

ПРИЗМА - Генератор Систем Понимания Изображений

И.В.Грибков, А.В.Захаров, А.В.Коганов, П.П.Кольцов,
Н.В.Котович, А.А.Кравченко, А.С.Куцаев, В.К.Николаев

Научно-Исследовательский Институт Системных Исследований РАН
Москва, Россия

Аннотация

В процессе разработки систем понимания изображений формируется пакет методов сегментации, формирования характерных признаков, идентификации объектов, анализа сцен и других инструментальных средств. В данной работе предлагается подход, позволяющий объединить эти средства в единую интегрированную систему, обладающую свойствами визуального программирования. Предлагаемая система *Призма* позволяет по заданному набору изображений и эталонов подобрать подходящие для него методы обработки, идентификации, параметры этих методов и сгенерировать специализированную и отчуждаемую от *Призма* систему понимания изображений.

Ключевые слова: Системы понимания изображений, ассоциативная память, эталоны, методы обработки.

1. ВВЕДЕНИЕ

Рассматривается задача обработки и понимания изображений в следующей постановке: система понимания изображений - СПИ снабжена набором эталонных изображений, представленными специфическим для СПИ образом, и задача заключается в том, чтобы на предъявленном изображении находить все присутствующие на нем эталонные объекты и соотношения между ними. Такими системами являются:

- Системы обнаружения объектов на телевизионных и тепловизионных изображениях.
- Системы распознавания машинописных и рукописных текстов.
- Системы обработки аэрофотосъемки.
- Системы обработки траекторий элементарных частиц и многие другие.

Практически любая СПИ может быть представлена как система, состоящая из следующих блоков:

- Панель управления.
- Преобразователь сигналов панели управления во внутренние параметры системы.
- Блок формирования представления изображения.

- Блок идентификации.
- Память системы.

Существующие СПИ характеризуются большим разнообразием этих блоков, причем различия часто носят принципиальный характер. К настоящему моменту не существует ни теоретических, ни практических подходов, позволяющих рассчитывать на разработку и реализацию универсальной системы понимания изображений, то есть такой системы, которая после настройки способна решать произвольную, поставленную перед ней задачу понимания изображений. Единственной такой системой является система зрения человека. К сожалению, она все еще недостаточно изучена, а те факты, которые известны, и попытки их интерпретировать и использовать заставляют сомневаться в том, что можно построить аналогичную систему на основе вычислительных устройств. С другой стороны ясно, что не существует практических задач для решения которых требовалась универсальная СПИ.

Вместе с тем в ряде областей, например в перечисленных выше, реализованы достаточно надежно работающие СПИ. При этом, с одной стороны структуры таких систем подчиняются вышеприведенной схеме, а с другой - процессы их разработки имеют одинаковые этапы. Это позволяет ставить задачу разработки системы, *производящей специализированные системы понимания изображений.*

2. АНАЛОГИЧНЫЕ РАБОТЫ

В последние годы рядом авторских коллективов - [2]-[11] интенсивно разрабатываются интегрированные системы понимания изображений. Такие системы получили закрепившееся название «Среда Понимания Изображений» - IUE - Image Understanding Environment. Плодотворность интегрального, синтетического подхода осознана многими, приведем в качестве примера мнение доктора G. Simpson (University of Southampton): «Ideally, we would be able to integrate many different image understanding technologies into a common framework and exploit the synergies among them to get the best possible information out.». Некоторое отличие нашего подхода заключается в том, что *Призма* проектируется не как среда понимания

изображений, а как среда для разработки систем понимания изображений.

В ряде работ по автоматизации параллельного программирования [12]-[15] предлагаются так называемые гибридные системы, реализующие подход при котором средствами визуального программирования в единое целое интегрируются разнородные элементы - как программы, написанные на разных языках, так и различные вычислительные средства. В *Призме* реализована примерно эта же схема: графический интерфейс скрывает от пользователя уровень программирования, и позволяет сосредотачиваться на специфических для формируемой СПИ проблемах.

