COMPUTING UNBOUND pdf

1: Galileo Unbound - David D. Nolte - Oxford University Press

Prof. Nigel Smart. Nigel Smart, Unbound Co-Founder, is a Professor at University of Bristol UK. He is a world-renowned expert in applied cryptography, and was the Vice President of the International Association of Cryptologic Research.

The Tesla M, based on the Fermi GF architecture anounced in , offered global memory bandwidth of up to Gigabytes per second and peak double-precision floating-point performance of Gigaflops. Actually achieving optimal performance on diverse GPU architectures can be challenging, since it relies on the implementation of carefully-crafted kernels that incorporate extensive knowledge of the underlying hardware and which take full advantage of relevant features of the CUDA programming model. This places a considerable burden on the CUDA developer seeking to port her application to a new generation of GPUs or looking to ensure performance across a range of architectures. Fortunately, many CUDA applications are formulated in terms of a small set of primitives, such as parallel reduce, scan, or sort. Before attempting to handcraft these primitive operations ourselves, we should consider using one of the libraries of optimized primitives available to CUDA developers. For example, CUB supports a set of device-wide primitives, which are called from the host, and in this regard, the functionality provided by CUB overlaps with Thrust to some degree. However, unlike Thrust, CUB also provides a set of kernel components that operate at the thread-block and thread-warp levels. Thread-block reduction â€" a simple CUB example A key feature of the CUB library, and one that makes CUB an attractive option for a wide range of performance-critical applications, is the fact that software components are not specialized for a particular GPU architecture or problem type. The library includes a templated BlockReduce class to perform reduction operations across a single thread block. It is declared as follows: Note that the binary operation that specifies the type of reduction being performed which, more often than not, involves computing the sum or the maximum or minimum of a data set is not included in the class declaration. Using the BlockReduce class and atomic operations, a kernel to compute the maximum value in an array of integers can be implemented as follows: The quantity and layout of this storage depend on the choice of algorithm, the type of the data, the number of threads per block, and the target GPU architecture. The optimal shared-memory configuration, which provides sufficient temporary storage and avoids unnecessary bank conflicts, is determined at compile time using the template arguments selected in the client code. However, the shared-memory configuration details themselves are hidden from the client application. On line 12 of our kernel, the BlockReduceT constructor which takes as an argument the temporary storage allocated above is called, generating a temporary object, which then invokes its Reduce method. This class is defined such that if maxObject is an instance of the class Max, then maxObject a,b returns the maximum of a and b. Other binary operations supported in CUB include binary addition, the binary min operation, and variants of max and min that identify the position of the first occurance of the maximum or minimum value in a data array. The result of a thread-block reduction is returned to the first thread of each block which has threadIdx. Finally, each thread block calls a single atomic operation to update the global maximum. Note that line 15 of the kernel assumes that the value pointed to by max is initialized to some minimum value before the kernel is launched. The latter algorithm is specialized for commutative binary operations such as the Max operation in our example, where the relative ordering of inputs does not affect the output , while the other algorithms also support non-commutative binary operators. The background to these algorithms is decribed in detail in a series of publications by Merrill and collaborators. As described in those papers, a core feature of the algorithms used in CUB is that they balance concurrency with serial computation in order maximize performance on GPU hardware. In contrast, earlier algorithms targeting GPU architectures tended to involve high levels of concurrency, where the number of logical threads assigned to a problem scales with the problem size. However, in reduction and scan calculations, logical threads have to share data and synchronize with each other, and the cost of this inter-thread cooperation scales with the amount of concurrency in the algorithm i. Performance can be improved by choosing a level of concurrency that ensures that the GPU hardware is fully utilized while minimizing communication and synchronization overheads. To understand how the CUB routines utilize serial processing, consider the raking block-reduction algorithms mentioned above. In these algorithms, after an initial step, which we discuss below, each thread in the block writes data to shared memory. At the end of this step, a single warp-width of data remains to be reduced, and one warp-level reduction completes the calculation. Further serialization can be achieved by having each thread in the thread block perform a serial partial reduction in registers at the beginning of the block-level reduction routine. To do this, we modify our reduction kernel as follows: In essence, CUB provides an outline of the reduction algorithm, but leaves performance-critical details, such as the exact choice of algorithm and the degree of concurrency unbound and in the hands of the user. These parameters can be tuned in order maximimize performance for a particular architecture and application. Since the parameter values are specified in the client application at compile time, this flexibility incurs no runtime performance penalty. The CUB library provides most benefit if integrated into a client-application auto-tuning procedure. In this case, on each new architecture and problem type, the client application would launch a series of short jobs to explore the CUB tuning space and determine the choice of template arguments that optimize performance. Among the other primitives implemented in CUB are block-wide data-exchange operations and parallel histogram calculations, and all of these implementations are flexible enough to ensure high performance in diverse applications running on a range of NVIDIA architectures. Well, that completes our brief introduction to the CUB library.

