M

Machine Recognition of Objects

Tomaso Poggio and Shimon Ullman
Department of Brain and Cognitive Sciences, McGovern Institute, Massachusetts Institute of Technology, Cambridge, MA, USA

Related Concepts

►Object Class Recognition (Categorization); ►Visual Cortex Models for Object Recognition

Definition

Machine recognition of objects is the task of locating and recognizing a given object in an image and consists of the following steps: object detection, feature extraction, and recognition.

Background

Early computer vision recognition schemes focused primarily on the recognition of rigid three-dimensional (3D) objects, such as machine parts, tools, and cars. This is a challenging problem because the same object can have markedly different appearances when viewed from different directions. It proved possible to deal successfully with this difficulty by using detailed 3D models of the viewed objects, which were compared with the projected 2D image (e.g., [14, 18, 33]). Over the last decade or so, computational models have made significant progress in the task of recognizing natural object categories under realistic, relatively unconstrained viewing conditions. Within object recognition, it is common to distinguish two main tasks: identification, for instance, recognizing a specific face among other faces, and categorization, for example, recognizing a car among other object classes. We will discuss both of these tasks below and use "recognition" to include both.

The qualitative improvement in the performance of recognition models can be attributed to three main components. The first is the use of extensive learning in constructing recognition models. In this framework, rather than specifying a particular model, the scheme starts with a large family of possible models and uses observed examples to guide the construction of a specific model which is best suited to the observed data. The second component was the development of new forms of object representation for the purpose of categorization, based on both computational considerations and guidelines from known properties of the visual cortex. These two components, representation and learning, are interrelated: initially, the class representation provides a family of plausible models, and effective learning methods are then used to construct a particular model for a novel class such as "dog" or "airplane" based on observed examples. The third component was the use of new statistical learning techniques, such as regularization classifiers (SVM and others) and Bayesian inference (such as graphical models). We next discuss each of these advances in more detail.

Learning instead of design. A conceptual advance that facilitated recent progress in object recognition was the idea of learning the solution to a specific

K. Ikeuchi (ed.), *Computer Vision*, DOI 10.1007/978-0-387-31439-6,

classification problem from examples, rather than focusing on the classifier design. This was a marked departure from the dominant practices at the time: instead of an expert program with a predetermined set of logical rules, the appropriate model was learned and selected from a possibly infinite set of models, based on a set of examples. The techniques used in the 1990s originated in the area of supervised learning, where image examples are provided together with the appropriate class labels (e.g., "face" or "non-face"). A comprehensive theory of the foundations of supervised learning has been developed, with roots in functional analysis and probability theory [6, 26, 27, 36]. The formal analysis of learning continues to evolve and to contribute to our understanding of the role of learning in visual recognition.

New image representations. A recognition scheme typically extracts during learning a set of measurements or "features" and uses them to construct new object representations. Objects are then classified and recognized based on their feature representation. Feature selection and object representation are crucial, because they facilitate the identification of elements that are shared by objects in the same class and support discrimination between similar objects and categories. Different types of visual features have been used in computational models in the past, ranging from simple local-image patterns such as wavelets, edges, blobs, or local-edge combinations to abstract three-dimensional shape primitives, such as cylinders [21], spheres, cubes, and the like [4].

A common aspect of most past recognition schemes is that they use a fixed small generic set of feature types to represent all objects and classes. In contrast, recent recognition schemes use pictorial features extracted from examples, such as object fragments or patches, together with their spatial arrangement [1, 3, 19, 30]. Unlike generic parts, these schemes use a large set of features, extracted from different classes of objects. The use of large feature sets is also connected to an interesting new trend in signal processing, related to "over-complete" representations. Instead of representing a signal in terms of a traditional complete representation, such as Fourier components, one uses a redundant basis (such as the combination of several complete bases).

Representations using such features have been used successfully in recent computer vision recognition systems for two reasons. First, these representations can be learned and used efficiently; second, they proved to capture effectively the broad range of variability in appearance within a visual class.

An additional comment is appropriate. The representations described above are view based, as opposed to object-centered models. A representation based on image appearance can include not only 2D image properties but also 3D aspects such as local depth variations or 3D curvature.

New statistical learning methods. Over the last few years, the mathematics of learning has become the "lingua franca" of large areas of computer science and, in particular, of computer vision. As we discussed, the use of a learning framework enabled a qualitative jump in object recognition. Whereas the initial techniques used to construct useful classification models from data were quite simple, there are now more efficient algorithms originally introduced in the area of learning in the 1990s such as regularization algorithms (also called kernel machines), which include SVM [35, 36] and boosting [12]. By now, the area of learning has grown to include, in addition to discriminative algorithms, probabilistic approaches with the goal of providing full probability distributions as solutions to object recognition tasks. These techniques are mostly Bayesian and range from graphical models [13, 15] to hierarchical Bayesian models [16, 17]. At the same time, the focus of research is shifting from supervised to unsupervised and semisupervised learning problems, using techniques such as manifold learning [2]. Semisupervised problems, in which the training set consists of a large number of unlabeled examples and a small number of labeled ones, are gaining attention.

Application

A number of early schemes, mainly focusing on the class of human faces, obtained significant improvement over previous methods [5, 29, 31, 32, 38]. The techniques have evolved to reach practical applications, as evidenced by their use in current digital cameras. The more recent versions of these computational schemes have started to deal successfully with an increasing range of complex object

categories such as pedestrians, cars, motorcycles, airplanes, horses, and the like, in unconstrained natural scenes, to deal with a broad range of objects within each class (e.g., [1, 8, 19, 22–24, 30, 34, 39]). The algorithms that were refined over the last few years can deal successfully with a large number of different object classes, in complex and highly cluttered scenes. They are being applied to databases of hundreds [9] and even thousands of object classes [7]. Yearly competitions in computer-based recognition, such as the Pascal challenge [25, 28], witness continuous improvement in the range of classes and in scene complexity successfully handled by automatic object categorization algorithms [10, 11, 37].

References

1. Agarwal S, Roth D (2002) Learning a sparse representation for object recognition. In: Proceedings of the 7th ECCV, Copenhagen, pp 113–130
2. Belkin M, Niyogi P (2004) Semi-supervised learning on Riemannian manifolds. Mach Learn J 56:209–239
3. Belongie S, Malik J, Puzicha J (2002) Shape matching and object recognition using shape contexts. IEEE PAMI 24(4):509–522
4. Biederman I (1985) Human image understanding: recent research and a theory. Comput Vis Graph Image Process 32:29–73
5. Brunelli R, Poggio T (1993) Face recognition: features versus templates. IEEE Trans PAMI 15(10): 1042–1052
6. Cucker F, Smale S (2002) On the mathematical foundations of learning. Bull Am Math Soc 39:1–50
7. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) ImageNet: a large-scale hierarchical image database. In: IEEE computer vision and pattern recognition (CVPR), Miami
8. Fei-Fei L, Fergus R, Perona P (2003) A Bayesian approach to unsupervised one-shot learning of object categories. In: Proceedings of the ICCV, Wisconsin
9. Fei-Fei L, Fergus R, Perona P (2004) Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In: IEEE conference on computer vision pattern recognition (CVPR 2004), workshop on generative-model based vision, Washington, DC
10. Felzenszwalb D, McAllester D, Ramanan A (2008) Discriminatively trained, multiscale, deformable part model. In: Proceedings of the IEEE conference on computer vision pattern recognition (CVPR), Anchorage
11. Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D (2009) Object detection with discriminatively trained part based models. IEEE Trans Pattern Anal Mach Intell 32:1627–1645
12. Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 55(1):119–139
13. Geman S (2005) On the formulation of a composition machine. Technical report, Division of Applied Mathematics, Brown University
14. Grimson WEL (1990) Object recognition by computer. MIT, Cambridge
15. Jordan I (2004) Graphical models. Stat Sci (Special issue on Bayesian Stat) 19:140–155
16. Kemp C, Tenenbaum JB (2008) The discovery of structural form. Proc Natl Acad Sci 105(31):10687–10692
17. Lee TS, Mumford D (2003) Hierarchical Bayesian inference in the visual cortex. J Opt Soc Am A Opt Image Sci Vis 20(7):1434–1448.
18. Lowe DG (1987) Three-dimensional object recognition from single two-dimensional images. J Artif Intell 31: 355–395
19. Lowe D (2004) Distinctive image features from scale-invariant key-points. Int J Comput Vis 60(2): 91–110
20. Marr D (1982) Vision: a computational investigation into the human representation and visual information. W.H. Freeman, New York
21. Marr D, Nishihara HK (1978) Representation and recognition of the spatial organisation of three-dimensional shapes. Proc R Soc Lond B 200:269–294
22. Mohan A, Papageorgiou C, Poggio T (2001) Example-based object detection in images by components. IEEE Trans Pattern Anal Mach Intell 23:349–361
23. Papageorgiou C, Evgeniou T, Mukherjee S, Poggio T (1998) A trainable pedestrian detection system. In: IEEE international conference on intelligent vehicles, Stuttgart, vol 1, pp 241–246
24. Papageorgiou C, Oren M, Poggio T (1998) A general framework for object detection. In: Proceedings of the international conference on computer vision, Bombay, 4–7 Jan 1998
25. Pascal website. http://pascallin.ecs.soton.ac.uk/challenges/voc/
26. Poggio T, Smale S (2003) The mathematics of learning: dealing with data. Notices AMS 50:537–544
27. Poggio T, Rifkin R, Mukherjee S, Niyogi P (2004) General conditions for predictivity in learning theory. Nature 428:419–422
28. Ponce J, Berg TL, Everingham M, Forsyth DA, Hebert M, Lazebnik S, Marszalek M, Schmid C, Russell BC, Torralba A, Williams CKI, Zhang J, Zisserman A (2007) In: Ponce J, Hebert M, Schmid C, Zisserman A (eds) Toward category-level object recognition. Lecture notes in computer science. Springer, Berlin
29. Rowley H, Baluja S, Kanade T (1998) Neural network-based face detection. IEEE Trans Pattern Anal Mach Intell (PAMI) 20(1):23–38
30. Sali E, Ullman S (1999) Combining class-specific fragments for object classification. In: Proceedings of the 10th British machine vision conference, Nottingham, vol 1, pp 203–213
31. Sung K, Poggio T (1998) Example-based learning for view-based human face detection. IEEE Trans Pattern Anal Mach Intell 20(1):39–51

32. Turk M, Pentland A (1991) Eigen faces for recognition. J Cognit Neurosci 3(1):71–86
33. Ullman S, Basri R (1991) Recognition by linear combination of models. IEEE PAMI 13(10):992–1006
34. Ullman S, Vidal-Naquet M, Sali E (2002) Visual features of intermediate complexity and their use in classification. Nature Neurosci 5(7):1–6
35. Vapnik N (1995) The nature of statistical learning theory. Springer, New York
36. Vapnik N (1998) Statistical learning theory. Wiley, New York
37. Vedaldi A, Gulshan V, Varma M, Zisserman A (2009) Multiple Kernels for object detection. In: Proceedings of the international conference on computer vision, Kyoto
38. Viola P, Jones M (2001) Robust real-time object detection. Int J Comput Vis 56:151–177
39. Zhang J, Zisserman A (2006) Dataset issues in object recognition. In: Ponce J, Hebert M, Schmid C, Zisserman A (eds) Toward category-level object recognition. Springer, Berlin, pp 29–48

Many-to-Many Graph Matching

Fatih Demirci[1], Ali Shokoufandeh[2] and Sven J. Dickinson[3]
[1]Department of Computer Engineering, TOBB University of Economics and Technology, Sogutozu, Ankara, Turkey
[2]Department of Computer Science, Drexel University, Philadelphia, PA, USA
[3]Department of Computer Science, University of Toronto, Toronto, ON, Canada

Synonyms

Error-correcting graph matching; Error-tolerant graph matching; Inexact matching; Transportation problem

Definition

When objects exhibit large within-class variation and/or when image features are under- or over-segmented, the image features extracted from two exemplars belonging to the same category may no longer be in one-to-one correspondence but, in general, many-to-many correspondence. If the features are structured, i.e., captured in a graph, then computing the correct correspondence can be formulated as a many-to-many graph matching problem.

Background

The matching of image features to object models is typically formulated as a one-to-one assignment problem, based on the assumption that for every salient image feature belonging to the object to be matched, e.g., SIFT feature, image patch, contour fragment, there exists a single corresponding feature on the model (and vice versa). While the one-to-one correspondence assumption has been prevalent in the object recognition community throughout its entire evolution, including the paradigms of graph matching [9], alignment [13], geometric invariants [11], local appearance [14], and a recent return to local contour-based features [8], one-to-one feature correspondence is a highly restrictive assumption that breaks down as within-class variation increases and as the segmentation and extraction of more abstract image features suffer from over- or under-segmentation [7]. In the more general case, feature correspondence is not one-to-one, but rather *many-to-many*. If a feature set is described by a graph, with nodes representing features and edges capturing pairwise relations between features, computing the correct many-to-many feature correspondence can be formulated as *many-to-many graph matching*.

Consider two simple examples, shown in Fig. 1. In Fig. 1a, a set of multiscale blobs and ridges have been extracted from two exemplars (humans) belonging to the same category. In the top image, the straight arm yields a single elongated ridge, while in the bottom image, the bent arm yields two smaller and coterminating elongated ridges. In this case, simple object articulation (a form of within-class variation) has led to a violation of the one-to-one correspondence assumption. Instead, the correspondence is clearly two-to-one; enforcing one-to-one correspondence will lead to an incorrect matching of the entire arm to either the upper or lower arm, e.g., the red highlighted features. In Fig. 1b, two region segmentations of two exemplars belonging to the same class yield a set of region correspondences that are rarely one-to-one, but more typically many-to-many. Once again, enforcing a one-to-one feature correspondence will ensure an incorrect matching, and will miss the correct correspondence.

The problem of computing a one-to-one correspondence between a model feature graph and a cluttered image graph can be formulated as a largest isomorphic subgraph problem, whose complexity is NP-hard.

Many-to-Many Graph Matching, Fig. 1 Two graph matching problems in computer vision for which assuming a one-to-one feature correspondence will lead to incorrect correspondences, and which can only be solved if formulated as a many-to-many graph-matching problem. In (**a**), a multiscale blob and ridges decomposition [17] of the two humans yields a single ridge for the extended arm (*top*) and two coterminating ridges for the bent arm (*bottom*). In this example, articulation has violated the one-to-one feature correspondence assumption; if a one-to-one correspondence is enforced for the arm, e.g., the red highlighted features, it will be incorrect. In this case, the correspondence should be two-to-one (or more generally, many-to-many). In (**b**), two different cup exemplars (*bottom row*) have been region segmented (*top row*), yielding regions that are rarely in one-to-one correspondence (due to within-class variation or region over- and/or under-segmentation). Once again, the correct correspondence is not one-to-one, but rather many-to-many

The complexity of the many-to-many matching problem is even more prohibitive, for the space of possible correspondences is greater (any subset of features in the image may match any subset of features on the model). The intractable complexity of the many-to-many matching problem can only be reduced by exploiting the types of regularities suggested by the perceptual grouping community, such as proximity, continuity, conservation of mass, etc. In what follows, a formal statement of the problem is introduced, and a number of approaches to its solution is reviewed.

Theory

The main objective of the many-to-many graph matching problem is to establish a minimum cost mapping between the vertices of two attributed, edge-weighted graphs. In an attribute-weighted graph $G = (V, E)$, let $\mathbb{L}(v)$ denote the set of attributes associated with $v \in V$. Given a subset $U \subset V$, let $\mathbb{L}(U) = \cup_{u \in U} \mathbb{L}(u)$. For a set $U \subset V$, let $G|_U$ denote the subgraph of G induced on the vertices in U, and let $w(u, v)$ denote the weight of an edge $(u, v) \in E$. Finally, let $\mathbb{P}(G)$ denote the power-set 2^V for the vertex set of G. A *many-to-many mapping* between two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is a mapping among power-sets $\mathbb{P}(G_1)$ and $\mathbb{P}(G_2)$ and can be characterized as a function:

$$\mathcal{M} : (\mathbb{P}(G_1) \times \mathbb{P}(G_2)) \rightarrow \{0, 1\}. \quad (1)$$

For two sets, $U \in \mathbb{P}(G_1)$ and $V \in \mathbb{P}(G_2)$, there will be a cost $C(\mathbb{L}(U), \mathbb{L}(V))$ associated with mapping the labels in set $\mathbb{L}(U)$ to those in $\mathbb{L}(V)$. An example of a common cost function is the edit-distance between the labels in sets $\mathbb{L}(U)$ and $\mathbb{L}(v)$. Let $\mathcal{S}(G_1|_U, G_2|_V)$ denote

the structural distance between induced subgraphs $G_1|_U$ and $G_2|_V$. Observe that every mapping $\mathcal{M}$ has a natural representation as a matrix, with $\mathcal{M}_{U,V} = 1$ if the sets $U \in \mathbb{P}(G_1)$ and $V \in \mathbb{P}(G_2)$ are mapped to each other under $\mathcal{M}$, and $\mathcal{M}_{U,V} = 0$ otherwise. Combining these two cost functions will result in the cost function $C(\mathcal{M})$ associated with the mapping $\mathcal{M}$:

$$C(M) = \sum_{U \in \mathbb{P}(G_1), V \in \mathbb{P}(G_2)} \mathcal{M}_{U,V} \times C(\mathbb{L}(U), \mathbb{L}(V)) \times \mathcal{S}(G_1|_U, G_2|_V). \quad (2)$$

In defining an optimal many-to-many matching between two attributed graphs, G_1 and G_2, a many-to-many mapping $\mathcal{M}^*$ of minimum cost $C(\mathcal{M}^*)$ subject to specific requirements on the structure or cardinality of $\mathcal{M}^*$ will be obtained. For example, to prevent a trivial solution that sets $\mathcal{M}_{U,V} = 0$, for all U and V, one can require a matching such that its cardinality, i.e., $\sum_{U,V} \mathcal{M}_{U,V}$, exceeds a threshold while minimizing $C(\mathcal{M})$. Other functions, such as maximizing the number of vertices from V_1 and V_2 that participate in $\mathcal{M}$, can be used to evaluate the quality of the mapping. Note that cost functions $C(\mathbb{L}(U), \mathbb{L}(V))$ and $\mathcal{S}(G_1|_U, G_2|_V)$ may be used to enforce constraints such as consistency of mapped labels, limits of feasible label mappings, or allowed structural mapping of induced graphs $G_1|_U$ and $G_2|_V$ by imposing arbitrary large values or by being ill-defined.

The above description of the many-to-many matching results in an intractable computational problem. First, due to the exponential size of power-sets $\mathbb{P}(V_1)$ and $\mathbb{P}(V_2)$ in terms of number of vertices in G_1 and G_2, the size of the search space for the many-to-many matching problem is exponential. Even simplifying the problem to one-to-one mappings, by replacing the power-sets $\mathbb{P}(V_1)$ and $\mathbb{P}(V_2)$ with sets V_1 and V_2, respectively, will result in the multidimensional matching problem that is known to be NP-complete for arbitrary labeled graphs.

Related Work

Many-to-many graph matching has been studied extensively in a variety of contexts, including graph edit distance [2, 16], spectral methods [4, 18], optimization problems [20], metric embedding [6], abstract models [10], and grammars [1, 21]. The classical formulation of graph edit distance introduces a set of graph edit operations, such as insertion, deletion, merging, splitting, and substitution of nodes and edges. Given a set of graph edit operations and a cost function, the objective is to find the lowest cost sequence of graph edit operations that transform one graph into the other. The edit distance between two graphs critically depends on the costs of the underlying edit operations; typically, lower costs are assigned to the most frequent edit operations. A number of approaches have addressed the problem of defining an appropriate cost, e.g., [3].

Many-to-many graph matching has also been studied in the context of spectral methods by examining the spectral properties of graph adjacency matrices. In [4], the authors present an approach based on renormalization projections of vertices into a common eigensubspace of two graphs. Instead of finding the overall similarity of two graphs from the positions of vertex projections, this approach uses an agglomerative hierarchical clustering technique to produce many-to-many vertex correspondences.

Another spectral method is due to [18, 19], which constructs a low-dimensional "signature" of a directed graph's "shape" from the magnitudes of the eigenvalues of the graph's adjacency matrix. The eigenvalues are invariant to the reordering of a graph's branches and are shown to be robust under minor structural perturbation of the graph. This vector can be used for both structural indexing and for matching in the presence of noise and occlusion. If two signatures (vectors) are close, their corresponding (sub)graphs, possibly having different cardinalities, are in many-to-many correspondence.

Recently, the approach presented in [20] formulates the many-to-many graph matching problem as a discrete optimization problem. The algorithm starts by extending the optimization problem for one-to-one matching to the case of many-to-one matching. The algorithm then obtains many-to-many vertex correspondences through two many-to-one mappings. Since this formulation of the many-to-many matching requires the solution of a hard optimization problem, the authors propose an approximate algorithm based on a continuous relaxation of the combinatorial problem.

The concept of a low-distortion graph embedding has been used to obtain many-to-many vertex correspondences [6]. Specifically, low-distortion graph

embedding is employed to transform the problem of many-to-many graph matching to a many-to-many point matching problem in a geometric space. This transformation maps nodes to points and edge weights to interpoint distances, not only simplifying the original graph representation (by removing the edges), but also retaining important local and global graph structure; moreover, the transformation is robust under perturbation. Representing two graphs as sets of points reduces the many-to-many graph matching problem to that of many-to-many point matching in the geometric space, for which a number of efficient distribution-based similarity measures are available. The authors use the Earth Mover's Distance [15] algorithm to find such correspondences and show that the resulting many-to-many point matching realizes the desired many-to-many matching between the vertices of the input graphs.

A number of researchers, e.g., [10, 12] and [5], have explored many-to-many graph matching in the context of model-based abstraction from images. The work presented in [10] starts by forming a region adjacency graph from each image. The approach then searches the space of pairwise region groupings in each graph, forming a lattice. Each input image yields a lattice such that its bottom node represents the original region adjacency graph and its top node represents the silhouette of the object. The framework defines a common abstraction as a set of nodes, one per lattice, such that for a pair of nodes, their corresponding graphs are isomorphic. The lowest common abstraction (LCA) is defined as the common abstraction whose underlying graph has the maximum number of nodes. Thus, the resulting LCA carries the most informative abstraction common to each image. Although effective, this technique does not find a match between two graphs whose common abstraction does not exist.

The two algorithms presented in [12] and [5] use the many-to-many graph matching technique of [6] for automatically constructing an abstract model from examples. The work in [12] computes the multi-scale ridge/blob decomposition (AND-OR) graph for each input image and obtains the many-to-many node correspondences between each pair of graphs, yielding a matching matrix. By exploring this matrix, the algorithm first finds features that match one-to-one across many pairs of input images. The many-to-many matchings between these features are then analyzed to obtain the decompositional relations among them. The extracted features and their relations are used to construct the final abstract model.

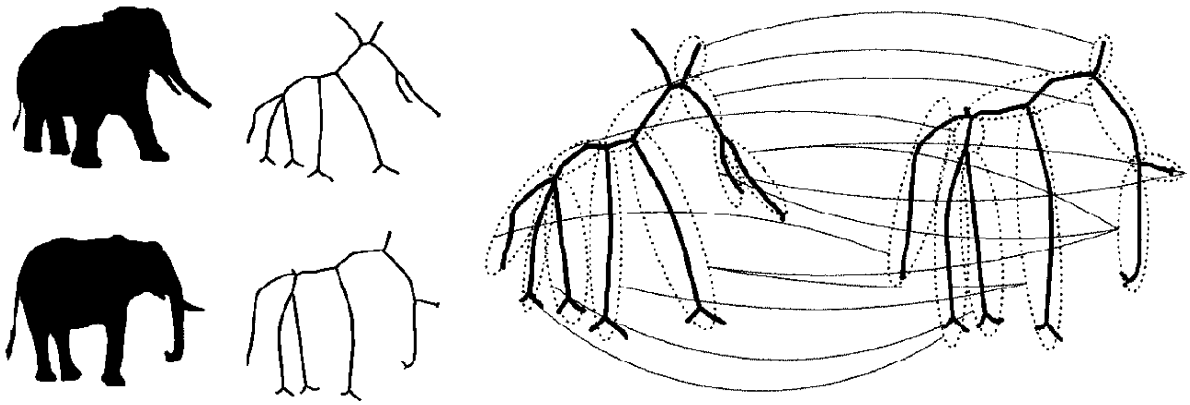
After obtaining many-to-many node correspondences based on [6], the algorithm in [5] computes the abstracted medial axis graph by first computing the averages of the corresponding pairs of subgraphs to yield the nodes in the abstracted graph, and then defining the overall topology of the resulting abstract parts to yield the relations. Each matching pair of subgraphs corresponds to a single node in the abstracted graph, and two abstracted nodes are connected by an edge if the corresponding subgraphs are adjacent in the original graphs. This procedure forms the basis of an iterative framework in which pairs of similar medial axis graphs are clustered and abstracted, yielding a set of abstract medial axis graph class prototypes.