3. ОБЩАЯ СТРУКТУРА СИСТЕМЫ

Структурно система *Призма* состоит из следующих элементов:

1. Базовые структуры системы
 - 1.1. Представления изображения.
 - 1.2. Характеристики изображения.
 - 1.3. Методы обработки - преобразования и идентификации.
 - 1.4. Параметры методов обработки.
 - 1.5. Цепи обработки.
2. Язык описания цепей обработки.
3. Ассоциативная память.
4. Базовая библиотека обработки изображений.
5. Блок формирования цепей обработки.
6. Блок настройки параметров цепи обработки.

3.1. Представления изображений.

Представлением изображения будем называть способ хранения изображения или результата его обработки в памяти. Можно сказать, что представление изображения это C++ класс, включающий в себя всю информацию, относящуюся непосредственно к изображению. При этом представление изображения может содержать в себе информацию о нескольких изображениях как таковых, например два изображения,

составляющие стереопару. В *Призме* существует некоторое очень небольшое количество базовых способов представления изображения, и от любого иного представления требуется, чтобы он имел два метода: *export* и *import*, которые позволяют преобразовывать данное представление в одно из базовых и обратно. Ясно, что в таком случае каждый формат может быть в два приема преобразован в любой другой, а при существовании N различных форматов для этого требуется $2*N$ конвертеров, при этом каждое представление изображения приносит с собой в систему два конвертера, необходимых для того, чтобы стать общей ее частью.

Пространство представлений изображения будем обозначать символом R , а сами представления - буквами r или R , возможно с индексами.

Эталонные изображения являются для *Призмы* входной информацией, поступающей вместе с тем набором изображений, для которого формируется СПИ. Эталонное изображение - это идеальный результат обработки изображения некоторым методом. Например, для методов сегментации эталонными изображениями являются границы выделяемого объекта. Процесс получения эталонных изображений трудоемок, однако без них совершенно невозможно количественно оценивать качество работы СПИ - см, например [1]. Изображения снабженные эталонами называются в дальнейшем аннотированными.

Пространство эталонных изображений будем обозначать символом E , а сами представления - буквами e или E , возможно с индексами.

3.2. Характеристики изображения.

Характеристики изображения делятся на внешние и внутренние. К внешним относятся те, которые не могут быть восстановлены по изображению. Такими характеристиками являются: условия получения изображения, тип изображения, условия съемки, прибор с помощью которого получено изображение и так далее.

Внутренние характеристики могут быть вычислены по изображению. К ним относятся: плотность распределения яркости, ее математическое ожидание и дисперсия, плотность распределения производных яркости и их статистические характеристики, спектр Фурье изображения и т. д.

Невозможно предусмотреть заранее все разнообразие как внешних, так и внутренних характеристик изображения, поэтому при манипулировании с ними в *Призме* используется следующий подход: вводится класс *BaseFeature*, который содержит поля, характеризующие тип изображения и функции сравнения типов.

Далее, каждая характеристика изображения, появляющаяся в *Призме* должна быть оформлена как наследник этого класса и обязательно иметь метод

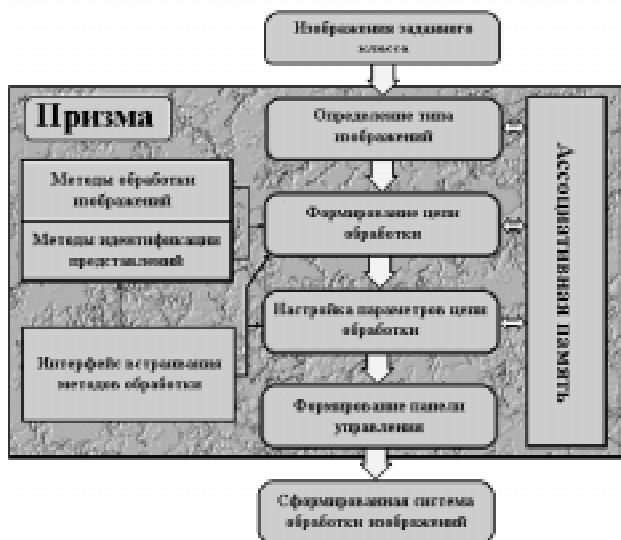


Рисунок 1
International Conference Graphicon 1998, Moscow, Russia, <http://www.graphicon.ru/>

float featcmp(NameFeature list),*

который сравнивает два экземпляра характеристик изображения одного и того же типа. Таким образом, характеристики изображения образуют дерево характеристик, так что двум характеристикам может быть сопоставлена степень их близости.

