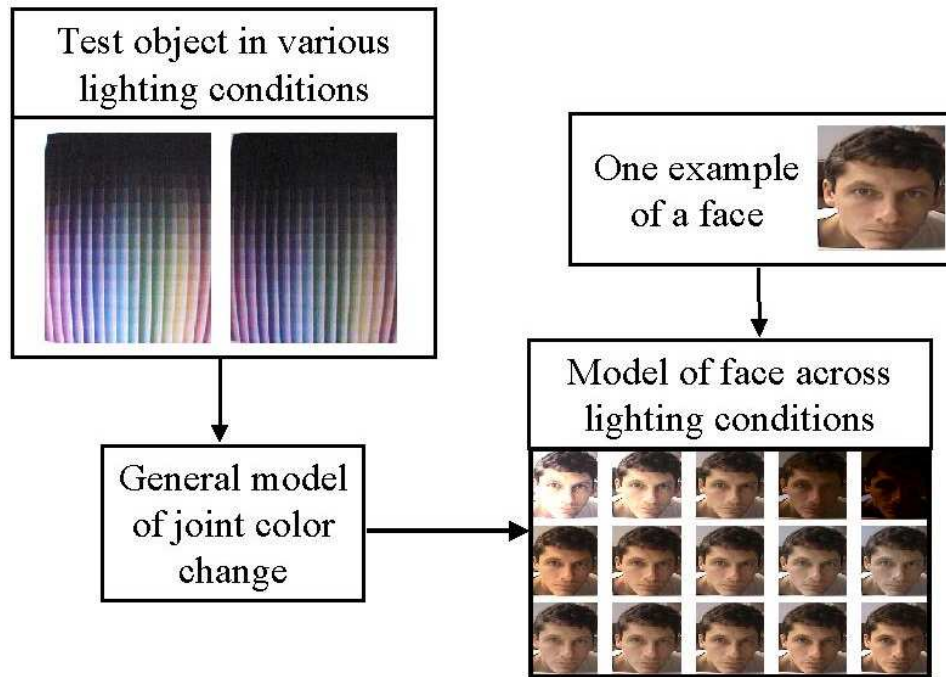


Figure 5: Diagram of method for sharing models of joint color variability. By representing typical joint color changes of a test object (upper left of figure), a general model of joint color change is developed. This model, combined with a single example of a new object (a face in this case) produces a model of the new object under common lighting changes (lower right).

(due to expressions) and color (due to temperature, illness, beards, makeup); and cars are difficult to identify since many of them are highly similar to each other, and yet any given car may change appearance dramatically when viewed in different lighting or from a different angle because of its highly reflective surface. In other words, for these groups of objects (cars and faces) and many others, the *within instance* variation can be very large, while the between instance variation can be small. An example of the object identification problem is illustrated in Figure 6.

Of course, one way to address the problem of object identification is to collect many examples of each object one wishes to identify. But this is exactly what we are trying to avoid, since this puts too heavy a burden on the trainer. We wish to be able to identify an object when we have only seen it once before. And because we are trying to do this in the domain of highly variable objects, it is even more challenging. We will allow ourselves the luxury, however, of seeing pairs of objects from the same *class* that either match or do not match. It is from these pairs that we hope to learn, for objects we have not yet seen more than once, a strategy for identification. We will first describe the system as it currently stands and has been reported on [12, 11]. Then we will discuss two extensions we hope to accomplish in future work to make the system both more general and more autonomous.

Learning with Hyper-Features. There is a single fundamental idea behind this work, that to recognize specific objects which are highly variable we must find parts of those objects that are both unique (with respect to other objects) and stable in appearance (i.e. that don't change under different viewing conditions).

What makes this problem interesting is that for each new object, the part of that object which is most

stable and most unique, or *salient*, may be in a different location than on other objects of the same class. For example, one car may be identifiable by its unusual pinstripes, while another car may have door handles in an unusual location. One face may have a large nose, while another has bushy eyebrows. The key question for this research is, “If we only get to see a single example of an object, and cannot study its variability over different appearances, how can we predict the stability of a feature of that object?”

