Open-Ended Inference of Relational Representations in the COSPAL Perception-Action Architecture

David Windridge, Josef Kittler

Centre for Vision, Speech and Signal Processing
Dept. of Electronic & Electrical Engineering, University of Surrey, Guildford, UK
Tel: +44 1483 876043, d.windridge@eim.surrey.ac.uk

Abstract. The COSPAL architecture for autonomous artifical cognition utilises incremental perception-action learning in order to generate hierarchically-grounded abstract representations of an agent's environment on the basis of its action capabilities. We here give an overview of the top-level relational module of this architecture.
The first stage of the process hence involves the application of ILP to attempted action outcomes in order to determine the set of generalised rule protocols governing actions within the agent's environment (initially defined via an *a priori* low-level representation). In the second stage, imposing certain constraints on legitimate first-order logic induction permits a compact reparameterisation of the percept space such that novel perceptual-capabilities are always correlated with novel action capabilites. We thereby define a meaningful *empirical* criterion for perceptual inference.
Novel perceptual capabilities are of a higher abstract order than the *a priori* environment representation, allowing more sophisticated exploratory action to be taken. Gathering of further exploratory data for rule induction hence takes place in an iterative cycle. Application of this mechanism within a simulated 'shape-sorter' puzzle environment indicates that this approach significantly accelerates learning of the correct environment model.

1 Introduction

1.1 Bootstrapped perceptual representations in the COSPAL cognitive architecture

The aim of EU COSPAL[1] project is to create an open-ended cognitive architecture for real-world implementation via incremental perception-action learning [3]. Perception-action learning hence seeks to address the frame-related difficulties[2] associated with autonomous cognitive agents by the expedient of creating

[1] COSPAL is an acronym for 'COgnitive Systems using Perception-Action Learning'.
[2] The frame problem [7] refers to the open-endedness of logical predication associated with typical real-world actions. This is caused principally by the domain description being very much richer than that of the action domain.

Proceedings of the ICVS International Cognitive Vision Workshop – ICVW 2007
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany

perceptual representations *only* when they are capable of being differentiated by the agent's actions.

When, as in the COSPAL architecture, learning is open-ended (such that action competences can be built-up in a hierarchical fashion), the adoption of a perception-action paradigm ensures that the increasingly abstract and symbolic representations at the top-level of the hierarchy are always grounded in meaningful actions at the lower-level of the hierarchy. We thus spontaneously infer new perceptual categories in a manner that *simultaneously* allows for the continuous refinement of models in the objective domain, in a way that would be paradoxical or ill-founded for non perception-action learners[3]. We term this *cognitive bootstrapping* [1,2].

Ideally, we would like the higher-levels of representation to include formal structural entities as well as the stochastic labels associated with the existing COSPAL visual and motor-control systems. In this way, the system can infer appropriate perceptual entities in complex rule-based environments (for example, a game of chess), by autonomously calibrating the lower-levels of visual representation (such that, for instance, individual chess-squares and pieces are preferentially segmented out) on the basis of their significance for the inferred high-level action protocols (the rules of chess, in this case).

Our goal in the current paper is hence to implement an Inductive Logic Programming (ILP) module within the COSPAL architecture consistent with the cognitive bootstrapping ideal. This will involve an iterative three-stage process involving: (1) Induction of the logical rules underlying action feasibility, (2) Remapping of the perceptual variables to best represent the class of legitimate actions, (3) Active exploration of the environment on the basis of this representation.

We will show that the gradual iterative refining of the percept space so as to best represent feasible actions inherently accelerates the process of learning the environment protocols, giving the randomised exploratory actions an increasingly 'intentional' character.

1.2 Experimental Instantiation Within the Shape-Sorter Environment

The test domain of the COSPAL ILP module is thus a simulated three-dimensional 'shape-sorter' puzzle. Here, variously shaped pieces can be positioned freely around the puzzle's surface and also placed within unique holes corresponding to their shape; pieces can also be stacked. The active agent embodied within this environment is a robotic gripper arm capable of positioning itself anywhere in the volume above the board and shapes.