Пространство характеристик изображения будем обозначать символом F , а сами характеристики - буквами f или F , возможно с индексами.

3.3. Методы обработки.

Методы обработки делятся на три класса: преобразования изображений, методы идентификации и вспомогательные.

Преобразованиями называются функции вида:

int methTran(ReprImage Rinp, MethParam* P, ReprImage* Rout);*

где $Rinp$, $Rout$ - входное и выходное представление изображений, а P - параметры метода. На методы обработки, включаемые в систему, накладываются минимальные требования.

Методами идентификации будем называть функции вида:

float methIden(ReprImage* r, int num, ReprImage* R)*

где r - представление изображения, R - массив в количестве num представлений изображений (эталонных), а возвращаемое значение является массивом мер сходства между r и элементами R . Такие массивы будем называть идентифицирующими.

Пространство методов обработки изображения будем обозначать символом M , а сами представления - буквами m или M , возможно с индексами.

3.4. Параметры методов обработки.

Параметры методов должны быть оформлены в виде класса, в котором присутствуют функции экспорта и импорта параметров и вычисления расстояния между различными значениями.

Эти функции необходимы для того, чтобы параметры различных методов могли рассматриваться как единое пространство, в частности, чтобы был возможен универсальный алгоритм построения параметров метода, использующий подмножества параметров других методов и т. д.

Пространство параметров будем обозначать символом P , а сами представления - буквами p или P , возможно с индексами.

3.5. Цепи обработки.

Цепью обработки называется граф с двумя выделенными вершинами - входной, в которой находится обрабатываемое изображение и выходной - в

котором находится результат обработки. Каждый узел этого графа является или методом преобразования или методом идентификации. Ребра графа показывают как между методами происходит передача параметров. Цепь обработки и параметры методов формируются с помощью простого и удобного графического интерфейса, предоставляемого системой.

Цепи обработки образуют пространство, обозначаемое C .

3.6. Ассоциативная память.

Ассоциативная память - это массив пар $\langle f, c \rangle$, где $f \in F$, $c \in C$. На первом этапе обработки для каждого изображения из заданного набора строится его внутренний тип, в результате чего формируется массив пар $\langle r, t \rangle$ ($r \in R$, $f \in F$, f может содержать и внешние характеристики изображения). Для каждой такой пары $\langle r, f \rangle$ в ассоциативной памяти отыскивается пары $\langle f^*, c \rangle$, такие что расстояние между f и f^* минимально. Таким образом формируется набор цепочек C^* , отправляясь от которого, пользователь может начать формировать цепь обработки оптимальную для заданного массива изображений. После того, как для заданного набора изображений построен набор оптимальных цепочек, в ассоциативную память добавляется соответствующий набор пар $\langle f, c \rangle$.

4. ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС

В режиме формирования цепи обработки пользователь имеет возможность сформировать цепь обработки и выбрать для каждого метода соответствующие параметры, являющиеся начальными приближениями и уточняемыми при работе в режиме настройки параметров. Примерный вид окна обработки в момент задания начальных параметров цепи обработки показан на рисунке 2. С правой стороны окна открыт диалог ввода параметров.

4.1. Дерево методов обработки

Методы преобразования изображения естественным образом образуют дерево. На первом его уровне



Рисунок 2

находятся группы методов, которые отличаются друг от друга типом изображений, к которым они могут применяться: обработка отдельного изображения, пары изображений, серии кадров. На следующем уровне методы разделяются на группы в соответствии с применяемыми способами обработки: статистические, фильтрационные, морфологические, фурьеподобные и т. д. На третьем уровне находятся конкретные реализации данного способа обработки изображения. Система *Призма* предоставляет пользователю доступ к методам обработки с учетом этой естественной структуры.

4.2. Окно ассоциативной памяти

При включении соответствующей опции в правой нижней половине окна открывается панель ассоциативной памяти, на которую система помещает цепи обработки, накопленные системой к этому моменту, обрабатывавшие наборы изображений, наиболее близкие к текущему. Эти цепи упорядочены по степени близости к текущему набору изображений.

4.3. Формирование цепи обработки

На правой верхней панели изображается формируемая цепь обработки. Пользователь имеет возможность начать формирование цепи обработки с нуля или скопировать наиболее подходящую цепь из окна ассоциативной памяти. В любой момент он может добавить/удалить выделенную вершину цепи обработки.