We answer this question by seeking out parts of the image of an object with certain properties. Let the object we have seen be Object A, and the object we are comparing it to be Object B. The first property we want is that if the patch from Object A matches closely the patch from Object B, then this should indicate the objects are likely to be the same object. Note that not all patches have this property. For example, if the patch from Object A is just a white constant patch, then the knowledge that this happens to match the corresponding patch in Object B *is not a good indication* that the two objects are the same, since, for example, the probability of one car having a constant white patch in the same relative location as another car is quite high.

Similarly, a patch is more informative if when the patch of Object A does *not* match the patch from Object B it is a strong indication that the objects do *not* match. Said another way, we are looking for patches which we suspect, when compared with another object, will give us a lot of information about whether the two objects are the same or not.

Consider the case of identifying cars. Because at test time, we will have only a single image of a car to compare to, we must derive our notion of the utility of patches ahead of time, from other pairs of cars that match or mismatch. Since there is such an enormous number of patches possible, we cannot learn about every patch, but must learn a function of features of patches that determines whether that patch will be useful or not. It is these features which tell us about the likely utility of a patch for matching that we call *hyper-features*.

The method for learning patch saliency from a training set of matched and mismatched pairs is quite complex and uses modern statistical techniques (generalized linear models [24], mutual information estimation [9], and recent feature selection techniques [10]) to learn a function from the hyper-features of a particular patch to an estimate of the future informativeness of that patch. Every patch in the image, from a grid of patches at different scales, is evaluated and assigned a numerical informativeness value measuring the expected information content of the patch.

The left of Figure 7 shows a sample face from the face experiments, along with the salient patches selected by the algorithm. It is important to note that the algorithm selects these patches *before seeing a potential match*. Thus, it selects these patches based only on their appearance and position in *a single image*



Figure 6: *The Identification Problem: Which of these cars are the same?* The two cars on the left, photographed from camera 1, also drive past camera 2. Which of the four images on the right, taken by camera 2, match the cars on the left? (Our system gets both correct.)

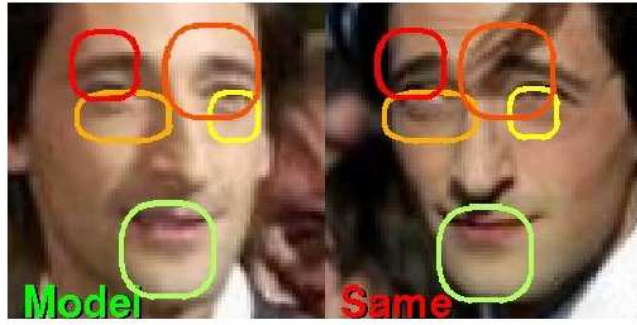


Figure 7: The patches outlined in the left picture were predicted to be useful for recognition by our saliency estimation method. Note that these patches were selected by looking *only at the image on the left* and yet the algorithm successfully determined that these patches were both relatively unique to this individual and also relatively stable. Notice for example, that there is no patch on the nose, which changes significantly with perspective. When the image on the right was presented, the quality of the matches for the salient patches indicated that this was the same person.

(the image on the left in this case). When the patch on the right is presented at test time, the relatively good match between the selected patches and the corresponding patches in the other image causes the system to vote for a “match” rather than a “mismatch”, which happens to be correct in this case. Our system has outperformed all other algorithms on this very difficult class of problems, and additional details can be found in our publications [12, 11].

Planned Improvements. While this system has effectively learned a function that enables it to understand the salient and stable features of an object from just a single image, it still has several significant drawbacks that we plan to address in ongoing work. Currently, for each new object category (cars or faces, for example) the system must be provided with a large training set of matched and mismatched pairs so that it can learn the function to predict saliency from hyper-features. Furthermore, for both training and testing, these pairs of images must be *aligned* or *registered* so that the correspondence between patches in the two images is clear. We hope to overcome both of these issues. First we address the problem of facilitating the collection of training data.