The actions (representing the *a priori* motor space) initially available to the simulated COSPAL agent are thus limited to positional translations of the

[3] Without the perception-action relationship object-model errors can simply be subsumed by the perceptual inference.

gripping arm, which is assumed to perform a 'grasping' action at the starting position of the attempted translation and a 'releasing' action at the final position of the proposed translation (we thus, at this stage, eliminate the possibility of object *rotation*). Because of the perception-action equivalence, this *a priori* motor-space corresponds exactly to the *a priori* perceptual domain, which is hence characterised as a discretised finite three-dimensional Cartesian space equipped with topological adjacency relations. Each discrete volumetric position is associate with a particular label (denoting occupancy by a particular class of object, though this concept is not yet representable by the agent).

Actions are consequently initially specified by the six-tuple instruction:

'$move(x1, y1, z1, x2, y2, z2)$' indicating a transition from position $(x1, y1, z1)$ to position $(x2, y2, z2)$, both defined by three-dimensional vectors.

The environment is initially modelled by the injective functional relationship between the Cartesian space (X, Y, Z) and the set of labels $\{L\}$. We will later seek to model the environment on the basis of *affordance*; the ability of the agent to permute this functional relationship. To do this we need to appropriately generalise the legitimate actions within this environment. Action *legitimacy* is hence determined in the most general and least environmentally-specific terms: by the success or failure of the action to do what was intended. Thus, we assess the legitimacy of the action '$move(x1, y1, z1, x2, y2, z2)$' on the basis of its *stability* (the final state must not undergo further changes not induced by the agent) and its *utility* (the final state must be different than the initial state). The movement from $(x1, y1, z1)$ to $(x2, y2, z2)$ must hence involve the gripping and releasing of an object at a location *in which it is supported*. This can only happen if an (unencumbered) object exists at $(x1, y1, z1)$ and a free position exists at $(x2, y2, z2)$, with a supporting surface immediately below it - at $(x2, y2, z2 - 1)$ on the assumption of a shape 'depth' of 1.

A supporting surface is thus any one of the following entities: the puzzle, any other shape, a hole that does not match the moved shape, or a hole that *does* match the shape, but has a different orientation to that of the shape itself. (Positions are discretised so that partial overlaps between object and holes are not permitted[4]). The subset of the $(|x1| \times |y1| \times |z1|)^2$ possible transitions within the *a priori* motor-space that are legitimately performable are thus approximately $|shapes| \times |x2| \times |y2|$ in number.

Given that the *a priori* percept classes existing prior to cognitive bootstrapping are positional occupancy labels, where positional relations are determined by certain prior adjacency and topology relations, we also require a corresponding *a priori* structure capable of determining relations between the individual labels. These take the form of class and relationship predicates capable of distinguishing: *positional occupancy labels*, *shape-labels*, *hole-labels*, *hole-shape label correspondences*, and *orientation labels* (these are hence in addition to the

[4] Obviously, a real physical shape-sorter puzzle would be more complex than simplified representation, permitting, for instance stable, but vertically-tilted states for the moved object; we are here attempting to ensure that legitimate transitions form a transparently closed class (a *group*, mathematically).

Proceedings of the ICVS International Cognitive Vision Workshop – ICVW 2007
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany

(x, y, z) *adjacency labels* associated with positional predicates). In specifying these predicates, it is important to appreciate that the prior perceptual structures have as yet no action-determined *meaning*; they act merely as label allocation functions.

In implementing cognitive bootstrapping in this domain our aim is, firstly, the determination of the legitimate transition rules (ie, the object model), and secondly the remapping of the *a priori* percept states such that *only the legitimate action state transitions are perceived*. That is, we would like to find a compact but maximally descriptive percept space in which all states are accessible by *performable* actions. We shall demonstrate that this maximally descriptive space is always of a higher-level of abstraction than the *a priori* space.

In alternating between perceptual remapping and exploratory action carried out *in terms of the inferred percepts* we shall hence also implement a particular instance of *active learning* [4], and, as such, will expect to achieve significantly faster learning within the objective domain (which is to say, faster convergence on the legitimate action states).

We therefore now turn to a description of the implementation of the simulated experimental environment in logical terms, and follow this with a description of the use of inductive logic programming (ILP) for concept generalisation, allowing the remapping of existing percepts into a more compact space in which all proposed actions are assumed to be achievable.