Для каждого метода обработки пользователь должен задать начальные значения его параметров. Это тоже

может быть сделано либо с помощью копирования подходящих параметров из окна ассоциативной памяти, либо непосредственным вызовом диалога, которым должен быть снабжен каждый метод обработки при включении его в систему, либо установкой параметров по умолчанию.

4.4. Прогон и отображение результатов

После того как цепь обработки сформирована, пользователь переходит в режим настройки ее параметров. В этом режиме выполняется поиск оптимального набора параметров.

Пользователь может выбрать автоматический или ручной вариант и в каждом из них организовать свою работу разными способами.

В ручном режиме процесс выбора оптимальных параметров пользователь берет в свои руки и выполняет прогон за прогоном, изменяя параметры по своему усмотрению.

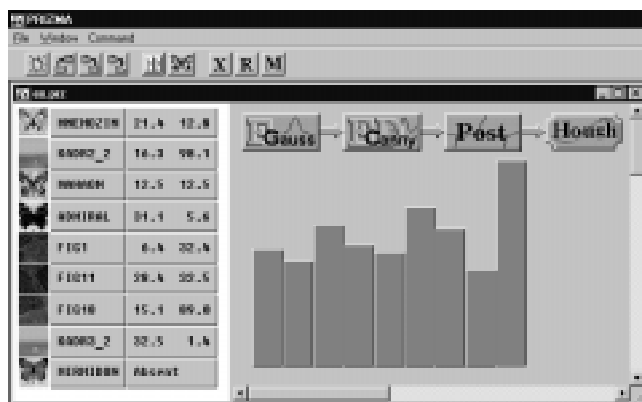


Рисунок 3

В автоматическом режиме пользователь выбирает некоторый метод поиска оптимального решения, запускает его и вплоть до завершения процесса остается в роли наблюдателя. Существует возможность переключения с режима на режим, приостановки итерационного процесса и изменения параметров или методов оптимизации.

При работе с аннотированными обучающими изображениями имеется возможность отслеживать ошибки, появляющиеся на узлах обработки, изучать скорость накопления ошибок, их зависимость друг от друга. Но самое главное - появляется возможность оптимизировать не всю цепочку в целом, а ее отдельные этапы, что эффективнее.

4.5. Итерационный характер настройки, кластеризация

Если в процессе выбора цепи и настройки ее параметров не удастся подобрать параметры для всего заданного набора изображений, *Призма* предоставляет возможность разделить все изображения на части и для

каждой из них отдельно подобрать цепь обработки и ее параметры. Наиболее естественной стратегией является отделение от всего потока изображений той части, для которой получаются удовлетворительные результаты, дообучение этой группы изображений в отсутствие остальных и переход к обработке оставшейся части. Именно при такой последовательности действий *Призма* предоставляет пользователю наиболее удобные средства, вплоть до автоматического выполнения процесса кластеризации.

4.6. Работа с отдельным изображением

В ситуациях, когда остается какое-то количество изображений, которые не удастся обработать, система *Призма* предоставляет пользователю возможность выделить интересующий его фрагмент изображения и рассмотреть каждый этап его обработки детально как в графическом, так и в текстовом виде и сохранить в выбранном виде эту информацию. Если такое изображение является аннотированным, то в дополнение к этому пользователь может получить изображение, на котором результаты обработки сравниваются с эталоном.

5. МЕТОДЫ ПРИЗМЫ

В настоящий момент в систему *Призма* интегрированы следующие методы обработки изображений:

1. Обработка отдельных изображений выполняется по классической схеме: предобработка, сегментация, выделение характерных признаков, сравнение с эталоном. При этом используются следующие методы.
 - 1.1. Фильтры Гаусса, J.Canny [16], Marr & Hildreth [20]-[22], Gabor [31]-[33], сохраняющие значения производных Butterworth-фильтры I. Weiss [23]-[26], адаптивные фильтры [28]-[30].
 - 1.2. Методы определения границ J.Canny [16], методы растущих областей, локально-многоканальный определитель границ M. Fleck [27], методы активного контура.
 - 1.3. Методы идентификации: преобразование Хау [18], методы основанные на преобразовании Фурье, методы нормализации, супрематическое и лучевое разложения.
2. Обработка стереопар. Целью анализа стереопары изображений является получение карты дальности. Этот подход иногда позволяет выделять неоднородно окрашенные объекты как одно целое.
3. Выделение движущихся объектов как на неподвижном, так и на подвижном фоне.