In the domain of grammars, objects are represented as variable hierarchical structures. Each part in this representation can be defined in terms of other parts, allowing an object to be modeled by its coarse-to-fine appearance. Overall, grammar-based models including AND-OR graphs support structural variability. To represent intra-category variation and to account for many-to-many correspondence, the grammar creates a large number of configurations from a small vocabulary set. To scale to a large number of object categories, the AND-OR graph, learning, and inference algorithms are defined recursively. Some examples of this type of approach include [1, 21].

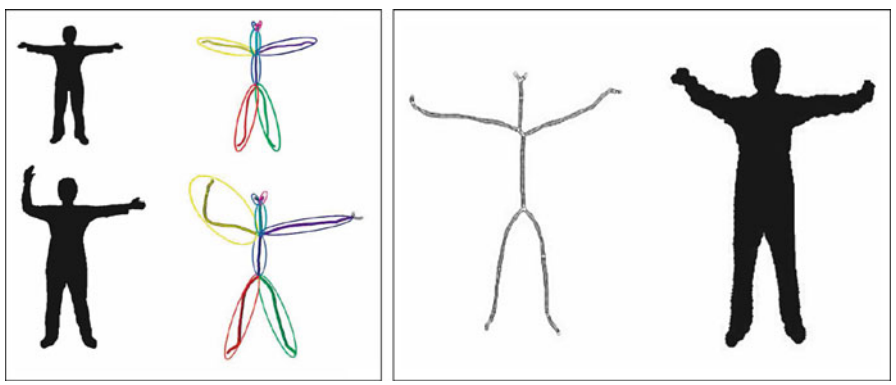
Experimental Results

In this section, some example results from some of the many-to-many matching approaches described in the Related Work section are illustrated. After representing silhouettes as skeleton graphs in Fig. 2, the algorithm proposed in [6] obtains many-to-many node correspondences through metric embedding, as discussed earlier. Based on the many-to-many correspondences of this algorithm, Fig. 3 demonstrates an example for the abstract shape created by the approach presented in [5]. The left part presents input silhouettes, their skeleton graphs, and many-to-many correspondences. The right part presents the abstract skeleton graph and its shape reconstructed from this graph.

Graph edit distance is another important class of many-to-many graph matching algorithms. Figure 4



Many-to-Many Graph Matching, Fig. 2 Example many-to-many correspondences computed by [6]. After representing two silhouettes as skeleton graphs, the graphs are embedded into geometric spaces of the same dimensionality. The embedded points are then matched using the Earth Mover's Distance algorithm. The *right part* illustrates the many-to-many correspondences between the vertices of the input graphs. Each *dashed ellipsoid* represents a set of vertices from the original graph



Many-to-Many Graph Matching, Fig. 3 A shape abstraction example of [5] based on many-to-many correspondences obtained by [6]. The *left image* shows input silhouettes and their skeleton graphs in which the same color is used to show the corresponding parts. Using these correspondences, the abstract skeleton graph and its silhouette are created as shown on the *right*

Many-to-Many Graph Matching, Fig. 4 Graph edit distance algorithms compute many-to-many correspondences of a pair of graphs by finding the lowest cost sequence of graph edit operations needed to transform one graph into another. In the example, same colors indicate the matching skeleton parts, while gray colors show spliced or contracted edges (The example is taken from Ref. [16])

shows the result of matching the skeleton graphs for two input shapes using the graph edit distance algorithm described in [16]. Same colors indicate the matching skeleton parts while gray colors show spliced or contracted edges. Observe that the many-to-many correspondences are intuitive in these figures.

References

1. Bunke H (1982) Attributed graph grammars and their application to schematic diagram interpretation. IEEE Trans Pattern Anal Mach Intell 4:574–582
2. Bunke H (1997) On a relation between graph edit distance and maximum common subgraph. Pattern Recognit Lett 18(8):689–694
3. Bunke H, Shearer K (1998) A graph distance metric based on the maximal common subgraph. Pattern Recognit Lett 19:255–259
4. Caelli T, Kosinov S (2004) An eigenspace projection clustering method for inexact graph matching. IEEE Trans Pattern Anal Mach Intell 26:515–519
5. Demirci F, Shokoufandeh A, Dickinson S (2009) Skeletal shape abstraction from examples. IEEE Trans Pattern Anal Mach Intell 31:944–952
6. Demirci F, Shokoufandeh A, Keselman Y, Bretzner L, Dickinson S (2006) Object recognition as many-to-many feature matching. Int J Comput Vis 69(2):203–222
7. Dickinson S (2009) The evolution of object categorization and the challenge of image abstraction. In: Dickinson S, Leonardis A, Schiele B, Tarr M (eds) Object categorization: computer and human vision perspectives. Cambridge University Press, New York, pp 1–37
8. Ferrari V, Jurie F, Schmid C (2010) From images to shape models for object detection. Int J Comput Vis 87(3): 284–303
9. Fischler MA, Eschlager RA (1973) The representation and matching of pictorial structures. IEEE Trans Comput 22(1):67–92
10. Keselman Y, Dickinson S (2005) Generic model abstraction from examples. IEEE Trans Pattern Anal Mach Intell 27(7):1141–1156
11. Lamdan Y, Schwartz J, Wolfson H (1990) Affine invariant model-based object recognition. IEEE Trans Rob Autom 6(5):578–589
12. Levinshtein A, Sminchisescu C, Dickinson S (2005) Learning hierarchical shape models from examples.In: Proceedings of the EMMCVPR, St. Augustine. Springer, Berlin, pp 251–267
13. Lowe D (1985) Perceptual organization and visual recognition. Academic, Norwell
14. Lowe D (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
15. Rubner Y, Tomasi C, Guibas LJ (2000) The earth mover's distance as a metric for image retrieval. Int J Comput Vis 40(2):99–121
16. Sebastian T, Klein P, Kimia B (2004) Recognition of shapes by editing their shock graphs. IEEE Trans Pattern Anal Mach Intell 26:550–571
17. Shokoufandeh A, Bretzner L, Macrini D, Demirci MF, Jönsson C, Dickinson S (2006) The representation and matching of categorical shape. Comput Vis Image Underst 103(2):139–154
18. Shokoufandeh A, Macrini D, Dickinson S, Siddiqi K, Zucker SW (2005) Indexing hierarchical structures using graph spectra. IEEE Trans Pattern Anal Mach Intell 27(7):1125–1140
19. Siddiqi K, Shokoufandeh A, Dickinson S, Zucker S (1999) Shock graphs and shape matching. Int J Comput Vis 30: 1–24
20. Zaslavskiy M, Bach F, Vert J (2010) Many-to-many graph matching: a continuous relaxation approach. Lecture Notes in Computer Science, http://arxiv.org/abs/1004.4965, DBLP, http://dblp.uni-trier.de 6323:515–530
21. Zhu S, Mumford D (2006) A stochastic grammar of images. Found Trends Comput Graph Vis 2:259–362

Matte Extraction

Jiaya Jia
Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong, China

Synonyms

Digital matting; Pulling a matte

Definition

An alpha matte has the same size as the input image. It contains respective weights to linearly blend latent foreground and background colors for each pixel to form the observed color. Estimating the alpha matte together with the foreground color image is generally referred to as matte extraction or digital matting.

Background

Classifying each pixel in an input image to either foreground or background is called *binary segmentation*, which is a fundamental computer vision problem. Digital matting relaxes the hard separation assumption

and takes ubiquitous foreground and background color blending in image formation, which happens along almost all object boundaries, into consideration. Results from matte extraction can be used to generate a new composite.

Color blending in natural images has a variety of causes, such as color interpolation during image production and light photons received by the camera sensor containing both background and foreground color for some pixels. Without additional information, digital matting is an ill-posed problem with many unknowns. So generally, either multiple frames are taken or a certain amount of user interaction is involved to sample foreground and background color in image and video matting.

Theory

In the digital matting framework, separating the background image B and foreground image F with respect to an alpha matte α from an input natural image I is expressed as

$$I = \alpha F + (1 - \alpha)B. \quad (1)$$

If $\alpha(x, y) = 1$, the pixel with coordinate (x, y) is definitely in the foreground. $\alpha(x, y)$ being 0 defines an absolutely background pixel. $\alpha(x, y)$ can also be in between 0 and 1, indicating a certain level of color mixing. Digital matting aims to estimate α and F (sometimes also B) from I. Existing methods follow one of the following lines.

Blue Screen Matting

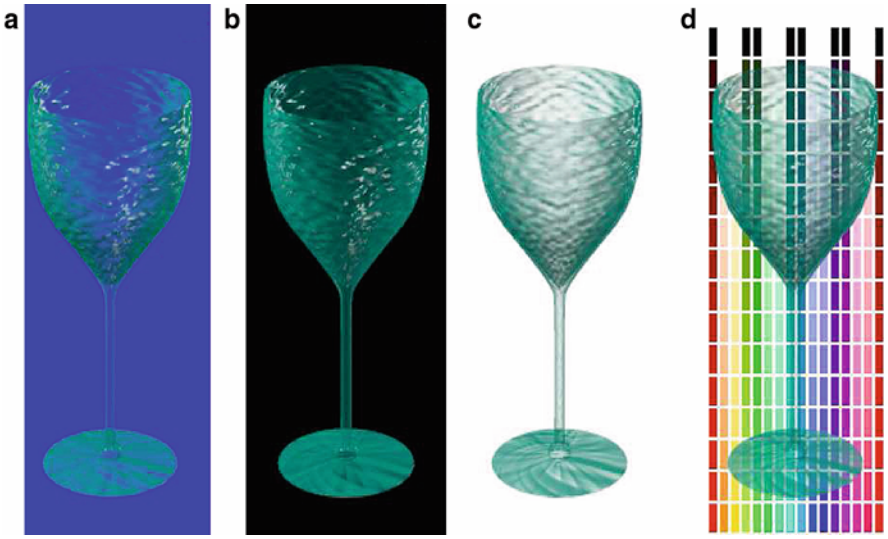
Blue screen matting [1], which is widely employed in movie and commercial production, needs to set up a controlled environment and uses a single or multiple constant-color backgrounds (as shown in Fig. 1). The blue screen matting problem is directly solvable. Its triangular matting technique, which captures images with two backgrounds containing different shades of the backing color, is particularly noteworthy because a closed-form solution exists. This technique can produce very accurate matting results usable as ground truth data. Blue screen matting can be applied in a frame-by-frame fashion to video foreground object extraction.

Natural Image Matting

In natural image matting, background B is generally unknown and possibly contains complex structures. In this case, simultaneously estimating α, F, and B becomes an ill-posed problem. Several methods [3–5] need user input of additional segmentation information to constrain it. *Trimap* is a popular format that partitions the image into three regions, i.e., "definitely foreground" (DF for short), "definitely background" (DB for short), and "unknown region", as shown in Fig. 2b. In DF and DB, α is set to 1 and 0, respectively. Digital matting only estimates α, together with the foreground and background color, in the unknown region by gathering color information from DF and DB.

There have been several methods proposed to sample color from DF and DB. In knockout [6], F is computed as the weighted average of foreground color along the perimeter of the DF region. B is computed similarly but with a final refinement step. Ruzon and Tomasi [3] sample F and B from local windows and then parameterize them as a mixture of unoriented Gaussians. Alpha values are computed by maximizing a function that interpolates the mean and variance of the Gaussians. Bayesian matting [4] gathers color samples from DF and DB using sliding windows and fits them with oriented Gaussian distributions. A *maximum a posteriori* (MAP) estimation of α, F, and B is applied. The final α values are chosen from the foreground and background color pairs that maximize the probability. Global Poisson matting [5] contributes a gradient-domain alpha matte estimation. When the condition of locally smooth color change in DF and DB is violated, user interaction is involved to improve the matting result with the supply of a group of filters.

The quality of results of these methods partly depends on how accurate the trimap is since color is sampled and the alpha matte is estimated within windows. Many later approaches instead require the user to only draw several foreground and background scribbles to coarsely indicate DF and DB and leave all unspecified pixels in the unknown region. This scheme simplifies user interaction but provides looser constraints for digital matting, as shown in Fig. 3. Representative work that can robustly solve for alpha mattes based on it includes (1) the iterative-optimization method [2], which samples color from user-drawn scribbles, builds the Markov Random Field (MRF), and solves for segmentation and matte



Matte Extraction, Fig. 1 Blue screen matting [1]. (**a**) Object against known constant *blue*. (**b**) Object against constant *black*. (**c**) Pulled foreground. (**d**) New composite

Matte Extraction, Fig. 2 A trimap matting example. (**a**) Input image. (**b**) The user-provided trimap where definitely foreground and background are in *white* and *black*, respectively. The *gray* *pixels* are unknown ones. (**c**) Alpha matte estimate by global Poisson matting

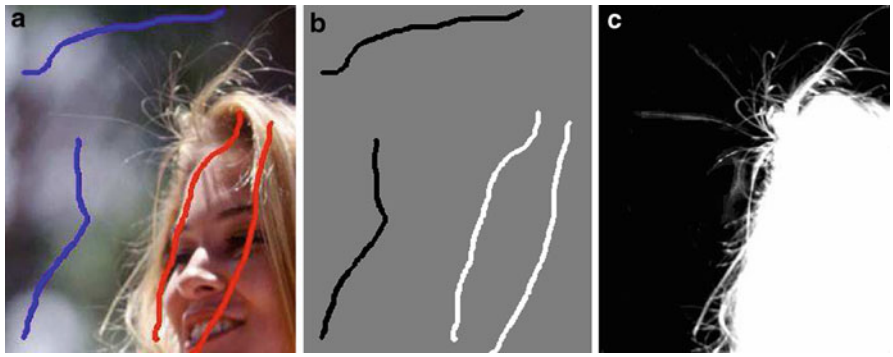
extraction using belief propagation, and (2) closed-form matting [7] that introduces a color line model and based on it derives a quadratic cost function only involving α and a matting Laplacian, enabling linear optimization. In addition, Rhemann et al. [8] extract high-resolution mattes by trimap segmentation and by employing gradient preserving alpha priors. The soft-scissor method of Wang et al. [9] can achieve real-time matting along with user painting the foreground boundary.

There are also automatic image matting methods. A soft color segmentation method was proposed in [26], where a global objective function is modeled by global and local parameters. These parameters are alternately optimized until convergence. It can be applied to matting without intensive user interaction. Spectral matting [10] is a single image approach. It shows that the smallest eigenvectors of the matting Laplacian span individual matting components, making their estimation equivalent to finding linear transformation of the eigenvectors. Flash matting [11] captures a pair of flash/no-flash images and assumes only the foreground region is lit by flash. This method can automatically extract foreground in a joint Bayesian matting framework.

Albeit some inevitable limitations as described in respective papers, all the above techniques advance image matting from different aspects. Other recommended readings also include [12–14].

Video Matting

Digital matting was extended to videos in various ways. Typical video matting methods deal with foreground regions with hair, trees, or smoke, where color blending exists for a large amount of pixels. To preserve temporal coherence among frames, with



Matte Extraction, Fig. 3 Matting with scribbles. (**a**) Input image with user-drawn scribbles. (**b**) The initial trimap. (**c**) Alpha matte result of Wang and Cohen [2]

the input of a monocular video, Bayesian video matting [15] propagates manually specified trimaps from keyframes to other frames using optical flow and suggests completing background using mosaic construction. This method is improved in [16] by incorporating stronger prior terms. The geodesic matting method [17] introduces temporal neighbors for each pixel and infers foreground and background scribbles in the video based on user input in only sparse frames. With the setup of special devices or systems, defocus matting [18] and the camera-array method [19] make use of multiple cameras that take pictures with different focus settings and with the existence of parallax respectively to profit video matting.

The other set of methods [20, 21] adopt video matting in a refinement step to improve foreground boundary estimation after video cutout where unknown regions are generally narrowbands around the boundaries.

Application

Digital matting exploits pixel-wise color blending and is an indispensable technique for high-quality object extraction from images and videos. The matte estimate together with the computed foreground color can then be used to form a new composite with another background image. Simple composition applies linear color blending again based on (Eq. 1), while several other approaches, include drag-and-drop pasting [22], context-sensitive blending [23], and compositional matting [24], explore the structure relationship between the source and target images and combine color blending with other schemes.

Matte extraction and the corresponding composite construction are fundamental tools for image/video editing and finds many applications in computer graphics and vision. Movie and commercial production relies on it to naturally insert objects into a virtual or real scene. Digital matting can possibly be combined with other decomposition, recognition, and tracking techniques to further improve the performance and expand the usability.

Experimental Results

Rhemann et al. [25] established a digital matting evaluation website containing data classified into high, strong, medium, and little transparency groups. Training data are also provided. This website contains updated experimental results of many approaches.

References

1. Smith AR, Blinn JF (1996) Blue screen matting. In: Proceedings of the ACM SIGGRAPH, New Orleans, 259–268
2. Wang J, Cohen MF (2005) An iterative optimization approach for unified image segmentation and matting. In: Proceedings of the ICCV, Beijing, 936–943
3. Ruzon MA, Tomasi C (2000) Alpha estimation in natural images. In: Proceedings of the IEEE conference on computer vision pattern recognition (CVPR), San Diego, 18–25
4. Chuang YY, Curless B, Salesin DH, Szeliski R (2001) A bayesian approach to digital matting. In: Proceedings of the IEEE conference on computer vision pattern recognition (CVPR), Kauai, vol II, 264–271
5. Sun J, Jia J, Tang CK, Shum HY (2004) Poisson matting. ACM Trans Graph 23(3):315–321

6. Berman A, Vlahos P, Dadourian A (2000) Comprehensive method for removing from an image the background surrounding a selected object. US Patent 6,134,345
7. Levin A, Lischinski D, Weiss Y (2006) A closed form solution to natural image matting. In: Proceedings of the IEEE conference on computer vision pattern recognition (CVPR), New York, 61–68
8. Rhemann C, Rother C, Rav-Acha A, Sharp T (2008) High resolution matting via interactive trimap segmentation. In: Proceedings of the IEEE conference on computer vision pattern recognition (CVPR), Anchorage
9. Wang J, Agrawala M, Cohen MF (2007) Soft scissors: an interactive tool for realtime high quality matting. In: Proceedings of the ACM SIGGRAPH, San Diego
10. Levin A, Rav-Acha A, Lischinski D (2008) Spectral matting. IEEE Trans Pattern Anal Mach Intell 30: 1699–1712
11. Sun J, Li Y, Kang SB, Shum HY (2006) Flash matting. In: Proceedings of the SIGGRAPH, Boston, 772–778
12. Grady L, Schiwietz T, Aharon S, Westermann R (2005) Random walks for interactive alpha-matting. In: Proceedings of the VIIP, 423–429
13. Wang J, Cohen MF (2007) Image and video matting: a survey. Found Trends Comput Graph Vis 3:97–175
14. Rhemann C, Rother C, Kohli P, Gelautz M (2010) A spatially varying psf-based prior for alpha matting. In: Proceedings of the IEEE conference on computer vision pattern recognition (CVPR), San Francisco, 2149–2156
15. Chuang YY, Agarwala A, Curless B, Salesin DH, Szeliski R (2002) Video matting of complex scenes. In: Proceedings of the SIGGRAPH, San Antonio, 243–248
16. Apostoloff N, Fitzgibbon A (2004) Bayesian video matting using learnt image priors. Comput Vis Pattern Recognit 1:407–414
17. Bai X, Sapiro G (2009) Geodesic matting: a framework for fast interactive image and video segmentation and matting. Int J Comput Vis 82:113–132
18. McGuire M, Matusik W, Pfister H, Hughes JF, Durand F (2005) Defocus video matting. In: Proceedings of the ACM SIGGRAPH, Los Angeles, 567–576
19. Joshi N, Matusik W, Avidan S (2006) Natural video matting using camera arrays. In: Proceedings of the SIGGRAPH, Boston, 779–786
20. Li Y, Sun J, Shum HY (2005) Video object cut and paste. In: Proceedings of the ACM SIGGRAPH, Los Angeles, 595–600
21. Wang J, Bhat P, Colburn RA, Agrawala M, Cohen MF (2005) Interactive video cutout. In: Proceedings of the ACM SIGGRAPH, Los Angeles, 585–594
22. Jia J, Sun J, Tang CK, Shum HY (2006) Drag-and-drop pasting. In: Proceedings of the ACM SIGGRAPH, Boston, 631–637
23. Lalonde JF, Hoiem D, Efros AA, Rother C, Winn J, Criminisi A (2007) Photo clip art. In: Proceedings of the ACM SIGGRAPH, San Diego
24. Wang J, Cohen MF (2007) Simultaneous matting and compositing. In: Proceedings of the IEEE conference on computer vision pattern recognition (CVPR), Minneapolis
25. Rhemann C, Rother C, Wang J, Gelautz M, Kohli P, Rott P (2009) A perceptually motivated online benchmark for image matting. In: Proceedings of the IEEE conference on computer vision pattern recognition (CVPR), Miami
26. Tai Y-W, Jia J, Tang C-K (2007) Soft color segmentation and its applications. IEEE Trans. Pattern Anal. Mach. Intell. 29, 9 (September 2007), 1520–1537

Maximum Likelihood Estimation

Thomas Brox
Department of Computer Science, University of Freiburg, Freiburg, Germany

Synonyms

Maximum likelihood estimator

Definition

Maximum likelihood estimation seeks to estimate model parameters that best explain some given, independent measurements according to a noise model.

Background

Many problems in computer vision can be formulated as finding the parameters of a predefined model given measurements or training examples.

For example in image segmentation one may want to describe a region by a simple region model, e.g., by a constant intensity value μ. There are many measurements, namely all the pixel intensities in the region. Assuming that these pixel intensities are independently generated from the constant intensity model according to a Gaussian distribution, the goal is to find the most likely parameter μ given these measurements. In this simple example, the optimal parameter μ is the mean of all intensities.

There are many more similar problems in computer vision, for instance, in the scope of optical flow estimation, camera calibration, image denoising, or pattern recognition. In the special case of a Gaussian noise model, maximum likelihood estimation comes down to a least squares approach.

Maximum likelihood estimation is often criticized because it ignores a-priori information, which can be interpreted as assuming a uniform prior density on the parameter space. This becomes especially problematic

when the model is described by many parameters and there are relatively few measurements. In cases where good a-priori assumptions can be made, maximum likelihood estimation should be replaced by maximum a-posteriori estimation, which takes the prior density into account.

Theory

Given a probabilistic model that is described by a parameter vector $\mathbf{w} \in \mathbb{R}^D$ and given N independent measurements $\mathbf{x}_n \in \mathbb{R}^K$, $N \geq D$, one aims at maximizing the likelihood

$$p\left(\mathbf{x}_1, \ldots, \mathbf{x}_N | \mathbf{w}\right) = \prod_{n=1}^{N} p\left(\mathbf{x}_n | \mathbf{w}\right). \tag{1}$$

For numerical reasons, rather than maximizing this probability, it is common to maximize its logarithm, the so-called log-likelihood:

$$\begin{aligned} \mathbf{w}^* &= \operatorname*{argmax}_{\mathbf{w}} \log p(\mathbf{x}_1, \ldots, \mathbf{x}_N | \mathbf{w}) \\ &= \operatorname*{argmax}_{\mathbf{w}} \sum_{n=1}^{N} \log p(\mathbf{x}_n | \mathbf{w}). \end{aligned} \tag{2}$$

Application

Applying this to a simple regression problem, where a line is to be fitted to a couple of points, one has the constraints

$$w_1 x_{1,n} + w_2 = x_{2,n}, \quad n = 1, \ldots, N. \tag{3}$$

Assuming a Gaussian distribution with constant covariance yields

$$\sum_{n=1}^{N} \log p(\mathbf{x}_n | \mathbf{w}) \propto \sum_{n=1}^{N} (w_1 x_{1,n} + w_2 - x_{2,n})^2. \tag{4}$$

The connection to least squares estimation can be seen immediately, but one could as well assume a Laplace distribution, which is more robust to outliers among the measurements and would lead to

$$\sum_{n=1}^{N} \log p(\mathbf{x}_n | \mathbf{w}) \propto \sum_{n=1}^{N} |w_1 x_{1,n} + w_2 - x_{2,n}|. \tag{5}$$

A necessary condition for a maximum of this expression is that the gradient with respect to the parameter vector must vanish:

$$\begin{aligned} \tfrac{\partial}{\partial w_1} \textstyle\sum_{n=1}^{N} |w_1 x_{1,n} + w_2 - x_{2,n}| = 0 \\ \tfrac{\partial}{\partial w_2} \textstyle\sum_{n=1}^{N} |w_1 x_{1,n} + w_2 - x_{2,n}| = 0 \end{aligned} \tag{6}$$

leading to the nonlinear system

$$\begin{aligned} \textstyle\sum_{n=1}^{N} \frac{1}{2} \frac{(w_1 x_{1,n} + w_2 - x_{2,n}) x_{1,n}}{|w_1 x_{1,n} + w_2 - x_{2,n}|} = 0 \\ \textstyle\sum_{n=1}^{N} \frac{1}{2} \frac{(w_1 x_{1,n} + w_2 - x_{2,n})}{|w_1 x_{1,n} + w_2 - x_{2,n}|} = 0, \end{aligned} \tag{7}$$

which can be solved by iteratively keeping the denominators fixed, solving the resulting linear system and updating the denominator. Gaussian distributions lead to linear systems that can be solved directly. More details and examples on maximum likelihood estimation can be found in [1, 2].