2 First Order Logical Implementation of the Shape-Sorter

It is evident that an inference system capable both of proposing novel exploratory actions and of evaluating their outcomes must be one of *generalisation*. Moreover, this generalisation is, at its highest level, inherently *relational* given the nature of the shape-sorter environment: stochastic generalisation is then limited to the lowest level of the perceptual hierarchy. For the present purposes, we shall assume ideal stochastic generalisation such that there is no ambiguity (and no redundancy) amongst the base perceptual classes (shape, position, etc). Our problem is thus purely one of *rule inference*[5].

Specifically, since the shape-sorter puzzle is protocol-based, our problem is the inference of action rules that are given in terms of general variables, for which a particular label constitutes a variable *instantiation*. We are hence implicitly considering a first-order logic system, in which a given move represents a logical *proposition* that may or may not be fulfillable in terms of the logical axioms describing the shape-sorter environment. This strongly suggests an implementation of the shape-sorter within an inductive environment such as *PROLOG*, within which the negation or affirmation of movement propositions with respect to the environment axioms is representable as a *goal*.

[5] In the full implementation, rule inference is permitted to directly influence the stochastic clustering of the lower hierarchical layer, such as by unifying clusters with identical logical relations in the manner of [5] (though the mechanism outlined in [5] does not undertake a comparable perceptual remapping phase).

We hence set out to define the shape-sorter puzzle in logical, rather physical terms, such that it becomes possible to later use Inductive Logic Programming to infer the logical axioms defining the system given only a few specific exploratory instances. This, in essence, is to define a *semantic parser* for the shape-sorter puzzle in PROLOG.

2.1 PROLOG Implementation

We thus define the shape-sorter protocol in terms of the *a priori* cognitive categories given earlier, which we shall render as the PROLOG predicates: $free\_position(X,Y,Z), is\_hole(X), hole\_shape\_match(A,B), orientation(X,O)$ and $position(A,X,Y,Z)$ (where A and B represent *entity labels*, X, Y and Z represent *ordinal position labels*, and O is an *angle label*). We also introduce an elementary topological relation applicable to each of three ordinates indicating directional adjacency: $inc\_x(X1,X2)$, $inc\_y(Y1,Y2)$ and $inc\_z(Z1,Z2)$, such that, for example, $inc\_x(X1,X2)$ is only satisfied when $X2 = X1 + 1$. (Angles and positions are hence both finite and discrete, being limited to 10 and $120 = |X| \times |Y| \times |Z| = 3 \times 8 \times 5$ possibilities, respectively). Again, we emphasise that these predicates are labelled so as to assist comprehension; there are as yet no action-determined meaning associated with the terms. Critically, these base percept categories have the potential to delineate higher-level concepts such as *the space above an object* A via concatenation, ie: $position(A,X,Y,Z)$, $inc\_z(Z,Z1)$, $free\_position(X,Y,Z1)$ (though this is has not yet been made explicit: this will be the aim of perceptual remapping). The logical rules thus correspond to the physical rules of the shape-sorter environment in an broadly intuitive fashion. The rules governing move legitimacy in this simplified shape-sorter are thus rendered in PROLOG as the three-clause sequence:

$$move(X1,Y1,Z1,X2,Y2,Z2) :- position(A,X1,Y1,Z1), inc\_z(Z1,Z3), free\_position(X1,Y1,Z3),$$
$$free\_position(X2,Y2,Z2), inc\_z(Z4,Z2), position(B,X2,Y2,Z4), not(hole\_shape\_match(A,B)), not(A == B).$$

$$move(X1,Y1,Z1,X2,Y2,Z2) :- position(A,X1,Y1,Z1), inc_z(Z1,Z3), free\_position(X1,Y1,Z3),$$
$$free\_position(X2,Y2,Z2), inc\_z(Z4,Z2), position(B,X2,Y2,Z4), is\_hole(B), hole\_shape\_match(A,B),$$
$$orientation(A,O1), orientation(B,O2), not(O1 == O2).$$

$$move(X1,Y1,Z1,X2,Y2,Z2) :- position(A,X1,Y1,Z1), inc_z(Z1,Z3), free\_position(X1,Y1,Z3),$$
$$position(B,X2,Y2,Z2), inc\_z(Z2,Z3), free\_position(X2,Y2,Z3), is\_hole(B), hole\_shape\_match(A,B),$$
$$orientation(A,O1), orientation(B,O2), O1 == O2.$$

(with commas separating *simultaneously* satisfied logical constraint conditions, and distinct clauses separating *alternative* logical satisfaction constraints.)