At present, to provide pictures of matching cars for training our hyper-feature system, we have two cameras taking pictures (automatically) from two locations on the same street. The views taken from each camera provide a training pair. However, not all cars that go by camera 1 are seen in camera 2. Thus a human observer must manually assemble pairs of cars from the two cameras that match and provide them as training data. We propose to obviate the human oversight by *tracking the same vehicle as it changes position* with respect to a single camera system. Then the first view and last view of the tracked object would be different enough to provide a training pair such as those used in the experiments above, *with no human supervision*. While this could in principle be done with a single immobile wide field-of-view camera, we believe we can collect better data by using a mobile tracking camera that pans a camera through a wide angle to follow a vehicle through a wide range of motion. We are currently adding dynamic tracking capability [33] to robots in our lab, so the software effort to achieve this is already well underway. In addition, we have requested funding for a two degree-of-freedom pan-tilt camera and video acquisition system that can be dedicated to this purpose and other goals in this grant. Finally, our video-enabled mobile manipulation system (see Figure 9) can be used for such acquisitions. Using these systems, we can track vehicles, people, or other

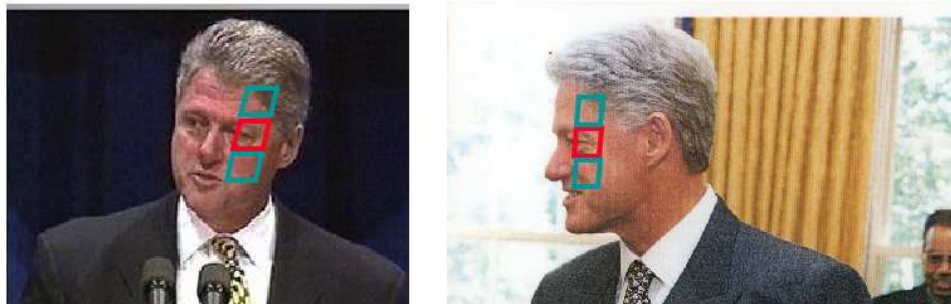


Figure 8: When images cannot be aligned, there may not be enough matching keypoints (center of red region) to establish an object match. However, by using the local coordinate system established by the affine keypoint (red parallelogram), additional patches can be defined that may provide enough information to confirm a match (green regions).

moving objects through wide angles and large changes in relative position, enabling us to gather an arbitrary amount of “matching” pairs in a mostly unsupervised manner.

One way to generate data which gives different perspectives on objects that are *not* moving on their own is to have a robot manipulate the objects to get different perspectives on them. The UMass robotics group has a strong record in grasping and manipulation [32, 17], and as described in the next section, we plan to use this expertise, along with the robotics equipment obtained under a recent NSF CRI equipment grant (see below), to enable the automatic “study” of objects by robots. This will be one of the basic capabilities that I plan to impart to our mobile manipulation platform to increase its autonomy and allow it to be more self-taught. More details on these capabilities are given in the next section.

The second goal in improving the hyper-feature work is to eliminate the need for registration of the image pairs before they can be compared. Not only will this eliminate a potentially unreliable part of the current system, but it will also allow us to learn to identify objects even when there is no reasonable registration of the two images of the object. For example, there is no reasonable registration of two pictures of a face when one is taken from the side and the other is taken from the front. However, there still may be features that are visible in both photos and which are enough to establish a match.

To achieve this second goal, we propose using recent “invariant keypoint” techniques [22, 25] to establish a small number of correspondences between two pictures. Then each corresponding pair allows us to establish corresponding *local coordinate systems* [20] on the pictures. These local coordinate systems allow comparisons of corresponding patches in the two images *without* putting the full images in correspondence. At the same time, even regions that do not qualify as “keypoints” and thus are difficult to put in correspondence directly can be compared by using the locally established coordinates. This idea is illustrated in Figure 8.

While our hyper-feature system already demonstrates significant autonomy by understanding important properties of an object from one example, we believe our proposed extensions will dramatically increase this autonomy in learning about the world, and have a major impact on general learning.

4 New Projects and Future Directions

In this section, I discuss new initiatives in two very different areas, robotics and optical character recognition. Despite their differences, these projects share the theme of this research—that they focus on allowing

machines to learn on their own, and to develop new capabilities with minimal human supervision.

Self-Taught Robots. A vision system that cannot move or manipulate objects in the world can only learn about those objects or scenes that are put in front of it. To become a truly explorative learner, it must be able to move and experiment with its environment. A modern mobile robot, equipped with a sophisticated manipulator such as a Barrett Whole Arm Manipulator (see Figure 9) and a vision system, is a powerful tool for exploration and autonomous learning about the real world [6]. Still, however, most robotics applications are painstakingly programmed using only the most rudimentary learning and vision capabilities.