References

1. Duda RO, Stork DG, Hart PE (2000) Pattern classification, 2nd edn. Wiley, New York
2. Bishop CM (2006) Pattern recognition and machine learning. Springer

Maximum Likelihood Estimator

▸Maximum Likelihood Estimation

Mesostructure

▸Bidirectional Texture Function and 3D Texture

Methods of Image Recognition in a Low-Dimensional Eigenspace

▸Eigenspace Methods

Microgeometry

▸Bidirectional Texture Function and 3D Texture

Micro Scale Structure

▸Surface Roughness

Mirrorlike Reflection

▸Specularity, Specular Reflectance

Mirrors

Jürgen Beyerer
Fraunhofer Institute of Optronics,
System Technologies and Image Exploitation IOSB,
Karlsruhe, Germany

Definition

A mirror is an optical device used for beam-forming or imaging based on the directional reflection of electromagnetic radiation.

Background

Computer vision applications apply mirrors in a twofold manner: for optical imaging and for illumination purposes. Furthermore, mirrors themselves could be test objects in visual inspection systems. This leads, with regard to the 3D shape of the mirror, to the shape-from-specular-reflection problem, and in the context of visual inspection systems to deflectometry.

Theory and Application

Mirrors consist of a smooth substrate with a metal coating (e.g., Au, Ag, Al) and/or dielectric layers. In the case of a surface mirror, the reflection takes place at a metal coating on the front side that has to be protected against scratches. The main advantage of a surface mirror is the lack of beam displacement due to the glass substrate. Alternatively, the backside of a glass substrate can be coated with a metal layer and with an additional protection against humidity and mechanical damage. Backside mirrors are usually more robust than surface mirrors, but lack their optical characteristics mentioned above.

The physical effect leading to the reflection of electromagnetic waves on metal surfaces can be simply described as "short circuit" of the electrical field.

Dielectric mirrors are composed of multiple thin layers of dielectric materials. They exhibit very high reflectance values, whereas the reflectance depends on wavelength, incident angle, and polarization. Advanced multilayer structure designs can be used to obtain certain functionality [10]:

- A broader reflection bandwidth
- A combination of desirable reflectivity values in different wavelength ranges
- Special polarization properties (for non-normal incidence, thin-film polarizers, polarizing beam splitters)
- Non-polarizing beam splitters
- Edge filters, e.g., long-pass filters, high-pass filters, band-pass filters
- Tailored chromatic dispersion properties

Such mirrors are especially used in laser applications.

Furthermore, thin metal layers allow semitransparent mirrors to be realized for coaxial illumination.

The electromagnetic theory of light is fundamental for the physical understanding of specular reflections [3]. Thereby, the law of reflection describes the geometric aspects, and the Fresnel equations the reflection coefficients, i.e., the radiometric behavior.

The law of reflection states the relationship of the incident $\mathbf{s}_i$ and reflected $\mathbf{s}_r$ light rays with the normal of the specular surface $\mathbf{n}$:

$$\mathbf{s}_i \times \mathbf{n} = \mathbf{s}_r \times \mathbf{n}\,, \tag{1}$$

with $\|\mathbf{s}_i\| = \|\mathbf{s}_r\| = \|\mathbf{n}\| = 1$.

Equation 1 leads, with $\|\mathbf{s}_\mathrm{i} \times \mathbf{n}\| = \sin\theta_\mathrm{i} = \sin\theta_\mathrm{r} = \|\mathbf{s}_\mathrm{r} \times \mathbf{n}\|$, directly to the following two conditions:

- The angle of the incident ray equals that of the reflected ray ($\theta_\mathrm{i} = \theta_\mathrm{r}$).
- The incident and reflected ray are coplanar with the surface normal.

In computer graphics and ray-tracing the law of reflection is often used in the form of a Householder transformation:

$$\mathbf{s}_\mathrm{r} = \boldsymbol{H}\,\mathbf{s}_\mathrm{i} \quad \text{with} \quad \boldsymbol{H} := \boldsymbol{I} - 2\mathbf{n}\mathbf{n}^\mathrm{T}, \tag{2}$$

with the identity matrix $\boldsymbol{I}$.

The bidirectional reflectance distribution function (BRDF; $\rho(\theta_\mathrm{i}, \varphi_\mathrm{i}; \theta_\mathrm{r}, \varphi_\mathrm{r})$) describes the reflectance characteristics of a surface, i.e., the ratio of incident and reflected radiance in dependency of incident and observation angles, Nicodemus et al. [9]. According to Horn and Sjoberg [5] the BRDF for an ideal mirror is according to Horn and Sjoberg [5] (c.f., Fig. 1):

$$\begin{aligned}\rho(\theta_\mathrm{i}, \varphi_\mathrm{i}; \theta_\mathrm{r}, \varphi_\mathrm{r}) &= \frac{\mathrm{d}L_\mathrm{r}}{L_\mathrm{i}\cos\theta_\mathrm{i}\,\mathrm{d}\Omega_\mathrm{i}} \\ &= 2\,\delta(\sin^2\theta_\mathrm{r} - \sin^2\theta_\mathrm{i})\,\delta(\varphi_\mathrm{r} - ((\varphi_\mathrm{i} + \pi) \bmod 2\pi))\,.\end{aligned} \tag{3}$$

In general, the amount of reflected light intensity depends on the wavelength, incident angle, and polarization of the incident light and on the characteristics of the surface itself, e.g., refraction index, shape and roughness.

The Fresnel equations describe the relationship between reflected intensity, incident angle, and polarization state of the incident electromagnetic wave of a smooth surface. These formulas are applicable in the two cases of dielectric and strongly absorbing materials (metals), and establish the theoretical basis for the creation of polarized light with mirrors.

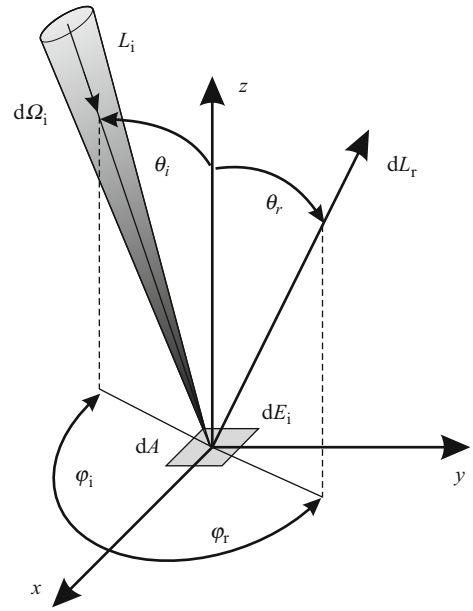
In Fig. 2, the reflectance of some metals is plotted against the wavelength. Most metals have a strong reflectance in the infrared spectrum. For laser applications, mirrors with gold coatings are often sufficient.

With dielectric films even higher reflectance values can be achieved.

Furthermore, the reflectance depends on the surface quality. The dependency on surface roughness σ (root-mean-squared roughness), wavelength λ, and the reflectances R_σ for rough and R for ideal smooth surfaces can be stated as [2]:

$$R_\sigma = R\exp\left[-\left(\frac{4\pi\sigma\cos\theta_\mathrm{i}}{\lambda}\right)^2\right]. \tag{4}$$

The roughness requirements in the far-infrared spectrum are lower than in the visible range. With large incident angles θ_i and surfaces with very small roughness, mirrors applicable even for X-radiation applications can be manufactured.



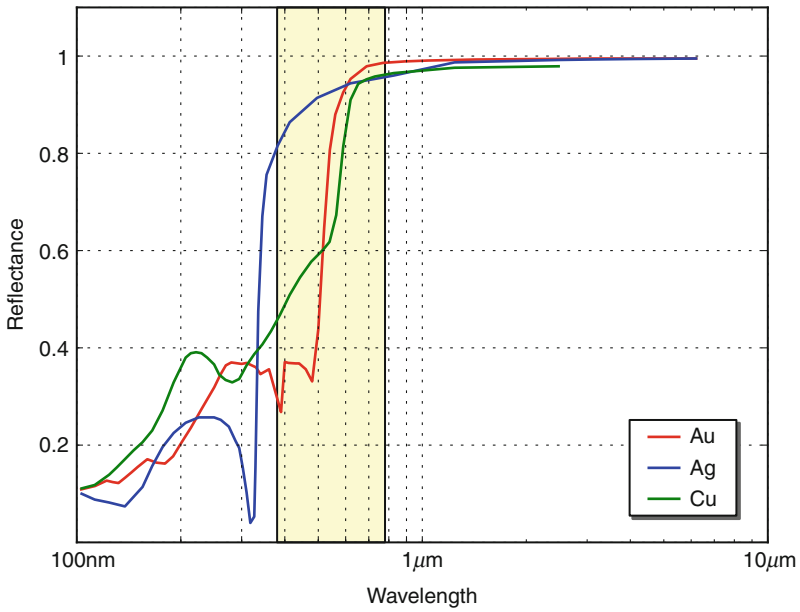
Mirrors, Fig. 1 Geometry of reflection and the BRDF, thereby $\mathrm{d}\Omega_\mathrm{i}$ denotes an infinitesimal solid angle of the incident radiation, L_i, L_r the incident and reflected radiance, and $\mathrm{d}E_\mathrm{i}$ the irradiance on the surface element $\mathrm{d}A$

The most familiar type of mirror is the plane mirror, which has a flat surface. This mirror is mostly used for beam deflection purposes. In Fig. 3a, the geometry of reflection on a plane mirror is shown.

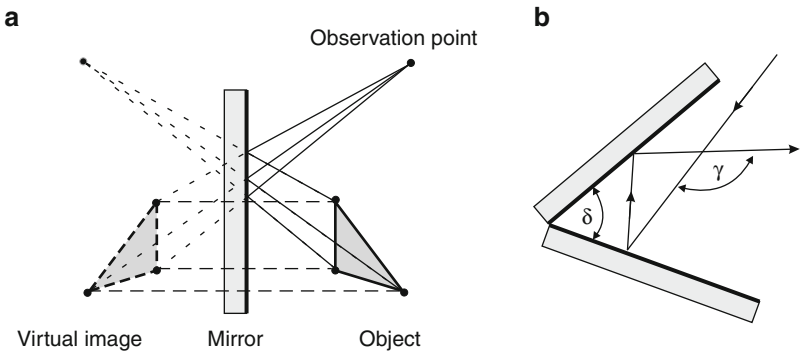
Thereby the image of an object is virtual with magnification equal to one, upright, right-left inversed, without aberrations, and symmetric to the mirror plane.

Figure 3b shows two plane mirrors in an angular mirror setup, whereby $\gamma = 2\delta$. A special case is a triple mirror with three pairwise orthogonal planes ($\delta = 90°$), which is used as a retroreflecting element.

Curved mirrors are also used, such as spherical, ellipsoid, paraboloid, or conical mirrors. Figure 4



Mirrors, Fig. 2 Reflectance vs. wavelength curves for gold (Au), silver (Ag), and copper (Cu) at normal incidence [1]



Mirrors, Fig. 3 Plane (**a**) and angular mirror (**b**)

shows convex and concave mirrors for optical imaging. The focal distance of a spherical mirror with radius r is given by:

$$f = \frac{r}{2}. \tag{5}$$

The mirror equation:

$$\frac{2}{r} = \frac{1}{s} + \frac{1}{s'} \tag{6}$$

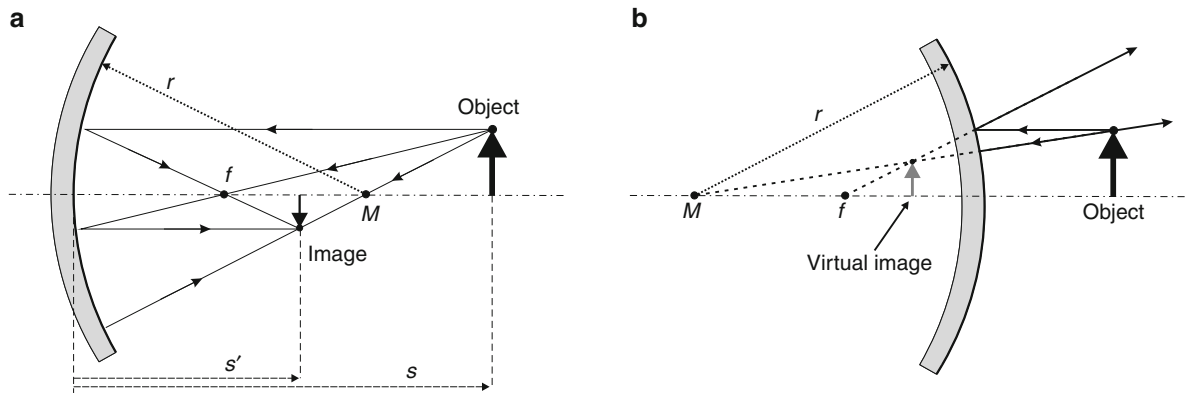
describes the relationship between object and image distances (s, s') with the mirror radius r.

A big advantage of mirrors above lenses is the lack of aberrations, but with the disadvantage of higher centering and adjustment requirements.

Torrance and Sparrow [13] and Phong [11] have, among many others, introduced surface models which can be used to describe specular reflections. Modeling of mirrors or partially reflecting surfaces is of ongoing interest for computer graphics applications.

Open Problems

The main principle for the visual inspection of mirrors is to use a highly controllable environment, where a screen presenting a well-designed pattern is observed via the specular reflecting surface. Knowing that pattern, it is possible to inspect the surface qualitatively and – at least with certain additional knowledge – to reconstruct the surface quantitatively. This reconstruction problem is ill-posed in a mathematical sense, and several regularization approaches have been proposed. The reconstruction of large and complex formed mirrors is still a challenge in the field of computer vision [6–8, 14, 15].



Mirrors, Fig. 4 Optical imaging: concave (**a**) and convex (**b**) mirror

Although the reconstruction problem is ill-posed, humans can usually estimate the shape of mirrors quite well, c.f., Fleming et al. [4]. The visual perception of mirror-like objects is an ongoing research effort.

Another area of ongoing research is the development of mirrors for the extreme ultraviolet (EUV) spectral range, used in EUV lithography tools. These mirrors can be standard Mo/Si mirrors or multilayer setups [12].

References

1. Bass M (ed) (2009) Handbook of optics, vol I–V. Mcgraw-Hill Professional, New York
2. Beckmann P, Spizzichino A (1963) The scattering of electromagnetic waves from rough surfaces. Pergamon Press, London
3. Born M, Wolf E (1999) Principles of optics: electromagnetic theory of propagation, interference and diffraction of light. Cambridge University Press, Cambridge, MA
4. Fleming RW, Torralba A, Adelson EH (2004) Specular reflections and the perception of shape. J Vis 4: 798–820
5. Horn BKP, Sjoberg RW (1979) Calculating the reflectance map. Appl Opt 18(11):1770–1779
6. Ihrke I, Kutulakos KN, Lensch HPA, Magnor M, Heidrich W (2008) State of the art in transparent and specular object reconstruction. In: Theoharis T, Dutre P (eds) STAR - State of the Art Report, EUROGRAPHICS 2008, Crete, Greece. The Eurographics Association, pp 87–108
7. Ikeuchi K (1981) Determining surface orientations of specular surfaces by using the photometric stereo method. IEEE Trans Pattern Anal Mach Intell 3(6):661–669
8. Nayar SK, Ikeuchi K, Kanade T (1991) Surface reflection: physical and geometrical perspectives. IEEE Trans Pattern Anal Mach Intell 13(7):611–634
9. Nicodemus FE, Richmond JC, Hsia JJ, Ginsberg IW, Limperis T (1977) Geometrical considerations and nomenclature for reflectance. Final report national bureau of standards, Washington, DC. Inst. for Basic Standards
10. Paschotta R (2008) Encyclopedia of laser physics and technology. Wiley-VCH, Berlin
11. Phong BT (1975) Illumination for Computer Generated Pictures. Commun ACM 18(6):311–317
12. Soer WA, Gawlitza P, van Herpen MMJW, Jak MJJ, Braun S, Muys P, Banine VY (2009) Extreme ultraviolet multilayer mirror with near-zero IR reflectance. Opt Lett 34(23): 3680–3682
13. Torrance KE, Sparrow EM (1967) Theory for off-specular reflection from roughened surfaces. J Opt Soc America 57(9):1105–1114
14. Wang Z, Inokuchi S (1993) Determining shape of specular surfaces. In: The 8th scandinavian conference on image analysis, Tromso, pp 1187–1194
15. Werling S, Mai M, Heizmann M, Beyerer J (2009) Inspection of specular and partially specular surfaces. Metrology Meas Syst 16(3):415–431

Mobile Observers

Jan-Olof Eklundh
School of Computer Science and Communication, KTH - Royal Institute of Technology, Stockholm, Swedon

Definition

A mobile (visual) observer is an agent or a system that perceives its environment using vision. In computer vision this typically is a mobile device such as a robot carrying one or more cameras.

Background

Gibson [1] claimed that a mobile observer is a prerequisite for natural vision. He discriminated between *ambient* or *ambulatory vision*, when the observer can move its head or body, and *snapshot* or *aperture vision* in cases when one or several images are recorded momentarily at certain fixation points. All those aspects are treated in computer vision, although current trends are on processing static images in the spirit of *snapshot* or *aperture vision*. Computer vision researchers began to study *visual motion* in the 1970s, when it became possible to connect video cameras to computers. This work did not really concern mobile observers, but such existed even earlier, when cameras were used as input devices to robots, e.g., in the work on "Shakey" [2]. Nowadays, mobile observers most often occur in the context of mobile robots, but recent developments on *wearable vision* have widened the interest in the topic. *Ambient vision* is what you have for instance in the case of *pan-tilt heads*, which are used in a large range of applications.

Theory

There have been attempts to find the notions of active and mobile observers theoretically. In biological vision Gibson's work is of a landmark nature, but there are many other proposals as well, e.g., relating to functionalism [3]. In computer vision the problem has been considered from the point of view of active vs. passive vision [4–6]. In [7–9] the theoretical aspects are more directly addressed. However, even with these attempts, one can hardly say that there exists any complete theory for a mobile observer.

Problems and Applications

The mobile observer obtains a stream or sequence of images as input rather than single images. This provides rich information about the environment as well as of the movements of the observer. However, observer motion also implies that there is image motion in almost every point in the sequence. In a static world, observer motion creates essentially all the variations over time in the images, i.e., those that are due to change of viewpoint and not, e.g., in illumination. If there are things in the environment that also move, the two types of motion are confounded in the images.

A mobile observer can derive (static) scene geometry through *structure-from-motion* algorithms. Moreover, *ego-motion*, i.e., the motion of the observer, can be estimated. Generally such methods assume a static background that is prominent in the field of view. Independently moving objects can then also be detected, and under certain conditions their motion can be estimated. *Ego-motion estimation* obviously plays an important role here. There are many types of algorithms for this, e.g., based on *optical flow*, monocular or binocular *feature tracking*, or *image stabilization*. There also exist algorithms for using *omnidirectional* or *composite cameras*, which highlights the fact that effects of ego-motion are manifested in a wide field of view. For instance, small rotations of an observer moving straight ahead can be estimated from peripheral flow, something that is useful in driving and in guiding of autonomous robots.

A mobile observer can be active or passive. In the first case, it purposively guides its motion and/or the way it directs its gaze on the basis of tasks it is involved in and as a reaction to what it observes. *Gaze control* and *fixation* in dynamic situations have been studied extensively in the field of *active vision*. In some cases these mechanisms have been used to control observer motion, e.g., for exploring a scene or an object and to facilitate recognition. Then *viewpoint planning* becomes an issue. However, a more general case is when the observer motion is only loosely dependent of what is seen, except for possible control of gaze. For instance, a mobile observer can induce depth cues through parallax by (small) camera motions that are not pure rotations. Another example is given by a robot moving from one point to another while observing an object along its path, analogously to a person riding in a car. Many applications contain elements of both active and passive observations, for instance in robot navigation including *obstacle avoidance* and *mapping* (as in *SLAM*), *hand-eye control* in grasping and manipulation, and in general for an ambulant observer, such as those studied in the context of *wearable* or *egocentric vision*.

Open Problems

The study of mobile observers from a computational perspective involves a broad range of problems traditionally addressed in computer vision. However,

there are certain issues that become central. For instance, the *correspondence problem* is ubiquitous. In applications such as those described above, the tight connection between *perception and action* is apparent. Visual sensing involving motor control raises problems on time criticality and real-time computations [9]. Other problems arise because the mobile observer continuously samples the visual world. Meaningful behavior based on the huge amounts of information requires methods for *attention* and *visual search*. In all, although some of the problems encountered in the study of mobile observers largely overlap those generally treated in computer vision, there are others that are specific to this area.

References

1. Gibson JJ (1979) The ecological approach to visual perception. Houghton Mifflin, Boston
2. http://www.ai.sri.com/shakey/
3. O'Regan JK, Noe A (2001) A sensorimotor account of vision and visual consciousness. Behav Brain Sci 24:939–1031
4. Bajcsy R (1985) Active perception vs. passive perception. In: Proceedings of the 3rd IEEE workshop on computer vision, Bellaire. IEEE CS Press, pp 55–59
5. Aloimonos Y, Weiss I, Bandyopadhyay A (1987) Active vision. In: Proceedings of the 1st ICCV, London. IEEE CS Press, pp 35–54
6. Ballard DH (1991) Animate vision. Artif Intell 48:57–86
7. Tsotsos JK (1992) On the relative complexity of active vs. passive visual search. Int J Comput Vis 7:127–141
8. Bennett BM, Hoffman DD, Prakash C (1989) Observer mechanics: a formal theory of perception. Academic, San Diego
9. Soatto S (2011) Steps toward a theory of visual information. CoRR. MIT, Cambridge

MoCap

▸Motion Capture

Model-Based Object Recognition

Min Sun and Silvio Savarese
Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, MI, USA

Synonyms

Object models; Object parameterizations; Object representations; Visual patterns

Related Concepts

▸Human Pose Estimation; ▸Object Class Recognition (Categorization); ▸Object Detection

Definition

Model-based object recognition addresses the problem of recognizing objects from images by means of a suitable mathematical model that is used to describe the object.

Background

In model-based object recognition, an object model is typically defined so as to capture object's geometrical and appearance properties at the appropriate level of specificity. For instance, an object model can be designed to recognize a generic "face" as opposed to "someone's face" or vice versa. In the former case, which is often referred to as the object categorization problem, the main challenge is to design models that are capable of retaining key visual properties for representing an object category, such as a "face," at the appropriate level of abstraction. Such models can be then used to recognize novel object instances from a query image. Moreover, a model must be able to generalize across variations in the object's visual characteristics due to viewpoint and illumination changes as well as due to occlusions or deformations. Meeting all of these desiderata can be extremely challenging. This makes object recognition an open, yet key, problem in computer vision.

Object Models for Recognition

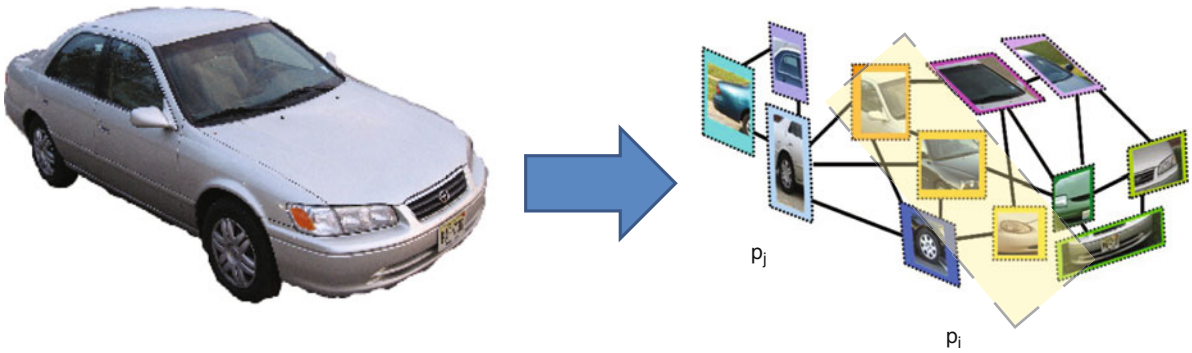
The design of an object model must reflect its ability to (i) capture geometrical and appearance characteristics of the object at the appropriate level of specificity and (ii) generalize across variations in viewpoint, illumination, occlusions, and deformations. The complexity of the representation can be reduced by making assumptions on the type of object specificity or the degree of viewpoint, occlusions, and deformation variability. Ultimately, the strategy in designing an object model will depend on the relevant application scenario.

Object models that are designed to recognize objects at the highest level of specificity – e.g., "my face" as opposed to "a face" – are often referred to as single-instance object models. These models are capable of recognizing a specific object instance while guaranteeing the ability to handle occlusions and a large degree of viewpoint variability. Research on 3D object recognition, from early contributions [1–9] to the most recent ones [10–13], follows these assumptions. Since single-instance object models do not need to accommodate any intra-class variations, they often consist of a rigid collection of visual features associated to a number of 2D or 3D templates. In recognition, by matching features of the query image with those associated to the models, it is possible to identify the object of interest and determine its 3D pose with respect to a common reference system. This matching process is usually subject to a geometrical validation phase that helps verify that the appearance, and geometric properties of the query object are consistent with the estimated pose transformation between observation and object model. While critical for ensuring sufficient discrimination power for recognizing single-instance objects as well as for enabling large viewpoint variability, tight geometrical constraints become inadequate when shape and appearance intra-class variability must be accounted for.

