Viewpoint Invariant Matching via Developable Surfaces

Bernhard Zeisl, Kevin Köser, Marc Pollefeys

Computer Vision and Geometry Group
ETH Zurich, Switzerland
{zeislb,kkoeser,pomarc}@inf.ethz.ch

Abstract. Stereo systems, time-of-flight cameras, laser range sensors and consumer depth cameras nowadays produce a wealth of image data with depth information (RGBD), yet the number of approaches that can take advantage of color and geometry data at the same time is quite limited. We address the topic of wide baseline matching between two RGBD images, i.e. finding correspondences from largely different viewpoints for recognition, model fusion or loop detection. Here we normalize local image features with respect to the underlying geometry and show a significantly increased number of correspondences. Rather than moving a virtual camera to some position in front of a dominant scene plane, we propose to unroll developable scene surfaces and detect features directly in the "wall paper" of the scene. This allows viewpoint invariant matching also in scenes with curved architectural elements or with objects like bottles, cans or (partial) cones and others. We prove the usefulness of our approach using several real world scenes with different objects.

1 Introduction and previous work

Utilizing image based features to compactly describe image content or to identify corresponding points in images has become a de facto standard in computer vision over recent years. Applications where a feature based approach proved particularly successful are for example structure-from-motion, image registration, visual SLAM systems or object detection and recognition. However, a major problem when trying to find correspondences between widely separated views is that the appearance of objects can change drastically with viewpoint. To remedy this problem techniques have been developed which normalize images or image regions such that they become (at least approximately) invariant to viewpoint changes. In case one matches two images (without depth information) against each other the most popular method is to use local image features that compensate for the first order effects of viewpoint change by normalization, i.e. affine transformations (cf. to [1, 2]) or slightly weaker models (e.g. [3]). Since scale, orientation and (anisotropic) stretch are all effects that could have been caused by a viewpoint change they need to be factorized out and therefore it is not possible to distinguish, e.g. real-world circles from ellipses or a small round dot from a huge sphere any more. This is a general dilemma of discriminative power

Fig. 1. Examplary objects and their developed surfaces. To normalize wrt. viewpoint changes we propose to detect and describe features in the unrolled surface textures rather than in the original images.

vs. invariance. For an in depth discussion of invariant feature constructions we refer the reader to [4].

When depth information is available one can normalize wrt. the given 3D structure. In [5, 6] the authors have shown that it is possible to virtually move a camera to a frontal view and then render a canonical representation of a local image feature. Similarly, normalization can also be obtained by extracting vanishing points in Manhattan scenarios [7, 8] and virtually rotating the camera. However, all these approaches have strong limitations: While [5] still requires an affine detector and thus the number of features obtained is limited, other approaches [6–8] rely on the existence of dominant scene planes. In contrast to this assumption we observe that many structures in our environment are also curved, e.g. like cylinders, cones or consist of free-form shapes. Many man-made objects are made by bending sheets or plates and thus - by construction - form *developable surfaces* that can virtually be "unrolled" when their geometric structure (depth) is known (see Fig. 1). For the particular application of pose-robust face recognition, Liu and Chen [9] coarsely approximate a human head via a 3D ellipsoid and back-project images onto its surface. Recognition is then conducted in the flat, but stretched and distorted texture maps. In comparison, we are interested in objects possessing developable surfaces and by this allow to create an undistorted texture map.

Further, in this work we follow the idea to develop such observed scene surfaces and to extract image features in the flat 2D wall-paper version of that very same surface, allowing for less invariant (and more discriminative) detectors/descriptors. In case scale is known (e.g. from a Kinect camera) the affine or perspective invariance requirement of the original problem is reduced to rotation in the image plane and can be reduced even further for surfaces such as cones or cylinders. We strongly believe that this technique is useful in several applications such as robotic scenarios, where a robot has to identify and manipulate an object, for automatically registering overlapping 2.5D or 3D models or loop detection in large structure-from-motion or SLAM systems.

The paper is organized as follows: Developable surfaces are discussed in Sec. 2, while Sec. 3 presents our algorithmic approach and Sec. 4 states implementation details. In Sec. 5 we illustrate experiments and results on both, data from active and passive consumer depth sensors, followed by concluding remarks.