Recently, as a co-PI on a NASA funded project for mobile manipulation, I have begun intensive collaborations with the Laboratory for Perceptual Robotics (LPR) at UMass. My students and I are rapidly adding capabilities like visual tracking [33], object recognition [22], visual servoing [19], motion segmentation [13], and depth estimation [18] to the lab's humanoid robot system, Dexter. Because the basic hardware, control software, and architecture for the robot has been in place for years, adding these capabilities is not much more difficult than implementing "software-only" vision algorithms. Dexter's flexible hardware and communications architecture allows the addition of a virtually unlimited number of video computing nodes, which receive digital FireWire video inputs and publish results of their computations asynchronously using a standard communications protocol (NDDS). Any control algorithm can subscribe to these communications and use the published information with minimal computational overhead. Thus, it is straightforward to add new capabilities to the robot. In addition, the author's recent NSF equipment grant with robotics Professor Oliver Brock and others provides funding for a new vision-enabled mobile manipulation platform for dedicated research (Figure 9).

Since the inception of AI, it has been argued not only that implementing vision on a robot would be the ultimate test for computer vision, but that it may be *necessary* to embody a vision system in order to duplicate human levels of performance. While this point of view faded into the background for the last 20 years, it is starting to re-emerge [8, 15, 14, 13, 1].

It seems an opportune time to reconsider the marriage of robotics, computer vision, and modern learning techniques to achieve more sophisticated ends. This work has already started at UMass, for example in learning visual features to develop categories of objects for specific grasps [8, 31], and elsewhere in using egomotion to segment objects [13] as well as correlating periodic signals across visual and other sensors [1].

The interplay between vision and robotics, especially for robots that can move and manipulate, is so rich that dozens of fundamental investigations could be described. However, a short list of our first projects will include

- manipulating objects to autonomously study invariances,
- manipulating objects to separate specular (reflective) and matte components of appearance,
- correlating specular components of an object with grasp slip,
- improving visual servoing using our own hyper-feature object recognition software and other recent object recognition algorithms, and
- using motion-on-contact to establish the depth of an object and autonomously learn depth cues (similar to [13]).

Some of these tasks support the others. For example, it is difficult to separate specular and non-specular components of an object without manipulating it, but if we can manipulate it to achieve this, then we can use the specularity as a feature to try to predict grasp slip. Thus, the basic capabilities obtained by leveraging vision and object manipulation immediately contribute to higher level goals.

Some may see a foray into robotics as an unnecessary distraction from the main work of Computer Vision. On the contrary, we see it as a critical source of real problems, and an opportunity for finding new solutions based upon the ability to probe the world and see immediate feedback. To be self-taught, one must

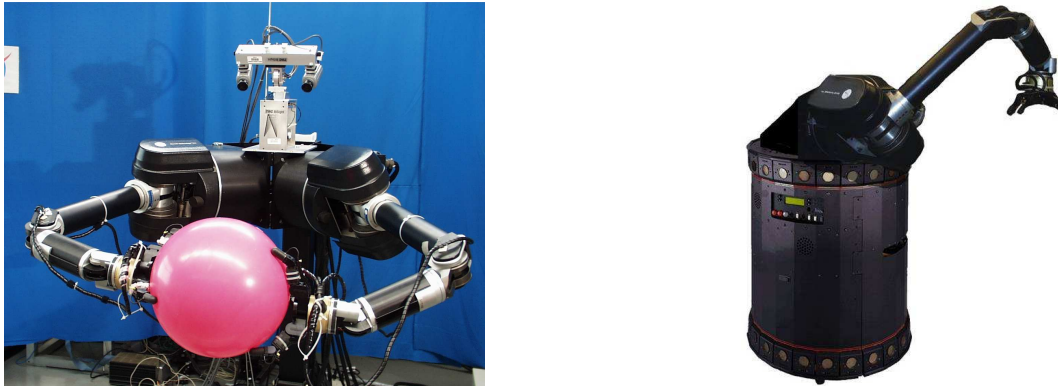


Figure 9: **Left.** The UMass stationary upper torso humanoid robot, Dexter. **Right.** The mobile manipulation platform to be assembled from a mobile base and a Barrett Whole Arm Manipulator under the author’s recent NSF CRI award.

have *some* source of actions and feedback, and thus real-world mobile manipulator seems an ideal platform to achieve many of the goals of this research.