Object models that are designed to recognize objects at a lower level of specificity – e.g., "a face" as opposed to "my face"– are often referred to as categorical object models. The ability to generalize across instances in the same category is critical and is typically achieved by characterizing the object as a collection of features whose appearance and geometrical properties tend to systematically occur in the category of interest. For instance, if the goal is to recognize a car, appearance properties such as the "color of the body" are not adequate to help obtain the right level of generalization (abstraction), whereas the orientation of edges associated to a wheel can capture more general appearance cues across instances. Appearance properties are typically captured by image descriptors such as [10, 14] associated to interest points that are detected at different locations and scales of the image. A popular design choice is to describe the object appearance by histograms of vector-quantized descriptors [15–17]. The ability of image descriptors such as [10] to be invariant to affine illumination transformations makes the appearance models robust to variability in illumination conditions. Geometrical properties are captured by retaining the spatial organization of features in the image and include simple characterizations based on the 2D location of either feature points or aggregation of features (e.g., edges, parts, fragments) with respect to a given object reference point [18–22]. Object models constructed upon constellation of parts such as [18–20] are suitable to accommodate object variations due to occlusions and simple 2D planar geometrical deformations (isometries or affinities). Suitable machine learning and probabilistic inference techniques such as expectation maximization (EM) [23], latent SVM (LSVM), [54] Markov random field (MRF) [24, 25], conditional random field (CRF) [26], generalized Hough voting [27], and RANdom SAmple Consensus (RANSAC) [28] are used to automatically select appearance and geometrical properties so as to reach the appropriate level of generalization and discrimination power.

Most of the object models for object categorization mitigate the complexity of the representation by assuming that objects are viewed from a limited number of poses and learn an object model that is specialized to identify the object from a specific viewpoint. These are often referred to as view-dependent object models. If similar views in the training set are available, the recognition problem is reduced to match the new query object to one, or a mixture, of the learnt view-dependent object models [29, 30]. The drawback of view-dependent object models is that (i) they can accommodate very limited viewpoint variability – mostly changes in scale or 2D rotation transformations – and (ii) different poses of the same object category result in completely independent models, where neither features or parts are shared across views. Because each single-view models are independent, these methods are often costly to train and prone to false alarms, if several views need to be encoded.

Object models that can accommodate both large viewpoint changes and large intra-class variability (low degree of specificity) overcome the above limitations by introducing a representation that seeks to effectively captures the intrinsic three-dimensional nature of the object category. These models are typically divided into two types: 2-1/2D layout models and 3D layout models [33]. In the 2-1/2D layout models [31, 32, 34], object diagnostic elements (features, parts, contours) are connected across views to form an unique and coherent 2-1/2D model for the



Model-Based Object Recognition, Fig. 1 Example of 2-1/2D layout models as introduced in [31] and generalized in [32]. *Left panel*: An image of an object category of interest. *Right panel*: In the 2-1/2D layout model, object parts are connected to form a graph structure. Each node P_i captures diagnostic appearance of the object part which is assumed to be locally planar. Each edge describes an homographic transformation that captures the viewpoint transformation between parts. The homographic transformation is illustrated by showing that some parts are slanted with respect to others

object category (Fig. 1). Relationships between features or parts capture the way that such elements are transformed as the viewpoint changes. These methods share some key ideas with pioneering works in 3D object recognition [1–6, 8, 9] as well as with the theory of aspect graphs [7, 35]. In the 3D layout models [36–41], object elements are organized in a common 3D reference frame and form a compact 3D representation of the object category. Such 3D structures of features (parts, edges) can give rise, for instance, to either a 3D generalization of 2D pictorial structures or constellation models or to hybrid models where features (parts or edges) lie on top of 3D object reconstructions or CAD volumes.

Open Problems

Although object recognition has been a core problem in computer vision for more than four decades and several powerful models have been proposed, state-of-the-art methods are still far from the level of accuracy, efficiency, and robustness that the human visual system achieves in recognizing, detecting, and categorizing objects from images. Recently, several new paradigms have been explored to address the above limitations. One major effort involves large-scale object recognition. With the introduction of ultra-large-scale datasets such as the ImageNet [42] – a collection of millions of images organized into a hierarchical ontology of thousands of categories – it is now possible to evaluate methods for object categorization that seek to (i) efficiently process these many images and categories and (ii) understand objects at different level of specificity; this is also referred to as to the fine-grain categorization problem [43–45]. Another major effort is related to the introduction of a recent paradigm whereby objects are modeled and recognized by means of their attributes. As pioneered by [46–48], visual attributes such as "it is metallic"; "it has wheels" can be used to obtain more effective and descriptive characterizations of object categories (i.e., a car or a truck). This has the benefit of (i) making the "boundaries" between different categories more fluid than in traditional parameterizations, (ii) enabling more powerful methods for fine-grained categorization [44], and (iii) providing critical building blocks for transferring visual properties across categories (transfer learning, one short learning) [46, 48].

Other important problems for future work include the ability to (i) overcome the traditional paradigm whereby objects are identified as just bounding boxes in images but rather provide a richer characterization in terms of their accurate outlines or segments, 3D properties (pose or 3D shape) [36, 41], as well as attributes; (ii) find a common ground between bottom-up representations (from pixels to features), akin to recent developments on convolutional neural networks [49, 50], and top-down models as recently advocated in [51] and (iii) describe the interplay between objects and their components at different levels of semantic resolution [52, 53].

References

1. Binford T (1971) Visual perception by computer. IEEE conference on systems and control
2. Marr D (1978) Representing visual information. Computer vision systems
3. Palmer S, Rosch E, Chase P (1981) Canonical perspective and the perception of objects. Atten Perform 9:135–151
4. Tarr M, Pinker S (1989) Mental rotation and orientation-dependence in shape recognition. Cogn Psychol 21: 233–282
5. Poggio T, Edelman S (1990) A neural network that learns to recognize three-dimensional objects. Nature 343:263–266
6. Ullman S, Basri R (1991) Recognition by linear combinations of models. TPAMI 13:992–1006
7. Koenderink J, Doorn AV (1979) The internal representation of solid shape with respect to vision. Biol Cybern 32: 211–216
8. Huttenlocher DP, Ullman S (1987) Object recognition using alignment. In: ICCV
9. Lowe D, Binford T (1985) The recovery of three-dimensional structure from image curves. TPAMI 7: 320–326
10. Lowe DG (1999) Object recognition from local scale-invariant features. In: ICCV
11. Rothganger F, Lazebnik S, Schmid C, Ponce J (2003) 3D object modeling and recognition using affine-invariant patches and multi-view spatial constraints. In: IEEE conference on computer vision pattern recognition (CVPR)
12. Brown M, Lowe DG (2005) Unsupervised 3D object recognition and reconstruction in unordered datasets. In: 3DIM
13. Ferrari V, Tuytelaars T, Gool L (2006) Simultaneous object recognition and segmentation from single or multiple model views. IJCV 67:159–188
14. Mikolajczyk K, Schmid C (2002) An affine invariant interest point detector. In: European conference on computer vision (ECCV)
15. Dance C, Willamowski J, Fan L, Bray C, Csurka G (2004) Visual categorization with bags of keypoints. In: European conference on computer vision (ECCV) international workshop on statistical learning in computer vision, Prague
16. Grauman K, Darrell T (2005) The pyramid match kernel: Discriminative classification with sets of image features. In: ICCV
17. Fei-Fei L, Fergus R, Perona P (2004) Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: IEEE conference on computer vision pattern recognition (CVPR)
18. Fergus R, Perona P, Zisserman A (2003) Object class recognition by unsupervised scale-invariant learning. In: IEEE conference on computer vision pattern recognition (CVPR)
19. Felzenszwalb PF, Huttenlocher DP (2005) Pictorial structures for object recognition. IJCV 61(1):55–79
20. Leibe B, Leonardis A, Schiele B (2004) Combined object categorization and segmentation with an implicit shape model. In: European conference on computer vision (ECCV) workshop on statistical learning in computer vision
21. Lazebnik S, Schmid C, Ponce J (2006) Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: IEEE conference on computer vision pattern recognition (CVPR)
22. Savarese S, Winn J, Criminisi A (2006) Discriminative object class models of appearance and shape by correlatons. In: IEEE conference on computer vision pattern recognition (CVPR)
23. Dempster A, Laird N, Rubin D (1977) Maximum likelihood from incomplete data via the em algorithm. J R Stat Soc 39:1–38
24. Koller D, Friedman N (2009) Probabilistic graphical models: principles and techniques. MIT
25. Wainwright MJ, Jordan MI (2008) Graphical models, exponential families, and variational inference. Found Trends Mach Learn 1:1–305
26. Lafferty J, McCallum A, Pereira F (2001) Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML
27. Ballard D (1981) Generalizing the hough transform to detect arbitrary shapes. Pattern Recognit 13:111–122
28. Fischler MA, Bolles RC (1981) Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 24:381–395
29. Schneiderman H, Kanade T (2000) A statistical approach to 3D object detection applied to faces and cars. In: IEEE conference on computer vision pattern recognition (CVPR)
30. Weber M, Einhaeuser W, Welling M, Perona P (2000) Viewpoint-invariant learning and detection of human heads. In: International conference on automatic face and gesture recognition
31. Savarese S, Fei-Fei L (2007) 3D generic object categorization, localization and pose estimation. In: ICCV
32. Su H, Sun M, Fei-Fei L, Savarese S (2009) Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In: ICCV
33. Hoiem D, Savarese S (2011) Representations and techniques for 3D object recognition and scene interpretation. In: Synthesis lecture on artificial intelligence and machine learning. Morgan Claypool, San Rafael
34. Thomas A, Ferrari V, Leibe B, Tuytelaars T, Schiele B, Goo LV (2006) Towards multi-view object class detection. In: IEEE conference on computer vision pattern recognition (CVPR)
35. Bowyer K, Dyer CR (1990) Aspect graphs: An introduction and survey of recent results. Int J Imaging Syst Technol 2:315–328
36. Sun M, Bradski G, Xu BX, Savarese S (2010) Depth-encoded hough voting for joint object detection and shape recovery. In: European conference on computer vision (ECCV)
37. Hoiem D, Rother C, Winn J (2007) 3D layoutcrf for multi-view object class recognition and segmentation. In: IEEE conference on computer vision pattern recognition (CVPR)
38. Liebelt J, Schmid C (2010) Multi-view object class detection with a 3D geometric model. In: IEEE conference on computer vision pattern recognition (CVPR)
39. Pepik B, Stark M, Gehler P, Schiele B (2012) Teaching 3D geometry to deformable part models. In: IEEE conference on computer vision pattern recognition (CVPR)
40. Arie-Nachimson M, Basri R (2009) Constructing implicit 3D shape models for pose estimation. In: ICCV

41. Xiang Y, Savarese S (2012) Estimating the aspect layout of object categories. In: IEEE conference on computer vision pattern recognition (CVPR)
42. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: IEEE conference on computer vision pattern recognition (CVPR)
43. Yao B, Bradski G, Fei-Fei L (2012) A codebook-free and annotation-free approach for fine-grained image categorization. In: IEEE conference on computer vision pattern recognition (CVPR)
44. Duan K, Parikh D, Crandall D, Grauman K (2012) Discovering localized attributes for fine-grained recognition. In: IEEE conference on computer vision pattern recognition (CVPR)
45. Perona P (2010) Visions of a visipedia. Proc IEEE 98: 1526–1534
46. Farhadi A, Endres I, Hoiem D, Forsyth D (2009) Describing objects by their attributes. In: IEEE conference on computer vision pattern recognition (CVPR), Miami
47. Ferrari V, Zisserman A (2007) Learning visual attributes. In: NIPS
48. Lampert CH, Nickisch H, Harmeling S (2009) Learning to detect unseen object classes by between-class attribute transfer. In: IEEE conference on computer vision pattern recognition (CVPR)
49. Lee H, Grosse R, Ranganath R, Ng AY (2009) Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: ICML
50. Yann LeCun FJH, Bottou L (2004) Learning methods for generic object recognition with invariance to pose and lighting. In: IEEE conference on computer vision pattern recognition (CVPR)
51. Arbelaez P, Maire M, Fowlkes C, Malik J (2011) Contour detection and hierarchical image segmentation. IEEE Trans Pattern Anal Mach Intell 33(5):898–916
52. Zhu L, Chen Y, Yuille A (2006) Unsupervised learning of a probabilistic grammar for object detection and parsing. In: NIPS
53. Todorovic S, Ahuja N (2008) Unsupervised category modeling, recognition, and segmentation in images. IEEE Trans Pattern Anal Mach Intell 30(12):2158–2174
54. Felzenszwalb P, Girshick R, McAllester D, Ramanan D (2010) Object detection with discriminatively trained part-based models. TPAMI 32:1627–1645

Monte Carlo Annealing

►Simulated Annealing

Morphology, Form Analysis

►Statistical Shape Analysis

Motion Blur

Neel Joshi
Microsoft Corporation, Redmond, WA, USA

Synonyms

Camera-shake blur; Object motion blur

Related Concepts

►Blur Estimation; ►Defocus Blur

Definition

Motion blur is due to motion of scene objects or the camera while the camera shutter is open, thus causing scene points to be imaged over a large area of camera sensor or film. The motion blur is a projection of the motion path of the moving objects onto the image plane. The motion path of a point can be due to translation and rotation of the camera or scene objects in three dimensions. There can be different paths for different parts of the scene, and in light-limited situations, when using long exposures, these paths can be quite large, resulting in very large blurs.

Background

Image blur can be described by a point spread function (PSF). A PSF models how an imaging system captures a single point in the world – it literally describes how a point "spreads" across an image. An entire image is then made up of a sum of the individual images of every scene point, where each point's image is affected by the PSF associated with that point. For an image to be "sharp" means that one ideally does not want any image blur. Thus, the PSF should be minimal, i.e., a delta function, where each scene point should correspond only to one image point. In practice, PSFs can take on a range of shapes and sizes depending on the properties of an imaging system. When this PSF is

Motion Blur, Fig. 1 With motion blur, the amount of blur depends on the relative motion between the camera and the scene objects; it depends on the focal length and of the lens and the scene depth and motion trajectories. An example of camera motion blur is shown in the *middle*, where the blur kernel is drawn for each corner of the image (From Joshi et al. [3]). An example of object motion blur is shown on the *right* (From Jia [11])

large relative due to camera or scene motion and relative to the image resolution and pixel size, an image with motion blur is captured.

The fundamental cause of motion blur is that a camera does not sample light from a single moment in time, but instead captures images by integrating the light over an exposure window. The relative motion between camera and scene objects is the primary factor in motion blur, as illustrated in Fig. 1. The path of motion during exposure affects the PSF and thus the blur shape and size. Properties such as exposure duration, lens focal length, and pixel size play an additional role.

Theory

Image blur is described by a point spread function (PSF). The PSF models how an imaging system captures a single point in the world.

The most commonly used model for blur is the linear model, where the blurred image b is represented as a convolution of a kernel k, plus noise:

$$b = i \otimes k + n, \tag{1}$$

where $n \sim \mathcal{N}(0, \sigma^2)$, which represents and additive Gaussian noise model. In this model, the blur is assumed to be constant over the entire image, i.e., spatially invariant; however, that is often not true in practice [1, 2]. If there is depth variation in the scene, the motion blur can change with that depth due to parallax. In these cases, one can think of the blur kernel, k, as being a function of image position, i.e., $k(x, y)$.

To model spatially varying blur, the spatially invariant kernel and convolution in (Eq. 1) can be replaced by a sparse re-sampling matrix that models the spatially variant blur, and the convolution process is now a matrix-vector product:

$$\boldsymbol{b} = A\boldsymbol{i} + \boldsymbol{i}. \tag{2}$$

Each column of A is the unraveled kernel for the pixel represented by that column. Thus, the blurred response at a pixel in the observed image is computed as a weighted sum, as governed by A, of the latent sharp image i formed into a column vector.

Representation

To model motion blur, first, let us consider the image a camera captures during its exposure window. The intensity of light from a scene point (X_t, Y_t, Z_t) at an instantaneous time t is captured on the image plane at a location (u_t, v_t), which is a function of the camera projection matrix P_t. In homogenous coordinates, this can be written as

$$(u_t, v_t, 1)^T = P_t(X_t, Y_t, Z_t, 1)^T. \tag{3}$$

If there is camera motion, P_t varies with time as a function of camera rotation and translation causing points in the scene to project to different locations at each time. If there is scene motion, (X_t, Y_t, Z_t) also varies with time, which also affects where the points project on the image plane. The integration of these projected observations creates a blurred image, and the projected trajectory of each point on the image plane is that point's point spread function (PSF). The camera projection matrix can be decomposed as

$$P_t = K\Pi E_t, \tag{4}$$

where K is the intrinsic matrix, Π is the canonical perspective projection matrix, and E_t is the time-dependent extrinsic matrix that is composed of the camera rotation R_t and translation T_t. In the case of image blur, it is not necessary to consider the absolute motion of the camera, only the relative motion and its effect on the image. This can be modeled by considering the planar homography that maps the initial projection of points at $t = 0$ to any other time t [3], i.e., the reference coordinate frame is coincident with the frame at time $t = 0$:

$$H_t(d) = \left[K(R_t + \frac{1}{d} T_t N^T) K^{-1} \right] \tag{5}$$

$$(u_t, v_t, 1)^T = H_t(d)(u_0, v_0, 1)^T, \tag{6}$$

for a particular depth d, where N is the unit vector that is orthogonal to the image plane.

If the scene is not moving, given an image I at time $t = 0$, the pixel value of any subsequent image is

$$I_t(u_t, v_t) = I(H_t(d)(u_0, v_0, 1)^T). \tag{7}$$

This image warp can be rewritten in matrix form as

$$\boldsymbol{I_t} = A_t(d)\boldsymbol{I}, \tag{8}$$

where $\boldsymbol{I_t}$ and $\boldsymbol{I}$ are column-vectorized images and $A_t(d)$ is a sparse re-sampling matrix that implements the image warping and re-sampling due to the homography. Each row of $A_t(d)$ contains the weights to compute the value at pixel (u_t, v_t) as the interpolation of the point $(u_0, v_0, 1)^T = H_t(d)^{-1}(u_t, v_t, 1)^T$. Thus, an alternative formulation for image blur is the integration of applying these homographies over time [3]:

$$\boldsymbol{B} = \int_0^s [A_t(d)\boldsymbol{I}\, dt]. \tag{9}$$

This leads to the spatially variant blur matrix in (Eq. 2):

$$A(d) = \int_0^s A_t(d) dt. \tag{10}$$

For camera motion blur, A is a function of depth. If there is scene motion, the full model can be extended to handle the time-varying mapping of scene points, (X_t, Y_t, Z_t), to the image plane.

Application

Estimation of camera motion blur [1, 3–9] and estimation of object motion blur [10–13] are extensively researched areas.

Estimated blur kernels are typically used for improving image quality by reducing blur using image deblurring and deconvolution methods [1, 10, 14, 15]. There are also methods that reduce motion blur or make the blur more easily removable but changing how images are captured [13, 16].

References

1. Joshi N, Szeliski R, Kriegman DJ (2008) Psf estimation using sharp edge prediction. In: Computer vision and pattern recognition, 2008 (CVPR 2008). IEEE conference on, Anchorage, pp 1–8
2. Levin A, Weiss Y, Durand F, Freeman W (2009) Understanding and evaluating blind deconvolution algorithms. In: Computer vision and pattern recognition, 2009 (CVPR 2009). IEEE conference on, Miami (Beach), IEEE Computer Society, pp 1964–1971
3. Joshi N, Kang SB, Zitnick CL, Szeliski R (2010) Image deblurring using inertial measurement sensors. ACM Trans Graph 29:30:1–30:9
4. Bascle B, Blake A, Zisserman A (1996) Motion deblurring and super-resolution from an image sequence. In: ECCV '96: Proceedings of the 4th European conference on computer vision-vol II, Springer, London, pp 573–582
5. Fergus R, Singh B, Hertzmann A, Roweis ST, Freeman WT (2006) Removing camera shake from a single photograph. ACM Trans Graph 25:787–794
6. Yuan L, Sun J, Quan L, Shum HY (2007) Image deblurring with blurred/noisy image pairs. In: SIGGRAPH '07: ACM SIGGRAPH 2007 papers, ACM, New York, pp 1
7. Ben-Ezra M, Nayar S (2004) Motion-based motion deblurring. IEEE Trans Pattern Anal Mach Intell 26(6): 689–698
8. Tai YW, Du H, Brown MS, Lin S (2008) Image/video deblurring using a hybrid camera. In: Computer vision and pattern recognition, 2008 (CVPR 2008). IEEE conference on, Anchorage, pp 1–8
9. Park SY, Park ES, Kim HI (2008) Image deblurring using vibration information from 3-axis accelerometer. J Inst Electron Eng Korea. SC, Syst control 45(3):1–11
10. Levin A (2006) Blind motion deblurring using image statistics. In: Weiss Y et al (eds) Advances in neural information processing systems. MIT, Cambridge
11. Jia J (2007) Single image motion deblurring using transparency. Computer vision and pattern recognition, 2007 (CVPR '07). IEEE conference on, Minneapolis, pp 1–8
12. Qi Shan WX, Jia J (2007) Rotational motion deblurring of a rigid object from a single image. In: ICCV '07, Rio de Janeiro

13. Levin A, Sand P, Cho TS, Durand F, Freeman WT (2008) Motion-invariant photography. ACM Trans Graph 27:71:1–71:9
14. Richardson WH (1972) Bayesian-based iterative method of image restoration. J Opt Soc Am (1917–1983) 62:55–59
15. Levin A, Fergus R, Durand F, Freeman WT (2007) Image and depth from a conventional camera with a coded aperture. In: SIGGRAPH '07: ACM SIGGRAPH 2007 papers, ACM, New York, p 70
16. Raskar R, Agrawal A, Tumblin J (2006) Coded exposure photography: motion deblurring using fluttered shutter. ACM Trans Graph 25:795–804

Motion Capture

Nils Hasler
Graphics, Vision & Video, MPI Informatik, Saarbrücken, Germany

Synonyms

MoCap; Motion capturing; Motion tracking; Performance capture

Related Concepts

▸Kinematic Motion Models; ▸Multiview Stereo

Definition

Motion capture is the process of recording the motion of a subject, processing it on a computer, and mapping it onto a virtual character.

Background

Parameterizing human motion is not just of academic interest, e.g., for studying the muscoskeletal system of humans, but has many applications in the industry. Historically, the first motion capture systems have been developed in the 1970s and 1980s to perform gait analysis in clinical settings. Today, however, sports sciences and the entertainment industry make heavy use of the technology as well. The setups are also not constrained to human gait analysis any longer. Instead, full-body motion of several humans, animals, stage props, and virtual cameras can be processed in real time.

When capturing motion, it is commonly assumed that the body or object can be decomposed into rigidly moving parts connected by joints. That way, the pose of a human, animal, or mechanical stage prop can be parameterized by a small set of joint angles organized in a hierarchical tree, the skeleton. This hierarchy can be inferred given trajectories of markers attached to the body. Yet, since this step is computationally expensive, the skeleton is normally supplied and scaled to the size of the actor beforehand.

Classification

Motion capture systems can be categorized by their use of sensors. Optical systems use cameras operating in the visible or infrared spectrum, whereas nonoptical systems are based on various other modalities.

Nonoptical Systems

Various nonoptical motion capture systems have been proposed using different sensors. They are grouped here because compared to optical motion capture, they occupy a marginal position. Yet, all approaches discussed here, have in common that they solve the main disadvantage of optical systems, the sensitivity to occlusion.

Mechanical tracking systems measure the angles of the joints mechanically, i.e., by attaching goniometers to the joints of the subject. Estimating the pose given the joint angles is straightforward, but several problems exist with the approach. The mechanical alignment of the goniometers with the body joints can be difficult, especially for joints with more than one degree of freedom, e.g., the shoulder, the devices are cumbersome, and limb lengths have to be measured very accurately for every subject to prevent drift.

Magnetic fields can be used to estimate the orientation of a magnetic sensor relative to the source of the field. By modulating magnetic coils in the vicinity of the capture volume and measuring the field at different points in time, position and orientation of the sensor can be inferred. The approach has the advantage that it does not suffer from occlusion because the human body is transparent to magnetic fields. However, magnetic fields attenuate rapidly over distance and are sensitive to interference with electrical equipment. The latter is a severe shortcoming as motion

capture systems are frequently used in conjunction with other equipment such as motion picture cameras, computers, or stage lights.

Inertial sensors measure acceleration and angular velocity of the limbs they are attached to. Aside from the acceleration of the body, accelerometers measure the gravitational acceleration. After compensating for gravitation, integrating measurements over time yields the pose of the subject. This approach is, like the previous methods, invariant to occlusion, but the numerical integration of measurements leads to drift in position and orientation. It can, however, be used effectively in conjunction with a drift-free method to compensate for drift and bridge occlusions with the acceleration data.

Optical Systems

The most common systems today are optical, i.e., they use one or more calibrated camera(s) to estimate the pose of the subject. Most commercial systems today require the actor to wear markers to simplify the tracking. One of the first marker-based systems used pieces of paper that glow under ultraviolet illumination. Nowadays, either small retroreflective balls or light-emitting diodes (LEDs) operating in the visible or infrared spectrum are used.