Fig. 2. Developable surfaces present in our environment. Note, that in the left image only cones are highlighted although planes and cylinders exist as well.

2 Developable Surfaces

As our approach builds on the notion of developable surfaces, we start by briefly introducing the underlying concept. In general a surface with *zero* Gaussian curvature at every surface point is developable [10] and can be flattened onto a plane without distortion (such as stretching or shortening).

To determine the Gaussian curvature of a surface, suppose we are given a smooth function s that maps 2D parameters u, v to points in 3D space, i.e. $s : \mathbb{R}^2 \to \mathbb{R}^3$ such that $s(u, v) = (x, y, z)^T$. The graph S of this function is a two-dimensional manifold and our surface of interest in 3D space. The derivatives $s_u = \frac{\partial s}{\partial u}$ and $s_v = \frac{\partial s}{\partial v}$ of s with respect to the parameters u and v define tangent vectors to the surface at each point. Their cross product yields the normal vector $n = s_u \times s_v$ to the surface. The second partial derivatives of s with respect to u, v are now used for constructing the shape operator

$$II = \begin{pmatrix} L & M \\ M & N \end{pmatrix} = \begin{pmatrix} n^T s_{uu} & n^T s_{uv} \\ n^T s_{uv} & n^T s_{vv} \end{pmatrix}, \tag{1}$$

which is also called the second fundamental form of s. The principal curvatures κ_1, κ_2 of the surface at a given position are defined as the eigenvalues of II. They measure how the surface bends by different amounts in different directions at a particular point. Finally, the determinant $\det(II) = \kappa_1 \kappa_2$ denotes the Gaussian curvature; in case it vanishes everywhere on the surface (at least one of the eigenvalues is zero) the surface is developable. The intuition is that in direction of zero curvature the surface can be described as a line. Hence, the surface development is just an unrolling of all corresponding lines into one plane. We refer the interested reader to [10] for more details.

For example a cylinder is developable, meaning that at every point the curvature in one direction vanishes. Its mean curvature is not zero, though; hence it is different from a plane. Contrary, a sphere is not developable, since its surface has constant positive Gaussian curvature at every point. Other basic developable shapes are planes, cylinders, cones and oloids[1] and variants thereof, such as cylindroids, or oblique cones. Intuitively, they are flattened by rolling the object on

[1] An oloid is defined as the convex hull of two equal disks placed at right angles to each other, so that the distance between their centers is equal to their radius.

a flat surface, where it will develop its entire surface. In fact, all surfaces which are composed of the aforementioned objects are developable as well. In practice, many objects in our environment are made by bending sheets or plates and thus form developable surfaces. Fig. 2 illustrates several real-world developable surfaces; note that even such complex structures as the church roof top (Fig. 2 very right) are (piece-wise) developable.

3 Exploiting Developable Surfaces for Viewpoint Invariant Matching

In the following we present our approach of matching two views of a rigid scene, separated by a wide-baseline, by means of developable surfaces. However, we point out that the same techniques are applicable for identifying and recognizing a single object in a database, for loop detection or for automatically registering multiple overlapping textured 3D models. As input to our algorithm we assume two RGBD images with sufficient overlap. Given pixel-wise depth measurements $d_{u,v}$ and camera intrinsics K a 2.5D point cloud is obtained per view via

$$(x, y, z)^{\mathrm{T}}_{u,v} = K^{-1}(u, v, 1)^{\mathrm{T}} d_{u,v} \quad \forall u, v \in I, \tag{2}$$

with image coordinates u, v. Then our method progresses in four steps, which are (a) detection and parameter estimation of certain developable surfaces in the depth data, (b) generating flat object textures by means of developing the detected surfaces, (c) detecting/describing features in the unrolled images (i.e. in the surface) and matching against the other views, and (d) verification of found correspondences. We will explain them in more detail in following Sec. 3.1 to 3.4.

3.1 Multi-Model Estimation

As described in the previous Sec. 2, many different developable surfaces exist. In this paper we focus on three basic shapes, the plane, the cylinder and the cone, because these shapes possess a low parametric representation and thus are detected reliably in depth data. Identifying these surfaces falls into the category of multi-model estimation and several techniques have been suggested to cope with it, including randomized hough transform, sequential RANSAC or more recently J-Linkage [11], multi-structure segmentation [12] or more problem-specific machine-learning inspired geometric classification approaches in the spirit of [13].