Self-Taught Character Recognition. As a final example of my approach, I return to a classic problem, optical character recognition (OCR). Commercial systems for optical character recognition are typically programmed with hundreds or thousands of fonts, so that when a document is scanned, characters can be immediately recognized with high accuracy before any context is used at all. Relatively simple *language models* augment these systems to disambiguate characters that are still difficult to distinguish due to the original document’s degradation or poor scanning quality. These commercial OCR systems have impressive accuracy when used on documents with known fonts or with relatively little noise. However, when the font is unknown, or when the documents or scanning have been significantly degraded, these methods still have unacceptable accuracy.

It may be surprising to learn that humans can often do perfect OCR *even when the characters that make up a font are completely unknown and have arbitrary shapes*. That is, they can do OCR with *zero training examples* of a new font even though they cannot recognize any of the individual letters in isolation. This is evidenced by the fact that humans can easily solve *cryptograms* in which each letter has been substituted (consistently) with some other letter. Clearly the shape of a character now carries no direct information about its identity. Rather, it is the *similarity* among characters that provides a clue. In particular, a cryptogram can be solved by clustering characters into equivalent categories, and then studying the frequency and joint occurrence of characters in each cluster. This approach of visual clustering, followed by statistical decoding, has been suggested in the OCR literature [5, 3, 4]. It is appealing in that it allows knowledge learned from one task (the study of language statistics for a known character set) to be applied in solving a problem where no labeled training data are available, hence allowing the computer to essentially teach itself a new font.

While this approach has been pioneered by others (Breuel, in particular), there is still lots of room for its improvement in the analysis of highly degraded documents and handwriting. In particular, I propose to

- collaborate with the Machine Learning group at UMass to use the most recent developments in statistical text and document analysis (e.g. [2]) to increase the power of the language model decoding (I have already started a collaboration with Professor Andrew McCallum on this project),
- use more sophisticated visual clustering techniques to improve performance on the visual component of this problem, including the “skimming” technique described below, and

- develop accurate and fast approximation techniques to make the results of this method competitive (in speed) with faster but less accurate training based approaches.

My idea of “skimming” is to cluster only characters that can be clustered with high confidence. An iterative procedure is used in which initial clusters are reduced in size by sorting elements by likelihood within their own cluster, and “skimming off” all but the most typical members. The idea is to produce clusters with near-perfect purity (having only a single character per cluster) even in noisy environments. Subsequently, a new cluster model is developed, and high-confidence points are slowly added. With a strong language model and highly pure clusters, I speculate that it is not necessary to cluster every character, but only to cluster a subset of the characters with high confidence. This use of pure clusters can be compared to methods people use to understand sloppy handwriting, focusing first on easy to group characters, using these to do inference, and building from these high confidence regions.

5 Educational Initiatives

I propose educational initiatives in two areas. The first is in the area of minority and low-income outreach, involving a group of students at an urban Massachusetts school. The second area involves curriculum development and curriculum guidance at the college and graduate levels at UMass, Amherst.

AVID outreach. The Advancement Via Individual Determination (AVID) program is a non-profit organization started 25 years ago in the San Diego public schools. Its goal is to increase the percentage of historically underserved teenagers who apply, attend, and remain in college to obtain college degrees. AVID has been remarkably successful in achieving this goal—minority and low-income students who enter the program go to college and stay in college *at the same rate as middle class white students*. Specifically, more than 90% go to college, and of those, 89% remain after two years.

My wife, Carole Learned-Miller, is principal of the Dr. Martin Luther King, Junior School in Cambridge, Massachusetts, where she manages the AVID program for grades 6-8. The King school has a large minority population (e.g. 65% of students are black, both African American and others), and more than 70% have “free lunch” status, which is to say that they come from low-income families. The King school is the first school in Massachusetts to adopt AVID.

AVID recommends that students be frequently exposed to opportunity and exciting careers through a series of speakers and field trips. As part of my educational outreach, *I will lead an annual field trip from Cambridge to Amherst and Northampton for the eighth grade of the King school*. The free trip will comprise

- tours of and hands-on demonstrations in the Computer Vision Laboratory and the Laboratory for Perceptual Robotics,
- presentations by at least one minority graduate student and one professor about a career in science,
- a presentation about successful minorities in Artificial Intelligence, highlighting top minority researchers and professors in AI throughout the country, and
- a visit to the Engineering Program at the all-women Smith College campus (Northampton, MA) as a special motivation for girls to get involved in science and engineering.