Passive systems are normally equipped with infrared lights located in rings around the cameras. This setup evokes a distinct dot for each retroreflective marker in the captured video frames. To further improve image quality, infrared filters in front of the cameras help to reduce spurious highlights in the visible spectrum. There are two main drawbacks of passive marker-based systems. Since all markers look alike, their trajectories can easily be confused, and all optical systems have in common that they suffer from occlusions.

Active markers, e.g., pulsed LEDs, have the advantage that markers can be identified by the blinking pattern. That way, confusing marker trajectories becomes impossible. Additionally, active marker systems are not restricted to studio environments because the blinking patterns can be distinguished effectively from interference introduced by sunlight. One disadvantage of active markers is that they have to be powered. Carrying a battery pack is not a big burden for an actor, but some stage props such as arrows cannot easily be fitted with batteries.

Markerless systems are subject of intense scientific research. However, first commercial systems are available as well. Yet, it is still unclear, which of the proposed approaches will prove to be the most effective in the long run. Most vision-based approaches today use a combination of edge features, silhouette constraints, texture analysis, and feature tracking or optical flow. Pose optimization is performed using gradient descent, particle filters, simulated annealing, belief propagation, or a combination of these methods. The main advantage of markerless systems is that the amount of preparation of the subject is minimal. Common drawbacks of current systems are that they lack robustness or restrict the setting in other ways than by adding markers, e.g., it is assumed that the background is static. Markerless systems also tend to be computationally expensive. The most advanced systems today are able to capture a single actor in real time, compared to five actors with a marker-based system.

Theory

Generally, the different motion capture systems require different workflows to set up the system and then to fit a skeleton to the captured data. Since optical systems are the most common, in the following, the procedure for this particular pipeline is outlined.

As a first step, all camera-based systems today need to be calibrated. That is, the relative positions of the cameras, their orientations, and their internal parameters (distortion) have to be estimated. This is commonly achieved by either placing a three-dimensional calibration object with known dimensions inside the capture volume or by covering the volume with a calibration wand, a typically one-dimensional object with known dimensions and marker positions. The advantage of wand-based calibration is that it is easier to cover the entire capture volume using only a small object. Covering the capture volume as exhaustively as possible is important to ensure accuracy of the calibration. Calibration objects, in contrast, tend to be as large as possible for a similar reason. In either case, the parameters of the cameras are extracted using a variation of Structure from Motion.

When using marker-based systems, the next step is to extract marker positions in the video frames. Normally, centroids and extents of the markers are extracted. The extents can be used as a measure of

quality of the marker. For example, distant markers tend to be smaller and should be considered less reliable. For large systems, with dozens of cameras running at high framerates (>[100]Hz) and high resolution (>[4]Megapixel), doing this processing close to the cameras is essential because the required bandwidth for transmitting the entire frames to a central processing unit would be prohibitively expensive.

Subsequently, the 2D dots can be combined into 3D markers using the calibration matrices of the cameras. For passive markers, this step is ambiguous. So additional heuristics have to be taken into account. For example, dots can be tracked in 2D or 3D to propagate the identity of a marker, established in a previous frame to the current frame. 3D markers should also be consistent with as many of the 2D dots as possible. Finally, skeleton fitting can be posed as a nonlinear minimization problem

$$\underset{\xi}{\operatorname{argmin}} \sum_{i=1}^{M} (m_i - s_i(\xi))^2, \tag{1}$$

where m_i is an estimated 3D marker position and $s_i(\xi)$ is the corresponding marker attached to the skeleton, as a function of the pose parameters ξ. This problem can be solved in many different ways. Commonly, gradient descent, Gauss-Newton, or Levenberg-Marquardt optimization is used. These approaches make use of the tracking assumption, i.e., the solution of the previous frame is used as a starting point for the current frame. Other methods, like particle filters or simulated annealing, are less likely to lose track by getting stuck in a local minimum but are computationally more expensive.

Alternatively, the 3D marker reconstruction step can be skipped. Instead, the nonlinear optimization is performed in image space of the cameras. That is, the distances between the projections of the skeleton's markers and the 2D markers are minimized rather than their distance in world space.

Markerless systems have to solve a very similar optimization problem. Only their method of acquiring correspondences between 2D image features and the tracked object is more sophisticated. Frequently, different characteristics are combined. Common features include edges, silhouettes, texture statistics, and corners.

Application

Historically, the first applications of motion analysis can be found in biomedicine. Gait analysis is used to assess pathological conditions of patients and to plan treatment such as orthopedic surgery and for follow-up monitoring. Similarly the methodology is applied in sports science, where the motion of athletes is optimized or monitored and automatically evaluated during endurance exercises.

More recently, motion analysis and motion capture have been applied in the entertainment industry. In its earliest form this involved a process called rotoscoping, i.e., an artist traced the outline of an actor in every frame of a reference video to create 2D animations of a subject. Nowadays, marker-based systems are frequently used to capture motions for use in both computer games and motion picture productions because creating animations by hand with the fidelity required in this industry is a very time-consuming task. Most commercial marker-based systems are still limited to controlled studio environments. Although recently, systems using active markers have been proposed that can be used outdoors or in onset conditions.

Open Problems

Although the main challenges of marker-based motion capture are generally considered solved, there is still room for improvement. Directors ask for ever more actors to be tracked simultaneously in real time; systems that work in outdoor settings are still only available using active markers or do not support real-time feedback. Retroreflective markers cannot be used outdoors because interference with other light sources prevents detecting markers reliably. Vision-oriented systems may be able to solve these issues and in most cases make less assumptions about the scene. Many approaches use no markers and significantly fewer cameras. In some cases, even moving backgrounds can be handled, or cameras are not assumed to be static. However, vision-based systems tend to be computationally expensive and less robust. Overall, combining the general capabilities of vision systems with the speed and robustness of marker-based approaches would significantly advance the state-of-the-art.

References

1. Sutherland DH (2002) The evolution of clinical gait analysis: part II kinematics. Gait Posture 16(2):159–179
2. Zhou H, Hu H (2008) Human motion tracking for rehabilitation – a survey. Biomed Signal Process Control 3:1–18
3. Moeslund TB, Hilton A, Krüger V (2006) A survey of advances in vision-based human motion capture and analysis. Comput Vis Image Underst 104(2):90–126
4. Menache A (1999) Understanding motion capture for computer animation and video games. Morgan Kaufmann, San Diego

Motion Capturing

▸Motion Capture

Motion Deblurring

▸Blind Deconvolution

Motion Tracking

▸Motion Capture

Multi-baseline Stereo

David Gallup
Google Inc., Seattle, WA, USA

Definition

Multi-baseline stereo is any number of techniques for computing depth maps from several, typically many, photographs of a scene with known camera parameters.

Background

The goal of any stereo algorithm is to reconstruct the 3D surface geometry of a scene from multiple photographs. Multi-baseline stereo can be seen as a generalization of binocular stereo, and it is one instance of a broader class of multi-view stereo algorithms. The classic binocular stereo problem focuses on using two views of a scene (the minimal case), whereas multi-baseline stereo uses more than two and typically many more views of the scene. More views not only provide a better signal to noise ratio but also eliminate most repetitive structure errors and offer new ways to handle occlusions.

Another type of multi-view stereo is volumetric stereo, which explicitly models the scene's surface in a volume, and is sometimes called object-based. Multi-baseline stereo on the other hand is image-based, and seeks to reconstruct the scene by assigning depth values to the pixels of one or more of the input images. Often this leads to better sampling of the input data and greater memory efficiency. The disadvantage is that special care must be taken at depth discontinuities.

Theory

Multi-baseline stereo shares the same theoretical concepts as other stereo problems. The inputs consist of a set of n input images with camera parameters. Camera parameters define a projection function which projects a 3D point into a pixel in the image. The goal is to compute a depth map for one or more of the input images. For simplicity we will focus on computing a depth map for a single view called the reference view.

Computing a depth map from images can be viewed as a maximum a posteriori estimation problem. Given images $I_1, \ldots, I_n$, compute the depth map Z that maximizes $P(Z|I_1, \ldots, I_n) = P(I_1, \ldots, I_n|Z)P(Z)$. The likelihood $P(I_1, \ldots, I_n|Z)$ describes how well the depth map fits the input data, and the prior $P(Z)$ describes desired properties of the depth map such as smoothness. This formulation can be expressed as an energy function:

$$E(Z) = \sum_{p \in Z} E_{\text{data}}(Z(p)) + \sum_{(p,q) \in \mathcal{N}} E_{\text{smooth}}(Z(p), Z(q)). \quad (1)$$

The data term E_{data} measures how well the depth value $Z(p)$ matches the input images. These *matching*

scores can be computed by comparing the intensity value in the reference view $I_{\text{ref}}(p)$ to the intensity values of the projections of $Z(p)$ in the matching views, $I_k(proj_k(Z(p)))$. Some views may be occluded, meaning that $Z(p)$ may not be visible from that view and $I_k(proj_k(Z(p)))$ will not match $I_{\text{ref}}(p)$. Occlusion handling aims to remove the influence of these occluded views and is a critical part of stereo. The smoothness term E_{smooth} penalizes variations between neighboring depth values (given by the set $\mathcal{N}$).

Matching Scores

Consider a known 3D point X on the surface of the scene as shown in Fig. 1. Let $x_0, \ldots, x_k$ be the projections of X into each image. The image intensities at these points should be *photo-consistent* (have the same appearance) since they are all images of the same point on the surface. This will not necessarily be true for 3D points off the surface of the scene. Brightness constancy is the assumption that the intensity of light does not change from viewpoint to viewpoint, a property of Lambertian surfaces. This can be measured by taking the absolute difference or squared difference between the intensity value in the reference view and the intensity value in matching view. Typically a single pixel does not carry enough information to make an informative matching score, and so often the differences between patches of nearby pixels are also incorporated into the matching score. This yields the sum of absolute differences (SAD) and sum of squared differences (SSD) scores.

The brightness constancy assumption can be violated for many reasons, for example, different exposure settings between images and specular surfaces that reflect light at different intensities depending on the angle. To account for some of these changes, patches can be normalized by removing the mean intensity of the patch and scaling the values so that variance is 1. This yields the normalized cross-correlation score (NCC). Let M and N be two rectangular image patches of size $w \times h$. These patches can be flattened to form vectors $\mathbf{m}$ and $\mathbf{n}$ of size $w \cdot h$. The NCC score is then

$$NCC(M, N) = \frac{(\mathbf{m} - \bar{\mathbf{m}}) \cdot (\mathbf{n} - \bar{\mathbf{n}})}{var(\mathbf{m}) var(\mathbf{n})}, \qquad (2)$$

where $\bar{\mathbf{m}}$ is the mean of $\mathbf{m}$ and $var(\mathbf{m})$ is its variance.

When patches originate from highly slanted surfaces, they may need to be corrected before being correlated. The surface's tangent plane can be defined given the point's surface normal, and images can be aligned by projecting them onto this plane. This transformation can be expressed as a homography. This adds two additional angle parameters per pixel to the problem. One method to account for the surface normal is to compute the matching score as the best score over all surface normals [1]. A much faster alternative is to consider only a small number of likely candidates [2]. Another is to iteratively estimate a depth map, using surface normals given by the depth map from the previous iteration.

Summing the SAD, SSD, or NCC scores from multiple views reduces the influence of noise as well as disambiguates mismatches due to repetitive structures since it is less likely that a mismatch will occur in all views simultaneously [3].

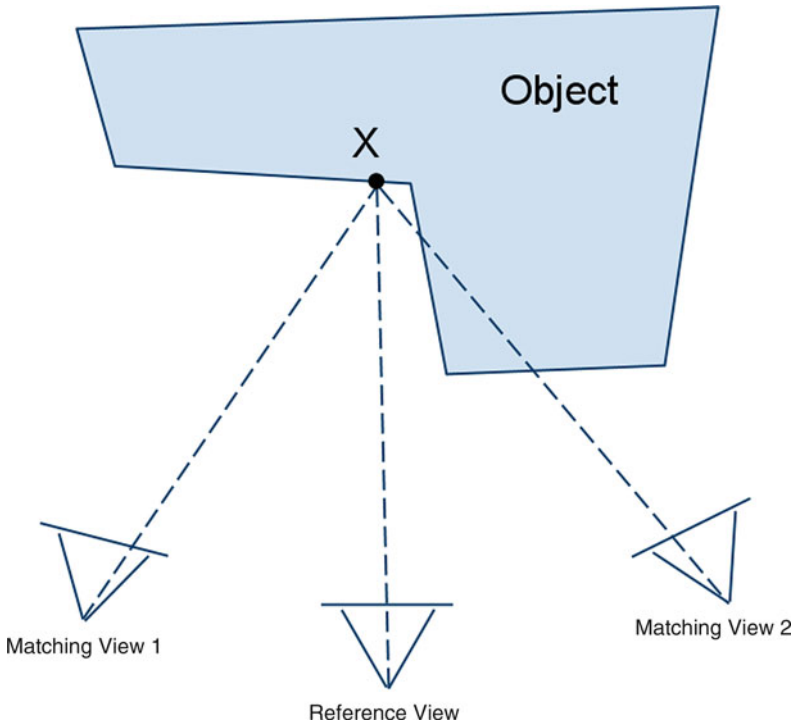
Occlusion Handling

Summing scores from multiple views treats all images equally. In fact some images may be occluded, meaning a different surface is seen from that viewpoint, and the matching score is arbitrarily bad. Handling these cases is important for stereo, and multi-baseline stereo has advantages in this regard. One method based on robust statistics is to assume that at least k views are unoccluded and to discard the rest. The sum of the best k views can be obtained by sorting and summing. Another method is to assume an object will be occluded from one side and not another. Assuming cameras are arranged in a line, the unoccluded half-set will be either the left or right half-set of cameras [4]. See Fig. 2.

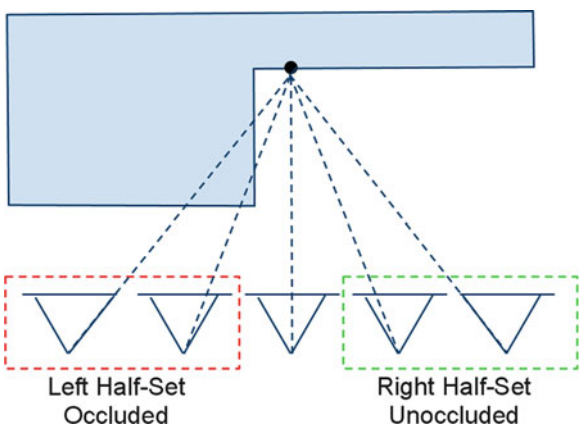
Other methods seek to detect occlusions explicitly by using the reconstruction itself to identify occluded views. This is a chicken and egg problem: the occlusions must be known to reconstruct the scene, and the reconstruction of the scene must be known to detect the occlusions. Some methods model occlusions probabilistically [5], and others start with a safe configuration and update the reconstruction convervatively [6].

Optimization

Optimizing Eq. 1 is difficult due to the non-convex data and smoothness terms. Some methods ignore the



Multi-baseline Stereo, Fig. 1 A surface point X is projected into the images. Matching view 2 is occluded and should not contribute to the matching score



Multi-baseline Stereo, Fig. 2 Simple occlusion geometry. Typically, either the left half-set or the right half-set will be free of occlusion

smoothness term, and each pixel is optimized independently by exhaustive search. The Z function is discretized into points along viewing rays that project to individual pixels in the matching views. Testing all points for each pixel can be done efficiently using graphics hardware which is well suited for this type of *embarrassingly parallel* computation [12]. Methods that do use the smoothness term obtain better results, and if the benefits of normalization (NCC) are not needed, patches need not be used and single pixels can be used. The ambiguity of single pixel matching is resolved because the smoothness term regularizes the solution. The optimization problem is *NP-hard*, but there are effective approximation algorithms based on graph cuts [8] and belief propagation [9].

Depth Resolution

The resolution of stereo as a depth sensor depends on the distance of the surface from the cameras. The depth resolution is the distance between pixels in a matching view projected onto a viewing ray in the reference view. In the binocular case, the resolution is

$$\Delta z = \frac{z^2}{bf}, \tag{3}$$

where Δz is the resolution, z is the distance to the reference view, b is the baseline or distance between camera centers, and f is the focal length of the cameras. Depth measurement uncertainty depends on the matching uncertainty which describes how precisely the two observed patterns can be registered and depends on

factors like texture and image noise. Depth measurement uncertainty is proportional to depth resolution. With enough views, multi-baseline stereo has the advantage that the baseline can be treated as a variable rather than a constant. This gives greater control over the depth resolution and computation time of the stereo algorithm [10].

Application

Multi-baseline stereo is applied to many 3D reconstruction problems such as 3D city modeling [11], view synthesis [12], and digital archiving. Multi-baseline stereo is often used in real-time applications where high-quality depth can be computed from a large number of views without the need for computationally intensive optimization techniques. This is especially true for video applications, where the camera moves in a fairly linear manner, and so more general multi-view stereo techniques are unnecessary.

References

1. Zabulis X, Daniilidis K (2004) Multi-camera reconstruction based on surface normal estimation and best viewpoint selection. In: Proceedings of the 2nd international symposium on 3D data processing, visualization and transmission (3DPVT), pp 733–740
2. Gallup D, Frahm J-M, Mordohai P, Yang Q, Pollefeys M (2004) Real-time plane-sweeping stereo with multiple sweeping directions. In: IEEE conference computer vision and pattern recognition (CVPR), Minneapolis
3. Okutomi M, Kanade T (2002) A multiple-baseline stereo. In: IEEE Trans Pattern Anal Mach Intell, 15(4): 353–363
4. Kang S, Szeliski R, Chai J (2001) Handling occlusions in dense multi-view stereo. In: IEEE conference computer vision and pattern recognition (CVPR), Kauai
5. Hernandez C, Vogiatzis G, Cipolla R (2001) Probabilistic visibility for multi-view stereo. In: IEEE conference computer vision and pattern recognition (CVPR), Minneapois
6. Kolmogorov V, Zabih R (2002) Multi-camera scene reconstruction via graph cuts. In: European conference on computer vision 2002 (ECCV), Copenhagen
7. Yang R, Pollefeys M (2003) Multi-resolution real-time stereo on commodity graphics hardware. In: IEEE conference on computer vision and pattern recognition (CVPR), Madison
8. Boykov Y, Veksler O, Zabih R (2001) Fast approximate energy minimization via graph cuts. In: IEEE Trans Pattern Anal Mach Intell, 23(11): 1222–1239
9. Felzenszwalb P, Huttenlocher D (2001) Efficient belief propagation for early vision. In: IEEE conference on computer vision and pattern recognition (CVPR), Kauai
10. Gallup D, Frahm J-M, Mordohai P, Pollefeys M (2008) Variable baseline/resolution stereo. In: IEEE conference on computer vision and pattern recognition (CVPR), Anchorage
11. Pollefeys M, Nister D, Frahm J-M, Akbarzadeh A, Mordohai P, Clipp B, Engels C, Gallup D, Kim S-J, Merrell P, Salmi C, Sinha S, Talton B, Wang L, Yang Q, Stewenius H, Yang R, Welch G, Towles H (2008) Detailed real-time urban 3D reconstruction from video. Int J Comput Vision, 78(2–3): 143–167
12. Yang R, Pollefeys M, Yang H, Welch G (2003) A unified approach to real-time, multi-resolution, multi-baseline 2D view synthesis and 3D depth estimation using commodity graphics hardware. Int J Image Graph, 4(4): 627–651

Multi-camera Calibration

▸Calibration of Multi-camera Setups

Multi-camera Human Action Recognition

Gaurav Srivastava[1], Johnny Park[1], Avinash C. Kak[1], Birgi Tamersoy[2] and J. K. Aggarwal[3]
[1]School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA
[2]Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX, USA
[3]Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX, USA

Synonyms

Activity analysis; Behavior understanding

Related Concepts

▸Gesture Recognition; ▸Human Pose Estimation

Definition

Multi-camera human action recognition deals with using multiple cameras to capture several views of humans engaged in various activities and then combining the information gleaned from the cameras for the classification of those activities.

Background

Research on human activity recognition gathered momentum in the mid- to late 1990s; much early work is summarized in a review by Aggarwal and Cai [1]. There emerged two dominant approaches during this period: (1) state-space modeling of human actions [2, 3]; and (2) template matching [4, 5]. The focus during that early phase of this research was primarily on recognizing human activities on the basis of the images collected by a single camera. While this is still an active research area in computer vision (see Aggarwal and Ryoo [6] for a survey), it unfortunately suffers from several serious shortcomings, many of them owing to the limitations inherent to images that are recorded from just one viewpoint. Human activities, in general, are much too complex in 3D to be described by cues extracted from single-viewpoint 2D projections. While it is true that the human eye (even just a single eye) can do a wonderful job of categorizing human activities, trying to replicate that in a computer would be far too ambitious a research project for a long time to come. It is not yet fully understood how the human brain fills in the information that it cannot see directly in order to recognize objects and movements despite severe occlusion and noise. While it is a proper exercise in humility to be awed by the capabilities of the human brain, it is nonetheless good to keep in mind that even a human can be fooled in its perception of an activity when the perception is limited to a single viewpoint. Magicians frequently take advantage of such limitations of human perception in order to produce their magical effects.

In addition to the problems caused by the fact that a single camera provides only single-viewpoint 2D projections of the scene, other reasons for the more recent interest in multi-camera approaches to activity recognition stem from the current global interest in wide-area surveillance, on the one hand, and in the design of intelligent environments for the living spaces of the future on the other. In both of these applications, the goal is to characterize a human activity as it is evolving with time and as it is occupying space that may not be limited to the coverage provided by a single camera. Consider a habitat of the future for the aged and the infirm where you may wish to use a network of cameras that silently watch for any undesirable human behavior, such as someone suddenly collapsing on the floor or tripping over a piece of furniture. To recognize such human activities, the camera system would need to analyze the sensed data over a period of time and, even more particularly, over some a span of physical space. Multi-camera imagery would obviously lend itself much better to the sort of data analysis that would need to be carried out for the required inferences.

With a view to gaining insight into the various aspects of multi-camera human action recognition, in the remainder of this chapter, section "Theory" describes the theoretical details of the different types of approaches that have been proposed by researchers to accomplish multi-camera action recognition. Owing to this chapter's intent of providing accessibility to the general readers, significant details about the algorithms are not discussed; rather, only the distinguishing highlights of the different approaches are presented. Section "Application" presents a discussion on some of the application areas that will benefit from multi-camera action recognition as compared to the single camera modality. Since human action recognition is a complex and challenging problem, there are quite a few open problems that need to be addressed before this research becomes useful for mainstream society. Section "Open Problems" enumerates some of these open problems related to human action recognition in general and also those that are specific to the area of multi-camera action recognition. Finally, in section "Experimental Results," this chapter is concluded with a performance comparison between the different multi-camera approaches on a common benchmark dataset.

Theory

The first step in human action recognition involves creating a library of models for different human actions to be recognized. An action model characterizes the unique motion patterns associated with an action. These models can be created from a temporal sequence of either 2D images or 3D reconstructions obtained by combining multiple camera images of a human actor. Researchers have proposed different approaches to creating the action models. For example, an action model may comprise of a set of local motion features extracted from the spatio-temporal volume of the action image sequence. Another example of an action model is a set of exemplar human poses represented

either as 2D silhouettes or 3D reconstructions. Yet another example is a set of spatiotemporal trajectories of different human body parts. Once the action models are created, new instances of human actions can be recognized. Given a test image sequence that contains an unknown human action, the same technique used for creating the action models is applied to the test sequence to generate its action representation. This action representation is then compared against each action model in the library. Finally, the test sequence is assigned the label of the most similar action model. In terms of the underlying operating principle, there is no difference between human action recognition and any other type of object recognition: A test object whose category label is to be ascertained is assigned the label of the closest matching model whose category is known.