3.2 Developing Surfaces

Subsequent to the initial model estimation and parameterization is the generation of a flat texture per detected model. We describe obtained mappings for principal geometric shapes in the following.

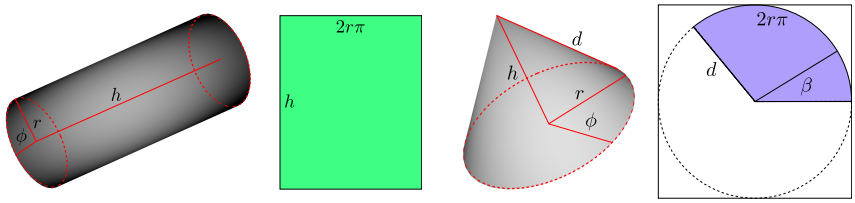

Fig. 3. (left) Cylinder with height h and radius r and its developed texture of dimensions $h \times 2\pi r$. (right) Cone with height h and radius r and its developed texture with radius d. Note that angles ϕ around cone axis and β in developed texture differ by $\frac{r}{d}$ and thus a 360 turn of the cone does not describe a full circle.

Planes From the previous model estimation (Sec. 3.1) we know a four parameter description $\pi_S = (\mathbf{n}^{\mathrm{T}}, d)$ with normal vector $\mathbf{n}$ and distance to the origin d, as well as a bounding box (or mask) for the region of interest on the 3D plane. Two orthogonal vectors $\mathbf{u}, \mathbf{v}$ in the plane are chosen as a basis $\mathcal{B}_S$ and we sample the plane in equidistant steps, i.e. we define a grid in the plane. Original image plane π_I and surface plane π_S together with the origin of $\mathcal{B}_S$ define a unique mapping $\mathbf{P}$. It is used to project each of the grid vertices (u_i, v_j) into the original image to obtain the appropriate color. The resolution is chosen such that we do not lose any image details; this means we project the four bounding box corners into the original image and evaluate the Jacobian matrix of the texture warp for some arbitrary grid resolution. Afterwards we increase or decrease the grid resolution such that the smallest minification between the developed surface and the original image is 1 (for details on texture mapping see [14]). In practice the transformation $\mathbf{P}$ will be a homography and the result is equivalent to [6]. Here, the frontal view of the plane coincides with the developed plane, however we will now generalize this to other developable surfaces.

Cylinders After model detection a cylinder is parameterized as $(\mathbf{c}, \mathbf{a}, r)$ with cylinder base center $\mathbf{c}$, axis vector $\mathbf{a}$ of length h and radius r. In order to unroll the cylinder we represent 3D points on the surface in their cylinder coordinates (r, ϕ, z) (with $\mathbf{c}$ as the origin). By removing the radius coordinate we obtain a 2D parameterization (ϕ, z) of the surface. The projection of surface points into the image plane π_I is thus defined by a unique mapping $\mathbf{P}$. The angular resolution in ϕ is determined to match the resolution along the cylinder axis and to obtain an image of aspect ratio $h \times 2\pi r$ for a full 3D development of the cylinder (see Fig. 3). In case scale is known (e.g. when a Kinect camera is used) and when it is desirable not to normalize over scale (e.g. because of similar features at different scales) we choose a metric surface resolution. Otherwise, we evaluate the local magnification/minification between the original image and the surface texture and ensure that no resolution is lost during unrolling. Given 2D coordinates (ϕ_i, z_j) in the unrolled surface texture, each corresponding 3D surface point (r, ϕ_i, z_j) is identified and projected into the original image via $\mathbf{P}$

to obtain the color. This mapping is very efficiently implemented on the GPU or using standard backward mapping on the CPU.

Cones A cone is parameterized and developed very similarly to the cylinder, taking into account that the surface tappers smoothly towards the apex. To obtain a flat surface texture, it is positioned with a line from the apex to the base circle of length $d = \sqrt{r^2 + h^2}$ (see Fig. 3) in the plane for development (imagine laying it on a piece of paper). Afterwards the apex is fixed and the cone rolled around it, resulting in a circle segment. The created circular texture contains the apex in the center, where the radial lines connect the apex with the cone's base circle (consequently of radius d, up to resolution). Thus 2D texture coordinates (β_i, d_j) are directly related to points on the cone surface. Similar to the cylinder, we backward map texture coordinates across the 3D surface into the original image to obtain the colors for the surface texture, maximizing its resolution.