There are ample funds available through the Cambridge Public Schools to finance such field trips, making them free for students. In addition, both my wife (a Smith alumna) and the UMass CS department have numerous contacts at Smith College to facilitate the coordination of the Smith College visit. In addition, I have already received a commitment from a senior African American UMass graduate student, Gary Holness, to

speak the first year. My wife and her colleagues at the King school believe that these are among the most important events for teenagers, and that this event will provide an important and exciting piece of support for the well-regarded AVID program at the King school.

Curriculum Development and Guidance. In developing the type of capabilities discussed in this proposal, the field of Computer Vision must incorporate more sophisticated techniques from mathematics, statistics, and other fields. Today's computer science students are not generally well-equipped to understand these latest developments. To prepare students for these new directions in Computer Vision, I would like to address three curriculum issues. I will

- develop *and advertise* two graduate “sub-curricula”, one for Machine Learning and one for Vision,
- develop a new undergraduate course in computer vision that emphasizes the interdisciplinary nature of the subject and the modern material,
- develop a new graduate level seminar called “Learning to See”, which emphasizes the type of projects presented in this proposal.

Graduate sub-curricula. Preparing to do top quality research in Computer Vision today is a demanding enterprise. In addition to learning traditional computer science skills in algorithms, theory of computation, and programming, students may find that they need sophisticated knowledge of statistics, probability and other areas of higher math (e.g. differential geometry) that they have not encountered. Organizing an approach to learning this huge array of material is daunting for new graduate students, especially when they may not even know what they are missing.

I will lay out sub-curricula specifically geared toward machine learning and computer science to address these issues. The curricula will be organized around the idea of filling in background in order to understand a sequence of real, carefully selected research papers, rather than simply a course sequence. Students will evaluate for themselves whether or not they need a particular course by their understanding of these papers. As an example, many incoming students have had little exposure to signal processing and linear systems. If they cannot understand completely a paper that uses concepts like Fourier transforms and linear superposition, it will be suggested that they fill in this background with a signal processing course in the Electrical Engineering Department. I developed such a “curriculum map” in graduate school and it was used and appreciated by a number of my peers there. I plan to expand this and publish it on the web for general use. Finally, the idea is not for these courses to be required, but for students to have a complete understanding of the broad background required in modern Computer Vision, and one route for acquiring it.

Undergraduate vision. I have taught and will continue to teach at both the undergraduate and graduate levels at UMass. Computer vision has experienced rapid change in the last ten years, and like any immature subject, needs to be continuously adapted to new research and developments. This year, I co-taught an undergraduate course in computer vision. My colleague handled many of the more traditional aspects while I focussed on incorporating probabilistic, statistical, and machine learning techniques. A major challenge is to cover recent developments without sacrificing essential classical material.

Graduate vision. In the fall of 2005, I will teach a course entitled “Learning to See”. This new course is dedicated to understanding the type of work presented in this proposal. It will focus not only on modern learning techniques, but on all aspects of learning to see, such as what information sources are available in the environment from which we can learn, and how humans achieve such remarkable results in visual tasks.

A major part of the course will be an introduction to non-parametric statistics and information theory, essential aspects of this type of work that are difficult for most students to pursue due to the large numbers of prerequisites.