A common assumption in single camera action recognition is that the test action and the model actions have been captured from identical or very similar camera viewpoints. The quality of match between the test action and the model actions is a function of the conformity between their camera views. As Souvenir and Babbs [7] point out, it would be impractical for any human motion analysis system to impose the constraint that humans engaged in an activity are facing the same direction relative to the camera view at all times. In order to alleviate this problem, several methods have been developed which use multiple cameras in the training and/or testing phases of action recognition. These methods provide viewpoint invariance, i.e., the ability to perform matching between a pair of action observations even if they have been acquired from different camera views. A good overview of multiple camera action recognition approaches can be found in [8]. Based on the distinct fundamental ideas of these methods, they have been categorized into the following three classes:

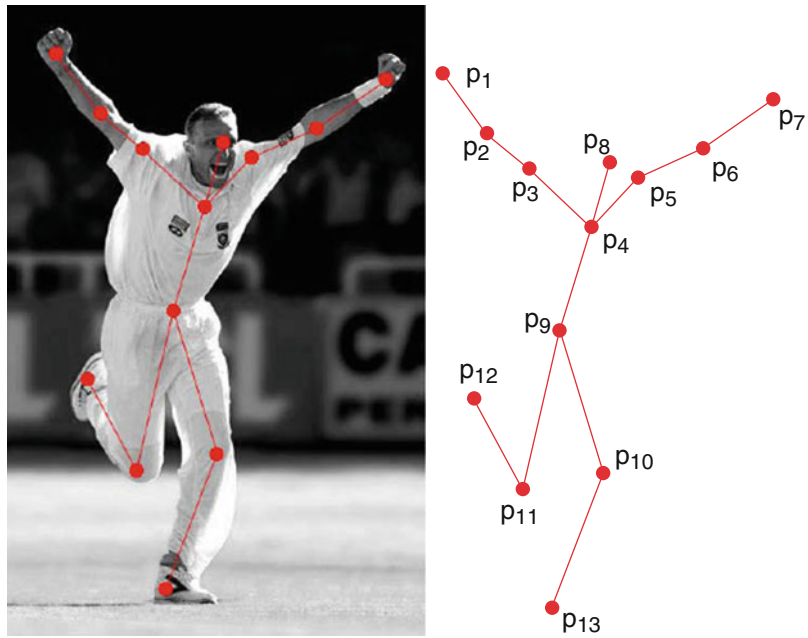
1. Multi-view geometry-based methods
2. View-invariant representation-based methods
3. Exhaustive search methods

Multi-view Geometry-Based Methods

The multi-view geometry-based approaches utilize the epipolar geometric constraints between multiple cameras for human action recognition. The intuition is that if two actors perform the same action, then assuming the same temporal rate of action execution, their postures at all corresponding time instants are related by epipolar geometric relationships. Such relationships between the two views are applied to point correspondences where the points are generally chosen as the anatomical landmarks on the human body, e.g., head, shoulders, hands, and feet (see Fig. 1). Action recognition is performed by measuring the similarity between the postures of a test action sequence and a model action sequence at every time instant. The similarity measure can be expressed in terms of the point correspondences and a matrix F known as the fundamental matrix that is computed using epipolar geometry (Fig. 2). Given at least eight pairs of point correspondences (x_i, x_i'), the fundamental matrix F satisfies the relation $x_i^T F x_i' = 0$, $i = 1, \ldots, n \geq 8$. In practical settings, the point correspondences between the action representations in two different views will generally not be precise, and hence, the quantity $x_i^T F x_i'$ will not be exactly zero. Nevertheless, the residual $\sum_i \left| x_i^T F x_i' \right|^2$ can be used as the matching cost or the similarity measure. If the matching cost is below a certain threshold, then the point correspondences come from the same action represented in the different views. Generally, the test action postures and the action models are derived from different persons with different body sizes and proportions; therefore, certain anthropometric constraints may also need to be imposed to normalize the landmark points to a common coordinate frame before the epipolar geometric constraints can be applied. Such and other similar constraints on the point correspondences have been used for matching of different action instances in [9, 11–13].

View-Invariant Representation-Based Methods

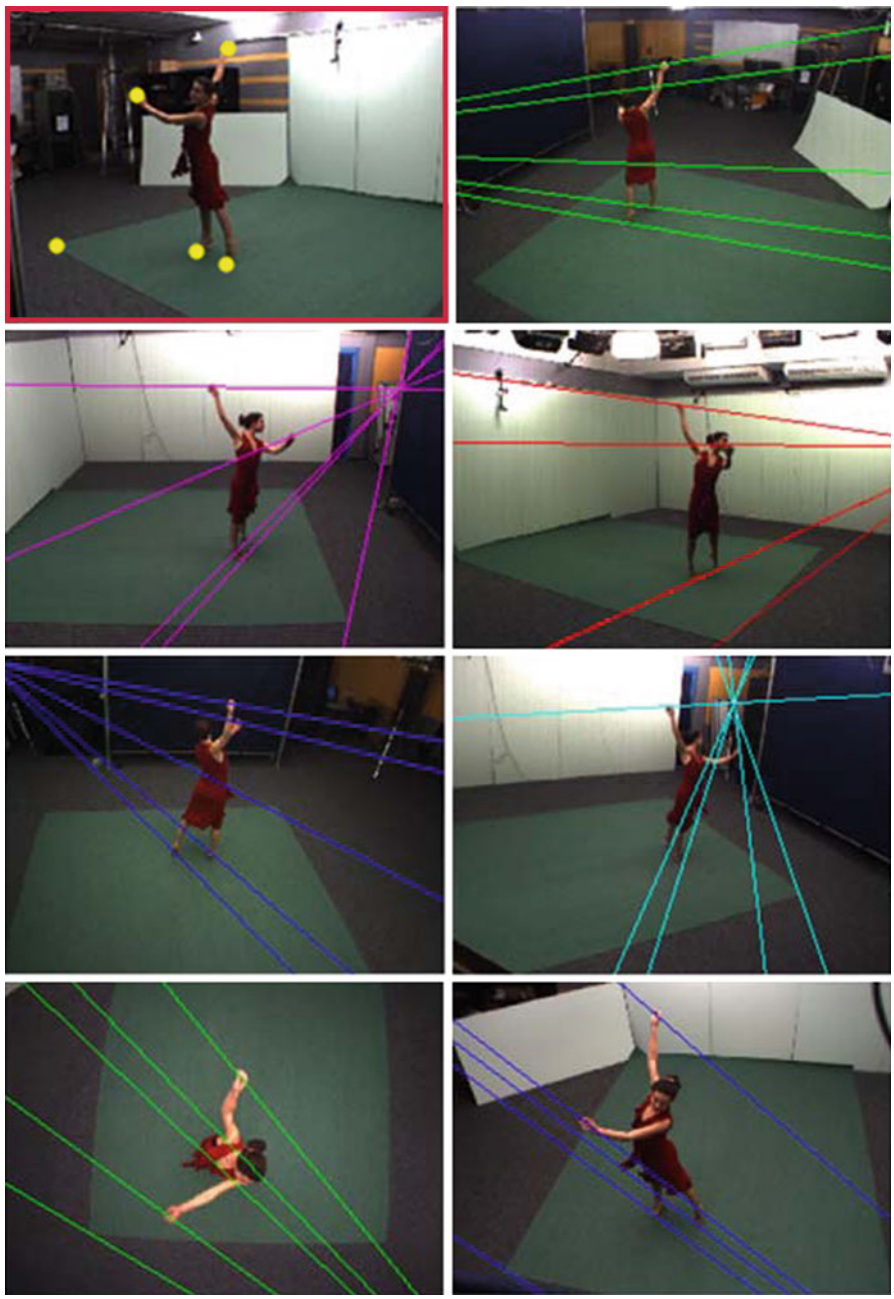
In addition to the multi-view geometry-based approaches, there are other interesting methods that accomplish viewpoint invariant action recognition based on machine learning techniques. It is a conceivable scenario that a discriminative model of an action is available for one camera viewpoint (source view), and the action recognition needs to be performed in another view (target view) for which such a model is not available. This issue has been addressed in [14], where a transfer learning approach is used along with examples of corresponding observations of actions from both views in order to learn how the appearance of an action changes with the change of viewpoint.



Multi-camera Human Action Recognition, Fig. 1 The posture of a human actor at a specific time instant. It is represented as a set of anatomical land marks (Gritai et al. [9], ©2004 IEEE)

It is worth noting that such learning can be performed with one set of actions and then applied to a new unknown action for which the transfer model is not explicitly built. Transfer learning is applied by splitting the source domain feature space using the action descriptors and the action labels to produce certain split-based features. These split-based features are considered transferable between the source view and the target view and therefore are used to construct an action-discriminative split of the target domain feature space. For a test action sequence in the target view, its action descriptors are extracted, and action recognition is performed by a nearest neighbor matching with the action descriptors of the model actions which were transferred from the source view.

In [7], the variation in the appearance of an action with viewpoint changes is estimated by learning low-dimensional representations of the actions using manifold learning. The action descriptor used is the $\mathcal{R}$ transform surface which is a temporal extension of the well-known Radon transform (Fig. 3). In the figure, the horizontal axes correspond to time t and the polar angle θ used in Radon transform computations. The $\mathcal{R}$ transform surface is a high-dimensional data that lies on a non-linear manifold, and hence it can be embedded into a lower dimensional space. In this low-dimensional space, learning how the data varies as a function of the viewpoint provides a representation which allows to avoid storing action examples from all possible viewpoints. Action recognition is performed by obtaining a similarity measure between two $\mathcal{R}$ transform surfaces S_1 and S_2, such as the L^2 distance $\|S_1 - S_2\|$. Another interesting approach has been described in [15], where the temporal self-similarity between the frames of an action sequence was shown to be highly stable with a changing viewpoint (Fig. 4). Specifically, for the same action sequence recorded from very different views, the so-called temporal self-similarity matrices (SSMs) corresponding to the different views were shown to be very similar. This observation was consistent even when different image features were used for computing the self-similarity matrix. By constructing histogram-based descriptors from the elements of the SSM (as described in [15], Sect. IV), action recognition can be performed by applying classifiers like nearest-neighbor or support vector machine on these descriptors. Recently, Kusakunniran et al. [16] proposed a view transformation model based on support vector regression for solving the multi-view gait recognition problem. This view transformation model uses local regions of interest (ROIs) in one view to predict the motion information of the corresponding regions in a different view. In order to perform gait recognition, the gait features from the different actors'

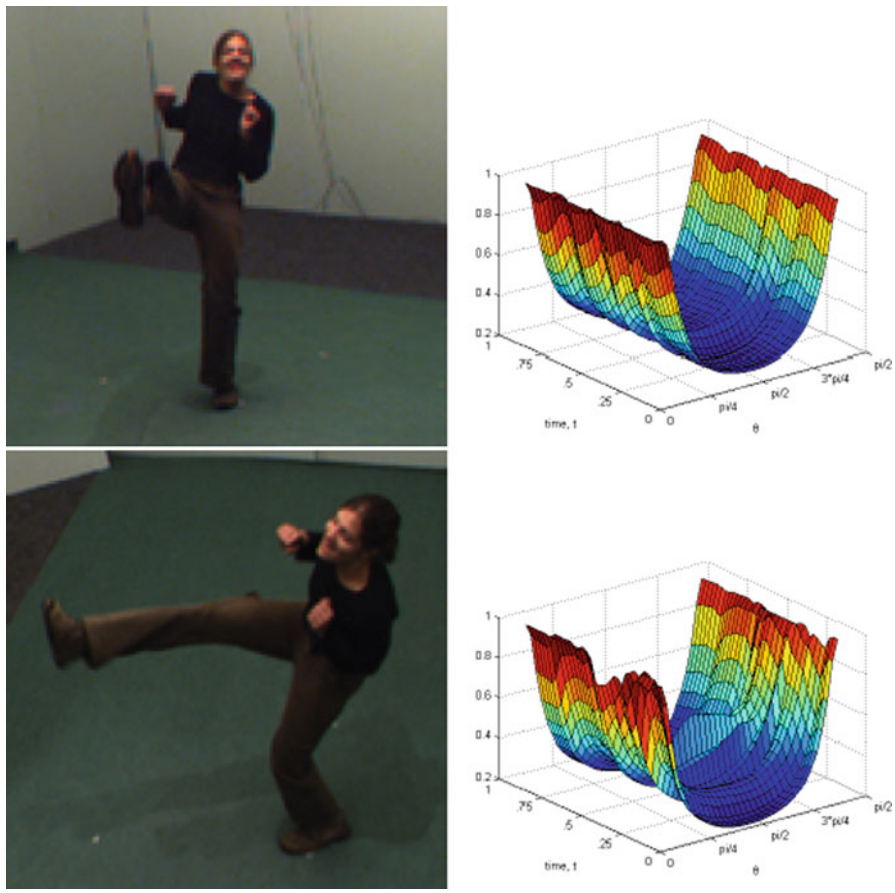


Multi-camera Human Action Recognition, Fig. 2 Landmark points (*yellow*) in one view. The other views show the epipolar lines which contain the points corresponding to the landmark points (Sinha and Pollefeys [10], ©2009 Springer)

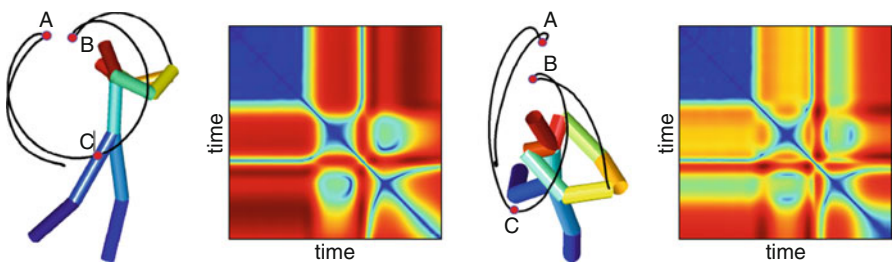
gait sequences and possibly different view angles are first normalized to a common viewing angle using the view transformation model followed by a similarity measure calculation between the normalized gait features using the Euclidean metric.

Exhaustive Search Methods

The third popular approach in the multi-camera human action recognition is to perform an exhaustive search in the space of multi-view action poses to find the best match for the test action poses. Such an exhaustive

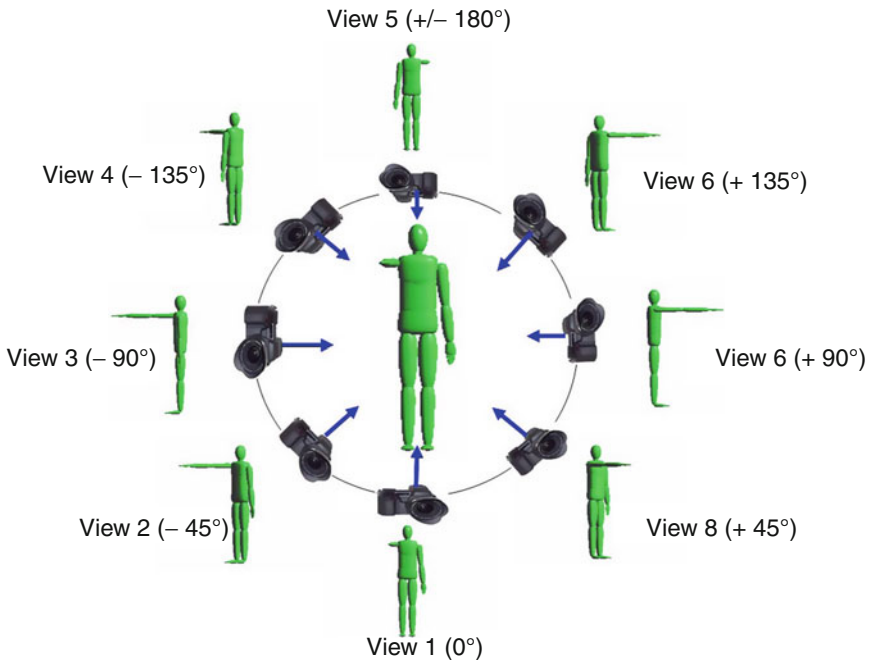


Multi-camera Human Action Recognition, Fig. 3 Kicking action from two different view points and their corresponding $\mathcal{R}$ *transform surface* action descriptors. They are quite similar despite view point variation (Souvenir and Babbs [7], ©2008 IEEE)



Multi-camera Human Action Recognition, Fig. 4 A golf swing action seen from two different views and their corresponding temporal self-similarity matrices (Junejo et al. [15], ©2011 IEEE)

search can be performed in 2D or 3D. During the training time, a set of multiple fixed cameras installed around the actor is used to record the multi-view sequences of his or her actions (Fig. 5). In the 2D exhaustive search approach, an observation is recorded for the unknown action and matched against each recorded view from the training session. The matching can be performed using the shape features derived from the 2D silhouettes, the motion features obtained from the optical flow, or histograms of local



Multi-camera Human Action Recognition, Fig. 5 Acquiring simultaneous multiview action sequences for training (Ahmad and Lee [17], ©2006 IEEE)

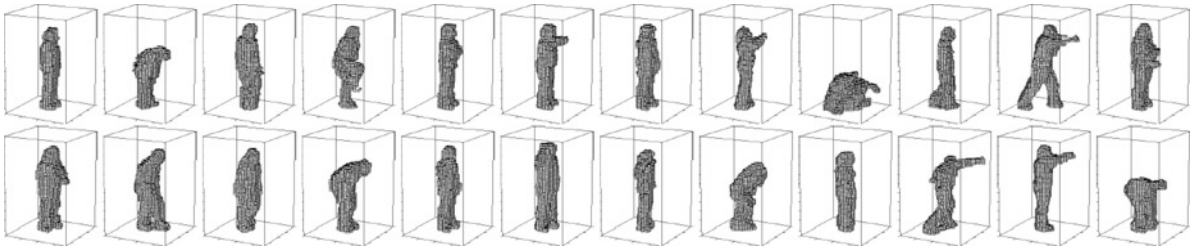
spatiotemporal cuboid features. In order to accomplish action recognition, this 2D search is performed at every time instant of the action sequence, and the model action resulting in the smallest feature distance over all the time instants is used to label the unknown action. A limitation of the 2D approach is that the same spatial configuration of cameras has to be used during the training and testing sessions. Generally, during the test time, the observations are recorded from a single view, but more than one view can also be used with the appropriate changes in the matching algorithm. The 2D approach has been used by several research groups [17–19].

The 3D exhaustive search approach provides more flexibility with regard to the placement of the cameras in the monitoring environment. Different spatial configurations of cameras can be used during the training and testing sessions. Here, instead of storing all the discrete views of the action during the training time, they are combined to produce an action model based on 3D reconstruction (Fig. 6). The key advantage of such a strategy is that there is no restriction on what camera view(s) can be used to record the test action sequence. If the camera parameters of the arbitrary camera view used during the testing session are known, then the model 3D action representations can be projected into that 2D view for matching with the observation. Some examples of such an approach are [20, 21].

Application

Vision-based human activity recognition has diverse applications. Generally, any practical application related to monitoring human activities requires that the monitoring can be done over an extended physical space beyond the viewing area of a single camera. It may also be required to monitor an event from several viewpoints simultaneously so as to obtain richer descriptions of complex human activities. Such requirements necessitate the use of multiple cameras for capturing the events. In this section, some of the application areas are briefly discussed where multi-camera human activity recognition or multi-camera event detection is currently being used or has strong potential for use in the near future.

1. *Wide area surveillance* – Facilities like government buildings, military installations, airports, subways, power plants, dams, and so on require round-the-clock surveillance. Some examples of the general human-related activities that need to be monitored are wide-area perimeter breaches by intruders, a person approaching the doors after hours, leaving of a suspicious unattended object by a person, extended loitering around the facility perimeter, and persons tampering with the facility security systems. In such scenarios, it is necessary to study not only the isolated activities of the persons but also



Multi-camera Human Action Recognition, Fig. 6 An action model consisting of 3D exemplars that were selected based on 11 types of actions by 10 human actors [20]. Given a test sequence of 2D action silhouettes, action recognition involves finding the best matching exemplar sequence and the best matching 2D projections (©2007 IEEE)

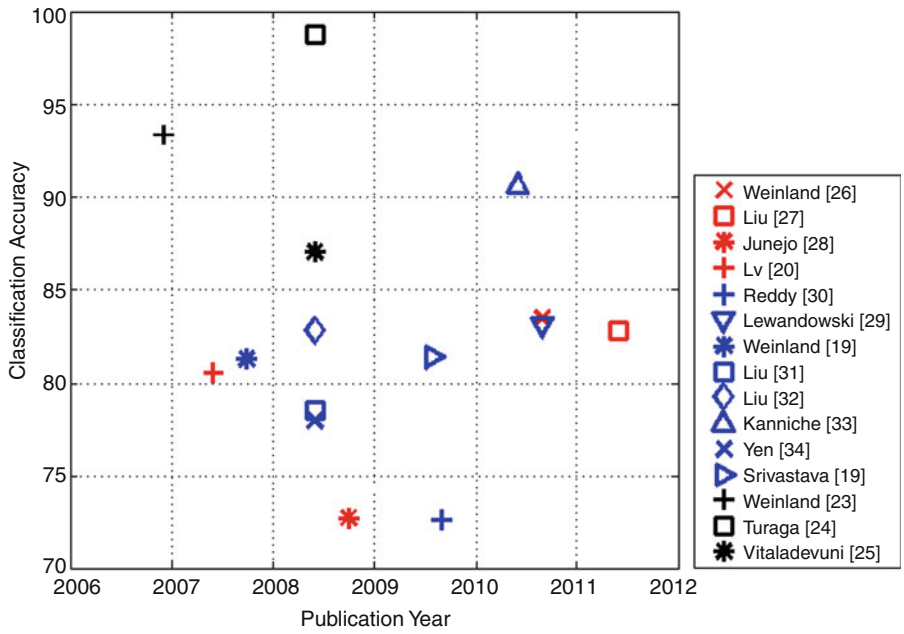
the patterns of their interactions with their surroundings, such as who are other persons they interact with, are they carrying any objects, how long have they been present or are they loitering around the security systems.

2. *Monitoring the elderly and children* – Assisted living homes and day care centers require constant monitoring of the activities of the elderly and children to avoid incidents such as an elderly person tripping over a furniture and falling down or a child playing too close to hazardous areas like a swimming pool or an electrical appliance. Commercial applications are available for day care centers that allow the installation of multiple wireless surveillance cameras in play areas, resting areas, or dining areas, where the video and audio feed from the cameras can be transmitted to a central computer connected to the Internet. Parents can thus check on the safety of their children at any time by viewing over the Internet.
3. *Three-dimensional human motion analysis for sports and medicine* – A 3D motion-based system can be used to perform a marker-based or markerless capture of the human motion while performing ordinary activities like walking or bending or sports activities like a golf swing, swimming strokes, tennis serves, gymnastics, and so on. In the field of medicine, such human motion data is used to characterize the dynamics of activities, discover the fundamental principles that govern movement, and understand the causes of movement disorders. In the area of sports, the motion data of the athlete can be analyzed by the coach for recommending changes in the dynamics to achieve more energy-efficient and agile performance, or it may be compared with the 3D reference motion of another expert athlete.
4. *Enhanced sports viewing experience* – Multiple cameras are routinely used in sports like football, cricket, and soccer to capture many views of a dynamic event for providing an enriched fly through experience of the scene to the viewers or for use by the referees when it is difficult to make judgments based on a single camera view.

Open Problems

In the last 15 years, substantial progress has been made in the field of human action recognition. The accomplishments have primarily been in the area of single-person action recognition in an uncluttered background and recognizing a set of simple activities like walking, jumping, waving hands, punching, and so on. This is evidenced by the popularity of datasets like KTH, Weizmann, and IXMAS for benchmarking the action recognition algorithms (please see [8], Sect. 6 for description of these datasets). These datasets capture simple actions performed by a single human, with a clean background, and negligible variations in spatial scale of the person or temporal speed of execution. The challenge for the research community, in most simple words, is to extend the current recognition algorithms to work on datasets comprising of video sequences captured in unconstrained settings such as the Hollywood movie dataset [22] or the YouTube video dataset [23].

Specifically in the area of multi-camera action recognition, an open problem is to obtain nonrigid point correspondences on the human body that are needed for applying the geometric constraints. The main underlying difficulty is the reliable detection and tracking of human body parts in an unconstrained visual setting. Similarly, a limitation on the view



Multi-camera Human Action Recognition, Fig. 7 Action recognition performance of different algorithms on IXMAS dataset

invariant representations like self-similarity matrices [15] and $\mathcal{R}$ transform surface [7] is that they are constructed from the temporal variation of the features; hence, they need the full video sequences to be available offline. Real-time applications like video surveillance and human-computer interaction will benefit from the development of view-invariant features that can be computed online.

Experimental Results

This chapter is concluded with a chronological summarization of the different algorithms' action recognition performances on a benchmark dataset named the INRIA Xmas Motion Acquisition Sequences (IXMAS) [24].

The IXMAS dataset is the most commonly used dataset for evaluating the multi-camera action recognition algorithms. It contains 13 daily-life actions, such as *check watch*, *cross arms*, and *scratch head*. Each action is performed three times by 11 actors. The actions are captured using five calibrated and synchronized cameras. The actors are free to perform the actions in any orientation, making this a fairly challenging dataset.

Figure 7 presents the reported average recognition accuracies of several recent works on multi-camera action recognition. It is important to note that not all the approaches use the same evaluation methodologies, and hence, it is difficult to compare them merely based on these values.

For easier comparison, these works are categorized into three groups based on their evaluation methodologies: (1) the methods, which use 3D representations in the recognition stage [24–26], (2) the methods, which report camera-specific recognition accuracies [21, 27–29], and (3) the methods, which incorporate results from multiple cameras (e.g., using simple voting) [19, 20, 30–35]. These groups are distinguished by black, red, and blue markers in Fig. 7, respectively.