3 Inductive Logic Programming in the Shape-Sorter Domain

We now wish to construct a system capable of inferring a rule set such as the above from specific examples of exploratory moves along with their (positive or negative) outcomes. Since we are in the domain of first-order logic, we are consequently interested in *Inductive Logic Programming* [6]. A natural implementation of ILP for our application is Muggleton's *PROGOL*. PROGOL operates by constructing the *most specific clause* of the first of the set of positive examples from which we wish to construct the general rule. The most specific clause is the concatenation of all true predication applicable to this positive example, selected from the range of possible 'body predicate' mode declarations. Predicates are then randomly pruned from this clause giving rise to a more generalised set of clauses which are tested both for their consistency with the negative examples and their compression of the positive examples. The most effective of these is then selected as background knowledge and used to remove redundant positive examples, after which the process begins again with the first of the remaining positive examples.

Thus, we intend to perform exploratory actions arising from cognitive bootstrapping within the simulated environment defined by the PROLOG rules given in section 2.1, attempting inference of them via PROGOL. For the current demonstrative purposes, rather than considering temporal sequences or multiple random instantiations of a single simple puzzle configuration, we shall, in obtaining our test data, opt rather to perform single actions on a fixed, but large and varied, puzzle configuration (slightly simplifying the form of mode declarations).

4 Active Learning Via Cognitive Bootstrapping in the Relational Domain

In seeking to simultaneously infer optimal object and percept models we shall hence implement a system of *iterative alternation* between the exploratory and the environmental (object-model) inference phases. Cognitive bootstrapping then stands as an *intermediary* between these two phases. Specifically, it takes the current environmental inference (that is, the attempted inference via PROGOL of the shape-sorter PROLOG rules when given the cumulative outcomes of all of the previous exploratory moves), and seeks to redefine the percept space in a manner appropriate to this newly-assumed environmental model. This remapped percept space then, in turn, suggests a new set of exploratory moves (in effect, the percept remapping *re-parameterises* the environmental model), thereby testing *both* the environmental and perceptual hypotheses at the same time, while overcoming the potential paradox involved in their interdependent definitions. We now look at exactly how this perceptual remapping is achieved:

4.1 Remapping of the Percept Space

Suppose that the application of PROGOL to the cumulative exploratory data has given rise to the inference of a partially accurate rule. The following is a typical example of the sort of rule infered after four legitimate exploratory action examples have been collated (along with very many more negative exploratory action examples):

$move(X1, Y1, Z1, X2, Y2, Z2) : -position(A, X1, Y1, Z1), inc\_z(Z3, Z2), position(B, X2, Y2, Z3).$

(This corresponds to the constraint that an object must be placed on top of another object)[6].

We notice that this rule has introduced three new variables (A, B and $Z3$) beyond the existing six variables ($X1, Y1, Z1, X2, Y2$, and $Z2$) used to specify the *a priori* motor-space. As a consequence of the nature of PROGOL, the predicate terms within the body of the above clause must declared with a *specific input/output structure*. For instance, the body mode declaration for the 'position' predicate is : $-modeb(1, position(-entity, +xint, +yint, +zint))$, indicating that for a given 3-D positional input, a *single* entity class object label is given as output. However, we have so specified the mode declarations that there is also a '*position*' predicate body mode declaration given with *exactly opposite* input/output structure (which can, if necessary, be differentiated via an appropriate suffix). Furthermore, in consequence of the particular design of the shape-sorter logical protocols, this symmetry is common to *all* of the predicates that have both an input and an output (so that, for instance, an occupied position always defines a unique shape label, while a given shape label always defines a unique position). We have thus adopted a strictly *functional* definition of predication within the mode declarations.