3.3 Feature Detection and Matching

Feature detection is performed directly in the unrolled textures. This is conceptually different to [1, 5] which first detect features and then try to normalize these wrt. to viewpoint variations. It is related to [6–8], however these approaches only consider planes for normalization. The unrolled textures allow to reach perspective invariance with only normalizing in-plane rotation in the image (or similarity normalization in case absolute scale is unknown). Even better, since cylinder[2] and cone define an inherent reference direction with their axis, all features can be expressed with respect to this orientation rather than computing an orientation from the local region as for example in SIFT [3]. Consequently it is possible to extract very basic features on the surface, which is very fast on the one hand and on the other hand allows to distinguish local regions that differ only by scale, orientation or linear shape. All detected features on the different developed textures are combined to form the set of features for a RGBD image and are subsequently used for wide-baseline matching.

3.4 Correspondence Verification

Naturally the set of estimated matches contains numerous outliers, which do not satisfy the underlying camera pose change. Therefore, correspondences are checked using geometric verification (e.g. using RANSAC). As observed in [6], each feature does not only include information about the 3D position, but also the local normal. Additionally, when orientation is known from the cylinder or cone geometry or when local orientations are estimated from gradients [3], three characteristic directions are known at each 3D feature point. This allows for a stratified verification as in [6] or for a minimal solution in RANSAC that requires only a single correspondence.

[2] For the cylinder there is still a 180° ambiguity for the direction of the axis.

4 Implementation Details

We employ RANSAC to obtain model parameters for planes, cylinders and cones in the captured RGBD data. Since surfaces are mostly local and continuous we utilize a local sampling strategy, where consecutive samples are drawn within a 0.5m radius. Surface models are searched for in order of increasing complexity, i.e. initially planes are detected, followed by cylinders and cones. We limit the size of models to physically plausible extents for the expected outdoor or indoor environments. In addition, found models need to guarantee that they show sufficient support over their surface to avoid algorithmic plausible, but incorrect estimations. Consequently, we reject models whose support is only defined at isolated points or clusters. Once detected, we robustly determine the model size in the image and estimate the spatial extent (e.g. height of cylinder). Subsequently, initial model parameters are updated via a non-linear optimization on the evaluated inlier set. After each iteration 3D points supporting the estimated model are removed from the search space, which prohibits assignment to multiple models. This iterative procedure terminates as soon as no model with sufficiently large support is found any more.

As mentioned in Sec. 3.3, image feature estimation in the developed surfaces can be accomplished with a basic detector such as a Harris corner detector. However, since we aim to compare obtained matches from developed surfaces with matches in the original RGBD data, we chose standard SIFT as our detector and descriptor to guarantee comparability. (Note, that employing upright-SIFT would also treat our approach with favor due to its greater discriminative power.) Detected image features are matched against each other in their descriptor space. To eliminate ambiguous matches (e.g. between repetitive structures) all best matches are kept for which the distance ratio to the second best match falls below 0.6 (known as the ratio test in [3]). Then, each feature is additionally augmented by its position in 3D space, which is determined by the corresponding 3D surface model (according to the derived mappings in Sec. 3.2). For features in the original RGBD images we consider present pixel-wise depth measurements and neglect them in case no depth is available, e.g. due to occlusions.

To obtain a robust estimate of correct matches, we employ a correspondence rejection method via RANSAC, which samples from the feature correspondence set and estimates a 6 DoF transformation between the two views. Since correspondences between 3D points are explicitly defined, we utilize Procrustes analysis [15] (and do not need an iterative non-linear optimization scheme) for the estimation of rotation and translation. Finally, the estimated transformation is validated on all potential correspondences, which gives a final set of correct and consistent matches. These consistent matches are visualized in the different experiments presented in the following.

5 Results for Active and Passive Stereo Devices

In this section we demonstrate our novel technique for different scenes and cameras. Fig. 4, Fig. 5 and Fig. 6 illustrate obtained results for a synthetic setup,

Fig. 4. Wide baseline matching for a synthetic setup. (top row) Outer left and right image illustrate the found models in the 2.5D point cloud; images in-between show developed textures. (bottom row) SIFT matches consistent wrt. the underlying 6DoF transformation between the initial RGBD images and between developed surfaces.