References

1. Aggarwal JK, Cai Q (1999) Human motion analysis: a review. Comput Vis Image Underst 73(3):428–440
2. Bobick AF, Wilson AD (1995) A state-based technique for the summarization and recognition of gesture. In: Proceedings of the fifth international conference on computer vision, ICCV '95, Washington, DC. IEEE Computer Society, pp 382–389
3. Brand M, Oliver N, Pentland A (1997) Coupled hidden markov models for complex action recognition. In: IEEE

computer society conference on computer vision and pattern recognition (CVPR), Washington, 1997, p 994
4. Polana R, Nelson R (1994) Low level recognition of human motion (or how to get your man without finding his body parts). In: Proceedings of the IEEE workshop on motion of non-rigid and articulated objects, Austin, TX, USA, 1994, pp 77–82
5. Bobick A, Davis J (1996) Real-time recognition of activity using temporal templates. In: WACV '96., proceedings 3rd IEEE workshop on applications of computer vision, Sarasota, FL, USA, 1996, pp 39–42
6. Aggarwal JK, Ryoo MS (2011) Human activity analysis: a review. ACM Comput Surv 43(3):1–43
7. Souvenir R, Babbs J (2008) Learning the viewpoint manifold for action recognition. In: IEEE conference on computer vision and pattern recognition (CVPR), Anchorage, pp 1–7
8. Weinland D, Ronfard R, Boyer E (2010) A survey of vision-based methods for action representation, segmentation and recognition. Comput Vis Image Underst 115(2): 224–241
9. Gritai A, Sheikh Y, Shah M (2004) On the use of anthropometry in the invariant analysis of human actions. In: Proceedings of the 17th international conference on pattern recognition, Cambridge, 2004. ICPR 2004, vol 2, pp 923–926
10. Sinha SN, Pollefeys M (2009) Camera network calibration and synchronization from Â silhouettes in archived video. Int J Comput Vis 87(3):266–283
11. Syeda-Mahmood T, Vasilescu A, Sethi S (2002) Recognizing action events from multiple viewpoints. In: Proceedings of IEEE workshop on detection and recognition of events in Video, Vancouver, BC, Canada, 2001, pp 64–72
12. Rao C, Yilmaz A, Shah M (2002) View-invariant representation and recognition of actions. Int J Comput Vis 50(2):203–226
13. Parameswaran V, Chellappa R (2003) View invariants for human action recognition. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition (CVPR), Madison, 2003, vol 2, pp 613–19
14. Farhadi A, Tabrizi MK (2008) Learning to recognize activities from the wrong view point. In: Forsyth D, Torr P, Zisserman A (eds) Computer vision–ECCV, 2008. Springer, Berlin/Heidelberg, pp 154–166
15. Junejo IN, Dexter E, Laptev I, Pérez P (2011) View-independent action recognition from temporal self-similarities. IEEE Trans Pattern Anal Mach Intell 33(1):172–85
16. Kusakunniran W, Wu Q, Zhang J, Li H (2010) Support vector regression for multi-view gait recognition based on local motion feature selection. In: IEEE conference on computer vision and pattern recognition (CVPR), San Francisco, pp 974–981
17. Ahmad M, Lee SW (2006) HMM-based human action recognition using multiview image sequences. In: 18th international conference on pattern recognition, Hong Kong, 2006. ICPR 2006, vol 1, pp 263–266
18. Ogale A, Karapurkar A (2007) View-invariant modeling and recognition of human actions using grammars. In: Proceedings of workshop on dynamic vision, Beijing, China, pp 115–126
19. Srivastava G, Iwaki H, Park J, Kak AC (2009) Distributed and lightweight multi-camera human activity classification. In: 2009 third ACM/IEEE international conference on distributed smart cameras (ICDSC), Stanford, CA, USA, pp 1–8
20. Weinland D, Boyer E, Ronfard R (2007) Action recognition from arbitrary views using 3D exemplars. In: Computer vision, Rio de Janeiro, 2007. ICCV 2007, IEEE 11th International Conference, pp 1–7
21. Lv F, Nevatia R (2007) Single view human action recognition using key pose matching and viterbi path searching. In: IEEE conference on computer vision and pattern recognition, Minneapolis, 2007 (CVPR'07), pp 1–8
22. Laptev I, Marszalek M, Schmid C, Rozenfeld B (2008) Learning realistic human actions from movies. In: IEEE computer society conference on computer vision and pattern recognition (CVPR), Anchorage, pp 1–8
23. Liu J, Luo J, Shah M (2009) Recognizing realistic actions from videos in the wild. In: IEEE conference on computer vision and pattern recognition (CVPR), San Francisco, 1996–2003
24. Weinland D, Ronfard R, Boyer E (2006) Free viewpoint action recognition using motion history volumes. Comput Vis Image Underst 104(2–3):249–257
25. Turaga P, Veeraraghavan A, Chellappa R (2008) Statistical analysis on stiefel and grassmann manifolds with applications in computer vision. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Anchorage
26. Vitaladevuni SN, Kellokumpu V, Davis LS (2008) Action recognition using ballistic dynamics. In: IEEE conference on computer vision and pattern recognition (CVPR), Anchorage
27. Weinland D, Ozuysal M, Fua P (2010) Making action recognition robust to occlusions and viewpoint changes. In: Proceedings of the 11th European conference on computer vision (ECCV), Heraklion. Lecture motes in computer science
28. Liu J, Shah M, Kuipers B, Savarese S (2011) Cross-view action recognition via view knowledge transfer. In: IEEE conference on computer vision and pattern recognition (CVPR), Colorado Springs
29. Junejo I, Dexter E, Laptev I, Perez P (2008) Cross-view action recognition from temporal self-similarities. In: Proceedings of the 10th european conference on computer vision (ECCV), Marseille. ECCV'08
30. Lewandowski M, Makris D, Nebel JC (2010) View and style-independent action manifolds for human activity recognition. In: Proceedings of the 11th European conference on computer vision: part VI (ECCV'10). Springer, Berlin/Heidelberg, pp 547–560
31. Reddy K, Liu J, Shah M (2009) Incremental action recognition using feature-tree. In: Computer vision, 2009 IEEE 12th international conference, Kyoto, pp 1010–1017
32. Liu J, Ali S, Shah M (2008) Recognizing human actions using multiple features. In: IEEE conference on computer vision and pattern recognition (CVPR), Anchorage
33. Liu J, Shah M (2008) Learning human actions via information maximization. In: IEEE conference on computer vision and pattern recognition (CVPR), Anchorage

34. Kaaniche MB, Bremond F (2010) Gesture recognition by learning local motion signatures. In: IEEE conference on computer vision and pattern recognition (CVPR), San Francisco, pp 2745–2752
35. Yan P, Khan S, Shah M (2008) Learning 4d action feature models for arbitrary view action recognition. In: IEEE conference on computer vision and pattern recognition (CVPR), Anchorage

Multi-focus Images

Amit Agrawal
Mitsubishi Electric Research Laboratories, Cambridge, MA, USA

Synonyms

Focus bracketing

Definition

Multi-focus images are a set of images of the same scene focused at different depths in the scene.

Background

Conventional imaging systems have a finite depth of field (DOF), which depends on the aperture size and the focal length of the lens. The DOF is the depth range within which the scene points appear sharp in the captured image. For scene points within the DOF, the size of the defocus blur is smaller than the minimum acceptable circle of confusion. Often, a single photo is unable to capture the entire scene in sharp focus.

DOF can be increased by decreasing the aperture size (increasing the F-number). However, reducing the aperture size decreases the light throughput, resulting in a dark and noisy image. Multi-focus images offer a solution to increase the DOF without decreasing light throughput. By capturing different photos focused at different depths in the scene and combining them, the entire scene can be brought into focus. This capture procedure is also called "focus bracketing." This is similar to "exposure bracketing," where images are taken under different exposures and combined to obtain a high dynamic range image. However, a disadvantage with focus bracketing is that the entire scene has to remain static during the capture process.

Theory

For a thin lens with focal length f and lens to sensor plane distance u, the plane of focus is at a distance v from the lens, where

$$\frac{1}{v} = \frac{1}{f} - \frac{1}{u}. \tag{1}$$

Let c be the size of the acceptable circle of confusion and A be the aperture diameter. Then, f-number $N = f/A$. The DOF is then spanned between D_n and D_f, where

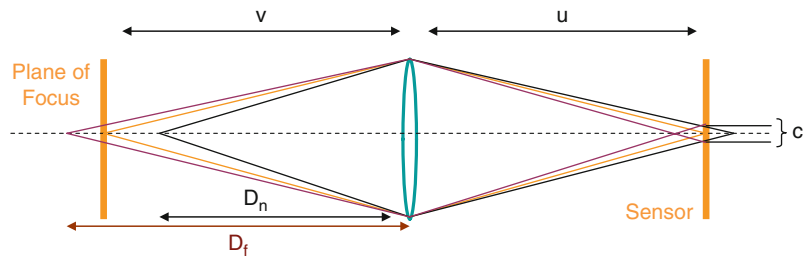
$$D_n = \frac{vf^2}{f^2 + Nc(v-f)}, \tag{2}$$

$$D_f = \frac{vf^2}{f^2 - Nc(v-f)}. \tag{3}$$

As shown in Fig. 1 the DOF region in front of the plane of focus is not equal to the DOF region behind it. Multi-focus images can be obtained by capturing several images, each under a different focus setting of the lens (change of u). Note that other camera parameters such as aperture size, exposure time, and zoom can also be modified along with the focus setting of the lens during the capture.

Minimizing the Capture Time

Since multi-focus images require the entire scene to remain static during the capture process, it is important to decrease the overall capture time. In [1], the problem of imaging a scene with a given depth of field at a given exposure level in the shortest amount of time was considered. The criteria for optimal capture sequence were derived. Since the light throughput is quadratic with respect to the aperture size but linear with respect to the exposure time, increasing the aperture size is more beneficial to reduce the overall capture time. Intuitively, one should use a large aperture and sweep the focus such that all scene points are sharp in at least one of the captured image.



Multi-focus Images, Fig. 1 DOF for a thin lens with focal length f lies between D_f and D_n

Combining Multi-focus Images

Multi-focus images can be combined to generate an image with larger DOF than any of the individual source images. This is also known as focus stacking and is especially useful in macro photography and optical microscopy. The resultant image is equivalent to the photo captured using a small aperture but will be significantly less noisy. Haeberli [2] showed how to combine multi-focus images by choosing each pixel intensity from the image where it appears to be the sharpest. The sharpness measure can be defined using local variance or local image gradients. Recent approaches have used an energy minimization framework to combine multi-focus images using fast techniques such as graph-cuts [3].

Depth from Defocus and Focus

An additional advantage of multi-focus images is that they can be used to estimate the scene depths, since the defocus blur of a scene point is related to its depth. Depth from defocus is an active area of research in computer vision. Depth from defocus techniques model the relationship between depth and defocus blur using a parametric function and use it to estimate depth from several defocused images. See [4] for a review on such techniques. On the other hand, depth from focus [5, 6] approaches use a focus measure to identify the focus setting for each pixel, which is converted to the metric depth via calibration.

Extended Depth of Field

Focus bracketing is an easy and practical solution to increase the DOF of imaging systems without any hardware modifications. The underlying problem of extending depth of field has several other interesting solutions such as (a) aperture modification [7, 8], (b) light-field based digital refocusing [7, 9–12], (c) lens modifications [13–15], and (d) sensor motion [16].

Application

Extending the DOF has applications in consumer and sports photography, as well as scientific imaging. Multi-focus images offer a viable solution if the scene is changing slowly compared to the capture time required for focus bracketing.

Open Problems

Applying multi-focus images to a dynamic scene is an open and interesting problem.

References

1. Hasinoff S, Kutulakos KN (2008) Light-efficient photography. In: Proceeding of the 10th european conference on computer vision (ECCV), Marseille, pp 45–59
2. Haeberli P (1994) A multifocus method for controlling depth of field. http://www.sgi.com/grafica/depth/index.html
3. Agarwala A, Dontcheva M, Agrawala M, Drucker S, Colburn A, Curless B, Salesin D, Cohen M (2004) Interactive digital photomontage. ACM Trans Gr 23(3):294–302
4. Chaudhuri S, Rajagopalan A (1999) Depth from defocus: a real aperture imaging approach. Springer, New York
5. Grossmann P (1987) Depth from focus. Pattern Recognit Lett 5:63–69
6. Nayar S (1992) Shape from focus system. In: Proceedings of the conference on computer vision and pattern recognition (CVPR), Orlando, pp 302–308
7. Veeraraghavan A, Raskar R, Agrawal A, Mohan A, Tumblin J (2007) Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. ACM Trans Gr 26(3):69:1–69:12
8. Levin A, Fergus R, Durand F, Freeman WT (2007) Image and depth from a conventional camera with a coded aperture. ACM Trans Gr 26(3):70
9. Ng R, Levoy M, Brï£¡dif M, Duval G, Horowitz M, Hanrahan P (2005) Light field photography with a hand-held plenoptic camera. Technical report, Stanford University
10. Ng R (2005) Fourier slice photography. ACM Trans Gr 24:735–744

11. Georgiev T, Zheng C, Nayar S, Curless B, Salasin D, Intwala C (2006) Spatio-angular resolution trade-offs in integral photography. In: Eurographics symposium on rendering, Nicosia, 263–272
12. Liang CK, Lin TH, Wong BY, Liu C, Chen H (2008) Programmable aperture photography: multiplexed light field acquisition. ACM Trans Gr 27(3):55:1–55:10
13. Green P, Sun W, Matusik W, Durand F (2007) Multi-aperture photography. ACM Trans Gr 26(3):68:1–68:7
14. Dowski ER, Cathey W (1995) Extended depth of field through wavefront coding. Appl Opt 34(11):1859–1866
15. Levin A, Hasinoff SW, Green P, Durand F, Freeman WT (2009) 4d frequency analysis of computational cameras for depth of field extension. ACM Trans Gr 28(3):97:1–97:14
16. Nagahara H, Kuthirummal S, Zhou C, Nayar S (2008) Flexible depth of field photography. In: Proceeding of the european conference of computer vision (ECCV), Marseille

Multimedia Retrieval

▸Video Retrieval

Multiple Similarity Method

▸Subspace Methods

Multiple View Geometry

▸Epipolar Geometry

Multiple View Stereo

▸Multiview Stereo

Multiplexed Illumination

Marina Alterman
Department of Electrical Engineering, Technion – Israel Institute of Technology, Haifa, Israel

Synonyms

Multiplexed sensing

Definition

In multiplexed illumination, multiple light sources are used simultaneously in different measurements of intensity data arrays. Then, the intensity under individual sources is derived by computational demultiplexing. This scheme enhances the results: it increases the signal-to-noise ratio of intensity data arrays, without increasing acquisition resources such as time. It also improves dynamic range.

Background

Measuring a set of variables is a common task. For example, in computer vision and graphics, there is a need to acquire multiple images under various lighting conditions; in spectroscopy, there is a need to measure several wavelength bands; in tomography, measurements are taken at a set of different directions; in microscopy, there is a set of focal planes or a set of measurements under several fluorescence excitation wavelength bands. Usually these variables are measured sequentially.

The measurements are subjected to noise, which may yield a low signal-to-noise ratio (SNR). In many cases, the SNR cannot be improved simply by increasing the illumination of individual sources or the exposure time. Simultaneously combining signals corresponding to multiple variables into a single multiplexed measurement may be more efficient. This way, in some cases, the total intensity of multiplexed signals increases relative to the noise. The acquired multiplexed measurements are demultiplexed by a computer, yielding an array of intensity values with a higher SNR.

Theory

Basic Multiplexing

Often, sensors seek measurement of a vector of observable intensity variables $\mathbf{i}$. Generally, the acquired raw measurements form a vector $\mathbf{a}$ of length N_{measure}. This raw vector is related to $\mathbf{i}$ by

$$\mathbf{a} = \mathbf{W}\mathbf{i} + \eta, \tag{1}$$

where η is a vector of measurement noise (the noise is uncorrelated in different measurements). Here $\mathbf{W}$ is a weighting matrix, referred to as a *multiplexing code*. One case is $\mathbf{W} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. This special case is referred to as *trivial* sensing: only a single component of $\mathbf{i}$ is acquired at a time. More generally, multiple components of $\mathbf{i}$ can be simultaneously summed up (multiplexed) and acquired in each raw measurement. The components of $\mathbf{i}$ included in the mth measurement are determined by the mth row of $\mathbf{W}$. After the measurements are taken, the vector $\mathbf{i}$ of intensities corresponding to the individual source can be decoded from the vector of measurements $\mathbf{a}$, using

$$\hat{\mathbf{i}} = \mathbf{W}^{-1}\,\mathbf{a} \tag{2}$$

(when $\mathbf{W}$ is invertible) or by an estimator such as least squares.

A simple case in which $N_{\text{sources}} = 3$ is depicted in Fig. 1. There, two sources are used simultaneously per acquired measurement. For general number of sources, the vectors $\mathbf{i}$ and $\mathbf{a}$ are related by a linear superposition as in Eq. (1). The elements $w_{m,s}$ of $\mathbf{W}$ represent [13] the normalized radiance of source s in measurement m. If $w_{m,s} = 0$, then source s is turned off completely at measurement m; if $w_{m,s} = 1$, then this source irradiates the object, at the source's maximum power. Generally, $0 \le w_{m,s} \le 1$.

The mean squared error (MSE) [5] of $\hat{\mathbf{i}}$ is

$$\text{MSE}_{\mathbf{i}} = \frac{\sigma^2}{N_{\text{sources}}}\,\texttt{tr}\left[\left(\mathbf{W}^T\mathbf{W}\right)^{-1}\right], \tag{3}$$

where $\texttt{tr}$ is a trace operation and σ^2 is the variance of η. Based on Eqs. (2) and (3), $\mathbf{i}$ can be reconstructed, with a potentially higher SNR [5, 11, 13] than $\mathbf{i}$ which is *trivially* sensed using $\mathbf{I}$. The *gain* of using multiplexing (termed multiplex *advantage*) is defined [5] as

$$\text{GAIN}_{\mathbf{i}} = \sqrt{\sigma^2/\text{MSE}_{\mathbf{i}}}\ . \tag{4}$$

Most related studies have aimed to maximize the SNR of the recovered images $\hat{\mathbf{i}}$. Thus, Refs. [5, 9, 11, 13] sought a multiplexing matrix that minimizes Eq. (3):

$$\hat{\mathbf{W}}_{\mathbf{i}} = \arg\min_{\mathbf{W}} \text{MSE}_{\mathbf{i}}\ . \tag{5}$$

An optimal multiplexing code should yield the highest SNR of the demultiplexed values. When the signal dependency of noise is not considered, the optimal multiplexing codes are based on Hadamard matrices [5].

According to the affine noise model [11], the detector noise variance is composed of two components, signal dependent and signal independent. The gray-level variance of the signal-independent noise is denoted by κ^2_{gray}. Considering a diffuse object, each light source yields a similar object radiance. Therefore, each source yields a similar level of noise. If C sources are activated in their maximum power, the total noise variance of the measured gray level is [11, 14]

$$\sigma^2 = \kappa^2_{\text{gray}} + C\mu^2\ , \tag{6}$$

where μ^2 is the photon noise variance, caused by object irradiance from a single source activated at its maximum power. If photon noise dominates the noise ($C\mu^2 \gg \kappa^2_{\text{gray}}$), Hadamard multiplexing codes degrade the decoded images. The reason is that simultaneous sources increase the image intensity, which in turn increases the photon noise. Therefore, there is a need for generalization of the multiplexing model to obtain new and improved multiplexing codes.

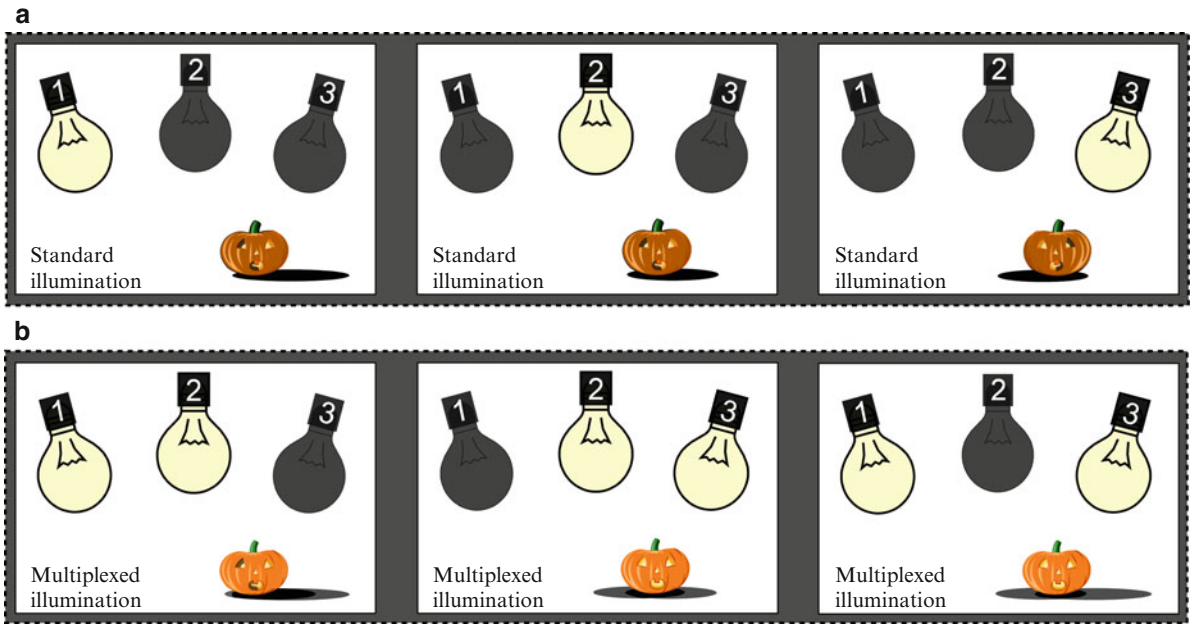
Generalized Multiplexing

References [11, 12] derive optimal multiplexing codes considering photon noise and saturation. Saturation occurs when the total illumination radiance exceeds a certain threshold. If all light sources yield a similar object radiance, then this threshold C_{sat} is expressed in units of light sources ($C = C_{\text{sat}}$). It is preferable to exploit the maximum radiance for every measurement. Thus, to account for saturation, the constraint

$$\sum_{s=1}^{N_{\text{sources}}} w_{m,s} = C_{\text{sat}} \tag{7}$$

is added to the optimization problem in Eq. (5). By scanning a range of C_{sat} values in Eq. (7) and using $C = C_{\text{sat}}$ in Eqs. (3) and (6), the value that yields the maximum gain Eq. (4) is found. This accounts both for saturation and photon noise.

However, is a demultiplexed array $\mathbf{i}$ the true goal of a vision system? Often not. Usually, the recovered intensity or reflectance array $\mathbf{i}$ is by itself an input to further analysis. For example, a multispectral imager



Multiplexed Illumination, Fig. 1 Three light sources illumination [13]. (**a**) Standard (trivial) illumination: single light source is active in each measurement. (**b**) Multiplexed illumination: two light sources are active in each measurement

may recover a scene's spectral datacube, and multiplexing is helpful in this. But the resulting multispectral datacube itself is of little interest *per se*: usually (e.g., in remote sensing) the user is interested in the underlying spatial distribution *of materials* or objects that created the spectral data. This is formulated as a mixing model

$$\mathbf{i} = \mathbf{X}\,\mathbf{c}\,, \tag{8}$$

where $\mathbf{X}$ is a *mixing* matrix. The end product of interest in this example is *not* demultiplexed intensities or spectral reflectance ($\mathbf{i}$) but a distribution of materials ($\mathbf{c}$). Similarly, in multispectral imaging of fluorescing specimen, intensities ($\mathbf{i}$) are just a means to obtain information about molecular distributions in the specimen ($\mathbf{c}$). Recovery of $\mathbf{c}$ based on $\mathbf{i}$ is called *unmixing*.

Reference [2] showed that unmixing can (and should) be fully integrated when optimizing the multiplexing codes. Otherwise, the true underlying variables of interest may *be harmed* by multiplexing. Let multiple sources be active in each measurement (intentional multiplexing). Using Eqs. (1) and (8), $\mathbf{c}$ can be estimated, for example, by using weighted least squares

$$\hat{\mathbf{c}} = \left[(\mathbf{WX})^T \boldsymbol{\Sigma}_{\text{noise}}^{-1} (\mathbf{WX})\right]^{-1} (\mathbf{WX})^T \boldsymbol{\Sigma}_{\text{noise}}^{-1} \mathbf{a}\,, \tag{9}$$

where $\boldsymbol{\Sigma}_{\text{noise}}$ is the covariance matrix of the raw measurement noise η. Then, the MSE of $\mathbf{c}$ is

$$\text{MSE}_{\mathbf{c}} = \frac{1}{N_{\text{materials}}} \mathtt{tr}\left\{\left[(\mathbf{WX})^T \boldsymbol{\Sigma}_{\text{noise}}^{-1} (\mathbf{WX})\right]^{-1}\right\}\,, \tag{10}$$

where $N_{\text{materials}}$ is the number of materials to unmix. To multiplex measurements in a way that optimally recovers $\mathbf{c}$ (*multiplexed unmixing*), a multiplexing matrix $\mathbf{W}$ that minimizes the MSE of $\mathbf{c}$ Eq. (10) is sought,

$$\hat{\mathbf{W}}_{\mathbf{c}} = \arg\min_{\mathbf{W}} \text{MSE}_{\mathbf{c}}\,. \tag{11}$$

Application

Multiplexed illumination is used in multispectral imaging (array of spectral bands) [3, 9], spectroscopy [4], and lighting (reflection from an array of light sources) [8, 11, 13]. It also has analogue formulations in coded apertures (array of spatial positions or viewpoints) and coded shuttering [1, 6, 7, 10] (array of spatiotemporal pixel values).