6. РЕЗУЛЬТАТЫ И ВЫВОДЫ

Система *Призма* была апробирована на трех наборах изображений: тепловизионные изображения, изображения местности, полученные с помощью аэрофотосъемки и набор изображений бабочек. Эксперименты, проведенные с системой позволяют сделать следующие выводы:

- Признаки изображений, накопленные в ассоциативной памяти системы, позволяют четко различать изображения каждого из этих классов.
- Для каждого из перечисленных классов удается построить цепь обработки и подобрать для нее оптимальные параметры.
- При предъявлении нового набора изображений, система с помощью ассоциативной памяти находит цепи обработки, ближайшие к предъявленному набору, и соответствующие параметры являются хорошим приближением к оптимальным значениям.

В целом предлагаемый подход показал свою жизнеспособность и эффективность.

В дальнейшем в *Призму* предполагается включить нейросетевые методы обработки изображений.

ЛИТЕРАТУРА

- [1] A.Hoover, G.Jean-Baptiste, X.Jiang, P.J.Flynn, H.Bunke, D.B.Goldgof, K.Bowyer, D.W.Eggert, A.Fitzgibbon and R.B.Fisher. An Experimental Comparison of Range Image Segmentation Algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol 18, No 7, pp. 673-690, 1996.
- [2] C.Kohl, J.Hunter, C.Loisselle, Towards a Unified IU Environment: Coordination of Existing IU Tools with the IUE, ICCV95(2-7).
- [3] T.E.Boult, S.D.Fenster, and J.W.Kim, Dynamic Attributes, Code Generation and the IUE, ARPA94(I:405-422).
- [4] D.T.Lawton, D.Dai, M.A.Frogge, W.R.Gardner, H.Pritchett, A.T.Rathkopf, I.Smith, and G.Ten, The IUE User Interface, DARPA93(289-300).
- [5] J.Dolan, C.Kohl, R.Lerner, J.L.Mundy, T.Boult, J.R.Beveridge, Solving Diverse Image Understanding Problems Using the Image Understanding Environment, ARPA96(1481-1504).
- [6] I.B.Gourevitch, N.S.Polikarpova, and Y.I.Zhuravlev, Logical Models of Images and Recognition Operators for Image Understanding Environment, SCIA97.
- [7] J.L.Mundy, The Image Understanding Environment Program, IEEE EXPERT(10), No. 6, December 1995, pp. 64-73.
- [8] R.A.Lerner, The Image Understanding Environment Progress Since IUW'96, DARPA97(715-722).
- [9] C.Kohl, J.L.Mundy, The Development of the Image Understanding Environment, CVPR94(443-447).