Prior NSF Support: None

References

- [1] A. Arsenio and P. Fitzpatrick. Exploiting amodal cues for robot perception. *International Journal of Humanoid Robotics*, 2005.
- [2] R. Bekkerman, R. El-Yaniv, and A. McCallum. Multi-way distributional clustering via pairwise interactions. In *International Conference on Machine Learning*, 2005.
- [3] T. M. Breuel. Classification by probabilistic clustering. In *International Conference on Acoustics, Speech, and Signal Processing*, 2001.
- [4] T. M. Breuel. Modeling the sample distribution for clustering ocr. In *SPIE Conference on Document Recognition and Retrieval VIII*, 2001.
- [5] T. M. Breuel and K. Popat. Recent work in the document image decoding group at Xerox PARC. In *Proceedings of the DOD-sponsored Symposium on Document Image Understanding Technology (SDIUT 2001)*, 2001.
- [6] Oliver Brock and Roderic Grupen. Integrating manual dexterity with mobility for human-scale service robotics—the case for concentrated research into science and technology supporting next-generation robotic assistants (whitepaper). <http://www-robotics.cs.umass.edu/oli/publications/src/whitepaper-mobmanip.pdf>, 2004. Signatories: Robert Ambrose, Oliver Brock, Rodney Brooks, Chris Brown, Joel Burdick, Andrew Fagg, Roderic Grupen, Jeffrey Hoffman, Robert Howe, Manfred Huber, Oussama Khatib, Vijay Kumar, Larence Leifer, Maja Matarić, Alan Peters, Kenneth Salisbury, Shankar Sastry, Bob Saveley, and Stefan Schaal.
- [7] G. Buchsbaum. A spatial processor model for object color perception. *Journal of the Franklin Institute*, 310, 1980.
- [8] J. Coelho, J. Piater, and R. Grupen. Developing haptic and visual perceptual categories for reaching and grasping with a humanoid robot. *Robotics and Autonomous Systems Journal, Special Issue on Humanoid Robotics*, 37(2-3):195–219, 2001.
- [9] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.
- [10] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004.
- [11] Andras Ferencz, Erik Learned-Miller, and Jitendra Malik. Building a classification cascade for visual identification from one example. In *ICCV*, 2005.
- [12] Andras Ferencz, Erik Learned-Miller, and Jitendra Malik. Learning hyper-features for visual identification. In *Advances in Neural Information Processing Systems*, volume 18, 2005.
- [13] P. Fitzpatrick. First contact: An active vision approach to segmentation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003.
- [14] P. Fitzpatrick. Object lesson: Discovering and learning to recognize objects. In *Proceedings of the 3rd International IEEE/RAS Conference on Humanoid Robots*, 2003.

- [15] P. Fitzpatrick and G. Metta. Towards manipulation driven vision. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2002.
- [16] P. Grother. NIST special database 19 handprinted forms and characters database. Technical report, National Institute of Standards, 1995.
- [17] S. Hart, R. Grupen, and D. Jensen. A relational representation for procedural task knowledge. In *Proceedings of the 2005 American Association for Artificial Intelligence (AAAI) Conference*, 2005.
- [18] B. K. P. Horn. *Robot Vision*. MIT Press, 1986.
- [19] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, 1996.
- [20] S. Lazebnik, C. Schmid, and J. Ponce. Semi-local affine parts for object recognition. In *British Machine Vision Conference*, 2004.
- [21] Erik G. Learned-Miller. Data driven image models through continuous joint alignment. To appear in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005.
- [22] David Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [23] J. J. McCann, J. A. Hall, and E. H. Land. Color mondrian experiments: The study of average spectral distributions. *Journal of the Optical Society of America*, A(67), 1977.
- [24] P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Chapman and Hall, 1989.
- [25] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. To appear in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005.
- [26] E. Miller, N. Matsakis, and P. Viola. Learning from one example through shared densities on transforms. In *IEEE Computer Vision and Pattern Recognition*, 2000.
- [27] E. Miller and K. Tieu. Color eigenflows: Statistical modeling of joint color changes. In *International Conference on Computer Vision*, volume 1, pages 607–614, 2001.
- [28] Erik G. Miller. *Learning from one example in machine vision by sharing probability densities*. PhD thesis, Massachusetts Institute of Technology, 2002.
- [29] Erik G. Miller and Christophe Chef d’hotel. Practical non-parametric density estimation on a transformation group for vision. In *IEEE Computer Vision and Pattern Recognition*, 2003.
- [30] Y. Moses, S. Ullman, and S. Edelman. Generalization to novel images in upright and inverted faces. *Perception*, 25:443–462, 1996.
- [31] J. Piater. Learning visual features to recommend grasp configurations. Technical Report 2000-40, University of Massachusetts, Amherst, 2000.
- [32] R. Platt, A. H. Fagg, and R. Grupen. Manipulation gaits: Sequences of grasp control tasks. In *Proceedings of the 2004 IEEE Conference on Robotics and Automation (ICRA)*, 2004.

- [33] B. Scassellati. A binocular, foveated active vision system. Technical Report MIT Artificial Intelligence Laboratory Technical Report 1628, Massachusetts Institute of Technology, 1998.
- [34] Kinh Tieu and Erik Miller. Learned color constancy. In *Advances in Neural Information Processing Systems*, volume 16, 2003.