2: What is Unbound? Webopedia Definition

Note: Citations are based on reference standards. However, formatting rules can vary widely between applications and fields of interest or study. The specific requirements or preferences of your reviewing publisher, classroom teacher, institution or organization should be applied.

Correspondence regarding this article should be addressed to Gary C. Abstract Five experiments examined the nature of object representation. Participants made same-different judgments between two multipart 3-D objects, according to rules where either the object parts and their spatial relationship had to be considered role-relevant, RR or just the object parts role-irrelevant, RI. Results indicate that it was easiest to judge two identical and orientationally aligned objects according to either rule, followed by judging those that shared identical parts located in different positions according to the RI rule. It was most difficult to judge the latter according to the RR rule when they were misaligned by rotation. These findings lend support to the hypothesis that object representations at the image level, part level, or full structural description level may be computed and used for making same-different judgements. The implications of our findings for object recognition in general and the role of spatial attention in particular are discussed. There is a growing debate on whether object recognition is best understood in terms of representations based on structural description versus representations resulting from image-based normalization processes. According to theories based on normalization, objects are represented in view-specific formats such that their recognition often requires a normalization process e. The transformed representation of the object can then be matched to a model of the object stored in long-term memory e. It is likely both kinds of representation i. For example, Hummel and Stankiewicz have recently proposed a model of object recognition that represents shape in a hybrid fashion i. The synchrony serves to dynamically bind features into parts and parts to their relations. Although the explicit representation of parts and their relations by a collection of units called the Independent Geon Array, or IGA is hindered by improper synchronization of parts, the holistic representation a collection of units called the Substructure Matrix, or SSM does not depend on synchrony for binding, and consequently it can drive recognition even in the event of binding errors. The model as a whole can recognize objects in familiar views without attention and very quickly i. Decomposing an object into its parts and explicitly representing the relations among those parts takes more time because dynamic binding requires visual attention. When a geon was the target, the subsequent display possessed one of the following three configurations: When a geon assembly was the target, the target as well as the subsequent displayed items comprised a vertically oriented geon connected to a horizontally oriented geon in an above-below spatial relation. The display presented following the target assembly consisted of two geon assemblies with one of the three configurations analogous to those used when a single geon was used as target. Illusory conjunctions were estimated by the difference in errors committed in the illusory-yes and illusory-no conditions. Reliable conjunction errors were found for both the single geon condition and the geon-assembly condition. The short exposure time that Shyi and Cheng used for viewing the displayed items severely compromised the computation of the complete structural description i. The goal of the present study was to further shed light on the kind of representations that may be computed and used in the course of object recognition. In particular, we attempted to empirically identify conditions that would implicate the use of different kinds of representation, when participants were asked to make same-different judgments between pairs of multipart 3-D objects. Our hypothesis is that 3-D objects can be matched on the basis of one of the three levels of representation. The first level involves the output from analyzing the 2-D images. Note that this level of processing only gives rise to the constituent components or parts of an object. Its outputs do not include the spatial relations that each part or component has relative to other parts or components. The third level of processing has representations of full structural description as its output. Not only the parts or components are identified but also the spatial relations that each part has in relation to other parts. The three levels of outputs do not necessarily imply a fixed sequence of processing. Nonetheless, this hypothesis does imply an order of relatively difficulty and efficiency with which each level of representations can be computed, with the 2-D image level being the easiest and quickest, the full structural description level being the most difficult and time-consuming, and the part-level lying in between. The experiments reported here were designed to identify conditions where matching two multipart 3-D objects may rely upon representations computed at these three respective levels. There are a number of important features of SIAM which suggest a natural link between the model and that proposed by Hummel and Stankiewicz i. According to SIAM, for example, when structured scenes are compared, the parts of the scene must be aligned, or placed in correspondence, with parts from the other scene. Correspondences can include both the proper bindings between features e. Another important feature of the SIAM model is its emphasis on the time course of comparison, as an attempt to provide a unified account for the disparate results obtained from a variety of similarity tasks used in the past. The emphasis on time course in SIAM is consistent with the time required for dynamic binding, and thus structural descriptions, to be established. For the purpose of object recognition, the JIM2 model would suggest that image-like, view-specific representations are computed and used either when there is limited time or when the object to be recognized was shown in a highly familiar point of view. The computation of a full structural description would need more time than is required for the construction of a viewpoint-specific image. For the purpose of similarity comparisons, matches between local features would play a greater role when there is limited time available as is often the case with same-different judgment tasks. Global consistency among corresponding features would become increasingly influential as the time available increases as is often the case in similarity rating tasks. Experiment 1 Proctor and Healy asked participants to make same-different judgment between two 3-letter strings in accordance with one of the two rules. For the order-relevant rule, the strings would be classified same when the letters in each string were identical and occupied the same locations. For the order-irrelevant rule, the strings would be classified same as long as the letters in each