This will not necessarily be the case within general logical environments; however, in the case of physical environments such as the shape-sorter this symmetry permits us to *invert* the input/output structure. Hence, visually rendering the clause I/O structure (as in figure 1) and reading the diagram from left to right, it becomes apparent that the six initial input variables are mapped to two final output variables. Consequently, reading the vertex structure from *right* to *left* after having imposed opposite input/output structure in the individual predicates permits us to see that the clause structure undergoes a transition from the two input variables A and B, to the six original variables. In so far as it is permissable to regard variable instantiations as *ordinates*, it is hence possible to re-parameterise the original six-dimensional space as a *two*-dimensional space characterising the space of possible moves. In doing so we have lost none of the posible instatiations of legitimate actions: we have merely removed all of the logical redundancy. This then is the proposed *percept* space, where we have, in choosing the observables as $\{A$ and $B\}$ rather than $\{X1, Y1, Z1, X2, Y2,$ and $Z2\}$, effectively re-conceived the percept space in the higher-level terms of *objects* and *surfaces* rather than the lower-level concept of *positions*. We thus redefine the six dimensional action space: $move(X1, Y1, Z1, X2, Y2, Z2)$ as the two-dimensional space: $move(A, B)$. Randomised actions in the reconsituted percept space are thus now of the 'put object A onto surface B' type, as opposed to the 'move gripper from $(X1, Y1, Z1)$ to $(X2, Y2, Z2)$' type; that is, they are much more 'intentional'. In algorithmic terms, this percept remapping is simply a case of establishing which of the newly introduced variables are *non-nested* with respect to the ensemble of sets of variables defined by the various predi-

[6] This is in fact sufficient to correctly eliminate the vast majority of the $(|x1| \times |y1| \times |z1|)^2$ proposable transitions in the *a priori* space.

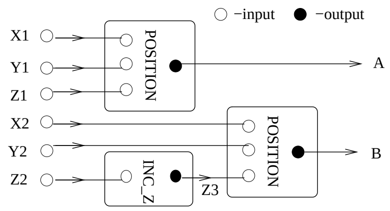

Fig. 1. Example schematic of clause structure.

cate groupings. This is the equivalent of determining which of newly introduced variables appears in only one of the predicate groupings when predicates with *only* input or output structures are excluded: individual clauses are assumed to occupy separate spaces.

4.2 Active and Passive Exploratory Phases

While the above method might thus be expected to increase the speed of convergence on the final object and percept models, it is evidently possible that it can cause convergence on a local, rather than global minimum, represented by an accurately inferred *subset* of the totality of permissable moves. Hence, we shall alternate the cognitive bootstrapping phase with a random exploratory phase that makes *no* high-level perceptual assumptions[7]. The active phase thus, in effect, acts to focus on those areas deemed permissable by the inferred rule such that data capable of falsifying it is obtained very much more quickly than would otherwise be the case (PROGOL requires only one instance to falsify a hypothesis). The random phase then acts to collect data that is indicative of general environmental rules, of which the active phase is perhaps investigating only a subset. The combination of the two approaches hence produces an exploratory method capable of rapidly ascending performance gradients, while at the same time undergoing random perturbations capable of finding alternative, perhaps more global, gradients to ascend.

As a calibration for the above method of alternation (which might be considered a primitive form of simulated annealing), we also provide a purely passive-learner in which PROGOL inference is applied *only* to random exploratory actions cumulatively derived from the *a priori* percept space *(X1, Y1, Z1, X2, Y2, Z2)*. For both types of learner, 10 exploratory actions are undertaken at each iteration.

[7] Note that there are potentially more efficient variants on this approach, such as embarking on a random exploratory phase only *after* active percept learning performance has reached a plateau (if some local criterion could be established to determine this, such as compressive capability with respect to the cumulative exploratory results). For this proof-of-concept demonstration, however, we opt for the most straightforward approach.

Proceedings of the ICVS International Cognitive Vision Workshop – ICVW 2007
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

5 Results and conclusions

5.1 Experimental findings

We give the average result of ten output runs (commencing after an initial rule-induction of 96% accuracy) in figure 2. The ratio of passive cycles to active cycles in the cognitive bootstrap learner is 5 to 1 (corresponding to 10 attempted actions during the cognitive bootstrap cycle followed by 50 actions during the random exploration cycle); this is compared with a purely passive learner. It is evident that the active learning procedure achieves convergence considerably faster than the passive learner, converging on a significantly higher accuracy figure at the extremity of the tested range.

Defining, more accurately, the respective absolute performance values on which the learners converge as the average performance value after they have come within 1 percent of their maximum values, we see that the performance figures are 99.57 percent for the active learner and 99.28 percent for the passive learner.