Scene type	Synthetic setup	Floor (Kinect)	Table (Kinect)	Trees (stereo)	Pylon (stereo)
Matches orig. RGBD images	37	9	27	3	56
Matches developed surfaces	255	79	195	22	227
Enhancement ratio	6.89	8.78	7.22	7.33	4.05

Table 1. Quantitative comparison of SIFT descriptor matches between original RGBD images and developed surfaces for different scenes.

indoors scenes captured with a Kinect camera and outdoor scenes taken with a Fuji3D stereo camera, respectively. Rectified textures of detected planes are not illustrated due to space limitations; though, they are included in the evaluation.

Comparing feature detection and matching in the original images and in the images of developed surfaces (see Tab. 1) we can record the following: While approximately the same number of features are detected and an equal amount of potential matches is obtained, evaluation shows that for the latter the amount of finally remaining *correct* matches is significantly larger. Between the original RGBD images many potential matches are wrong due to viewpoint distortions in the descriptor space and thus need to be rejected. This validates that our approach of viewpoint invariant description of developable surfaces is able to extract features, which are stable over a variety of largely different viewpoints and improves wide-baseline matching considerable. In addition, rather than interpreting our approach as a competitor to standard feature matching, one should see it as an additional cue for obtaining more stable features.

6 Conclusion and Future Work

We have presented a novel technique to exploit depth information for viewpoint invariant matching between RGBD images. It develops the surface and detects

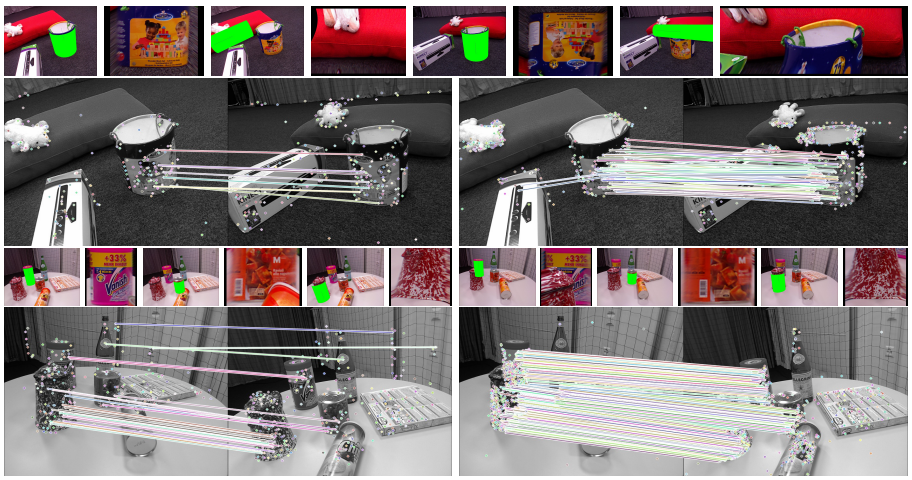

Fig. 5. Wide baseline matching for RGBD data captured with a Kinect sensor (Floor and Table scene). (1st and 3d row) Detected objects (green) and their respective developed surface for the two views. (2nd and 4th row) Consistent SIFT matches between original images and developed surfaces, respectively.