string were identical, regardless of whether or not they occupied the same locations. Proctor and Healy found that it was much easier for participants to make judgments based on the order-relevant than the order-irrelevant rule, both in terms of reaction time and error rate. The difference was most evident in cases where an identical set of letters was rearranged in the two strings e. While they could easily respond different according to the order-relevant rule, participants could not as easily respond same according to the order-irrelevant rule. Goldstone and Medin reported a similar finding. Participants in their study were asked to make same-different judgments, according to either a role-relevant rule or a role-irrelevant rule, to a set of colored squares forming a cross shape. These rules were analogous to those used by Proctor and Healy ; for instance, according to the role-relevant rule, for the colored patterns to be judged same, the same-colored squares had to occupy the same locations within the cross shape. Like Proctor and Healy, Goldstone and Medin also found that on average, the error rate and response latency for "different" judgments was substantially lower for the role-relevant than for the role-irrelevant group. Such findings are hard to reconcile with the predominant view that visual processing begins by breaking down objects into features, and originally features are processed in an unbound manner Treisman, , ; Wolfe, One possible reason that performance for role-relevant RR is better than for role-irrelevant RI is that participants may be able to respond as though sensitive to binding because they are using an image-like representation. This has limitations, however. If the objects were misaligned with respect to orientation, for instance, then the image-like representations would not be as dependable as when the objects were aligned in orientation. The main goal of Experiment 1 therefore was to identify conditions under which role-relevant judgments are easy to make and those where role-irrelevant judgements are easy to make. One hundred and nineteen undergraduate students from Indiana University served as participants in order to fulfill a course requirement. Fifty-eight and 61 participants were in the role-irrelevant and role-relevant groups, respectively. Objects were designed using three-dimensional rendering software Infini-D R2. Each object consisted of three components -- a large cut-off cone and two smaller geometric shapes attached to the upper left and lower right portions of the cone. Sample stimuli are shown in Figure 1. The smaller components were separated by approximately degrees in depth. The smaller shapes were chosen from the following set: These shapes were created so as to vary on two orthogonal geon aspects -- curvature and taper. Insert Figure 1 about here A set of 12 objects was created by combining every possible pair of small shapes under the constraint that no shape was paired with itself see Figure 1. Thus, letting the letters A, B, C, and D represent the four small shapes and letting the order of the letter indicate whether it is on the left or right side of the object, the following objects were constructed: This set of 12 objects was replicated at three different orientations: All rotations were performed in the plane of the screen, as shown in the lower panel of Figure 2. Each object was 6. On each trial, participants were shown two objects and were instructed to respond as to whether the two objects were the same or different. The role-irrelevant and role-relevant groups were given different instructions for deciding whether two objects were the same. For the role-relevant group, for two objects to be the same, they had to possess the same shapes, and these shapes had to be in the same positions within the two objects. Thus, the two objects under the 2 MIPs column in Figure 2 are the same for the role-relevant group, but the objects in the 2 MOPs column are different because they have the same shapes but in different positions. If two objects were identical but differently rotated, participants were instructed to respond "same. Thus, participants in this group should respond "same" when presented with the objects related by 2 MOPs, but the objects related by 1 MIP would be different because they do not have all of their shapes in common. There were four different types of relation between the two compared objects, each of which is represented in Figure 2. These four trial types are: On 2 MIPs trials, the two objects were identical, although one may have been rotated with respect to the other. On 2 MOPs trials, the objects had the same shapes, but the positions of the two small shapes were swapped. On 1 MIP trials, the objects shared one shape in addition to the large common base and this shape occupied the same position within the objects prior to any rotations. On 1 MOP trials, the objects shared one shape but this shape occupied different positions in the two objects. The correct responses for the role-irrelevant and -relevant groups were identical for all trial types except for the 2 MOPs trials, which were properly called "same" for the role-irrelevant group and "different" for the role-relevant group. When instantiating each of the four trial types, the standard object was chosen randomly from the set of 12 objects, and then was altered in one of the four ways. When the object could be transformed in more than one way to implement a particular trial type, the transformed object was selected at random. On each trial, a standard object was presented at a random location on the left half of a Macintosh 2CI computer screen for 1. The object was then removed, and after a Participants responded "same" by pressing the "s" key and "different" by pressing the "d" key. Participants were required to make their "same" or "different" response within ms of the offset of the second object. Thus, the offset of the second object can be viewed as a cue for participants to give a response, although it was also acceptable for participants to make their response prior to the offset of the second object. Participants received instantaneous feedback as to whether their response was correct and were shown the correct response. If participants made a response after ms of the offset of the second object, the word "Overtime" was displayed. Participants saw trials in all, consisting of four blocks of 72 trials. Each block consisted of a set of 12 trials repeated six times by factorially combining two levels of orientation same or different and three deadline levels. Within the set of 12 trials, there were several repetitions of each of the four trial types. In order to give role-irrelevant and role-relevant groups an equal number of same and different trials, the number of 2 MIPs, 2 MOPs, 1 MIP, and 1 MOP trials were 6, 2, 2, and 2 for the role-relevant group and 3, 3, 3, and 3 for the role irrelevant group. The ordering of all trials was randomized. The rotation of the standard object was randomly selected from 0, , and degrees. On "same orientation" trials, the identical amount of rotation was assigned to the transformed object.