References

1. Agrawal A, Raskar R (2009) Optimal single image capture for motion deblurring. In: Proceedings of the IEEE conference on computer vision pattern recognition (CVPR), Miami
2. Alterman M, Schechner YY, Weiss A (2010) Multiplexed fluorescence unmixing. In: IEEE ICCP, Cambridge
3. Ben-Ezra M, Wang J, Wilburn B, Li X, Ma L (2008) An LED-only BRDF measurement device. In: IEEE conference on computer vision pattern recognition (CVPR), Anchorage
4. Cull EC, Gehm ME, Brady DJ, Hsieh CR, Momtahan O, Adibi A (2007) Dispersion multiplexing with broadband filtering for miniature spectrometers. Appl Opt 46:365–374
5. Harwit M, Sloane NJA (1979) Hadamard transform optics. Academic, New York
6. Levoy M, Chen B, Vaish V, Horowitz M, McDowall I, Bolas M (2004) Synthetic aperture confocal imaging. ACM Trans Graph 23:825–834
7. Liang CK, Lin TH, Wong BY, Liu C, Chen HH (2008) Programmable aperture photography: multiplexed light field acquisition. In: ACM TOG. ACM Trans Graph, New York, pp 1–10
8. Narasimhan S, Koppal S, Yamazaki S (2008) Temporal dithering of illumination for fast active vision. In: Proceedings of the European conference on computer vision (ECCV), Marseille, pp 830–844
9. Park J, Lee M, Grossberg MD, Nayar SK (2007) Multispectral imaging using multiplexed illumination. In: Proceedings of the IEEE ICCV, Rio de Janeiro
10. Raskar R, Agrawal A, Tumblin J (2006) Coded exposure photography: motion deblurring using fluttered shutter. ACM Trans Graph 25:795–804
11. Ratner N, Schechner YY (2007) Illumination multiplexing within fundamental limits. In: Proceedings of the IEEE conference on computer vision pattern recognition (CVPR), Minneapolis
12. Ratner N, Schechner YY, Goldberg F (2007) Optimal multiplexed sensing: bounds, conditions and a graph theory link. Opt Express 15:17072–17092
13. Schechner YY, Nayar SK, Belhumeur PN (2007) Multiplexing for optimal lighting. IEEE Trans PAMI 29:1339–1354
14. Wuttig A (2005) Optimal transformations for optical multiplex measurements in the presence of photon noise. Appl Opt 44:2710–2719

Multiplexed Sensing

▸Multiplexed Illumination

Multisensor Data Fusion

▸Sensor Fusion

Multiview Geometry

▸Epipolar Geometry

Multiview Stereo

Sudipta N. Sinha
Microsoft Research, Redmond, WA, USA

Synonyms

Multiple view stereo

Related Concepts

▸Dense Reconstruction; ▸Multi-baseline Stereo

Definition

Multiview stereo refers to the task of reconstructing a 3D shape from calibrated overlapping images captured from different viewpoints. Various representations of 3D shape can be used. For example, dense 3D point cloud or surface mesh representations are common in applications that synthesize a new photorealistic image of the scene using computer graphic rendering techniques. The topics of multiview stereo and multi-baseline stereo matching share key concepts related to the recovery of dense 2D pixel correspondences in multiple images.

Background

Reconstructing 3D geometry from images (often also called *3D photography*) involves using cameras or optical sensors (and optionally illumination) to acquire the 3D shape and appearance of objects and scenes in the real world. Existing methods can be broadly divided into two categories – *active* and *passive*

methods. Active methods usually require additional special light sources, whereas passive methods work with natural lighting. Active methods often use special-purpose sensors such as laser range scanners and depth sensors (Kinect, time-of-flight cameras) and can often capture high-quality 3D models. However, developing passive methods for ordinary cameras is also important because of their greater ease, flexibility, and wider applicability.

Multiview stereo is a popular and well-studied passive 3D reconstruction technique. It is based on the principle that dense pixel correspondences in multiple calibrated images captured from different viewpoints makes it possible to estimate the 3D shape of the scene via triangulation, which involves intersecting rays backprojected from the corresponding pixels. Multiview stereo therefore requires the camera calibration parameters to be known. These parameters can either be precomputed offline or can be computed from the image sequence using structure from motion algorithms. Figure 1 shows an example of a 3D reconstruction obtained using multiview stereo.

The main challenge in multiview stereo lies in computing precise, dense pixel correspondence between images. Difficulties arise due to ambiguities in matching pixels in two images of the same scene. Multiview stereo works best when surfaces are textured and *Lambertian*, i.e., when the local appearance of a surface patch does not depend on the viewing angle. Glossy or specular surfaces are non-Lambertian and are more difficult to handle compared to diffuse surfaces. Occlusions can complicate the situation even further, especially in the *wide baseline* setting where the camera viewpoints are farther from each other compared to *the narrow baseline* setting, where the effect of occlusions is less pronounced.

For reconstructing a static scene with multiview stereo, a single camera can be used to capture images from multiple viewpoints over time. Sometimes, images are captured with a static camera with the object placed on a rotating turntable [1]. These techniques can also be applied to dynamic scenes, provided multiview video is captured from a calibrated, synchronized multi-camera rig. The Virtualized Reality project at CMU [2] was the first to demonstrate multiview 3D reconstruction of dynamic events within a large scene.

Theory

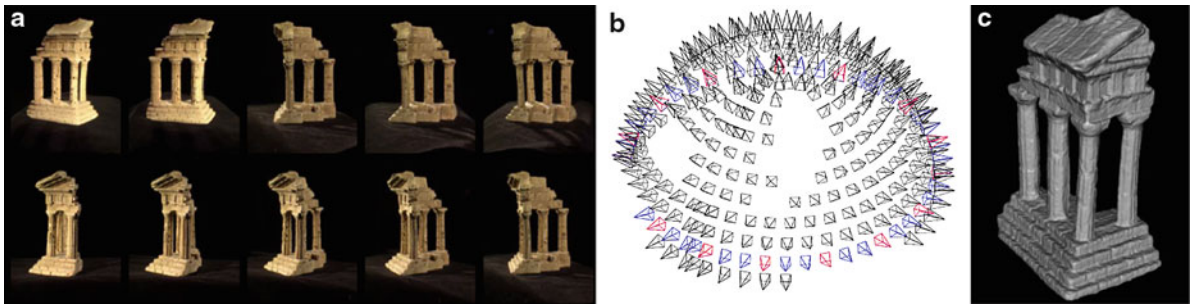
A taxonomy was recently introduced to broadly categorize multiview stereo approaches [3]. It proposed studying various methods based on the following properties: 3D shape representations, the photoconsistency measure, visibility handling, shape priors, and the reconstruction algorithm. A recent tutorial [4] and an earlier survey [5] are also excellent sources of information on this topic.

3D Shape Representation

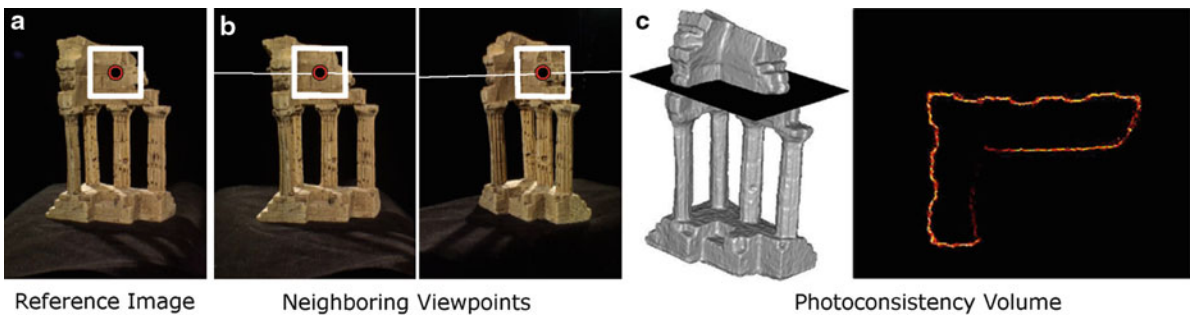
Many multiview stereo methods represent the 3D scene as a set of depth maps, one for each calibrated viewpoint, and recover the 3D shape that is most consistent with all the depth maps [6–8]. Other methods use explicit surface-based representations. Polygonal meshes are used both as an internal representation in some methods [1] and as the representation for the final 3D shape. Volumetric representations such as a uniform 3D voxel grid where each voxel is labeled as occupied or empty are common as well for their flexibility in approximating a wide variety of shapes [9–11]. Other volumetric representations such as adaptive tessellation of 3D space into tetrahedral cells can be more compact [12, 13]. Finally, patch-based representations are also possible [14, 15]. Recent multiview stereo methods represent scenes using a finite set of locally planar, oriented 3D patches, referred to as *surfels*. No connectivity information is stored as in a surface mesh. While surface meshes and volumetric representations are advantageous for reconstructing closed surfaces or 3D objects, depth maps or patch-based representations are often better choices for reconstructing large open scenes [7, 8, 15, 16].

Photoconsistency

It is a key ingredient in multiview stereo methods. It is a measure for the photometric similarity of the 2D projections of any 3D scene point in a set of calibrated images. For 3D points on surfaces visible in a set of cameras, the 2D projections in those images are expected to be similar or photometrically consistent. Such points are said to have high photoconsistency. On the other hand, arbitrary 3D scene points are likely to have low photoconsistency in most



Multiview Stereo, Fig. 1 Images captured with a camera rig. Calibrated cameras and the final 3d model produced by a state-of-the-art multiview stereo method rendered as a triangulated mesh



Multiview Stereo, Fig. 2 (**a**) A pixel in a reference image and and a *square patch* around it. (**b**) Corresponding pixels and associated patches in the nearby images that lie on respective *epipolar lines*. In calibrated images, the search for correspondences reduces to 1D, i.e., along the *epipolar line*. (**c**) A horizontal slice of the 3D cost volume is shown. It is constructed using the photoconsistency measure described in [10]

situations. Some earlier methods used the variance in pixel colors in images where a 3D point is visible, as a measure of photoconsistency. However, since comparing individual pixels can be ambiguous, a more reliable approach is to compare the similarity of image patches around the projections of a 3D point (see Fig. 2). For narrow baselines, it is sufficient to compare image-oriented square patches centered on pixels. However, by incorporating the geometry of the 3D patch, the matching windows can be correctly adjusted to account for scale differences in the images and slanted surfaces.

A pair of 2D image patches can be compared using normalized cross correlation (NCC) after resampling a $n = k \times k$ 2D grid at each patch and comparing the vector of color values denoted here as u and v, respectively:

$$\mathrm{NCC}(u,v) = \frac{\sum_{j=0}^{n}(u_j - \overline{u}).(v_j - \overline{v})}{\sqrt{\sum_{j=0}^{n}(u_j - \overline{u})^2 . \sum_{j=0}^{n}(v_j - \overline{v})^2}} \quad (1)$$

The NCC-based similarity measure lies in the range $[-1,1]$ where a value closer to 1 indicates that the vectors are similar in appearance. Other similarity measures such as the sum of absolute differences (SAD), the sum of squared differences (SSD), or nonparametric measures can be used to compute photoconsistency as well. Unlike NCC, SAD or SSD are affected by brightness changes in different images.

Photoconsistency Volume Computation

Many multiview stereo methods require the photoconsistency function to be evaluated densely on a 3D voxel grid for the volume containing the scene which is then referred to as the *cost volume*. Figure 2 shows an example. One simple way to construct this cost volume is to evaluate the photoconsistency of all 3D points (or voxels) using a pairwise similarity measure such as NCC and then compute the mean NCC score from multiple pairs of nearby images. There also exist direct methods to compute the cost volume [17] or approaches that are better at minimizing noise in the photoconsistency

estimates [10]. Being able to estimate the visibility of a 3D point allows one to select which images contribute to its photoconsistency measure. A coarse approximation of the shape such as the visual hull reconstruction obtained from silhouettes is often sufficient when reconstructing closed objects [1, 10, 12]. However, when a large number of images are available, occlusions can also be treated as outliers and explicit visibility reasoning is not required [10, 14].

Another general approach to construct the cost volume involves first estimating depth maps from each camera's viewpoint using a state-of-the-art multibaseline stereo matching algorithm and then merging them into a consistent volumetric representation. This sort of aggregation incorporates visibility information induced by the independently estimated depth maps. Intuitively, this can be thought of as a way of probabilistically carving out the 3D volume [6, 18], an idea that is closely related to the problem of fusion of range images [19].

Plane Sweep Stereo and Depth Map Fusion

A class of methods referred to as plane sweep stereo methods avoid building the cost volume on a uniform 3D grid. Instead, they sweep the scene with a set of parallel planes corresponding to the candidate depths considered for pixels in the reference image. These planes are typically fronto parallel to the reference camera, and their spacing is selected in uniform intervals with respect to their inverse depth from the camera. Neighboring images are warped on to these planes using 2D homographies, and the photoconsistency measure is computed at each pixel for each candidate depth plane. Plane-sweeping strategies are often chosen by incorporating some knowledge of scene structure and are popular for 3D recontruction in urban scenes [7]. Their main advantage over uniform voxel grids or other 3D tessellations lies in the fact that they represent large working volumes more efficiently, which is important for reconstructing large open scenes. Plane sweep stereo can be combined with depth map fusion which can be implemented using an image space representation [8]. Artifacts in the original depth maps are reduced in the fused depth maps from which triangulated surface meshes can be directly extracted. The fusion step works better when per-pixel confidence estimates associated with the depth estimates are available.

Optimization Methods

Broadly all multiview stereo methods formulate the reconstruction task in terms of a *local* or *global* optimization problem. A local method such as space carving [9] starts with an overestimate of the 3D scene, and uses a greedy strategy to remove voxels that are not photoconsistent one at a time. Similarly, in fast plane sweep stereo [7], each pixel's depth is computed independent of other pixels. These local methods are susceptible to noise and outliers and cannot easily reconstruct smooth geometric shapes. Part of the difficulty is that the 3D reconstruction task is inherently ill-posed since different 3D scenes can be consistent with the same set of images. By assuming that surfaces in the scene are primarily smooth, various global optimization methods seek the optimal 3D shape which maximizes both photoconsistency and smoothness. This is achieved by formulating multiview stereo as a global optimization problem with geometric regularization terms in the objective function or energy function. These are minimized either in the discrete or in the continuous setting.

Global methods for depth map estimation incorporate image-based smoothness constraints into the formulation by designing suitable energy functions that encourages neighboring pixels to take on identical or similar depth values. Such methods are often based on a 2D Markov Random Field (MRF) framework for which efficient optimization algorithms have been developed in recent years. The MRF framework can also be used for enforcing surface regularization in volumetric methods on a 3D uniform grid [10] where any binary labeling of voxels in the grid corresponds to some 3D shape. The optimal surface corresponding to the global minimum of the energy function can be efficiently computed using graph cut algorithms in many situations [1, 18]. In case of [12, 13], the minimal surface computed using graph cuts directly produces a triangulated mesh. These methods first solve a discrete optimization problem to obtain a globally consistent solution and then performs local refinement using continuous optimization to recover finer details in the structure [1, 12]. As an alternative, direct continuous optimization methods based on convex relaxations have also been developed to address discretization artifacts which can occur in graph cut-based methods [11].

Variational methods can also be used to extract an optimal smooth surface from the cost volume using

multiple iterations that progressively minimize a global objective function. In [1], this was achieved by evolving a deformable surface mesh using photoconsistency cues. However, a good initial guess for the 3D shape was required. Methods based on level sets are more flexible as the surface topology is allowed to change during the iterations. These approaches represent the surface as the zero level set of an evolving implicit function [20]. The energy function is minimized by modeling the evolution of the function using partial differential equations. These level-set methods have the advantage of producing smooth surface reconstructions but have the disadvantage of being susceptible to local minima unless a good initialization is available.

Patch-Based Multiview Stereo

Another class of popular multiview stereo approaches represent surfaces as a set of oriented patches without storing any connectivity information. This makes the representation flexibile and suitable for reconstructing both closed 3D shapes and open scenes. A watertight surface mesh can be extracted using the Poisson surface reconstruction algorithm as a post-processing step. Patch-based methods are related to region-growing methods as they typically start from a few seed 3D points with known depth. The algorithm proposed in [14] starts with a sparse structure from motion 3D point cloud and iteratively optimizes the depth and normals of the oriented patches using an NCC-based photoconsistency measure. The pixels in the reference image are processed in a matching score-based priority order that ensures that confident estimates are propagated first in the surface-growing phase. This step produces semi-dense depth maps which are then merged to generate a global set of oriented 3D patches. This method was used to perform 3D reconstruction from Internet images of popular landmarks and showed that dense correspondences could be reliably computed from such large heterogenous image collections as well.

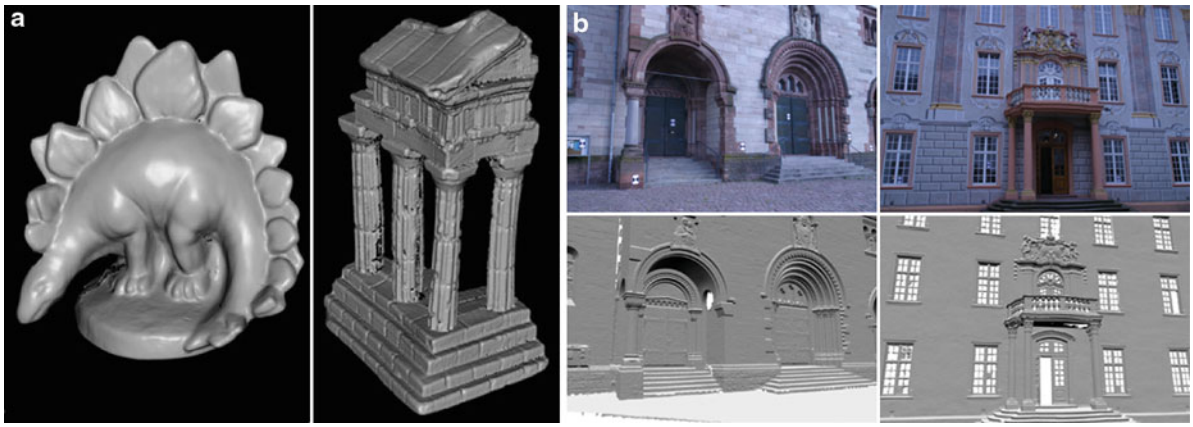
Patch-based multiview stereo (PMVS) [15] is another popular algorithm that also uses a seed and grow reconstruction strategy and consists of three important steps. First, seed patches are created from sparse 2D keypoint correspondences in neighboring overlapping views which can be matched with high confidence. These patches are iteratively expanded using a locally planar model to generate new patches. This is followed by a patch optimization and filtering step which refines the position and orientation of the patches and then removes noisy or outlier samples based on photoconsistency and visibility constraints. A publicly available implementation of this algorithm is available as part of the PMVS library [15]. Recently, this library was also extended to support large-scale reconstructions from Internet collections [16].

Efficient Multiview Stereo on GPUs

Some multiview stereo methods strive for high-fidelity reconstructions and can be very computationally intensive especially when high-resolution images are processed [1, 14–16, 21]. The main computational bottleneck in multiview stereo lies in the photoconsistency computation or computing matching cost over many pairs of windows. Typically, this can be accelerated by orders of magnitude on massively parallel hardware and is also perfectly suitable for the data-parallelism supported on modern programmable graphics hardware (GPUs) with SIMD architectures. Many variants of multiview stereo ranging from plane sweep stereo [7], depth map fusion [8] to level-sets-based methods [22] have been successfully ported to the GPU, and one or two orders of magnitude speedup have been demonstrated.

Benchmarks

The Middlebury multiview stereo datasets shown in Fig. 3a contain ground truth 3D models created by scanning the models using a laser stripe scanner and registering the 3D mesh to the calibrated images captured with a gantry. The benchmark has recently been quite popular for evaluating multiview stereo algorithms [3]. It uses two criteria to evaluate the reconstructions – *accuracy* and *completeness*. The model's accuracy is calculated by computing the distance between the points sampled on the reconstructed model and the nearest points on the ground truth model and reporting the distance (in mm) such that 90% of the points on the reconstructed model are within that distance from the ground truth model. Similarly, the



Multiview Stereo, Fig. 3 Multiview stereo benchmarks used for quantitative evaluation: (**a**) Middlebury *Dino* and *Temple* datasets from the Middlebury multiview stereo benchmark (http://vision.middlebury.edu/mview/). (**b**) Large scenes from the outdoor multiview stereo benchmark (http://cvlab.epfl.ch/strecha/multiview/denseMVS.html)

completeness measure is computed for a given threshold by finding the nearest point on the reconstructed mesh for each vertex in the ground truth mesh and the percentage of points on the ground truth model that is within a distance threshold (default value = 1.25 mm) of the reconstructed model.

Another benchmarks for evaluating multiview stereo reconstruction of large scenes is also available [23]. Precise laser scanned models are provided for ground truth comparisons. Unlike Middlebury where the scanned models are quite small (only 16 cm on the longest dimension), these datasets consist of high-resolution images and much larger outdoor scenes. Two of the scenes captured in this benchmark are shown in Fig. 3b. Several multiview stereo methods have demonstrated accurate result on these datasets.

References

1. Hernández C (2004) Stereo and silhouette fusion for 3D object modeling from uncalibrated images under circular motion. PhD thesis, Ecole Nationale Supŕieure des Télécommunications
2. Kanade T, Rander P, Narayanan PJ (1997) Virtualized reality: constructing virtual worlds from real scenes. IEEE MultiMed 4(1):34–47
3. Seitz SM, Curless B, Diebel J, Scharstein D, Szeliski R (2006) A comparison and evaluation of multi-view stereo reconstruction algorithms. In: Computer vision and pattern recognition (CVPR), vol 1, pp 519–528. New York, USA
4. Hernandez C, Vogiatzis G, Furukawa Y (2010) CVPR Tutorial. http://carlos-hernandez.org/cvpr2010/
5. Slabaugh GG, Culbertson WB, Malzbender T, Stevens MR, Schafer RW (2004) Methods for volumetric reconstruction of visual scenes. Int J Comput Vis 57:179–199
6. Goesele M, Curless B, Seitz SM (2006) Multi-view stereo revisited. CVPR '06, vol 2, pp 2402–2409. New York, USA
7. Gallup D, Frahm JM, Mordohai P, Yang Q, Pollefeys M (2007) Real-time plane-sweeping stereo with multiple sweeping directions. In: IEEE conference on computer vision pattern recognition (CVPR). Minneapolis, USA
8. Merrell P, Akbarzadeh A, Wang L, Mordohai P, Frahm JM, Yang R, Nistér D, Pollefeys M (2007) Real-time visibility-based fusion of depth maps. In: ICCV, pp 1–8. Rio de Janeiro, Brazi
9. Kutulakos K, Seitz S (2000) A theory of shape by space carving. Int J Comput Vis 38(3):199–218
10. Vogiatzis G, Esteban CH, Torr PHS, Cipolla R (2007) Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. IEEE Trans Pattern Anal Mach Intell 29(12):2241–2246
11. Kolev K, Klodt M, Brox T, Cremers D (2009) Continuous global optimization in multiview 3d reconstruction. Int J Comput Vis 84(1):80–96
12. Sinha SN, Mordohai P, Pollefeys M (2007) Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In: ICCV. Rio de Janeiro, Brazi
13. Vu HH, Keriven R, Labatut P, Pons JP (2009) Towards high-resolution large-scale multi-view stereo. In: Conference on computer vision and pattern recognition (CVPR). Miami Beach, USA
14. Goesele M, Snavely N, Curless B, Hoppe H, Seitz SM (2007) Multi-view stereo for community photo collections. In: ICCV, pp 1–8. Rio de Janeiro, Brazil
15. Furukawa Y, Ponce J (2007) Accurate, dense, and robust multi-view stereopsis. In: Computer vision and pattern recognition (CVPR), pp 1–8. Minneapolis, USA
16. Furukawa Y, Curless B, Seitz SM, Szeliski R (2010) Towards internet-scale multi-view stereo. In: IEEE

conference on computer vision pattern recognition (CVPR). San Francisco, USA
17. Hornung A, Kobbelt L (2006) Robust and efficient photo-consistency estimation for volumetric 3d reconstruction. ECCV'06, vol 2, pp 179–190. Graz, Austria
18. Hernández C, Vogiatzis G, Cipolla R (2007) Probabilistic visibility for multi-view stereo. In: IEEE conference on computer vision pattern recognition (CVPR). Minneapolis, USA
19. Curless B, Levoy M (1996) A volumetric method for building complex models from range images. In: Proceedings of the 23rd annual conference on computer graphics and interactive techniques. SIGGRAPH '96, pp 303–312. New Orleans, USA
20. Pons JP, Keriven R, Faugeras O (2007) Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. Int J Comput Vis 72(2): 179–193
21. Tola E, Strecha C, Fua P (2011) Efficient large-scale multi-view stereo for ultra high-resolution image sets. J Mach Vis Appl 23(5):903–920
22. Labatut P, Keriven R, Pons JP (2006) Fast level set multi-view stereo on graphics hardware. In: International symposium on 3D data processing visualization and transmission. Chapel Hill, USA
23. Strecha C, von Hansen W, Gool LJV, Fua P, Thoennessen U (2008) On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: IEEE conference on computer vision pattern recognition (CVPR). Anchorage, USA

Mutual Illumination

▸Interreflections