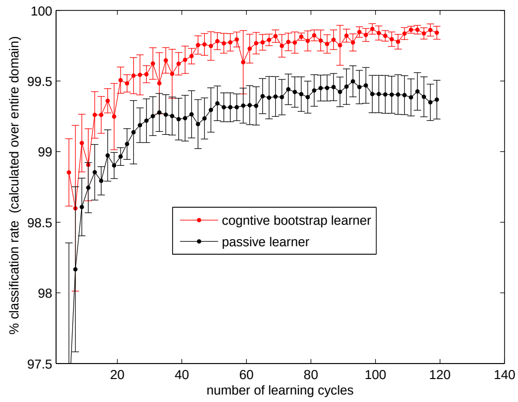

Fig. 2. Accuracy vs iteration number for the cogntive boostrap and passive learners.

5.2 Conclusions

The outlined experiment has thus demonstrated how it is possible to build a relational perception-action learner for the COSPAL architecture capable of simultaneously optimising a percept-domain while optimising its model of the external world described in terms of these percepts. Thus, cognitive bootstrapping aims to creates a space of perceived action possibilities that are *always* (in principle) realisable, and where redundant action possibilities are eliminated

Proceedings of the ICVS International Cognitive Vision Workshop – ICVW 2007
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

from perception. The outlined method hence constitutes an artifical realisation of Phenomenologists' goal (eg [8]) of relocating the concepts of *representation* and *symbolic meaning* in the interaction between an agent's capabilities and the world, as opposed having them specified by purely internal states (ie 'subjectively'), as they are in conventional machine vision.

In carrying-out this instantiation of relational cognitive bootstrapping in a COSPAL-like environment, we have also found evidence that, in so far as it may be regarded as form of active learning (that is, when the remapping of the percept space directly suggests novel exploratory actions), cognitive bootstrapping can give rise to significantly faster training within a perception/action domain.

Future work will involve coupling the system to the lower-level stochastic vision system such that high-level inferences can 'pre-filter' the lower level vision features so as to eliminate perceptual redundancy (as determined by the rule protocols) at these levels as well as the higher-levels (for instance, by meta-identification of logically indistinguishable predicate labels). In this way, once a system has begun to infer the rules of (say) a chess game via an existing set of visual primitives (colour segmentations), it can utilise these protocols to assist segmentation of these primitives in a manner that is more protocol-appropriate (say, by preferentially segmenting chess-pieces and board-squares).

Acknowledgements The work presented here was supported by the European Union, grant COSPAL (IST-2003-004176)[8]

References

1. David Windridge, Cognitive Bootstrapping: A Survey of Bootstrap Mechanisms for Emergent Cognition in Natural and Artificial Agents, CVSSP Technical Report VSSP-TR-2/2005, University of Surrey, UK, 2005
2. Bowden R, Ellis L, Kittler J, Shevchenko M, Windridge D., Unsupervised Symbol Grounding and Cognitive Bootstrapping in Cognitive Vision, (Lecture Notes in Computer Science, Vol. 3617, Sept 2005)
3. Gösta Granlund, Organization of Architectures for Cognitive Vision Systems, Proceedings of Workshop on Cognitive Vision, 2003, Schloss Dagstuhl, Germany
4. D. A. Cohn, Z.Ghabhramani, M. I. Jordan, Active learning with statistical models, Journal of Artificial Intelligence Research, 4–129, 145, 1996.
5. D. Magee, C. J. Needham, P. Santos, A. G. Cohn, and D. C. Hogg, Autonomous learning for a cognitive agent using continuous models and inductive logic programming from audio-visual input, Proc. of the AAAI Workshop on Anchoring Symbols to Sensor Data, 2004
6. Muggleton S., Inductive Logic Programming, Academic Press, 1992
7. J. McCarthy and P. J. Hayes (1969). Some philosophical problems from the standpoint of artificial intelligence. Machine Intelligence, 4:463-502.
8. M. Merleau-Ponty, Phenomenology of Perception, trans. Colin Smith, New York: Humanities Press, 1962.

[8] However, this paper does not necessarily represent the opinion of the European Community, and the European Community is not responsible for any use which may be made of its contents.