COMPUTING UNBOUND pdf

3: Computing Representations for Bound and Unbound 3-D Objects Matching1

Enter your mobile number or email address below and we'll send you a link to download the free Kindle App. Then you can start reading Kindle books on your smartphone, tablet, or computer - no Kindle device required.

He is a world-renowned expert in applied cryptography, and was the Vice President of the International Association of Cryptologic Research. In the past, Nigel worked at Hewlett-Packard Laboratories developing advanced encryption technologies. He has also been involved in developing many standards, and has worked with both industry and government on applying cryptography to solve critical security problems. January 7, What is Quantum Computing? In Part I of our blog series on the implication of quantum on cryptography, I described what a quantum bit was, and what operations could be performed on such data. Since their inception a lot of work has been carried out in determining algorithms for quantum computers. A classical problem in computing is given a list X of N items to search the list for an item which has a property, which we will call P. If we suspect only one such x exists then this will take N steps. Cryptographically think of X being the set of keys for AES and P x being the function which tests whether the key maps one plaintext to a given ciphertext. Quantumly, however we can do a lot better. This means that quantumly the AES cipher with bit keys only provides 64 bits of security as opposed to bits of security. For block ciphers this means that if we are wishing to protect against quantum attacks we need to double the key sizes. A hash function is designed to make finding collisions hard. This is needed in digital signature schemes which hash a message before signing. Since if you can find a collision x,y , then you can pass off a signature on the message x as a signature on the message y. Classically, if the output of the hash function is an element from a set X of size N, and the output is essentially random, then the best algorithm to find a collision will take sqrt N steps. Thus if our hash functions are to be quantum secure we need to use hash functions with bigger output length. What this algorithm actually does is find the length of a cycle in a finite abelian group. All sorts of interesting cryptographic problems can be reduced to finding a cycle length in a finite abelian group. The most important being factoring numbers and finding discrete logarithms. The problem is that factoring and discrete logarithms form the basis of all public key cryptographic systems in use today. Thus, the development of a quantum computer would render all deployed public key algorithms instantly insecure. So in summary, if a quantum computer was ever built we would have to increase the key lengths of our symmetric ciphers such as AES say to use AES as opposed to AES , and we would need to find new public key algorithms.