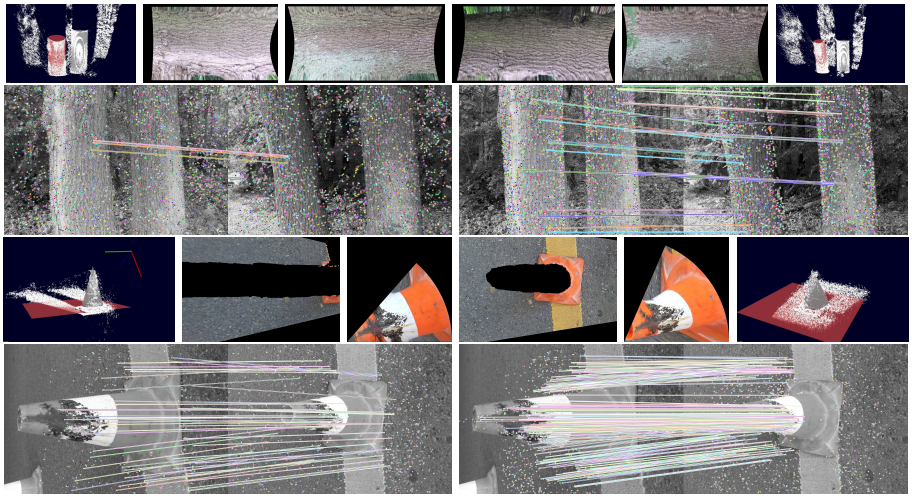

Fig. 6. Wide baseline matching between images taken by a Fuji3D consumer stereo camera (Trees and Pylon scene). (1st and 3rd row) Detected models in the 2.5 point cloud and their respective developed surfaces. (2nd and 4th row) SIFT feature matches between original scenes and developed surfaces, respectively. Note, that for the bottom experiment depth estimates are noisy and contain a considerable amount of errors, leading to degraded parameter estimation for the detected cone.

features in the surface's wall paper which removes perspective effects. While we have shown drastically increased number of matches as compared to classical SIFT, the feature detector can actually be chosen freely for the given application. Scale can easily be integrated (when known from the range sensor) to allow for using simple and more discriminative features such as Harris corners and in the special case of cylinders and cones even in-plane rotation is known. In any case, while we have demonstrated the approach as a competitor that outperforms standard SIFT by far, it should be clear that both can easily be combined for a real system. Compared to earlier viewpoint normalization approaches that relied on global scene planes we have shown that many other geometric shapes are feasible and demonstrated this using cones and cylinders. Possible further work will include the extension from detectable parametric surfaces to flattening of general (approximately developable) surfaces.

References

1. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A Comparison of Affine Region Detectors. IJCV **65**(1-2) (2005) 43–72
2. Mikolajczyk, K., Schmid, C.: A Performance Evaluation of local Descriptors. IEEE Trans. Pattern Anal. Mach. Intell. **27**(10) (2005) 1615–1630
3. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. IJCV **60**(2) (2004) 91–110
4. Gool, L.V., Moons, T., Pauwels, E., Oosterlinck, A.: Vision and Lie's Approach to Invariance. Image and Vision Computing **13**(4) (1995) 259 – 277
5. Köser, K., Koch, R.: Perspectively Invariant Normal Features. In: ICCV, Workshop on 3D Representation and Recognition. (2007)
6. Wu, C., Clipp, B., Li, X., Frahm, J.M., Pollefeys, M.: 3D Model Matching with Viewpoint-Invariant Patches (VIP). In: Proc. CVPR. (2008) 1–8
7. Robertson, D., Cipolla, R.: An Image-Based System for Urban Navigation. In: Proc. BMVC. (2004) 819–828
8. Cao, Y., McDonald, J.: Viewpoint Invariant Features from Single Images using 3D Geometry. In: Workshop on App. of Comp. Vision. (2009) 1–6
9. Liu, X., Chen, T.: Pose-robust Face Recognition using Geometry Assisted Probabilistic Modeling. In: Proc. CVPR. Volume 1., IEEE (2005) 502–509
10. Kuehnel, W.: Differential Geometry : Curves - Surfaces - Manifolds. American Mathematical Society, Providence, R.I (2006)
11. Toldo, R., Fusiello, A.: Robust Multiple Structures Estimation with J-Linkage. In: Proc. ECCV. (2008) 537–547
12. Wang, H., Chin, T.J., Suter, D.: Simultaneously Fitting and Segmenting Multiple-Structure Data with Outliers. IEEE Trans. Pattern Anal. Mach. Intell. **34**(6) (June 2012) 1177 –1192
13. Rusu, R., Holzbach, A., Blodow, N., Beetz, M.: Fast Geometric Point Labeling using Conditional Random Fields. In: Proc. IROS, IEEE (2009) 7–12
14. Heckbert, P.S.: Fundamentals of Texture Mapping and Image Warping. Technical report, U.C Berkeley (1989)
15. Eggert, D., Lorusso, A., Fisher, R.: Estimating 3-D Rigid Body Transformations: A Comparison of Four Major Algorithms. Machine Vision and Applications **9**(5) (1997) 272–290