- [10] J.L.Mundy, and G.Ten, The Image Understanding Environment: Overview, DARPA93(283-288).
- [11] K.E.Price, and G.Ten, The Image Understanding Environment: Image Features, DARPA93(311-315).
- [12] A.Beguelin, J.J.Dongarra, G.A.Geist, R.Manckek, V.S.Sunderam, Visualization and Debugging in a Heterogeneous Environment, IEEE Computer, 26(6) (1993) 88-95.
- [13] J.C.Browne, M.Azam and S.Sobek, CODE: A Unified Approach to Parallel Programming, IEEE Software, 6(4) (1989) 10-18.
- [14] P.A.Suhler, J.Biswas, K.M.Korner, J.C.Browne, TDFL: A Task-Level Dataflow Language, Journal of Parallel and Distributed Computing, 9 (1990) 103-115.
- [15] V.A.Guarna, Jr, D.Gannon, D.Jablonowski, A.D. Malony and Y.Gaur, Faust: An Integrated Environment for Parallel Programming, IEEE Software, 6(4) (1989) 20-26.
- [16] J.Canny, "A Computational Approach to Edge Detection," *IEEE Pattern Anal. Machin. Intell.*, vol PAMI-8, pp 679-698, Nov 1986.
- [17] J. Illinworth and J.Kittler, A survey of the Hough transform, *Comput. Vision. Graphics Image Processing*, vol 44, no 1, pp. 87-116, 1988.
- [18] È.Á. Æðéâéíâ, À.Á. Çàðàðíâ, Ì.Ì. Êñüüíâ, Í.Î. Êîðíâðè-, Á.Ê.Ïééíéâââ. Ìðââíáðââíðèâ è ðàññèçíâââíéâ äâóíâðíúò èçíâðâæâíéé. Á ñâðíéèâ "Ðàññèçíâââíéâ. Ìâéðíñâðè. Æèððòâèüíâÿ ðââèüíñòü", *Áñðíñü èðâððíâðèèè*, Ìñéââ 1997.
- [19] D.C. Marr and E. Hildreth, "Theory of edge detection," Proc. Roy. Soc. London., vol B 207. pp 187-217, 1980
- [20] E.C.Hildreth, "Implementation of a Theory of Edge Detection," M.I.T.A.I. Tech. Rep. TR-579, 1980.
- [21] E.C.Hildreth, "The Detection of Intensity Changes by Computer and Biological Vision System," *Comput. Vision, Graphics, and Image Processing*, vol. 22, pp. 1-27, 1983
- [22] I.Weiss, "Projective invariant of shapes," in Proc. DAPRA Image Understanding Workshop, Cambridge, MA, 1988.
- [23] I.Weiss, "Noise resistant invariants of curves," *IEEE Pattern Anal. Machin. Intell.*, vol 15, pp 943-948, Sept. 1993.
- [24] I.Weiss "High-Order differentiation filters that work," *IEEE Pattern Anal. Machin. Intell.*, vol 16, pp 734-739, July 1994.
- [25] I. Weiss, "Higher order differentiation filters that work," Univ. of Maryland CS-TR-2639, 1991.
- [26] M.M.Fleck, "Multiple Width Yield Reliable Finite Differences," *IEEE Pattern Anal. Machin. Intell.*, vol 14, pp. 412-429, April. 1992.
- [27] H.Jeong and C.I.Kim, "Adaptive Determination of Filter Scales for Edge Detection," *IEEE Pattern Anal. Machin. Intell.*, vol 14, pp 579-585, May 1992.
- [28] V.Torre and T.Poggio, "On edge detection," M.I.T. Artificial Intell. Lab., Cambridge, MA, Rep. AIM-768, 1984.
- [29] J.Babaud, A.P.Witkin, M.Baudin, and R.O.Duda "Uniqueness of the Gaussian kernel for scale-space filtering," *IEEE Tranz. Patt. Anal Machine Intell.*, vol PAMI-9, pp 726-741, Nov. 1987.
- [30] A.K.Jain, F.Farrokhnia, Unsupervised texture segmentation using Gabor filters, *Pattern Recognition*, 24, 12, pp.1167-1186 (1991)
- [31] D.Dunn, W.E.Higgins, J.Wakeley, Texture segmentation using 2-D Gabor elementary functions, *IEEE Trans. Pattern Anal. Machine Intell.*, 16, 2, pp.130-149 (1994).
- [32] T.S.Lee, Image Representation Using 2D Gabor Wavelets, *IEEE Trans. Pattern Anal. Machine Intell.*, 18, 10, pp.959-971 (1996).

PRISM - Generator of Image Understanding Systems

I.V.Gribkov, A.V.Zakharov, A.V.Koganov, P.P.Koltsov, N.V.Kotovich, A.A.Kravchenko, A.S.KoutsaeV, V.K.Nikolaev.

NIISI RAN
Moscow, Russia

Abstract At the development the image understanding systems the package of methods of segmentation, feature extraction, identification of objects, scene analysis and other tools is created. In the given work the approach is offered allowing to combine these tools in integrated system, having properties of visual programming. The offered system Prism allows for the given set of the images and their samples to choose processing and identification methods, suitable for it, to select optimal parameters of these methods and to generate the image understanding system, specialised and alienated from Prism.

Àððâñ: Ðíññèÿ, Ìñéââ, 117218, Ìàðèèíâñèéé ïðíñíâèð, äñ 36, êíðíóñ 1, ÌËËË ÑÀÍ.

E-mail: nik@genebee.msu.su, koltsov@niisi.msk.ru, nikolaev@niisi.msk.ru