4: RE: Computing values in Unbound Columns

From the Publisher: The exercises in this workbook provide beginning and advanced exercises for the most popular, accessible, and powerful software packages available for the Macintosh, to help students master both applications software and programming.

5: computing â€“ Linux Unbound

I'm trying a very simple Excel like function. I have a form with a datgrid view bound to a dataset. I retreive and display 3 columns from the dataset ColA,ColB And ColC the same is displayed in my gridview.

6: Introducing CUDA UnBound (CUB) | Microway

Computing Unbound: Hands-On Exercises for the IBM PC, with Two Optional Exercises for the Macintosh by Denise S Kiser starting at $ Computing Unbound: Hands-On Exercises for the IBM PC, with Two Optional Exercises for the Macintosh has 1 available editions to buy at Alibris.

7: Protect and Control Cryptographic Keys | Unbound is Transforming Trust

Unbound Digital Computer Repair and Networking - W Oakland Ave Ste 7, Johnson City, Tennessee - Rated based on 26 Reviews "When a virus hit.

8: What is Quantum Computing? | Unbound

At AWS, we've crafted a range of HPC solutions that allow you to unbound your innovation. We combine the latest compute, networking, storage, security, cloud orchestration, and visualization technologies, with a vibrant partner and ISV community.

9: Unbound | Charity Ratings | America's Most Independent Charity Watchdog | CharityWatch

Unbound protects secrets such as cryptographic keys, credentials or private data by ensuring they never exist in complete form so that you can perform trust-contingent operations anywhere. Eliminate the need for dedicated hardware, without compromising security or privacy, and adopt an agile approach to key management.

COMPUTING UNBOUND pdf

Practice in Biology (Practice In.) The correspondence of Isaac Basire, D. D. Bach chorales sheet music The horses of the Royal Canadian Mounted Police Nuclear Safety Research and Development Act of 1980 Recovery in Patent Infringement Suits The Christmas Almanac (Little Big Book) Being Religious Interreligiously The Mentchkins make friends Ringkasan novel laskar pelangi The Effects of Using Fire/t175 The father of British Canada Catalogue of stills, posters and designs Applied mineralogical thermodynamics The Passion of Maggie Higgins A Darker Shade of Pale Against circumcision Michael Kimmel Trigonometry tutorial step by step Roasting the Swan of Avon Curbside recycling increases when people do not have to sort their trash Susan Kinsella Medium (Australian Centre for Egyptology Reports) The adventures of a bear called Paddington Snakes of south-central Texas Hearings regarding Executive Order 13233 and the Presidential Records Act How to be outrageously successful with women Epilepsy syndromes Textual orientations How to Make Million in Real Estate in Three Years Starting with No Cash Bureaucrats and regulators: Delivering services and protecting consumers Jurisprudence notes ccs university Index to Vital Data in Local Newspapers of Sonoma County, California, Volume V Preamble 9: stepping away from the marriage of knowledge and production Skype brand identity book and guidelines Bible in 30 days Sbi neft form 2017 Fluorescent substrates for polyamine catabolic enzymes and transport Koichi Takao and Akira Shirahata Proposed legislation to authorize emergency assistance for refugees and displaced persons in and around I Rodger Latimers mistake Silver and Banqueting in Malta College applications step by step